Universidad del Desarrollo
Facultad de Ingeniería

"UTILIZACIÓN DE MÉTODOS DE NLP PARA OBTENCIÓN DE INSIGHTS SOBRE LA SATISFACCIÓN DE CLIENTES:
Del texto al valor"

POR: FRANCISCO MOISÉS INOSTROZA CARVAJAL

Proyecto de grado presentado a la Facultad de Ingeniería de la Universidad del Desarrollo para optar al grado académico de Magíster en Data Science

PROFESOR GUÍA:

Dra. Loreto Bravo

Diciembre 2021

SANTIAGO

TABLA DE CONTENIDO

Abstract

En la era de la digitalización, gran parte de las compañías cuentan con diversos canales para obtener feedback de sus clientes, por lo que analizarlo es de gran importancia para obtener ventajas competitivas, uno de los métodos claves que últimamente ha mostrado gran potencial son las encuestas de satisfacción, que entregan opiniones relevantes respecto a la voz del cliente. El problema recae cuando la retroalimentación viene de forma no estructurada, como lo es el texto, del cual no es fácil obtener información con métodos tradicionales. Por esta necesidad, se han desarrollado técnicas como el Procesamiento de Lenguaje Natural (PLN) que satisface gran parte de estas necesidades [5]. Este proyecto logra la obtención de Insights valiosos para la compañía, correlacionando distintas variables, lo que permitió identificar que los servicios prestados de fibra óptica e internet inalámbrico son los peores percibidos por los usuarios. También se logró reconocer que la performance de los canales digitales, la recarga de bolsas y los problemas de señal, son las principales causas que disminuyen la satisfacción de los usuarios.

Por otro lado, se construyó un modelo de clasificación de opiniones de los usuarios basado en “Bidirectional Encoder Representations from Transformers” (BERT) [11], que permite clasificar las opiniones de los usuarios en distintas categorías valoradas por el negocio, obteniendo un 68% de Accuracy, esto ayudando a automatizar la categorización de textos que actualmente se realiza de forma manual por una analista.

1. Introducción

No es novedad que en los últimos años las personas han aumentado la utilización de canales digitales para sus distintas necesidades, lo cual ha facilitado la obtención de datos para las empresas, permitiendo a éstas conseguir información clave que les ayude a conocer mejor a sus clientes con el objetivo de mejorar sus productos y servicios. El aprovechamiento de estos recursos genera una gran ventaja competitiva, lo que ha obligado a las compañías impulsadas por datos a hacer cada vez mayores esfuerzos para conocer la voz y experiencia de sus clientes, entendiendo sus perfiles, gustos, molestias, entre otras características, aportando valor a las distintas áreas de la organización para tomar acciones que permitan una mejora continua en la satisfacción de sus usuarios.

1.2 Definición del problema

Una de las problemáticas que produce el gran crecimiento de la generación de datos es el cómo éstos se manipulan y analizan, especialmente cuando se trata de datos no estructurados como lo es el texto. Cuando se analiza información escrita por usuarios, como lo son las encuestas de satisfacción, los métodos tradicionales de análisis no son efectivos, por lo que se debe recurrir a técnicas específicas que nos ayuden a resumir y clasificar información para obtener insights de forma automatizada, como lo es el Procesamiento de lenguaje natural (PLN) y sus técnicas asociadas, por ejemplo, el análisis de sentimientos y modelamiento de tópicos

1.3 Contribución del proyecto

Este proyecto tiene como objetivo generar insights accionables en base a las opiniones de los usuarios de una empresa de telecomunicaciones a partir de encuestas de satisfacción, con el fin de entregar información valiosa al "Centro de Excelencia" de la compañía para aportar a entender en detalle cómo afectan distintas variables a los usuarios de acuerdo con sus características (canal digital utilizado, producto, geografía, entre otros), aportando información clave para la compañía basado en los insights obtenidos, con el fin de aumentar el nivel de satisfacción de sus usuarios.

1.4 Organización de la tesis

Capítulo 2: Hipótesis y Objetivos: Se plantean la hipótesis de la investigación, objetivo general y específicos.

Capítulo 3: Datos y Metodología: En este capítulo se explica el dataset a utilizar, detallando su estructura y variables relevantes. También, se detalla la metodología **y** técnicas a emplear.

Capítulo 4: Resultados: Se realiza el procesamiento del texto, análisis de n-gramas, modelamiento de tópicos, análisis de sentimientos y clasificación de texto.

Capítulo 5: Conclusiones, Limitaciones y Planificación: Detalla las conclusiones obtenidas a partir de los resultados, las limitantes observadas y el trabajo futuro propuesto

2. Hipótesis y Objetivos

2.1 Hipótesis

Los datos de texto sobre encuestas de satisfacción, aportan valor para conseguir insights accionables sobre los clientes de un negocio.

2.2 Objetivo general

Obtener insights de clientes de una compañía de telecomunicaciones con datos de texto, en base a sus opiniones provenientes de encuestas de satisfacción.

2.3 Objetivos específicos

- Aplicación de técnicas de Procesamiento de Lenguaje Natural a opiniones redactadas por usuarios.
- Modelamiento de tópicos para categorizar los distintos tipos de opiniones.
- Generación de análisis de sentimientos en base a encuestas.
- Empleo de métodos de Deep Learning para clasificación de texto.

3. Datos y Metodología

3.1. Datos

La base de datos a utilizar, fue originada con el fin de comprender las opiniones y características de los usuarios de una compañía de telecomunicaciones, esta se obtuvo a través de encuestas realizadas por los canales digitales de la empresa (App y Web) a sus usuarios.

El Dataset original se extrajo sin pre procesamiento previo cuenta, con 63 distintas variables y 136.122 registros, donde cada registro representa a un usuario, para este análisis se utilizarán sólo 8 variables, que son las que se consideraron relevantes para el desarrollo del proyecto, los registros consideran encuestas desde enero 2020 hasta Noviembre 2021.

La estructura del Dataset a analizar es de 136.122 registros y 8 variables, las cuales son:

Fecha medición: Considera el año, mes, día y hora de la medición.

Mercado: Se refiere si el cliente cuenta con producto post pago, cuenta controlada, plan suscripción, Internet o TV.

Opinión: Es el texto de la opinión del cliente.

Coordenadas: Latitud y longitud del usuario al momento de responder la encuesta.

Duración: Tiempo en segundos que demoró el usuario en redactar su opinión**.**

Sistema operativo: Qué sistema operativo utilizó al momento de responder.

Funcionalidad: Alude a cuál fue la funcionalidad que originó su ingreso al canal digital.

Canal digital: Explica qué canal digital utiliza el usuario, App o Web.

También se utilizará un segundo dataset, con el objetivo de realizar la clasificación de texto, este se compone de 2 variables (la opinión de los usuarios y la categoría a la que pertenece dicha opinión) y 2.903 registros. Para determinar la categoría, esta la define un analista que lee las encuestas y en base a tu experiencia y criterio, categoriza dicha opinión dentro de las 8 categorías consideradas relevantes para el negocio, las cuales son las siguientes:

- **Servicio:** Opiniones relacionadas a los servicios entregados por la compañía.
- **UX:** Encuestas relacionadas a la interfaz y diseño del canal digital.
- **Funcionalidad:** Comentarios sobre las funcionalidades que entrega el canal.
- **Error no reproducible:** Se refiere a errores puntuales que se presentan a determinados usuarios.
- **Error reproducible:** Son errores masivos, que se presentan de forma generalizada.
- **Positivo:** Opiniones positivas sobre el canal.
- **Revisar:** Respuestas fuera de lo común, que requieren ser revisadas en mayor detalle.
- **Otro:** Cualquier opinión que no clasifique en las categorías especificadas.

3.2. Metodología

La metodología propuesta pretende utilizar las opiniones de los usuarios, obtenidas desde las encuestas de satisfacción, de forma de procesar esta variable de texto con técnicas de Procesamiento de Lenguaje Natural para conseguir resultados claves que nos permitan relacionar las características de los usuarios con sus opiniones.

Redes neuronales

Las redes neuronales, son un subconjunto del Machine Learning y el fundamento del Deep Learning, estas funcionan simulando las neuronas del cerebro humano y la conexión entre neuronas. Estas redes se componen de capas de entrada, capas ocultas y una capa de salida (output). Su funcionamiento se basa en pesos numéricos asignados a cada neurona, el cual representa el nivel de importancia de esta, también se requiere una función de activación y un valor de umbral para el cálculo y ponderación del valor de salida de cada neurona, del cual depende la fuerza de activación de cada una de estas.

BERT

BERT es un modelo de representación de lenguaje, el cual hace uso de Transformers que aprende relaciones contextuales entre palabras de un texto. Conceptualmente, Transformers incluye dos mecanismos principales: un codificador que lee la entrada de texto y un decodificador que produce una predicción, pero el objetivo específico de BERT es generar un modelo de lenguaje, por lo que se enfoca en el mecanismo codificador.

Modelamiento de lenguaje enmascarado

El modelamiento de lenguaje enmascarado, se refiere a la tarea de decodificar un token enmascarado en una oración, llenando los espacios en blanco a través del uso del contexto de esta, intentando predecir cual debería ser la palabra enmascarada, donde el modelo generará la sustitución más probable para cada uno permitiendo que el modelo aprenda una representación bidireccional de la oración en cuestión.

Transferencia de aprendizaje

La transferencia de aprendizaje (transfer learning), es un método de machine learning en el que se entrena un modelo para una tarea específica, el cual permite ser reutilizado para cumplir otras tareas relacionadas a partir del modelo entrenado inicialmente, esto ayudando a mejorar la eficiencia y recursos del modelo generado, ya que nace de un punto de partida más avanzado que un modelo entrenado desde cero.

Procesamiento de texto

Para el procesamiento deltexto se seguirán los siguientes pasos:

1) **Convertir texto en minúsculas:** El Dataset presenta palabras en mayúsculas y minúsculas, por lo que se estandarizará el texto en minúsculas.
2) **Quitar *Stopwords*:** Esto con el fin de remover las palabras más comunes que no aportan valor al mensaje.
3) **Utilización de expresiones regulares:** Esto con el fin de remover las puntuaciones y caracteres inútiles.
4) **Lematización de palabras:** Se lematizará el texto con el objetivo de estandarizar palabras que se escriben distintas, pero representan el mismo significado.

Luego de obtener el texto preprocesado se plantea realizar “minería de opinión”, a través de análisis de sentimientos, esto para identificar y clasificar qué emoción expresa el usuario de acuerdo con su opinión.

También se propone efectuar un modelamiento de tópicos, para poder categorizar las palabras claves extraídas que representan un tema similar.

5) **Análisis de palabras más frecuentes y n-gramas:** Se obtendrán las palabras más frecuentes encontradas en las encuestas y se generarán bi-gramas y tri-gramas, con el objetivo de identificar las frases más mencionadas por los usuarios.

6) **Modelamiento de tópicos:** Se realizará modelamiento de tópicos para agrupar los distintos tópicos encontrados en las reviews de los usuarios que utilizan los canales digitales de la compañía [9].

7) **Análisis de sentimientos:** Se estudiará la polaridad de las opiniones obtenidas en las encuestas con el objetivo se definir el sentimiento de los usuarios, con el objetivo de facilitar la obtención de Insights [10]

8) **Clasificación de texto:** Se utilizará un modelo de lenguaje Bidirectional Encoder Representation from Transformers (BERT), con el fin de clasificar las distintas opiniones en categorías, las cuáles son claves para priorizar el feedback de los usuarios, éstas actualmente se definen por un analista que lee las opiniones y las clasifica según su criterio.

Posterior a la obtención de resultados, se buscará encontrar correlaciones entre éstos y las distintas características de los usuarios, analizando de forma más

detallada qué factores y variables entregan opiniones más negativas, con el objetivo de utilizar estos insights que nos ayudarán a encontrar oportunidades de mejora. También se definirá el trabajo futuro que nos permitirá potenciar y mejorar los resultados obtenidos.

4. Resultados

La obtención de resultados se basa principalmente en 4 etapas, preprocesamiento de texto, análisis de las palabras más frecuentes y n-gramas, modelamiento de tópicos, análisis de sentimientos y clasificación de texto.

4.1 Palabras más Frecuentes

Luego de haber realizado el preprocesamiento de texto mencionado en la metodología, se obtuvieron las siguientes palabras más frecuentes como resultado, las cuales se representan en una nube de palabras [Fig.1]:

Figura 1: Nube de Palabras más frecuentes

Como podemos observar palabras como “problema”, “cobertura”, “servicio”, “fácil”, “internet”, “lento” se muestran a primera vista, lo que nos da algunos indicios del contenido de las encuestas.

4.2 N-gramas más frecuentes

Luego de observar las palabras más frecuentes, estas no son muy concluyentes por si solas, por lo que es necesario analizar la secuencia de éstas, para tener un mejor contexto de lo que se busca expresar, por esto se realizó un análisis de bi-gramas con el objetivo de lograr rescatar la mayor cantidad de información de los textos estudiados. Para esto, se utiliza la probabilidad de predecir qué palabra será la siguiente dadas la probabilidad condicional de las palabras anteriores [12].

Bi-gramas

Al realizar el análisis de bi-gramas dentro de las opiniones estudiadas, se obtuvieron los siguientes resultados [Fig.2]:

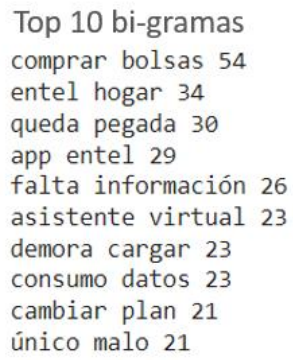

Figura 2: Top bi-gramas

Tri-gramas

Los tri-gramas más frecuentes obtenidos son los siguientes [Fig.3]:

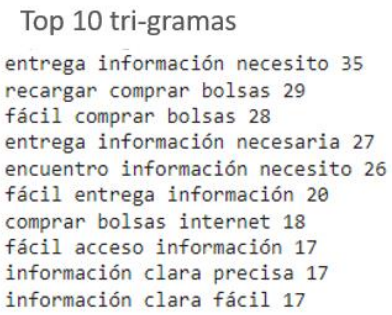

Figura 3: Top tri-gramas

4.3 Modelamiento de Tópicos

Con el objetivo de analizar más en detalle las opiniones de los usuarios, se realizó un modelamiento de tópicos de los temas presentes en el corpus [1], en este proceso iterativo se utilizaron 50 iteraciones, donde cada una de éstas pasa por el documento y reasigna los pesos de las palabras a uno de los tópicos hasta que se llega a un estado estable [2], luego de varias pruebas, se consideró que 2 tópicos es un número ideal para analizar este caso.

Para generar la visualización de tópicos se utilizó la librería “LDAvis” ya que nos proporciona una visión global de los temas y al mismo tiempo permite una inspección profunda de los términos por cada tópico [3].

En el panel de la izquierda, se visualizan los temas como círculos en el plano bidimensional, donde el área de cada uno representa la prevalencia general del tópico y la visualización de la derecha muestra la frecuencia del término de forma global y por tópico seleccionado. También cuenta con una métrica “λ”, que permite clasificar los términos según su relevancia, como primera vista se utilizó un “λ” de 0,6 ya que en general es el Valor óptimo sugerido [4]

Tópicos más destacados en general

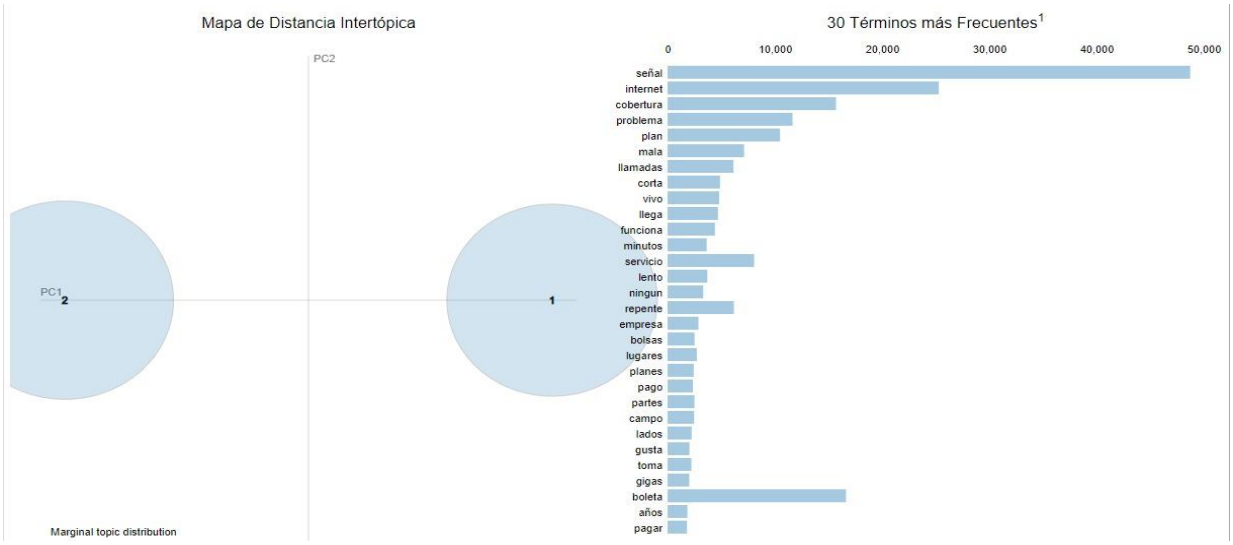

Figura 4: Modelamiento de tópicos

Observando la visualización de tópicos en general [Fig.4], podemos apreciar que la palabra “Señal” es el término más destacado, seguida por “Internet”, “cobertura” y “boleta”. También, de acuerdo con el mapa Inter tópico, observamos que los tópicos identificados, se encuentran relativamente bastante distantes.

Términos para tópico 1

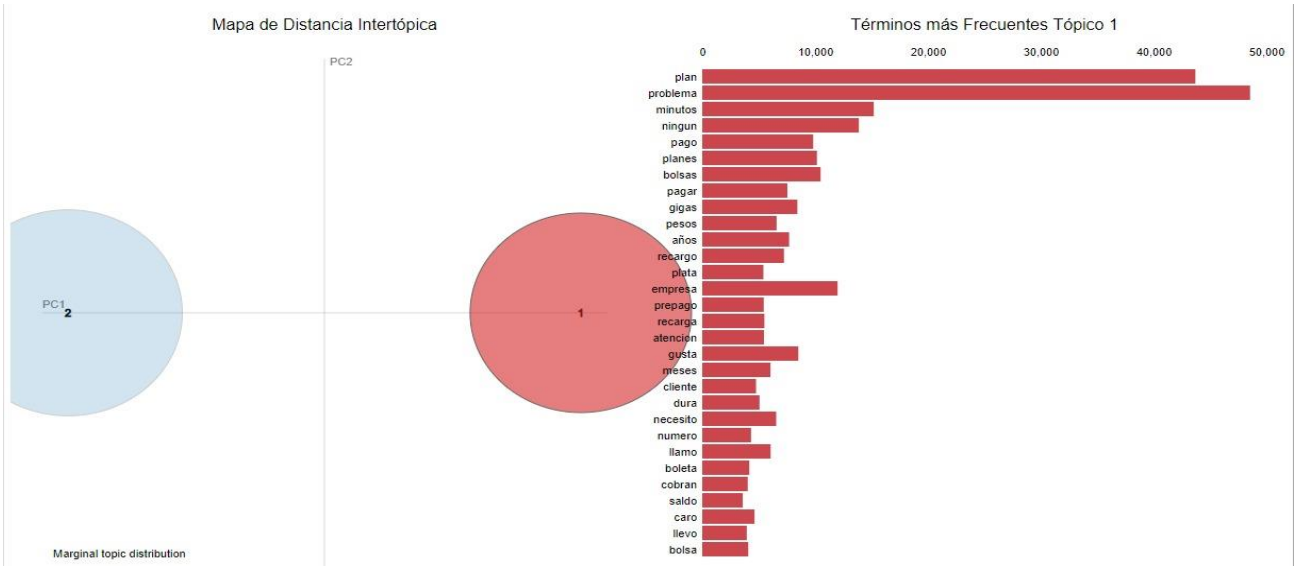

Figura 5: Términos tópico 1

Dentro del tópico 1 [Fig.5] los términos más relevantes son “plan”, “problema”, “minutos”, “empresa” y “pago”, por lo que se puede inferir que existen problemas con los pagos de boletas y planes, lo que nos da una idea del contexto del tópico 1.

Términos para tópico 2

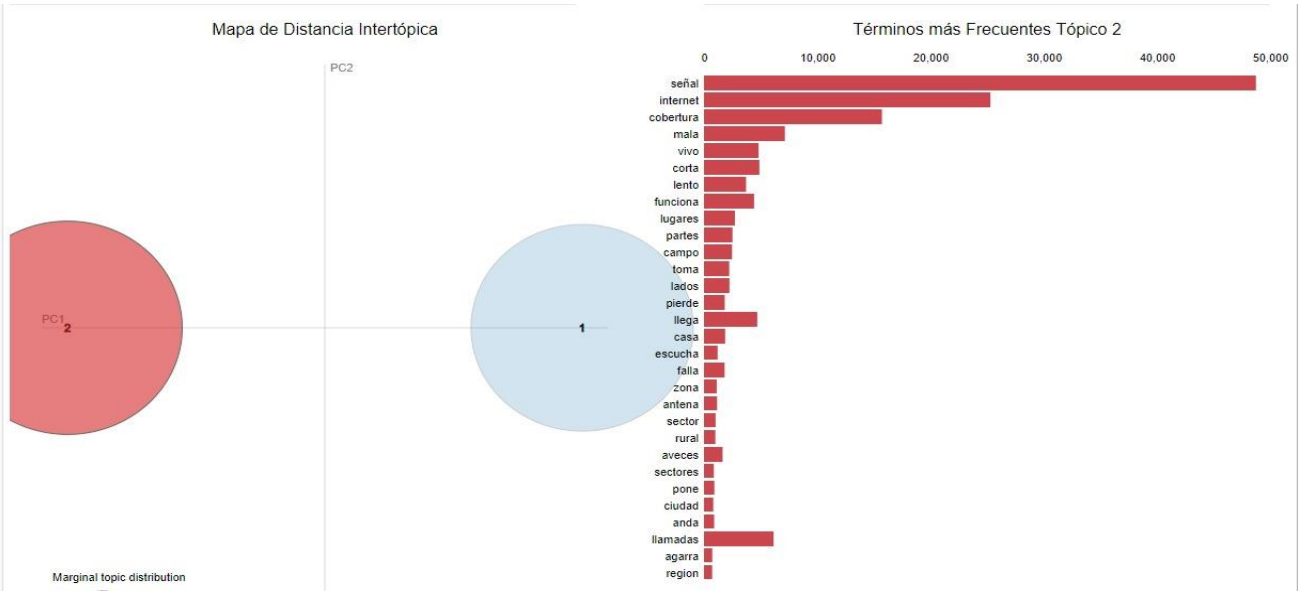

Figura 6: Términos tópico 2

En la visualización del tópico 2 [Fig.6], se aprecia que las 4 palabras clave de esta categoría son “señal”, “internet”, “cobertura”, “mala”, por lo que está claro que este tópico en general presenta insatisfacción en la cobertura y señal.

4.4 Análisis de Sentimientos

A pesar de ser un análisis comúnmente utilizado, se realizó un estudio de sentimientos ya que éste nos facilitará encontrar opiniones positivas o negativas para entender qué características, como funcionalidades, diseño, performance, entre otras están siendo bien o mal apreciadas por los usuarios, ayudando a entender la experiencia de estos [8].

Para realizar este análisis se utilizó la librería “vaderSentiment” [13].

Como distribución general de las opiniones, se obtuvieron los siguientes resultados [Fig.7]:

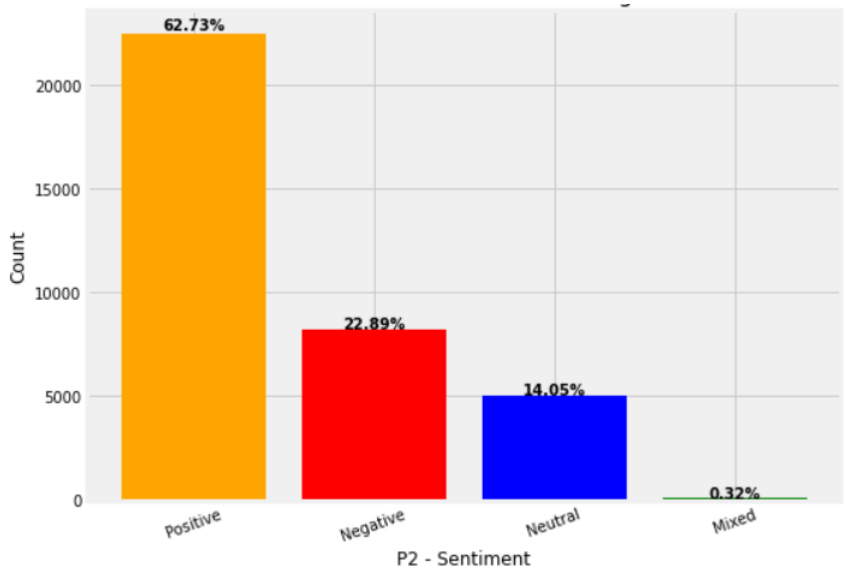

Figura 7: Distribución de sentimientos

De acuerdo con el gráfico, podemos observar que la mayor cantidad de las encuestas observadas presentan una opinión positiva (62,73%), mientras que sólo un 22,89% presenta un sentimiento negativo, por lo que podemos inferir que en general los usuarios están satisfechos con el uso de los canales digitales. También, se identificaron opiniones “neutrales” y “mixtas”, las cuales dentro de este contexto no son muy concluyentes.

N-gramas por sentimientos

Luego de realizar el análisis de sentimientos, es importante entender que “frases”, es este caso n-gramas se expresan con un motivo positivo y negativo, por esto, se generaron n-gramas por sentimientos, encontrando los siguientes resultados:

Bi-gramas por sentimiento

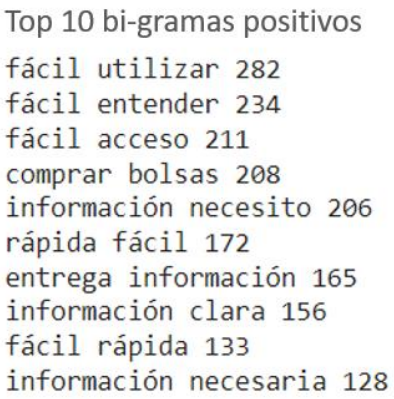

Figura 8

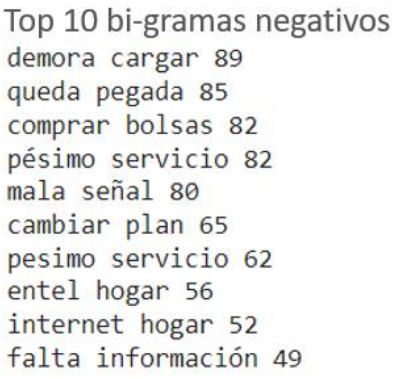

Figura 9

Como observamos en la Figura 8, los tres bi-gramas más frecuentes expresan una facilidad para acceder, entender y utilizar los canales, lo que nos indica que desde el punto de vista de la “User Experience y User Interface” (UX/UI), cumplen con los requerimientos de los usuarios. Por otro lado, analizando los bi-gramas negativos [Fig.9], estos nos expresan temas como “demora en la carga”, “queda pegada”, “comprar bolsas”, por lo que se puede inferir que existen problemas de performance o latencia, que tiene a los usuarios disconformes, por lo que esta evidencia es relevante para informar al área de desarrollo, con el objetivo de priorizar dentro de su “roadmap” una mejora en rendimiento de los canales.

Análisis de Sentimientos por Funcionalidad

Como segunda vista, se realizó el análisis aperturado por tipo de funcionalidad, esto con el objetivo de ver de forma más detallada cómo perciben las distintas funcionalidades los usuarios.

Nuevamente se observa que la gran cantidad de las encuestas respecto a las funcionalidades, presentan un sentimiento positivo [Fig.10], pero también se puede apreciar que el punto de "performance", "Problemas de señal" y "Problemas para recargar" muestran un sentimiento negativo, lo cual concuerda con lo visto anteriormente en el modelamiento de tópicos.

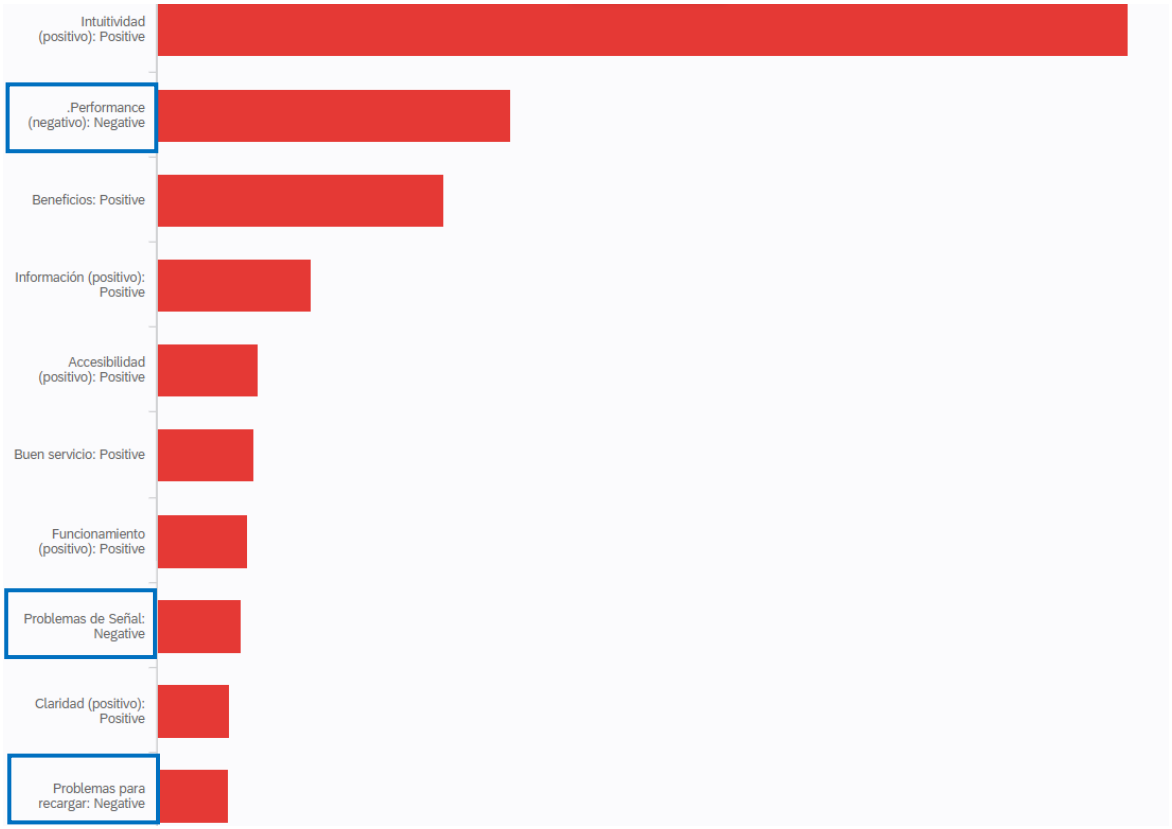

Figura 10: Sentimientos por funcionalidad

Como forma de potenciar este tipo de análisis, la generación de un Dashboard como herramienta para identificar tendencias por periodos o correlacionar con distintas variables sería un gran valor para el negocio y facilitar el estudio de la satisfacción de los clientes.

Sentimiento por mercado

Los mercados se refieren al servicio ofrecido por la compañía, dentro del dataset se encuentran 8 servicios, especificados a continuación:

Fibra: Servicio de fibra óptica
Inalámbrico: Internet inalámbrico
HG: TV Streaming
ILM: TV Inalámbrica
SS: Plan de suscripción móvil
CC: Plan de cuenta controlada móvil
PP: Servicio de prepago
PP plus: Servicio de prepago Plus

Otra vista que se analizó fue el sentimiento expresado por los usuarios con respecto al mercado al que pertenecen, es decir, al tipo de servicio contratado, donde se obtuvieron los siguientes resultados [Fig.11]:

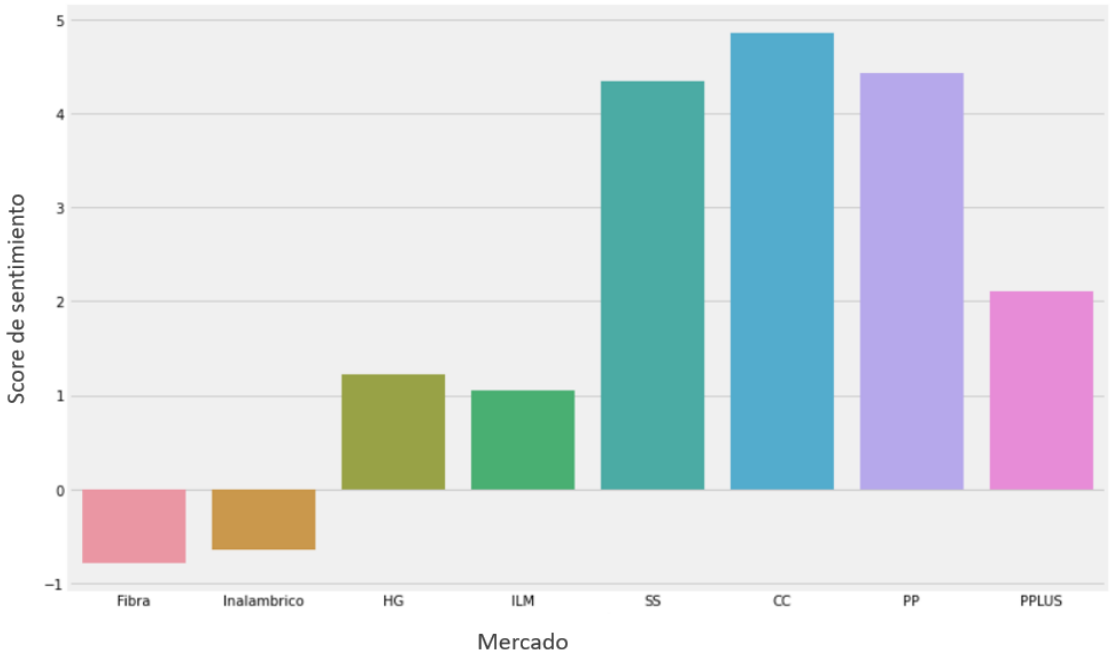

Figura 11: Sentimiento por mercado

Estudiando el gráfico, se aprecia claramente que los servicios de Fibra e Inalámbrico presentan un sentimiento negativo percibido por los usuarios, lo cual es un insight de gran valor para el negocio. Esto ayudará a dar foco a estos dos servicios en específicos para generar un plan de mejora en el corto o mediano plazo.

Correlación "Duración encuesta vs Sentimiento"

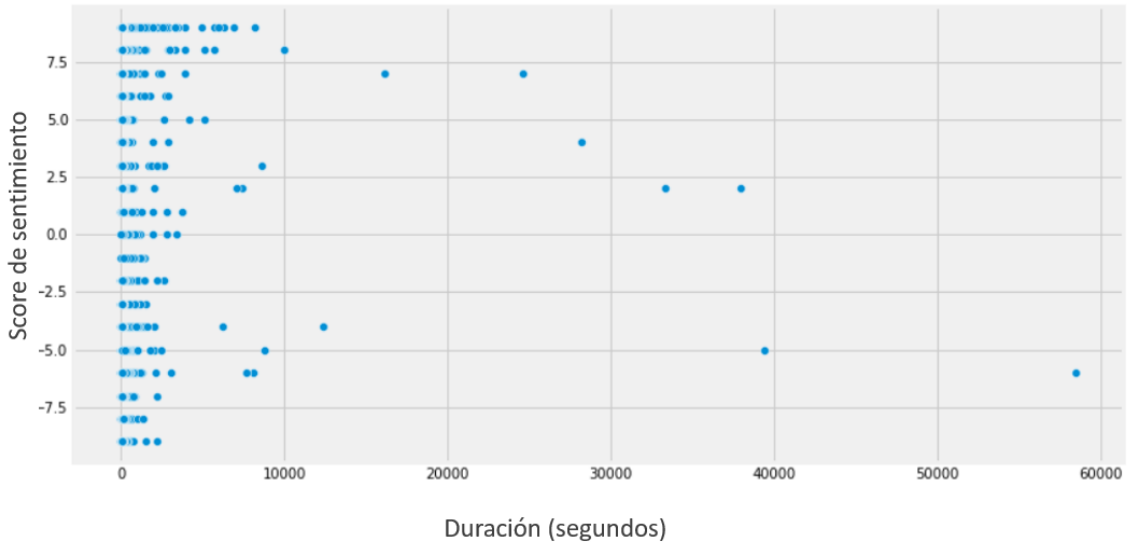

Figura 12: Gráfico de correlación

Con el objetivo de comprender si la variable "duración", es decir, el tiempo que demora el usuario (en segundos) esta correlacionado con el score de sentimiento, se generó un "Scatterplot" [Fig.12], que es la visualización clásica que nos ayuda a responder esta interrogante [14]. Como observamos en el gráfico, no se identificó un patrón de correlación entre estas 2 variables, por lo que podemos determinar que el tiempo que le toma a un usuario redactar su opinión, no afecta el Score de sentimiento.

Score de Sentimientos por día

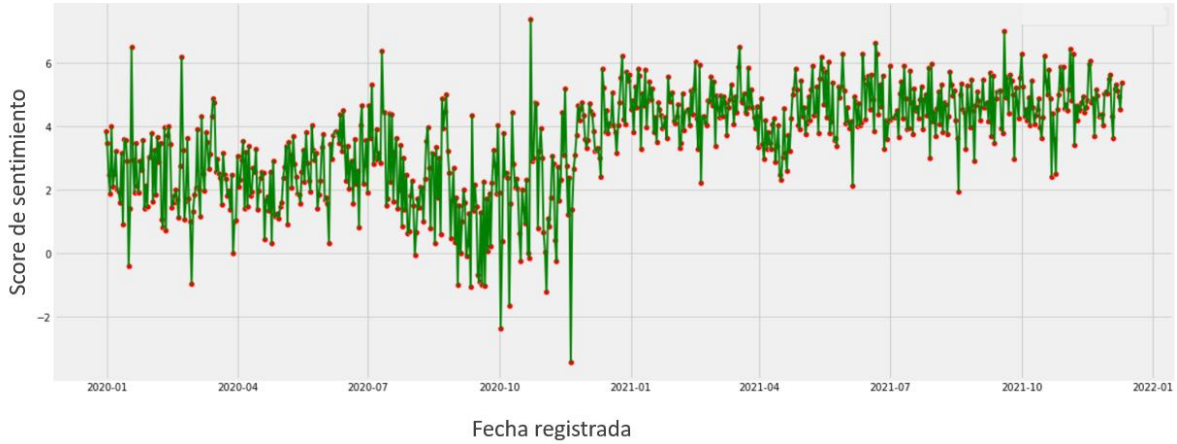

Figura 13: Sentimientos por fecha

Para estudiar la tendencia diaria del score de sentimientos, se utilizó un gráfico de líneas ya que nos ayuda a entender una tendencia durante el tiempo. En la figura 13, podemos observar una tendencia en general estable, pero con días peak (positivos y negativos), donde en noviembre de 2020, vemos una gran disminución en la satisfacción, el peak más bajo desde enero 2020, el cual se explica por una falla masiva que ocurrió dentro del canal App, por lo que la gran mayoría de encuestas recibidas aquel día fueron reclamos. Posterior a esta falla, observamos que, desde diciembre 2020 el sentimiento positivo de los usuarios logras llegar a un “piso”, apreciando una tendencia general mayor a la vista en el año 2020, esto debido a mejoras de rediseño del canal App, mejorando considerablemente el “Look & Feel” de esta.

4.5 Clasificación de Texto

La clasificación de texto se realizará con el objetivo de solucionar un problema de negocio, para esto se utilizará el modelo de lenguaje “Bidirectional Encoder Representations from Transformers” (BERT), técnica basada en redes neuronales para el preentrenamiento del procesamiento de lenguaje natural (PLN) desarrollada por Google [7]. Dentro del dataset se encuentra la variable “opinión”, las cuales se clasifican en 8 distintas categorías que podemos observar en el siguiente recuadro [Fig.14].

Opinión	Categoría
Lento el sistema, no se notan para nada todos los megas que tengo contratado	**Servicio**
Muy mal diseñado, para comunsarse directamente con un ejectuivo es casi imposible, para buscar información es visualmente básico	**UX**
Necesito un historico de pagos del año 2020 y no es posible obtenerlo por este medio	**Funcionalidad**
Me están cobrando cuentas falsas, he ido a las sucursales y via web y telefono no las sacan, son unos ladrones!	**Otro**
No cargan las pestañas, por ejemplo necesito descargar una boleta y no aparece	**Error reproducible**
Se encuentra la información en forma fácil y rápida	**Positivo**
En dos ocasiones he tenido problemas para cancelar la cuenta por este medio	**Error no reproducible**
Nada menos que a veces no me reconoce como cliente y lo soy desde hace más de una decena de años.	**Revisar**

Figura 14: Tabla de opiniones

El problema es que actualmente esta clasificación de literales (de los cuales existen miles) la realiza una persona de acuerdo con su criterio y experiencia, por lo que existen un problema de capacidad para poder abordarlos todos, además del tiempo utilizado en realizar esta clasificación.

Es de gran importancia tener este feedback de los clientes clasificado para priorizar cada opinión, ya que un problema sobre el diseño (UX) no es tan importante solucionar en el corto plazo como un error masivo (Error reproducible), por lo que es crítico identificar este tipo de errores en el menor tiempo posible, ya que este debe ser solucionado de manera inmediata porque quita operatividad al canal digital, lo que hace reducir considerablemente la satisfacción de los usuarios.

Modelo de Lenguaje

Para la realización del modelo de clasificación, se utilizará una dataset de entrenamiento el cual contiene las opiniones ya categorizadas, estos labels fueron generados por un analista que lee las encuestas redactadas por los usuarios y según su experiencia las categoriza.

Preprocesamiento de los datos

Los modelos de Machine Learning no se utilizan con textos sin formato, es necesario convertir el texto en números de través de la codificación, y para el caso específico de BERT, se requiere más procesamiento ya que es un modelo más complejo.

Como primera etapa para la generación del modelo, se codificaron las distintas categorías en números enteros, del 0 al 7. Luego, se dividió el dataset en tres partes, set de entrenamiento, set de validación y set de testing.

Para el entrenamiento, se utilizó la GPU para realizar el trabajo de cómputo por temas de eficiencia y velocidad [15], para el setup se utilizó la librería torch a la que se le indicó el uso de la GPU.

Se utilizaron Transformers “Huggingface” para generar el modelo de clasificación de texto BERT.

Como parte del procesamiento de texto, se transformó el texto sin formatos en números, para lograr esta tarea, se utilizó BERT tokenizer.

Como modelo pre entrenado, se consideró adecuado utilizar “dccuchile / bert-base-spanish-wwm-cased”, este es un BERT en español, entrenado sobre un gran corpus en español, tiene un tamaño similar a un BERT-base y fue entrenado con la técnica “Whole Word Masking” [16]. Luego de la tokenización, se agregaron tokens especiales utilizando el método “encode_plus”, considerando los siguientes tokens especiales [17]:

[SEP]: Para marcar el final de una oración

[CLS]: Se agrega este token al inicio de cada oración, para que BERT entienda que se está clasificando

[PAD]: Para rellenar las oraciones que no cumplen con la longitud requerida.

Con esto, se logró que los tokens se almacenen en un tensor y se rellenan con una longitud de 64 como podemos observar en la siguiente tabla [Fig.15]:

```
64
tensor([    4,  6592,  1970,  1108,  6092,  1008,  8161, 14429,  1051,  1858,
         7620,  1038,  1445,  1728,  8258,  1975,  2669, 11435,  1074,  1084,
         1051, 12072, 19757,  1019,  1038,  1084,  5332,  1038,  1084,  1423,
        28021,  3246,   981,  1008,  2573,  1009,  1009,     5,     1,     1,
            1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
            1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
            1,     1,     1,     1])
```

Figura 15: Tensores

Podemos apreciar la identificación de los tokens, observamos que, si la secuencia de texto es menor que la longitud máxima de la secuencia, esos tokens adicionales se completan a 1, el identificador de token 4 es el token [CLS], el 5 es el [SEP] y el resto son los tokens reales.

Escogiendo el largo de la secuencia: BERT trabaja con secuencias de longitud fija, por lo que usaremos una estrategia simple para determinar la longitud máxima, donde se guardará la longitud del token de cada texto:

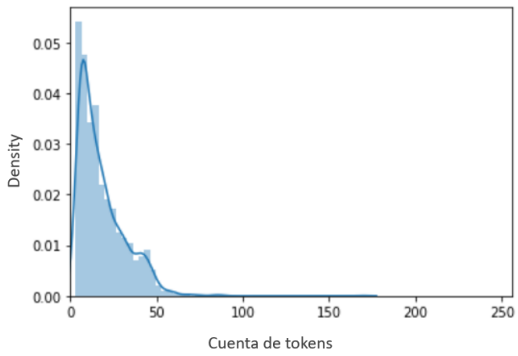

Figura 16: Distribución por tokens

En el gráfico de distribución de densidad, observamos que la mayoría de los textos contienen menos de 60 tokens, pero para asegurarnos, se seleccionará una longitud máxima de 80.

Al tener determinada la longitud máxima, se utilizará el modelo "BertFroSequenceClassification", este es el modelo normal con una capa lineal única, agregada en la parte superior que nos permitirá clasificar las oraciones [18]. A medida que alimentamos el modelo con los datos de entrada, todo el modelo BERT previamente entrenado y la capa adicional de clasificación se entrena para la tarea específica, tomando provecho de la "transferencia de aprendizaje" [19]. En la siguiente imagen [Fig.17], podemos observar el modelo generado:

```
from transformers import BertForSequenceClassification

#Cargando el modelo BERT pre entrenado
model = BertForSequenceClassification.from_pretrained(
    "dccuchile/bert-base-spanish-wwm-cased", # Utilizando el modelo pre-entrenado de bert-base-spanish-wwm-cased
    num_clases = 8, # El número de clases objetivo = 8
)

# Especificando a pytorch que ejecute este modelo en la GPU.
model = model.to(device)
```

Figura 17: Modelo generado

Para el propósito del tuneo de parámetros, se utilizaron los recomendados por los autores de BERT [20], definiendo los siguientes parámetros [Fig.18]:

```
# Parámetros de entrenamiento
batch_size = 32
learning_rate = 5e-5
epochs = 3
```

Figura 18: Parámetros de tuneo

Teniendo el modelo construido, se tokenizaron todos los datos utilizando Pytorch, creando un conjunto de datos de prueba y entrenamiento, se definió una clase para crear un conjunto de datos de pytorch, la cual incluye el tokenizador para convertir el texto en tokens.

También se definió una función llamada "data_loader" para cargar los datos en batches, tomando

características, etiquetas, tokenizador, longitud máxima se la secuencia y tamaño del batch como entradas.

Entrenamiento

Para el entrenamiento se utilizarán “data loaders” para cargar los datos en batches. Se definieron optimizadores y tasas de aprendizajes, utilizando los siguientes parámetros [Fig.19]:

```
# Creando optimizador
from transformers import AdamW
optimizador = AdamW(model.parameters(),
                  lr = learning_rate
                )

# Cree el programador de tasas de aprendizaje.
from transformers import get_linear_schedule_with_warmup

# El número total de pasos de entrenamiento es:
# [número de lotes] x [número de épocas].
pasos = len(train_DataLoader) * epochs

scheduler = get_linear_schedule_with_warmup(optimizer,
                                            num_warmup_steps = 0,
                                            num_training_steps = total_steps)
```

Figura 19: Parámetros de entrenamiento

Luego de tener el modelo construido y los parámetros definidos se realizó el entrenamiento obteniendo los siguientes resultados por cada Epoch:

Epoch 1:

```
======== Epoch 1 / 3 ========
Training...
  Batch    10  of     74.    Elapsed: 0:00:09.
  Batch    20  of     74.    Elapsed: 0:00:17.
  Batch    30  of     74.    Elapsed: 0:00:25.
  Batch    40  of     74.    Elapsed: 0:00:33.
  Batch    50  of     74.    Elapsed: 0:00:41.
  Batch    60  of     74.    Elapsed: 0:00:50.
  Batch    70  of     74.    Elapsed: 0:00:58.

  Average training loss: 1.47
  Training epcoh took: 0:01:01

Running Validation...
  Accuracy: 0.65
  Validation Loss: 1.03
  Validation took: 0:00:02
```

Epoch 2:

```
======== Epoch 2 / 3 ========
Training...
  Batch    10  of     74.    Elapsed: 0:00:09.
  Batch    20  of     74.    Elapsed: 0:00:17.
  Batch    30  of     74.    Elapsed: 0:00:25.
  Batch    40  of     74.    Elapsed: 0:00:34.
  Batch    50  of     74.    Elapsed: 0:00:42.
  Batch    60  of     74.    Elapsed: 0:00:50.
  Batch    70  of     74.    Elapsed: 0:00:59.

  Average training loss: 0.98
  Training epcoh took: 0:01:02

Running Validation...
  Accuracy: 0.72
  Validation Loss: 0.85
  Validation took: 0:00:02
```

Epoch 3:

```
======== Epoch 3 / 3 ========
Training...
  Batch    10  of     74.    Elapsed: 0:00:09.
  Batch    20  of     74.    Elapsed: 0:00:17.
  Batch    30  of     74.    Elapsed: 0:00:25.
  Batch    40  of     74.    Elapsed: 0:00:34.
  Batch    50  of     74.    Elapsed: 0:00:42.
  Batch    60  of     74.    Elapsed: 0:00:50.
  Batch    70  of     74.    Elapsed: 0:00:59.

  Average training loss: 0.67
  Training epcoh took: 0:01:01

Running Validation...
  Accuracy: 0.71
  Validation Loss: 0.85
  Validation took: 0:00:02
```

A modo de resumen, podemos observar la siguiente tabla con las métricas claves obtenidas [Fig.20]:

epoch	Training Loss	Valid. Loss	Valid. Accur.	Training Time	Validation Time
1	1.47	1.03	0.65	0:01:01	0:00:02
2	0.98	0.85	0.72	0:01:02	0:00:02
3	0.67	0.85	0.71	0:01:01	0:00:02

Figura 20: Métricas del modelo

También se consideró importante analizar el sobreajuste del modelo, por lo que la siguiente figura [Fig.21] muestra las métricas “Validation loss” y “Training loss”.

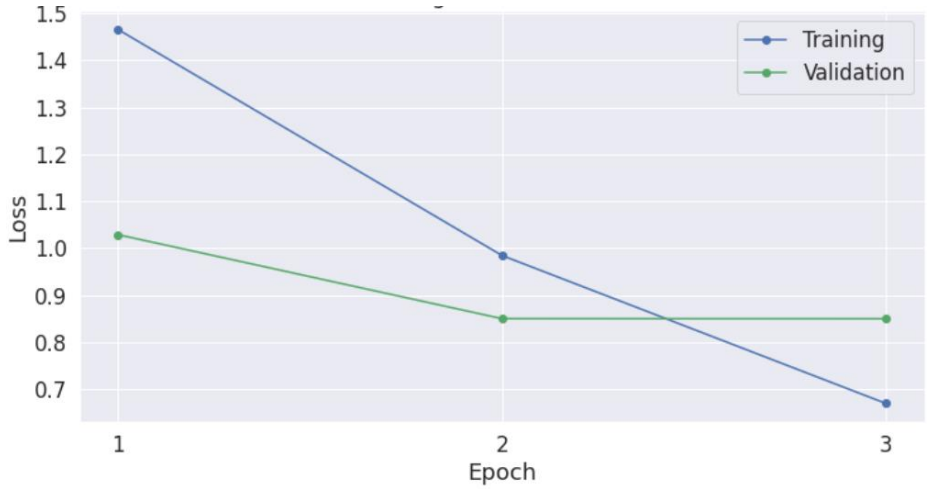

Figura 21: Análisis de sobreajuste

Analizando el gráfico, podemos observar que el modelo comienza a sobreajustarse en medio de las 2 Epochs, por lo que se detuvo el entrenamiento en la tercera Epoch [21].

Para estudiar los resultados en mayor detalle, se generó una matriz de confusión [Fig.22], que nos permitirá observar de manera mas fina, los errores y aciertos del modelo generado.

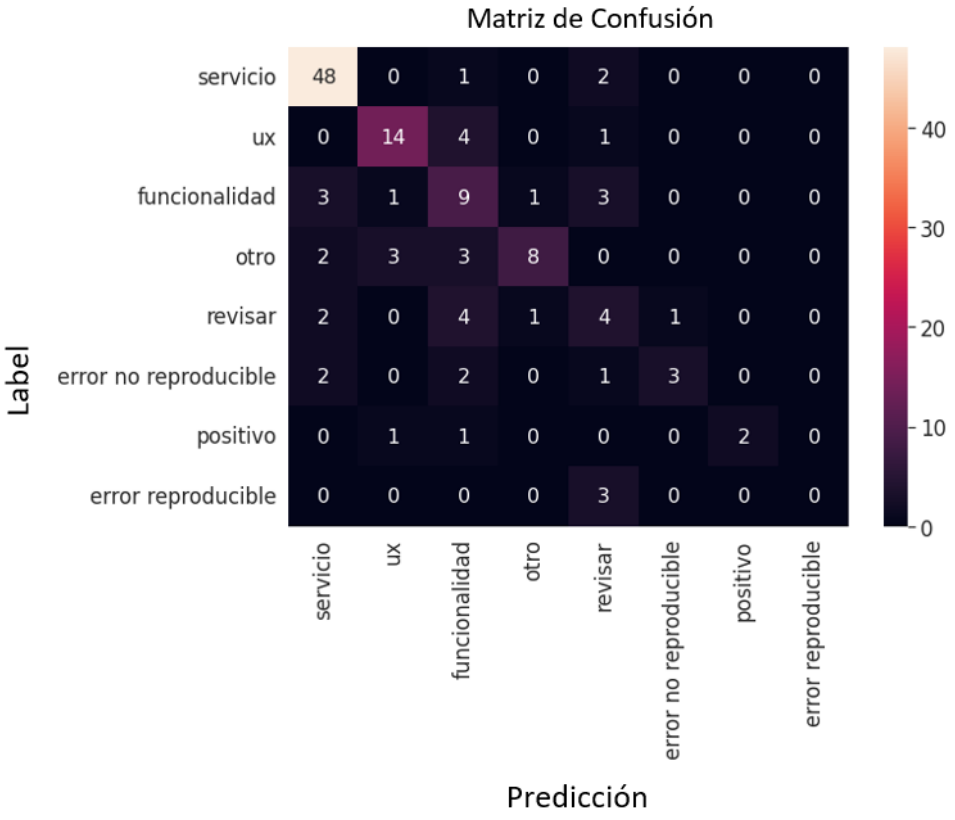

Figura 22: Matriz de confusión

De acuerdo con la matriz de confusión generada, podemos observar que los elementos representados en la diagonal, son las clases predichas correctamente y el resto de los elementos son las predicciones incorrectas. Se observa que, de 130 muestras de prueba, el modelo predijo 88 muestras correctamente y clasificó incorrectamente las 42 muestras restantes. También se generó un reporte de clasificación con las métricas claves que nos ayudan comprender el rendimiento del modelo, obteniendo los siguientes resultados [Fig.23]:

	precision	recall	f1-score	support
servicio	0.84	0.94	0.89	51
ux	0.74	0.74	0.74	19
funcionalidad	0.38	0.53	0.44	17
otro	0.80	0.50	0.62	16
revisar	0.29	0.33	0.31	12
error no reproducible	0.75	0.38	0.50	8
positivo	1.00	0.50	0.67	4
error reproducible	0.00	0.00	0.00	3
accuracy			0.68	130
macro avg	0.60	0.49	0.52	130
weighted avg	0.69	0.68	0.67	130

Figura 23: Métricas de performance

Conclusiones del modelo

El modelo predice bien las clases “servicio” y “ux” principalmente, considerando como métrica más relevante el f1-score, ya que pondera la precision y el recall. Esto se debe principalmente a que eran las clases con mayor presencia dentro del dataset de entrenamiento en comparación a las otras clases. Cabe recalcar que en general el modelo obtuvo un promedio ponderado de 67% y un accuracy de 68%, lo cual es aceptable. El rendimiento del modelo se puede mejorar aún más agregando más datos para equilibrar la distribución del dataset.

5. Conclusiones

Como conclusión de acuerdo con los resultados obtenidos se logró la obtención de Insights de valor para el negocio a través de los análisis propuestos con técnicas de procesamiento de lenguaje natural, los cuales nos permitirán identificar falencias y oportunidades de mejora de forma más rápida.

Respecto al preprocesamiento de texto, este se realizó de manera exitosa a pesar de que los textos se encontraran con faltas de ortografías y redactados en general con un lenguaje coloquial característico chileno, donde se obtuvieron resultados lógicos y de valor.

Con respecto al modelamiento de tópicos, luego de varias pruebas, se obtuvieron sólo dos tópicos distintos y coherentes, basados en temas de cobertura, señal (tópico 1) y pagos, boleta (tópico 2), esto nos entrega una vista general de lo que se habla en las encuestas, pero en general, no entrega mucho valor para el negocio.

Por otro lado, el análisis de sentimientos fue un punto muy relevante del proyecto, ya que nos entregó distintos Insights, con respecto a lo que se está haciendo bien y mal dentro de los servicios desde el punto de vista de los usuarios. Se demostró que gran parte de las encuestas (62,73%) presentan una opinión positiva de los canales digitales, también nos permitió identificar oraciones claves y frecuentes encontradas en los textos. Otro punto importante, fue que se logró identificar que funcionalidades específicas son las peor evaluadas por los usuarios, como la performance del canal, problemas de señal y problemas para realizar recargas, lo cual es un gran aporte para los Stakeholders de los

canales digitales. También, nos permitió identificar qué mercados o servicios son los mejor y peor evaluados dentro de los canales digitales, lo que permite realizar foco en estos (Fibra e Inalámbrico) optimizando recursos, acelerando las mejoras, y al mismo tiempo aprender de los servicios mejores evaluados para aprender de estos y transferir sus características a dichos canales.

Con respecto a la clasificación de texto, se obtuvieron resultados aceptables, pero con mucha oportunidad de mejora, obteniendo un 68% de Accuracy, esto nos puede ayudar a clasificar las encuestas en las categorías valoradas por la compañía, ahorrando tiempo a las personas que realizan esta clasificación de forma manual y aumentando la cantidad de opiniones categorizadas, ya que, al realizarse de forma manual, la capacidad es limitada.

Por otro lado, como limitaciones en la clasificación de texto se consideró que la cantidad de opiniones categorizadas era muy baja para obtener los mejores resultados, por lo que se propone realizar el entrenamiento con una mayor cantidad de labels y categorías mejora balanceadas, lo que aumentará la performance del modelo de clasificación generado.

Como trabajo futuro, se propone la construcción de un Dashboard como herramienta que nos facilite la visualización de estos resultados para estudiarlos en mayor detalle, hacer más accesible los datos y acelerar la toma de decisiones.

Revisión Bibliográfica

1. Purver, M.: Topic Segmentation. Spoken Language Understanding: Systems forExtracting Semantic Information from Speech, pp. 291–317 (2011)

2. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022

3. Carson Sievert, A method for visualizing and interpreting topics, pp 64

4. Sozzi Alessandra, Visualising topics as distributions over words. http://bl.ocks.org/AlessandraSozzi/raw/ce1ace56e4aed6f2d614ae2243aab5a5/

5. Kao Anne, Poteet Stephen, Natural Language Processing and Text Mining.

6. Pla Ferran, Hurtado Lluís, Sentiment Analysis in Twitter for Spanish.

7. Devlin Jacob, Chang Ming-Wei, Lee Kenton, Toutanova Kristina (2018), BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding :3-11.

8. Jussila Jari, Vuori Vila, Helander Nina, Okkonen Jussi, Reliability and Perceived Salue of Sentiment analysis for twitter data.

9. Ozdagoglu Guzin, Kapucugil Aysun, Fuat Ayhan, Topic Modelling-based Decision Framework for Analysing Digital Voice of the Customer.

10. Karim Mirsa, Das Smija, Sentiment Analysis on Textual Reviews.

11. Prabhu Sumanth, Mohamed Moosa, Misra Hemant, "Multi-class Text Classification using BERT.

12. Jurafsky Daniel, Martin James (2021), Speech and Language Processing, N-gram language models, Chapter 3.

13. Ray Taylor, Anreddy Sujan (2021), Using vanderSentiment to intuitively predict the sentiment of Social Media. https://ds3.ssrc.msstate.edu/2021/01/14/using-vadersentiment-to-intuitively-predict-the-sentiment-of-social-media-posts/

14. Rensink Ronald (2017), The nature of correlation perception in scatterplots: 776-797.

15. Kayid Amr, Khaled Yasmeen (2018), Performance of CPUs/GPUs for Deep Learning workloads.

16. Cañete José, Chaperon Gabriel, Fuentes Rodrigo, Ho jou-Hui, Kang Hojin, Pérez Jorge (2020), Spanish pre-trained BERT model and Evaluation data.

17. Webster Jonathan, Kit Chunyu (1992), Tokenization as the Initial Phase in NLP.

18. Bert for Sequence Classification. https://huggingface.co/transformers/v3.0.2/model_doc/bert.html#bertforsequenceclassification

19. Weiss Karl, Wang DingDing, Khoshgoftaar Thagi (2016), A survey on Transfer

Learning: 3-7.

20. Devlin Jacob, Chang Ming-Wei, Lee Kenton, Toutanova Kristina (2018), BERT: Pre-training of Deep Bidirectional Transformers for Language understanding: 13-14, apéndice A.3.

21.Ying Xue (2019), An Overview of Overfitting and its Solutions.