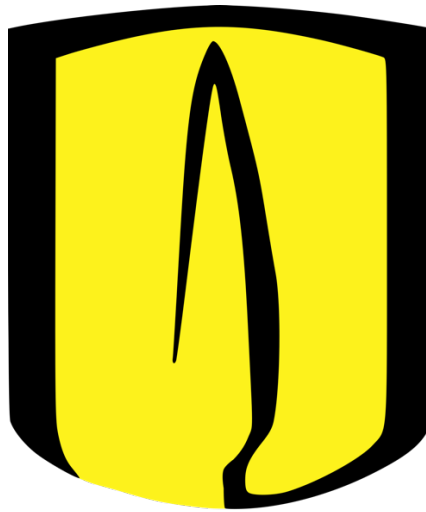


## **Documento – Proyecto Analítica de textos – Etapa 2**



### **Grupo G30: Tamarindo**

Jesús Jiménez – 202020431

Juan Camilo Bonet – 202022466

Thais Tamaio – 202022213

Universidad de los Andes  
Ingeniería de Sistemas y Computación  
Inteligencia de negocios

## Tabla de contenidos

<b>1.</b>	<b><i>Creación del pipeline.....</i></b>	<b>2</b>
<b>2.</b>	<b><i>Desarrollo de la aplicación.....</i></b>	<b>3</b>
<b>3.</b>	<b><i>Utilidad de la aplicación en un negocio .....</i></b>	<b>4</b>
<b>4.</b>	<b><i>Mejoras obtenidas con la ayuda del experto de estadística asignado .....</i></b>	<b>5</b>

### 1. Creación del pipeline

Como primer paso para el desarrollo de nuestra aplicación, decidimos implementar un pipeline para poder estandarizar y automatizar el proceso de limpieza y transformación de los datos, junto con la predicción de sentimientos utilizando el modelo entrenado.

Para implementar el pipeline se tiene primero una clase "Limpieza" que hereda de dos clases: "BaseEstimator" y "TransformerMixin". Esta clase se utiliza para limpiar los datos de un DataFrame de Pandas que contiene opiniones o comentarios en español.

En el método "init" de la clase se inicializa un conjunto de stopwords en español que se utilizará más adelante para eliminar las palabras comunes que no aportan información relevante en el análisis. Por otro lado, en el método "fit" no se realiza ninguna tarea y simplemente devuelve el objeto "self", lo cual es una práctica común en Scikit-learn para mantener la coherencia del pipeline. Además, en el método "transform" se llama al método "preprocess", que es donde se realiza la limpieza de los datos. El método "preprocess" recibe un DataFrame de Pandas y realiza una serie de tareas de limpieza en el texto de cada comentario, que se detallan a continuación:

1. **Remove los caracteres ASCII:** utilizando la librería "unicodedata", se eliminan los caracteres especiales que no son parte del alfabeto español.
2. **Cambiar las mayúsculas por minúsculas:** se convierten todas las letras a minúsculas para homogeneizar el texto.
3. **Eliminar los signos de puntuación:** utilizando la librería "string", se eliminan los signos de puntuación del texto.
4. **Reemplazar los números por su equivalente en palabras:** utilizando la librería "num2words", se reemplazan los números que aparecen en el texto por su equivalente en palabras en español.

5. Eliminar las stopwords: utilizando la librería "nltk", se eliminan las palabras comunes que no aportan información relevante en el análisis.
6. Finalmente, el método "transform" devuelve el DataFrame limpio y preparado para ser utilizado en el siguiente paso del pipeline.

Por otro lado, en el archivo Pipeline.ipynb se entrena y se exporta el pipeline como tal. Primero, se importa la clase "Limpieza" que se define en otro archivo y que se utiliza para limpiar los datos de texto. Luego, se define el pipeline que consta de tres etapas:

1. "preprocessor": la instancia de la clase "Limpieza" que se encarga de limpiar el texto.
2. "vectorizer": un vectorizador HashingVectorizer que convierte los comentarios limpios en vectores numéricos que se pueden utilizar como entrada para el modelo de clasificación.
3. "classifier": un clasificador RandomForestClassifier que se encarga de predecir la polaridad del sentimiento de los comentarios.

A continuación, se carga el conjunto de datos de comentarios de películas y se divide en conjuntos de entrenamiento y prueba utilizando la función "train\_test\_split". Luego, se ajusta el pipeline con los datos de entrenamiento mediante el método "fit" y se utiliza el método "predict" para hacer predicciones de sentimientos sobre los datos de entrenamiento y prueba. Finalmente, se guarda el joblib que contiene el pipeline entrenado.

## 2. Desarrollo de la aplicación

Para el back-end de la aplicación usamos python con el framework FastAPI. Este framework es conocido por su facilidad de uso, velocidad de ejecución y su robustez. FastAPI usa el servidor web Uvicorn para correr y exponer el API. La aplicación es capaz de recibir archivos CSV que contienen reseñas de películas en español, y también una única reseña en español. Luego, utiliza un modelo de aprendizaje automático previamente entrenado (que se encuentra en el archivo **prediction\_model.py**) para predecir si cada reseña es positiva o negativa.

El primer endpoint **make\_prediction\_file()** utiliza el método HTTP POST y recibe como entrada un archivo **UploadFile** que contiene múltiples reseñas en español. El archivo es leído en memoria, transformado en un objeto pandas DataFrame, y luego

se utiliza el modelo para hacer predicciones en el DataFrame completo. El resultado es una lista con las predicciones de cada reseña.

El segundo endpoint **make\_prediction()** también utiliza el método HTTP POST y recibe como entrada un objeto **ReviewModel** que representa una sola reseña en español. La reseña es transformada en un objeto pandas DataFrame, se utiliza el modelo para hacer la predicción y el resultado es una lista con la predicción de cada reseña.

Por el lado del front, usamos el framework Angular. Este framework permite la creación de aplicaciones de una sola página, con estructura robusta y escalable. Uno de los principales beneficios de utilizar Angular en el desarrollo del front-end de una aplicación es la capacidad de crear una experiencia de usuario interactiva y dinámica, lo cual lo hace una buena opción para nuestra aplicación.

La página se divide en tres componentes, el de cargar una sola reseña, el de cargar más de una reseña y el de listar los resultados de las reseñas cargadas. En la página de cargar una reseña, hay un campo de texto donde el usuario puede escribir su reseña y al darle click a publicar nos redirige a la ruta de listar, donde podemos ver el sentimiento predicho por el modelo. Por detrás, se está mandando el texto a la ruta que definimos en el back como **“/predictone”**.

Por otro lado, tenemos el componente de cargar varias reseñas. Aquí, encontramos un campo donde podemos cargar un archivo. Al recibir la reseña, se listan todos los resultados en la ruta de listar, tal como el componente anterior. Para este componente, se manda el archivo a la ruta del back-end definida como **“/predict”**. Por el lado del back nos encargamos de verificar que el archivo tenga la estructura correcta, en caso de que sea incorrecta, le informamos al usuario que hubo un error.

Para el despliegue, decidimos usar Google Cloud Platform. Creamos dos máquinas virtuales, una para el front y otra para el back con 4 y 16 GB de RAM respectivamente. Esto se debe a que la máquina del back-end tiene que hacer computaciones más intensivas, ya que tiene que analizar todas las reseñas que lleguen desde el front-end.

### 3. Utilidad de la aplicación en un negocio

El usuario/rol de la organización que utilizará esta aplicación podría ser una empresa de streaming que esté interesada en conocer la opinión de los usuarios sobre las

películas en su plataforma. Al utilizar esta aplicación, la empresa podrá obtener información valiosa sobre las películas que tienen reseñas positivas y negativas, lo que les permitirá tomar decisiones informadas sobre qué películas mantener en su plataforma y cuáles eliminar.

La conexión entre esta aplicación y el proceso de negocio que apoyará es que la empresa de streaming utilizará la información obtenida a través de la aplicación para tomar decisiones importantes sobre su oferta de películas. Esta información les permitirá identificar las películas que atraen a los espectadores y las que no, y hacer ajustes en consecuencia.

La importancia de esta aplicación para la empresa de streaming radica en que les permitirá tomar decisiones informadas sobre su oferta de películas, lo que a su vez podría mejorar la satisfacción del usuario y aumentar la retención de los clientes. Además, la automatización del proceso mediante el modelo de machine learning hace que la tarea de clasificación sea más rápida y precisa, lo que permite a la empresa tomar decisiones más rápidas y eficientes. En resumen, la existencia de esta aplicación puede tener un impacto significativo en la estrategia de la empresa y en su éxito a largo plazo.

#### 4. Mejoras obtenidas con la ayuda del experto de estadística asignado

Para esta segunda etapa del proyecto, contactamos a la experta de estadística con varios días de anticipación. En el correo con el que fue contactada, se le preguntó si le era más conveniente el realizar una reunión para mostrarle nuestro proyecto, o si prefería que le grabáramos un video explicándole todo lo necesario para recibir su retroalimentación. A esta solicitud, ella nos contestó que prefería que le mandáramos un video, lo cual hicimos ese mismo día. No obstante, nunca recibimos la retroalimentación respectiva, por lo que no pudimos completar esta parte del proyecto.

A continuación, se muestran evidencias fotográficas de nuestra interacción con la persona asignada:

TR

Thais Tamaio Ramirez

Para: Valery Alexandra Calixto Molina

CC: Jesus Alberto Jimenez Garizao; Juan Camilo Bonet De Viviero

Mié 26/04/2023 6:50

Hola Valery, ¿Cómo estás?

Te escribimos para preguntarte si nos podemos reunir esta semana o si prefieres que te mandemos un video mostrándote los resultados de nuestro proyecto, para así obtener tu retroalimentación y sugerencias generales.

Tu ayuda es muy valiosa para nosotros y agradeceríamos cualquier comentario o consejo que puedas brindarnos en relación al proyecto.

¡Muchas gracias!

Grupo G-30 (Thais Tamaio, Jesús Jiménez y Juan Camilo Bontet).

V

Valery Alexandra Calixto Molina

Para: Thais Tamaio Ramirez

Jue 27/04/2023 6:57

Hola Thais, buenos días, te comento que prefiero la opción del video porque nos da más flexibilidad con respecto a los tiempos ya que ya tengo agendada varias citas para mañana y el sábado, con gusto lo veré y haré un documento con distintos puntos que considero que pueden ayudar a una mejor realización de su trabajo y si aún cuentan con dudas podemos comunicarnos por este medio, quedo atenta.

...

TR

Thais Tamaio Ramirez

Para: Valery Alexandra Calixto Molina

Jue 27/04/2023 21:05

¡Hola Valery!

Acá te enviamos el link con los resultados de la segunda etapa de nuestro proyecto:

<https://youtu.be/yxuwo4ZLmSc>

De igual manera, te queremos mandar un vídeo que explica de fondo el proceso que seguimos para obtener el modelo utilizado en esta etapa, por si quieres saber más al respecto:

<https://youtu.be/2VAg0de9xE>

Muchas gracias por tu tiempo y ayuda. Quedamos pendientes de tu feedback o cualquier comentario sobre nuestro trabajo.

Grupo G-30 (Thais Tamaio, Jesús Jiménez y Juan Camilo Bontet).

Get [Outlook for iOS](#)

...