Informe Proyecto 1 Etapa 2

Automatización y uso de modelos de analítica de textos:

Análisis de sentimientos de películas

Inteligencia de Negocios

2023-10

Integrantes:
Juan Felipe Patiño 201922857
Juan Ricardo Diaz 201922167

1. Proceso de automatización

1.1. Proceso de preparación de datos

Para el proceso de preparación de datos tuvimos en cuenta lo realizado durante la etapa 1. Además, con el objetivo de automatizar esto por medio de un Pipeline tuvimos en cuenta el tipo de preparación de datos necesarios. Esta etapa es de las más importantes y críticas para el proceso de modelos analíticos en textos, y es clave que el entrenamiento seleccionado como mejor opción en la etapa anterior sea el mismo usado en esta etapa. En esta etapa buscamos asegurar que los datos sean relevantes, consistentes y en el formato esperado por el proceso. El primer paso es la recopilación de datos, lo cual manejaremos por la API a desarrollar, pero para el entrenamiento como tal usamos los datos que se entregaron. Para la limpieza tomamos la decisión de eliminar las palabras de menos de 2 usos ya que estas pueden causar un overfitting del modelo. Fuera de esto tomamos la decisión de vectorizar con ayuda de la librería nktl para generar una vectorización sobre todo el abecedario español. Esta vectorización se realizó con el método Bag of Words, el cual fue seleccionado en la etapa anterior como el mejor.

1.2. Construcción del modelo

La construcción del modelo y las decisiones asociadas fueron los pasos realizados en la etapa anterior de la entrega. Es por esto que con los resultados y decisión del modelo anterior construimos el pipeline para el sector de modelos. Recordemos que seleccionamos el un modelo de Random Forest Classifier, el cual tuvo un mejor desempeño respecto al resto, además utilizamos un grid search para los parámetros óptimos. Estos parámetros son de 200 estimadores, con un mínimo de 5 para un split, una profundidad de árbol máxima de 5 y bajo el criterio de entropía. De esa manera el pipeline construido contiene el modelo de mejor desempeño de la etapa anterior. Usualmente se realizará pasos como selección de características donde se filtran ciertas columnas, pero en este caso la única característica es el texto en español, y está es la que es va a recibir en el modelo, por lo cual no es necesario este paso para nuestra pipeline, pues está únicamente recibe el texto. Otros pasos relevantes, como lo son la preparación de datos de entrenamiento y prueba, el entrenamiento del modelo, la evaluación y ajuste del mismo son los pasos que ya realizamos en la etapa anterior del proyecto, por lo cual ya contamos con estos pasos realizados y se pueden referenciar en la entrega de la etapa 1.

1.3. Persistencia del modelo

La persistencia del modelo se refiere al proceso de guardar un archivo de modelo entrenado para que pueda usarse en el futuro sin tener que volver a entrenarlo desde cero. Después de entrenar el modelo y refinar su desempeño, lo persistimos por medio de un Pipeline de sklearn. Esto es importante porque ahorra tiempo y recursos en usos futuros, especialmente cuando se trabaja con grandes conjuntos de datos o cuando se desea evaluar instancias posteriormente, como es el caso de esta etapa. En particular, persistimos el modelo a un joblib para que la API pueda usar este modelo ya entrenado para evaluar nuevos inputs.

Si bien este no es el caso de nuestro API, es importante considerar que para un uso más adecuado del modelo con mejores resultados sería ideal continuar entrenando el modelo. Esto atendiendo a la evolución constante en la forma de comunicación de las personas por tuits, y como la connotación de una palabra puede diferir entre momentos de tiempo. Entonces sería deseable continuar entrenando el modelo con datos etiquetados a través del tiempo para refinar sus resultados.

1.4. Acceso por medio de API

El primer puente que se construye es el que permite al modelo, como un archivo joblib, ser accedido por medio de una clase de Python. Esta clase de modelo se encarga de, al inicializarse, cargar el modelo de pipeline construido y entrenado previamente, y luego de recibir consultas responder con 1 si es positiva o 0 de lo contrario. De esta manera tenemos esta primera clase que permite consultar el Pipeline. Ahora para el API como tal utilizamos Fast API la cual permite desarrollar APIs de manera rápida y fácil, la cual utilizaremos en conjunto al despliegue de en local host para acceder a este. Contamos un HTML el cual es el front como tal y en este incluimos parámetros de predicciones que nos permiten informar al usuario de los resultados y de si fue exitosa o no la operación en caso de ser un csv.

2. Desarrollo de la aplicación y justificación

Hacia el cliente, la aplicación desarrollada es un sistema que permite utilizar un modelo de análisis de texto de clasificación los tuits como opiniones positivas o negativas sobre películas. Este programa es útil para el gerente de cine porque le permite tomar decisiones informadas sobre cuántas funciones reproduce por película y a qué hora y por cuantos días.

El usuario/rol en la organización que utiliza esta aplicación es gerente de cine. El proceso comercial respaldado por esta aplicación es la toma de decisiones sobre los horarios de las películas y establecer fechas y horas de estas. Un gerente de cine debe saber lo que los clientes piensan sobre las películas para poder programarlas de manera efectiva y maximizar los ingresos del cine, ya que una película con muchas opiniones positivas va a llevar más personas al teatro que una que no tenga opiniones positivas. El vínculo entre esta aplicación y el proceso comercial que admite es fundamental para el éxito del cine pues maximiza la ocupación de salas de cine en cada función. Al conocer las reseñas de películas de los potenciales clientes, el gerente puede tomar decisiones informadas, esto significa que los ingresos del teatro se pueden minimizar mediante la programación de películas de manera eficiente.

Se desarrolló una aplicación web que permite interactuar con el modelo y probar .csv de datos. Estos deben contener la columna de “review_es” sobre la cual se realizará el procesamiento de datos. Cuando esto se haga de manera exitosa informa al usuario de este hecho y le permite descargar un csv que contiene lo mismo que el que subió más el sentimiento al respecto calculado por el modelo.

También permite ingresar un único texto que desea calificar, para el cual responde el tipo que es. En este caso interpreta el mensaje como solo uno, aún si el usuario intenta poner varios. Después le responde en un mensaje debajo de la caja de texto si es positivo o negativo.

Con respecto al trabajo interdisciplinario fue importante múltiples recomendaciones y aspectos. La primera fue la verificación de calidad de los datos, donde nos recalcaron la importancia de realizar un análisis preciso y relevante para la problemática, lo cual va de la mano de entender la limitaciones y así mismo medir las expectativas, entendiendo que no es el modelo más robusto ni mejor entrenado, y así mismo esperar resultados congruentes a este hecho. Aún así, nos insistió en verificar la validez y revisar que los resultados del modelo y lógicos sean los esperados, indicando que debemos mirar más allá de las medidas estadísticas y aplicarle lógica a la respuesta.

3. Resultados

Link del video:
https://youtu.be/zySIBWbHTyc

4. Trabajo en Equipo

4.1. Roles

Integrante	**Rol**
Juan Felipe Patiño	Líder de Proyecto, Ingeniero de Datos, Ingeniero de software responsable de desarrollar la aplicación final
Juan Ricardo Diaz	Ingeniero de software responsable del diseño de la aplicación y resultados

Juan Felipe Patiño, como Líder de Proyecto, estuvo a cargo de coordinar las fechas de reuniones y de trabajo, llenar los formatos de pre-entrega y dividir el trabajo a realizarse de manera equitativa y justa. Además, estuvo pendiente de los avances constantes y notificó al grupo sobre los detalles de la entrega. Finalmente en este rol fue el que subió y entregó todos los entregables pertinentes. Por otro lado como Ingeniero de Datos, tuvo la labor de el proceso de automatización del modelo, en especial el proceso de construcción del modelo y preparación de datos, traduciendo la etapa anterior a un pipeline autocontenido y desplegado como un modelo.joblib con su modelo.py para acceder a este, la cual se interconecta con la API posteriormente. Además, tuvo su rol como Ingeniero de Datos de la aplicación final lo cual incluye la unión de modelo joblib con la FastAPI y la validación funcional de este.

Por otro lado Juan Ricardo, como Ingeniero de software tuvo la responsabilidad del diseño de aplicación y resultados. Para esto se enfocó en el desarrollo de un frontend funcional.

Con los roles de cada uno asignados y desarrollados, cada integrante se planteó utilizar una misma cantidad de tiempo individual y un tiempo grupal de reuniones y de apoyo mutuo. Este tiempo de desarrollo por cada rol fue menor a 5 horas cada uno, dependiendo ya de la velocidad de desarrollo y condiciones del software y su entorno, mientras que de manera grupal es tuvieron múltiples reuniones, alrededor de 3 formales y múltiples informales, de cada una alrededor de 1 hora

Entre los retos principales estaba el tiempo disponible, ya que si bien se tuvo el enunciado disponible con antelación no fue hasta después de terminar la etapa anterior que se pudo iniciar a avanzar en esta. Además, para esta ocasión fue necesario un mayor grado de investigación sobre el desarrollo de front end local, para poder desplegar la aplicación. Entonces el reto de obtener el tiempo para desarrollar la

actividad de manera completa y de buen nivel fue un gran reto. Durante el transcurso de la segunda etapa se llevaron a cabo múltiples reuniones, donde vale la pena recalcar una primera de decisión de tema, una segunda de división de roles, dos reuniones posteriores de avances y retroalimentación mutua y una reunión final para finalización de entregables y posterior entrega.

4.2. División de puntuación

La división de puntos, acorde a la planeación y realización del proyecto se realizará acorde a los roles de cada uno. Por esto los puntos se dividirán 55% para Juan Felipe y el resto a Juan Ricardo.

5. Despliegue Cloud

Como se establece en la sección se dará un bono por el despliegue en cloud de la aplicación. Para esto utilizamos GCP y una MV de Ubuntu para clonar y correr el servidor. Este se encuentra en la siguiente dirección IP para uso público:

34.121.177.205