MODELO DE RECONOCIMIENTO CON API DE IBM: "PERSONALITY INSIGHTS". RASGOS DE LA PERSONALIDAD, NECESIDADES Y VALORES EN LOS TEXTOS.

Martínez Hernández, Niver Asaid.
nivermartinezmed@gmail.com
Universidad Nacional Autónoma de México

***Resumen*— El presente escrito plantea una implementación para un espacio experimental (sitio web) en el que se pueda utilizar una API en el intento de generar un desarrollo que persiga el consumo de las tecnologías de este tipo; proponiendo una versión que nos ofrezca aprender más de ellas mediante la solución de algunos modelos para comprender en este caso, los rasgos de la personalidad en un texto.**

***Índice de términos*— Personality Insights, Python, IBM, Values, API, Redes Neuronales, Modelos Computacionales, Psicología.**

I. INTRODUCCIÓN

La personalidad es un conjunto de varias características que configuran la manera de ser de una persona, algunas personas señalan que cada persona tiene una personalidad única e irrepetible. Sin embargo, existen al menos en el campo de la inteligencia artificial; algunos modelos que proponen describir algunas de estas propiedades de las personas. La herramienta utilizada en esta ocasión es una API de la empresa Norteamérica de tecnología, IBM™. Esta herramienta que proporciona un buen modelo de entrenamiento alojado en los servidores de la misma empresa nos permite hacer llamadas por medios de unas credenciales que se pueden obtener al crear una cuenta en IBM Cloud [1].

Figura 1. Los cinco grandes, es un concepto de la psicología y hoy en día de la publicidad que delimitan algunos factores de la personalidad de las personas.

Luego de crear la cuenta, se consume el servicio conocido como *Personality Insights [2]*, que es una de varias tecnologías que se pueden tener a disposición con la cuenta gratuita. Te dan alrededor de 1000 llamadas al mes para consultas de API's que se quieran implementar. Los documentos de los sitios oficiales de la empresa no son muy extensos respecto a los detalles técnicos de la implementación, sin embargo la documentación de la API no está mal, que inclusive menciona algunos datos sobre cómo se hacen los modelos de predicción de la personalidad desde el punto de vista psicológico como publicitario, porque a fin de cuentas es una herramienta como muchas otra utilizada para diversos fines, sin embargo en esta ocasión se planea utilizarla como un medio para detectar rasgos de personalidad en personas con problemas sobre todo de depresión.

II. PREPARANDO EL ENTORNO PARA LA IMPLMENTACIÓN.

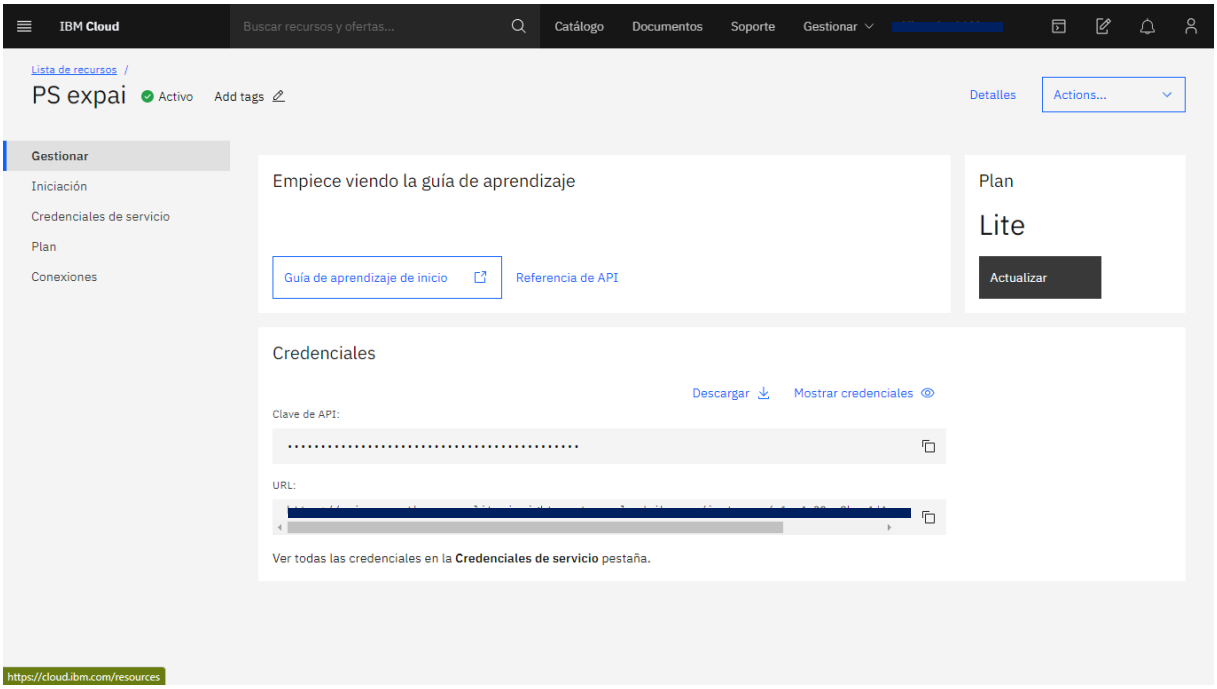


Figura 2. El sitio de IBM Cloud, donde podemos ver como se gestionan las credenciales para consumir los servicios que nos proporciona la API de IBM Watson. No sólo hay servicios de IA, sino que también cuentan con almacenamiento, máquinas virtuales, contenedores, etc.

Quizá generar un entorno de desarrollo sea de las tareas más engorrosas de preparar un proyecto, dado que muchas veces surgen complicaciones por los medios y el equipo que se están utilizando. Sin embargo, una vez librada esa parte se pueden obtener valores muy interesantes de estar buscando la forma de utilizar cierta tecnología.

El primer paso fue verificar que se podía consumir la API con Python, lo cual resulto bien, pero se tuvieron que realizar varios pasos previos desde generar la cuenta hasta obtener las credenciales para poder generar las llamadas con una autenticación correcta.

```
#Import personality Insights SDK
from watson_developer_cloud import PersonalityInsightsV3
```

Figura3. Celda de Jupyter Notebook que ilustra la línea que nos ayuda a importar la API que requerimos para Python.

Cuando se consume la API, encuentras que el método que te genera el perfil de lo que se está ingresando (en este caso un texto), va a retorno un archivo .json en el cual con esa estructura de árbol jerárquico se van subdividiendo los campos de interés que se pueden obtener de una persona, por ello se incluye la biblioteca para utilizar algunos métodos necesarios en está implementación. Por otro parte se proponía que como los textos sería interesante que ingresarán en idioma inglés que es como el modelo funciona mejor; podemos traducir con una API de Google, para saber en nuestro idioma que es lo que estamos ingresando.

III. ¿QUÉ NOS DICEN LOS POST DE UNA PERSONA?

Estuve revisando algunas propuestas para ver como conseguía información y recursos que me proporcionaran un mejor método para utilizar el modelo, pues en sí lo más interesante de esto es a mi parecer entender cómo se puede aplicar un campo de la inteligencia artificial para interpretar cosas de nuestra vida cotidiana. Watson, es un agente que se dice una de las inteligencias artificiales más desarrolladas de nuestros tiempos y con ello observamos que la precisión del modelo podía ser puesta a prueba

respecto a texto de personas con depresión que fui encontrando en foros, pero principalmente el sitio *Reddit* (cabe señalar que a pesar de tomar información de otras personas se ha dejado lejos de la vista pública). Se deja el enlace a algunos foros que son público [3] [4].

Para poder analizar el texto, tenemos que generar una instancia del servicio de IBM (*Personality Insights*). Los parámetros que nos pide el constructor de esta son las credenciales una url que aparece punteada para ocultar los caracteres y al *apikey.*

Una vez realizada esa instancia y con autenticación correcta si se ingresaron bien las credenciales, comienza la parte más interesante que es empezar a ingresar datos y que nos entregue algo el modelo. ¿Qué es ese algo?

Una vez que hemos logrado captar un texto de cualquiera de las maneras posible. Sigue darle un sentido a eso que está plasmado como oraciones y palabras. Por ello, utilizamos un método de la instancia que creamos con el servicio de rasgos, este método no permite obtener un diccionario con el que vamos a poder obtener información por medio de métodos más conocidos, lo cual es la magia de la programación. Si llegas a ese punto en donde ya estás en tu zona de dominio, puede que le entres con más fervor a interpretar el comportamiento de las distintas tecnologías, no muchas personas en esta área nos es fácil perderlo el miedo a las API's. Y en fondo son muy útiles.

Este diccionario parece ya más un HTML o un .json visto desde las impresiones que nos ofrece la variable:

```
#profile['warnings']
#profile['processed_language']
#profile['processed_language']
profile['personality']
```

```
[{'trait_id': 'big5_openness',
  'name': 'Openness',
  'category': 'personality',
  'percentile': 0.12815913523744238,
  'significant': True,
  'children': [{'trait_id': 'facet_adventurousness',
    'name': 'Adventurousness',
    'category': 'personality',
    'percentile': 0.02053905567538128,
    'significant': True},
   {'trait_id': 'facet_artistic_interests',
    'name': 'Artistic interests',
    'category': 'personality',
    'percentile': 0.39312075578453515,
    'significant': True},
   {'trait_id': 'facet_emotionality',
```

Fig4. El diccionario se puede operar más fácilmente y con ello podemos acceder a datos que nos son de utilidad para describir el perfil de una persona. Es muy similar al ,json.

El vector general que nos arroja esta instancia del servicio abarca los rubros para caracterizar al texto, de:

- Palabras contadas
- Palabras contadas por mensaje
- Procesamiento del lenguaje
- Personalidad.
- Valores
- Y alertas.

Ahora que ya tenemos en un diccionario la información que nos interesa, podemos llevarla a términos de otra estructura de datos que nos deja apreciar algunas características más estructuradas de un perfil asociado al porcentaje que se encuentra presente de ciertos rasgos de personalidad, valores y necesidades presentes en los textos.

IV. VISUALIZACIÓN DEL PERFIL.

Ahora generamos un diccionario de sólo las partes que nos interesa mostrar, tal que es el porcentaje de presencia de los rasgos y su correspondiente denominación. Lo llevamos hacia una estructura de tipo dataFrame y posteriormente ya en esa estructura, la misma tiene métodos de salida hacia una representación gráfica.

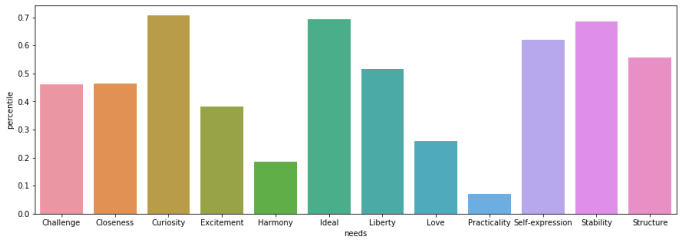


Figura5. Tabla generada con los resultados de las necesidades según el modelo y vemos que en este testo lo que predomina son emociones de curiosidad, ideales, estabilidad e introspección.

El segundo punto, ya es parte de los subtemas como tal, nos habla de las investigaciones relacionadas y nos da una breve descripción de ellas como una en la que los autores mencionan que "[…] los autores presentan un enfoque para identificar instancias de eventos y extraer información semántica relacionado con ellos, como los verbos que representan la acción del evento" [2].

Posteriormente, se realizaron más pruebas de distinta forma y se hizo con el fin de desplazar la aplicación hacia otras plataformas como el diseño web. Lo cual no es del todo o no fue una buena idea, porque dediqué mucho más tiempo de estarme peleando con cosas del HTML, CSS y hasta otras cosas que te encuentras por el camino, pero que luego resumida, pero bastante específica sobre el PLN, sobre la IA en general no es tan simple como quisiéramos.

También se propusieron otros módulos durante el desarrollo que están incluidos en el código fuente, pero que no resultaron sencillos de integrar con la versión web. Así que tanto el analizar audios, como el hecho de traducir los textos está deshabilitada por el momento.

V. EL CÓDIGO QUE SE HACE PASAR POR HTML, DATAFRAME Y CADENAS.

Se presentan algunos avances significativos que nos permiten tener un respaldo de lo que se logró al trabajar con una API como esta. Se generó un archivo de Python que al correrlo hace las consultas necesarias, se le tiene que pasar un texto ya sea dentro del mismo código o llamando alguno dentro de su mismo directorio que sea de preferencia .txt y llega hasta el punto de convertir la salida de dataFrames en un archivo de tablas con los valores estadísticos de los rasgos de la personalidad. A continuación, el resultado:

```
17  def start(text):
18      profile = service.profile(text, content_type = 'text/pl
19      x = convierte_df(action_back('personality',profile))
20      y = convierte_df(action_back('needs',profile))
21      z = convierte_df(action_back('values',profile))
22      crea_html(x,y,z)
23      return 0
24
25  def convierte_df(df_marks):
26    html = df_marks.to_html()
27    if html is None:
28      html = DEFAULT_VALUE
29    return html
30
31  def crea_html(p,n,v):
32      f = open('gener.html','wb')
33      mensaje = """<html><head></head><body><div><h1>Personal
34      s = mensaje.encode()
35      f.write(s)
36      f.close()
37      webbrowser.open_new_tab('gener.html')
38      return 0
```

Figura6. La función de convertir hacia HTML casi no se aprecia, pero en general es algo inusual quizá, integrar de esa manera par comunicar entre lenguajes, pero era en la búsqueda de lograr otro objetivo mayor.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.18362.836]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\niver>cd Desktop

C:\Users\niver\Desktop>python back.py
Iniciando...
back.py:15: DeprecationWarning: watson-developer-cloud moved to ibm-watson. To
get updates, use the new package.
  service = PersonalityInsightsV3(url=url, iam_apikey=apikey, version='2020-05-
31')

C:\Users\niver\Desktop>
```

Figura7. Se hace la llamada para instanciar el servicio PS de IBM y posteriormente se regresa al estado principal una vez que se generó el HTML que se ve así.

Una vez que se ejecuta este código el plan es que pueda desplegar resultados interactivos en una página web que veremos a continuación. El problema es que no contaba con la asincronía.

Necesidades

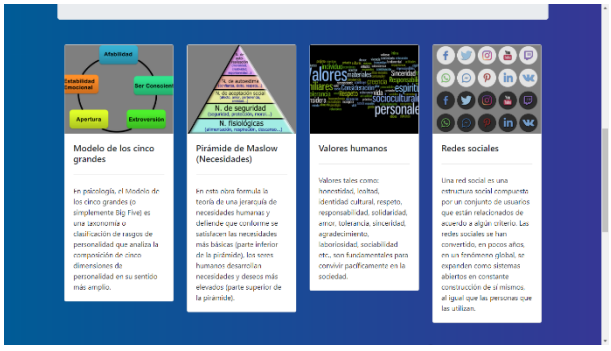
	needs	percentile
0	Challenge	0.177934
1	Closeness	0.630001
2	Curiosity	0.719963
3	Excitement	0.403437
4	Harmony	0.433785
5	Ideal	0.530917
6	Liberty	0.332839
7	Love	0.872584
8	Practicality	0.143497
9	Self-expression	0.320851
10	Stability	0.370934
11	Structure	0.371587

Figura8. Se muestra una tabla en HTML generada por el código para mostrar los pesos de cada necesidad humana según el texto analizado.

VI. SITIO WEB

En esencia mi plan con el sitio web es que resultase como un espacio de experimentación para tecnologías tipo API, porque si bien es importante conocer cómo aplicar los diferentes modelos computacionales de la inteligencia artificial, es aún más complicado saber que hay detrás de un modelo como este. Considero que de no tener nociones de Python o la más mínima idea de lo que son las redes neuronales, el *machine learning* y las API's. Esto hubiera sido realmente fatal, pero de lo que resultó.

Presentó una captura pantalla del sitio que se subió al servidor, la idea de la funcionalidad que espero que tenga y se logré resolver el problema de la asincronía y dejo el link [5].



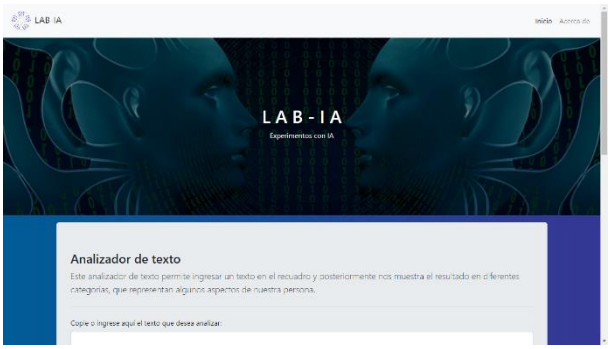


Figura8 a) Captura de una de las páginas de sitio web b) Captura del home donde se ejecuta la implementación principal.

VII. CONCLUSIONES.

Realizar el análisis de textos mediante Inteligencia Artificial, y sobre todo sobre un modelo que no está cercano mí, vive en algún servidor a cientos o miles de kilómetros de mí. Y lo más importante de todo esto es darse cuenta de que es más importante saber que hay detrás de cualquier modelo. Porque una red neuronal se comporta como lo hace. En ese sentido, considero que el aprendizaje logrado es una manera de conocer que los modelos predictivos tienen una confiabilidad tal que su función de rendimiento tiene que mejorar y siempre estarán aprendiendo para reforzar esa base de datos. Supongo que también sería bueno hacerlo, nos ayudaría a ser más efectivos en proyectos como este, estudiar cosas demás de las que conocemos uy entender que hasta los textos pueden hacernos ver cosas de nuestra personalidad que, hasta una AI, puede interpretar.

REFERENCIAS

[1] IBM Cloud. *Personality Insights*. Recuperado el 30 de mayo del 2020, en el sitio web: https://cloud.ibm.com/apidocs/personality-insights.
[2] IBM Cloud. Recuperado el 30 de mayo de 2002, en el sitio: https://cloud.ibm.com/.
[3] mrenjoydamoney. *Quarantine period about to end in from where I'm from, starting to realise something*.
[4] Reddit. Recuperado el 30 de mayo del 2020, en el sitio:https://www.reddit.com/r/ForeverAlone/comments/gt830p/quarantine_period_about_to_end_in_from_where_im/.

[5] Github. Recuperado el 30 de mayo de 2020, del sitio: https://github.com/NiverMtz/PS-ex_pai/back.py.
[6] Heroku. LAB-IA. Recuperado el 30 de mayo de 2020, del sitio: https://infinite-crag-13775.herokuapp.com/.
[7] Matheson, R. (2018). *Model can more naturally detect depression in conversations. MIT*. Recuperado el 30 de mayo de 2020, del sitio: http://news.mit.edu/2018/neural-network-model-detect-depression-conversations-0830.
[8] Arroll B, Goodyear-Smith F, Crengle S, et al. (2010). *Validation of PHQ-2 and PHQ-9 to screen for major depression in the primary care population*. *Ann Fam Med*. Recuperado el 30 de mayo de 2020, del sitio: https://pubmed.ncbi.nlm.nih.gov/20644190/.

Autor
Martínez Hernández Niver Asaid.
Estudiante en Ingeniería en Computación.
FI-UNAM.