Valoración cuantitativa de productos a través de procesamiento de lenguaje natural (NLP)

Diego Stiven Osorio Vélez

Luis Felipe Salazar Ucros

Seleccione tipo de documento para optar al título de Especialista en Analítica y Ciencia de Datos

Asesor
Sebastian Rodríguez Colina, Magíster (MSc)

Universidad de Antioquia
Facultad de Ingeniería

Especialización en Analítica y Ciencia de Datos

Medellín, Antioquia, Colombia

2021

Cita		Osorio Vélez y Salazar Ucros [1]
Referencia Estilo IEEE (2020)	[1]	D.S.Osorio Vélez y L. F. Salazar Ucros, *"Valoración cuantitativa de productos a través de procesamiento de lenguaje natural (NLP)"*, Trabajo de grado especialización, Especialización en Analítica y Ciencia de Datos, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2021.

Especialización en Analítica y Ciencia de Datos, Cohorte II.

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Jesus Francisco Vargas Bonilla.

Jefe departamento: Diego Jose Luis Botia Valderrama.

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

RESUMEN

Para las organizaciones es importante entender el grado de satisfacción que un cliente tiene hacia un producto o servicio. Obtener esta información de manera cuantitativa puede resultar costoso en muchas ocasiones, especialmente en empresas que mueven una gran cantidad de clientes, como es el caso de las e-commerce.

Por lo anterior, se hace necesaria la automatización del proceso de valoración cuantitativa del grado de satisfacción de un cliente. Este trabajo se enfocó en la estimación cuantitativa de esta adherencia al servicio a través de la categorización automática del sentimiento presente en un comentario hacia el producto. En este documento se expone y evidencia el proceso de desarrollo de una aplicación basada en deep learning para lograr el objetivo propuesto.

Palabras claves: Deep learning, procesamiento de lenguaje natural, análisis de sentimiento, necesidad, deseo, e-commerce, satisfacción de cliente.

I. INTRODUCCIÓN

El presente proyecto tiene como objeto plantear una solución a la valoración cuantitativa de productos a través del análisis de sentimiento basada en deep learning, soportada en el modelo BERT, el cual es capaz de interpretar el lenguaje humano en diferentes aplicaciones. Esta solución se enfoca en el análisis de sentimientos de los comentarios u opiniones que puedan tener los productos ofrecidos a los consumidores, con el fin de clasificarlos y de esta manera obtener información para la toma de decisiones sobre el posicionamiento tanto de la marca como del producto y su aceptación en el mercado.

La captura de los datos no estructurados como comentarios, opiniones o calificaciones se logra gracias a la interacción constante con el cliente; éstos se obtienen de diferentes fuentes en tiempo real generando una gran estructura de flujo de datos, para la cual se hace necesario agruparlos y etiquetarlos de forma automática, para poder realizar los análisis respectivos que ofrecerá como resultado nuevas estrategias de mercado. Es de anotar que las etiquetas a utilizar en esta solución serán comentarios positivos y negativos.

El interés de este trabajo consiste en poder utilizar las herramientas y tecnologías asociadas al machine learning en las compañías, aportando nuevos desarrollos como aplicaciones que faciliten el procesamiento eficaz y eficiente de la información encontrada en la web. Es por ello que se seleccionó el e-commerce más grande e influyente del mundo en la actualidad, el cual es Amazon, ya que allí es donde se comercializan la mayoría de los productos ofrecidos por las compañías y se puede acceder a bases de datos que pueden servir de insumo para el desarrollo de modelos.

II. PLANTEAMIENTO DEL PROBLEMA

En la actualidad con el avance de la tecnología informática, la globalización, el aumento en el volumen y flujo de datos en general, la competitividad de las industrias en los mercados ha sido mayor, específicamente el sector e-commerce en donde el término de "reputación online" ha cobrado relevancia dado que influye en las interacciones de los clientes con el portal, servicio o producto.

Lo anterior no es ajeno al concepto análisis de sentimiento, el cual se realiza utilizando técnicas de procesamiento de lenguaje natural conocidas en el campo del aprendizaje de máquinas, y se utiliza para la clasificación, monitoreo, seguimiento y valoración cuantitativa de contenido escrito subjetivo presente en comentarios, foros, webs, opiniones y documentos obtenidos en la internet que pueden denotar una connotación positiva o negativa.

Toda esta información refleja en parte la reputación de la empresa de cara al cliente y permite realizar seguimientos a sus operaciones obteniendo métricas e información relevante de negocio. Surge entonces la necesidad de realizar un producto que permita hacer valoraciones cuantitativas de las opiniones clasificándolas como positivas o negativas.

El objeto de estudio de este proyecto, como se mencionó con anterioridad, está enfocado en el e-commerce más grande del mundo hasta el momento, Amazon, en donde se espera realizar un análisis de sentimiento partiendo de las opiniones que los clientes escriben en la plataforma al finalizar la compra de un producto, por medio de una clasificación binaria y aplicando técnicas de procesamiento de lenguaje natural con un modelo de deep learning.

III. DISEÑO DE LA SOLUCIÓN

Para la solución se evalúa un modelo conocido con el nombre de “BERT” el cual está enmarcado en la línea de conocimiento del deep learning, debido a su efectividad a la hora de resolver problemas de procesamiento de lenguaje natural y su escalabilidad para manejar grandes volúmenes de datos. Este trabajo se estructuró basado en los siguientes aspectos:

a. Adquisición de bases de datos.
b. Selección de métricas de evaluación.
c. Selección de modelo, entrenamiento y compilación.
d. Resultados.
e. Despliegue de modelo.

A continuación, se van a detallar los elementos más relevantes del diseño.

A. Adquisición de bases de datos

Los datos para este trabajo fueron obtenidos a través de un api que conecta directamente con un repositorio de amazon web service, el cual tiene cerca de 683 millones de registros de opiniones sobre diferentes productos en categorías aleatorias, el api se encuentra disponible en el siguiente link: https://s3.amazonaws.com/fast-ai-nlp/amazon_review_polarity_csv.tgz.

La focalización en esta base de datos se centró específicamente en tres variables. La variable “sentiment” que describe si un comentario es positivo o negativo, “title” el cual hace referencia al título con el que se escribió la opinión y “review” en donde está el texto de la opinión hecha por el cliente.

Una vez seleccionada la base de datos, por cuestiones de optimización de los recursos se limita la lectura a 500.000 registros mediante la creación de funciones, con el fin de no saturar la memoria RAM disponible, y se procede a hacer un análisis exploratorio de la misma, lo cual permite evidenciar que se está trabajando con una base de datos balanceada.

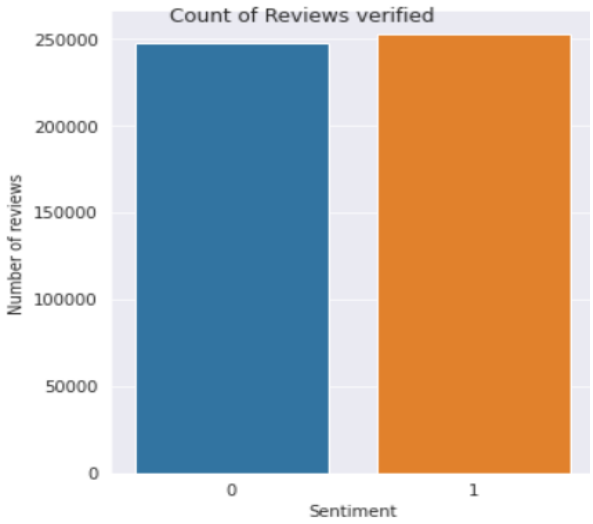

Figura 1. Diagrama de barras con conteo de sentimiento para la base de datos

Fuente:Amazon web service y elaboración propia (2021)

Adicionalmente se identificaron cuáles eran las palabras que más se repetían dentro de las opiniones y su distribución por oración.

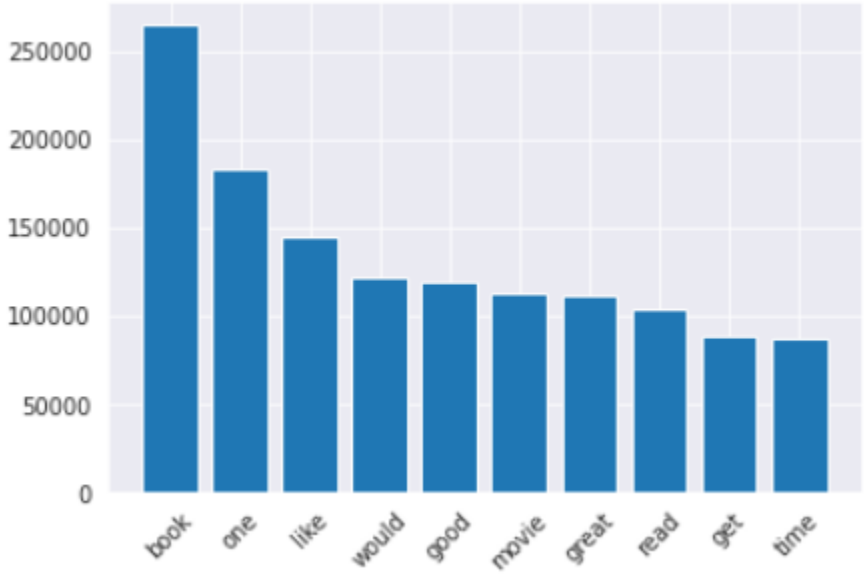

Figura 2. Palabras que más se repiten dentro de la base de datos.

Fuente: Amazon web service y elaboración propia (2021)

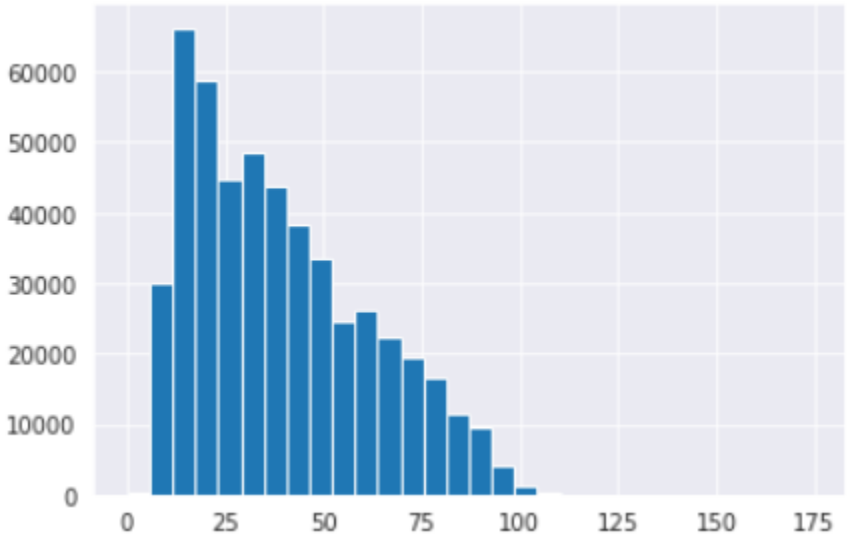

Figura 3. Histograma número de palabras por opinión.

Fuente: Amazon web service y elaboración propia (2021)

	sentiment	title	review
0	1	Stuning even for the non-gamer	This sound track was beautiful! It paints the ...
1	1	The best soundtrack ever to anything.	I'm reading a lot of reviews saying that this ...
2	1	Amazing!	This soundtrack is my favorite music of all ti...
3	1	Excellent Soundtrack	I truly like this soundtrack and I enjoy video...
4	1	Remember, Pull Your Jaw Off The Floor After He...	If you've played the game, you know how divine...
...	...	...	...
499995	0	Prepare for offence	Ridley's conjecture and supposition swirling a...
499996	1	This is a great song but BUY THE ALBUM.	Trust me, you'd much rather buy the album than...
499997	0	What happen to Fram gas filter's Quality ???	I used this same number of Fram filter before ...
499998	0	NO FILTER ON 2003 Hyundai XG350L!!!	There is no fuel filter on this year/make. Tru...
499999	0	What a pain!	This item was not easy to put up. It took a wh...

500000 rows × 3 columns

Figura 4. Lectura de datos recopilados para entrenar el modelo.

Fuente: Amazon web service y elaboración propia (2021)

B. Selección de métricas de evaluación.

Para el desarrollo de este proyecto es importante conseguir un modelo que logre tener una precisión por encima del 80%. Por lo anterior se espera si un comentario está clasificado como negativo o positivo en efecto lo sea y adicionalmente se espera maximizar la métrica de recall.

Partiendo de la matriz de confusión se definen las métricas seleccionadas.

1) Matriz de confusión:

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Tabla 1. Matriz de confusión conceptual

Fuente: https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico

2) Precisión:

Esta métrica es definida como los verdaderos positivos sobre el total de positivos, para este caso a través de esta métrica se evaluarán cuáles son las etiquetas que fueron correctamente clasificadas.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Figura 5. Concepto precisión.

Fuente: https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico

3) Recall:

Esta métrica calcula cuántos positivos reales captura el modelo como verdaderos, para este caso la métrica cobra mucha relevancia dado que mide la capacidad de encontrar positivos.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Figura 6. Concepto recall.

Fuente: https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico

C. Selección de modelo, entrenamiento y compilación

Para la selección del modelo primero se buscaron referencias para este tipo de problemas. Actualmente el modelo BERT cumple con las características pertinentes, por su arquitectura bidireccional y ha sido entrenado con grandes cantidades de datos; se estima que cerca de 3.300.000 palabras provenientes de Wikipedia y Google books han sido utilizadas en su entrenamiento de base, teniendo en cuenta el contexto del uso de las palabras.

Su sigla obedece a *Bidirectional Encoder Representations from Transformers* (BERT), que traduce "Representaciones de Codificador Bidireccional de Transformadores", es decir el transformer incluye dos mecanismos separados: un codificador que lee la entrada de texto y un decodificador que produce una predicción para la tarea, esta innovación se les atribuye a los investigadores de Google AI Languages.

Para la creación del modelo se trabajó con la GPU de Google colab con el fin de optimizar el tiempo de compilación y entrenamiento del modelo, es de anotar que este tipo de recursos tiene un tiempo limitado en cuanto a uso.

Dentro del script de Google colab se realizó el cargue de la información mediante el api de Amazon y se construyó una función que contiene el modelo "BERT" en su versión preentrenada, construida con tensorflow, la cual tiene cerca de 110 millones de parámetros.

```
# Construir modelo BERT
def build_model():
  text_input = tf.keras.layers.Input(shape=(), dtype=tf.string, name='text')
  preprocessing_layer = hub.KerasLayer("https://tfhub.dev/tensorflow/bert_en_uncased_preprocess/3", name='preprocessing')
  encoder_inputs = preprocessing_layer(text_input)
  encoder = hub.KerasLayer("https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4", trainable=False, name='BERT_encoder')
  outputs = encoder(encoder_inputs)
  net = outputs['pooled_output']
  net = tf.keras.layers.Dropout(0.1)(net)
  net = tf.keras.layers.Dense(1, activation='sigmoid', name='classifier')(net)
  return tf.keras.Model(text_input, net)
```

Figura 7. Construcción función modelo BERT.

El modelo se puede detallar con mayor precisión en el siguiente diagrama:

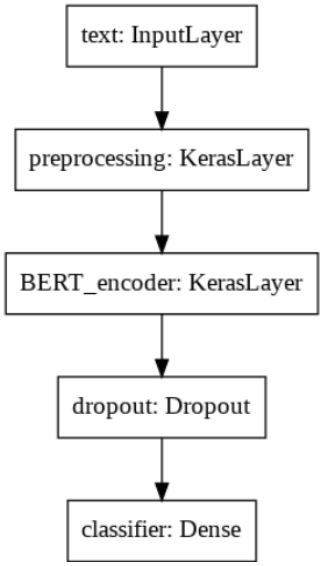

Figura 8. Diagrama flujo modelo BERT.

El modelo recibe un vector de entrada, en este caso el texto que representa la opinión, el cual pasa a la primera capa de preprocesamiento en donde se eliminan los acentos y convierte a minúscula

cada palabra, adicionalmente la entrada que recibe el modelo tiene un límite de 128 caracteres, si una opinión de entrada excediera los 128 caracteres, el modelo tokeniza hasta llegar al límite mencionado, estos pasos en la capa de preprocesamiento se conocen como:

- **"Input_word_ids"**: El cual tiene los identificadores de token de la secuencia de entrada.
- **"Input_mask"**: Tiene el valor de 1 en las posiciones de las palabras tokenizadas y 0 en las posiciones de relleno.
- **"Input_type_ids"**: Tiene las posiciones de los identificadores de entrada.

Estos tres últimos conceptos forman un diccionario conocido como codificador, el cual sirve como entrada de la capa BERT_encoder [Batch_size,128], la cual tiene un bloque de 12 capas ocultas que producen una salida de un vector de 768, y para evitar overfitting este vector de salida pasa por un dropout del 10% para finalmente pasar por una neurona de salida con función de activación “sigmoid” la cual genera la predicción entre 0 y 1.

D. Arquitectura del modelo:

```
Model: "model"
_________________________________________________________________________________________________
 Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
 text (InputLayer)            [(None,)]                 0           []

 preprocessing (KerasLayer)   {'input_type_ids':        0           ['text[0][0]']
                               (None, 128),
                               'input_mask': (Non
                              e, 128),
                               'input_word_ids':
                              (None, 128)}

 BERT_encoder (KerasLayer)    {'encoder_outputs':       109482241   ['preprocessing[0][0]',
                               [(None, 128, 768),                    'preprocessing[0][1]',
                               (None, 128, 768),                     'preprocessing[0][2]']
                               (None, 128, 768),
                               (None, 128, 768),
                               (None, 128, 768),
                               (None, 128, 768),
                               (None, 128, 768),
                               (None, 128, 768),
                               (None, 128, 768),
                               (None, 128, 768),
                               (None, 128, 768),
                               (None, 128, 768)],
                               'default': (None,
                              768),
                               'pooled_output': (
                              None, 768),
                               'sequence_output':
                               (None, 128, 768)}

 dropout (Dropout)            (None, 768)               0           ['BERT_encoder[0][13]']

 classifier (Dense)           (None, 1)                 769         ['dropout[0][0]']

=================================================================================================
Total params: 109,483,010
Trainable params: 769
Non-trainable params: 109,482,241
```

Figura 9. Arquitectura del modelo BERT.

Fuente: Elaboración propia (2021)

El modelo construido se compiló con el optimizador “Adam”, usando como función de pérdida “binary crossentropy”, evaluado con las métricas precisión y recall. Finalmente, se separan los datos de la siguiente manera: 75% para entrenar y 25% para evaluar.

La distribución de las etiquetas como se mencionó anteriormente es balanceada, igualmente al momento de separar los datos se puede observar en las figuras 10 y 11 que las etiquetas mantienen la misma distribución balanceada.

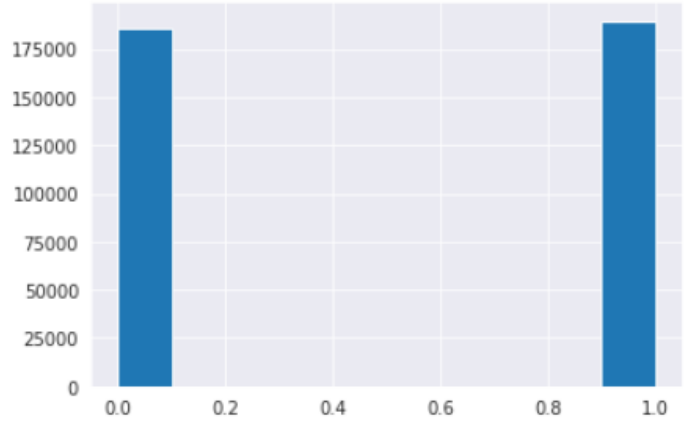

Figura 10. Distribución etiquetas datos de entrenamiento.

Fuente: Elaboración propia (2021)

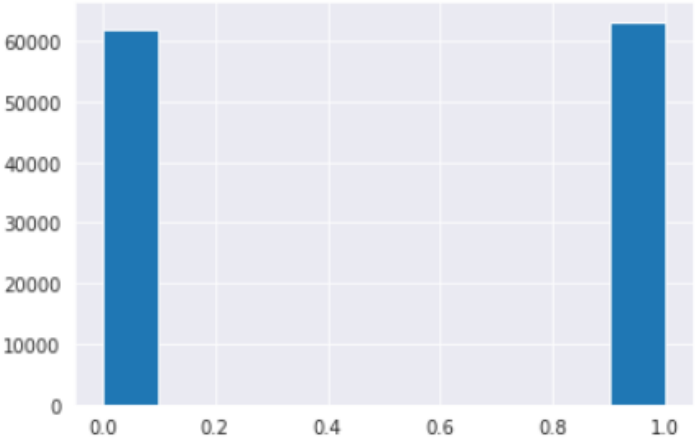

Figura 11. Distribución etiquetas datos de test.

Fuente: Elaboración propia (2021)

E. Resultados

Una vez entrenado el modelo, se evaluaron los datos test se obtuvieron los siguientes resultados:

1) Matriz de confusión:

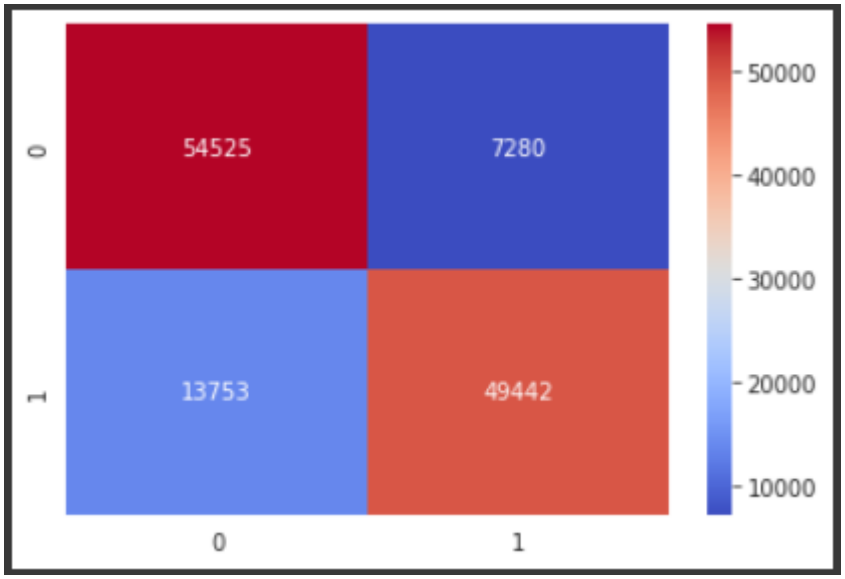

Figura 12. Matriz de confusión datos de prueba.

Fuente: Elaboración propia (2021)

Sentiment	Precision	Recall
0	0.80	0.88
1	0.87	0.78
Macro average	0.84	0.83

Tabla 2. Reporte de métricas modelo

Fuente: Elaboración propia (2021)

Lo anterior refleja resultados esperados en donde se busca una precisión del modelo por encima o igual de 80% en ambas métricas de negocio, tanto para los comentarios negativos como para los positivos. Lo anterior es de utilidad para Amazon, pues se podrá hacer seguimiento más detallado y tomar decisiones sobre los productos que se están ofreciendo en el mercado.

F. Despliegue

Con la intención de dejar el modelo entrenado en producción, se despliega el algoritmo como una API en AWS utilizando FastAPI y Docker, herramientas que van a permitir realizar esta tarea. Para llevar a cabo lo anterior primero se entrena y se guarda el modelo con una GPU de Google colab, después se crea una app en FastAPI para hacer predicciones con el modelo guardado y se utiliza Docker para crear una imagen de esta.

Luego se crea una cuenta en Amazon web services, allí se usa el servicio de almacenamiento y motor de cálculo en la nube, poniendo a disposición un bucket en "S3" y una máquina virtual en "EC2" (t2. large). Una vez se tiene la máquina virtual configurada, se guarda la imagen de Docker, la cual contiene la app creada en FastAPI, en un repositorio del servicio ECR (Elastic Container Registry) para posteriormente ser transferida a la máquina virtual creada anteriormente y la cual va a permitir, finalmente, utilizar el modelo como un servicio web.

A continuación, se propone una arquitectura simplificada de lo anterior.

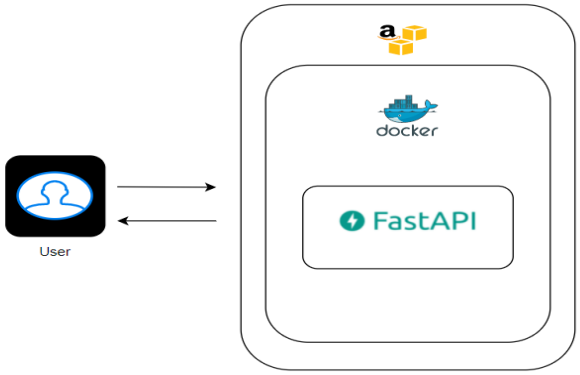

Figura 13. Prototipo de modelo desplegado.

Fuente: Elaboración propia (2021)

Consultar el detalle en repositorio de github.

IV. TRABAJO FUTURO

Se espera poder escalar o evolucionar el modelo implementando los siguientes ítems:

- Nuevos modelos en diferentes idiomas específicamente para opiniones en español, dado que estas permiten abordar el segmento de clientes potenciales como es el caso de LATAM.

- Agregar más información de salida sobre las opiniones que pasen por el modelo, una especie de contador de palabras, log de trazabilidad, estadística descriptiva y ajustar el modelo a que reciba un "Identificador de entidades", es decir, que identifique o haga un etiquetado sobre si el comentario es una referencia al producto como tal, o si es a la marca.

- Monitorear el modelo desplegado en la nube con el fin de evaluar, generar alertas y estadísticas sobre las predicciones, permitiendo dar cuenta de algún tipo de cambio de comportamiento en los clientes.

V. CONCLUSIONES

A través de este trabajo se puede concluir que se cumplió el objetivo definido al usar el modelo BERT para generar una valoración cuantitativa de productos, sin embargo, aún queda camino por recorrer en busca de mejorar las métricas alcanzadas en este experimento.

Por otra parte, el modelo BERT cuenta con una amplia versatilidad para resolver problemas de procesamiento de lenguaje natural, en este caso el enfoque fue abordar uno de sus usos, el cual es el análisis de sentimiento, evidenciando que la automatización de procesos como el etiquetado de opiniones de productos puede ser un factor diferenciador en las organizaciones, al influir directamente en las interacciones con los clientes.

Adicionalmente, fue un reto poder poner en producción en AWS este modelo, ya que se debe ser muy cuidadoso al momento de crear una máquina virtual, porque esta debe contar con las características necesarias para su funcionamiento sujeto a la optimización de recursos. Para este caso la capa gratuita que ofrece AWS no fue suficiente porque se quedaba corta en memoria RAM, por ende, se adquirió una máquina virtual denominada *t2. large* de características superiores.

Finalmente, la propuesta que se entrega es la posibilidad de poner en producción el etiquetado automático, y a partir de este monitoreo generar estadísticas descriptivas sobre este etiquetado de opiniones, reconociendo cuáles son los productos que mayor impacto están generando y los de menor impacto, con el objeto de tomar decisiones que permitan mejorar constantemente el servicio y la reputación online de la organización.

VI. REFERENCIAS

[1] Koo Pin Shung. Accuracy, precision, recall or F1?. 2018, [En línea]. Disponible en: https://bit.ly/3ljcfbk

[2] Michael Phi. Illustrades guide to LSMTs and GRUs: A step by step explanation. 2018, [En línea]. Disponible en: https://bit.ly/3188mii

[3] Haruna Isah, Tariq Abughofa, Sazia Mahfuz, Dharmitha Ajerla, Farhana Zulkernine, Shahzad Khan. A Scalable Framework for Multilevel Streaming Data Analytics using Deep Learning. 2019, [En línea]. Disponible en: https://bit.ly/3nZDEkk

[4] Data science academy. Modelo BERT para processamento de linguagem natural. 2021, [En línea]. Disponible en: https://bit.ly/32AvDKt

[5] Itelligent. Análisis de sentimiento. ¿Qué es, cómo funciona y para qué sirve?. 2017, [En línea]. Disponible en: https://bit.ly/3cTNLkB

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Tranformers for Languaje Understanding. 2019, [En línea]. Disponible en: https://bit.ly/3cVfy3R

[7] Nagesh Singh Chauhan. Métricas de evaluación de modelos en el aprendizaje automático. 2020, [En línea]. Disponible en: https://bit.ly/3I09IN8

[8] Forbes Advertorial. Latinoamérica, terreno fértil para el ecommerce. 2021, [En línea]. Disponible en: https://bit.ly/3I4BMPe

[9] Chiara Panico. La eficacia del análisis de sentimientos para la empresa: el caso de estudio Dell Technologies Inc. 2018, [En línea]. Disponible en: https://bit.ly/3cVoHJZ

[10] Tensorflow, “tfhub.dev/tensorflow/bert_en_uncased_preprocess/3 [Software].”. TensorFlow, 2021.

[11] Tensorflow, “tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4 [Software].”. TensorFlow, 2021.