

Ingeniería de Sistemas y Computación
Pregrado
ISIS3301 – Inteligencia de Negocios
Semestre 2024-10
Proyecto 1 – Etapa 2

Santiago Chamie Rey - 202122182
Daniel Escalante Pérez – 202122384
Daniel Felipe Vargas Ulloa - 202123899

Contenido

Proceso de automatización del proceso de preparación de datos, construcción de modelo, persistencia del modelo y acceso por medio de API	3
Construcción del pipeline	3
Otras cosas idk	3
Desarrollo de la aplicación y justificación.....	3
Backend:	4
Operaciones y sus endpoints:	4
Frontend:	5
Presentación de métricas:	5
Predicciones:	6
Re-entrenado:	7
Evaluación del Cliente:.....	8
Trabajo en equipo	8
Integrantes.....	8
Reuniones	9
Tareas.....	9
Reflexión grupal	9

Proceso de automatización del proceso de preparación de datos, construcción de modelo, persistencia del modelo y acceso por medio de API

Construcción del pipeline

En primera instancia, hicimos el automatizamos el proceso de preparación de datos y construcción del modelo mediante a un pipeline de la librería Sci-Kit Learn. El pipeline se divide en tres partes: la primera, el procesador de texto se encarga de toda la parte de preparación de los datos para ser vectorizados usando el método tf-idf, incluyendo la eliminación de caracteres y palabras sin valor para el modelo y la lematización de las palabras que sí; la segunda, el vectorizador, quien además de calcular los valores de tf-idf de las palabras del modelo, guarda dentro de una variable la lista de todas las palabras válidas que recibe el modelo, para que a la hora de predecir no nos encontremos con errores de este estilo; y la tercera, corre el modelo, con la opción de hacer una búsqueda de grilla en caso de querer explorar múltiples hiperparámetros, y eligiendo la mejor opción para asignar como el modelo final que se usará para predecir. Usamos un diccionario como grilla de parámetros, que entra al pipeline como parámetro y sirve para correr la búsqueda de grilla, con lo cual se puede correr el modelo sin problema alguno. El modelo es reentrenable y devuelve métricas correspondientes al modelo ejecutado de regresión logística, como la probabilidad asociada a cada clasificación.

Desarrollo de la aplicación y justificación

Después de una reunión con el equipo de estadística y la discusión de los resultados de la etapa uno se propuso el proceso de definir la aplicación. Se llegó al consenso de hacer los avances correspondientes a la aplicación.

Entre los roles pensados y discutidos se considero que los usuarios fueran los encargados de la dirección de los hoteles puesto que estos serán los encargados de buscar materializar los objetivos del negocio por medio de estrategias provenientes de la aplicación de analítica en este caso. No obstante, se consideró que la aplicación también podría usar por la dirección de marketing y de control de calidad, porque gracias a esta información se pueden publicitar elementos resaltados que hacen un sitio atractivo. Por parte de control de calidad, estos podrían ver aspectos que causan alejamiento por parte de los clientes de los sitios turísticos y hacer verificación de estos.

Ya con estos roles tomados en cuenta se pensaron en múltiples opciones para la aplicación. La primera de estas era un simple predictor de calificación de reseñas, pero al ver que esto no sería muy indicativo y útil para los negocios, se considero una segunda opción. En la segunda opción se pensó en tener una plataforma donde los hoteles vieran sus reseñas y pudieran realizar analítica alrededor de estas. No obstante, se prefirió tomar un enfoque más general para poder alimentar el modelo analítico con más datos que provienen de muchos hoteles diferentes. Con esto en cuenta, se decidió hacer una aplicación para realizar inteligencia hotelera en la cual se tendría la posibilidad de calificar automáticamente reseñas, ver que tan buenas eran las predicciones de la aplicación, ver que palabras influyen más o menos en cada calificación y poder reentrenar la aplicación con los datos deseados para realizar

inteligencia sobre los dominios deseados. Todo esto, buscando facilidad de uso, ser estéticamente agradable y que todo fuera fácilmente entendible e intuitivo.

La aplicación estaría apoyando el proceso de negocio de evaluación de los servicios (control de calidad) que provee el hotel, como, por ejemplo, limpieza, comida, ayuda ante quejas, dudas y problemas, entretenimiento entre otras. Por otra parte, el área de marketing podría usar la aplicación para la ideación de campañas y de productos atractivos al ver que es popular entre los clientes. Un ejemplo de esto es lo que se vio en la etapa anterior, donde resaltaron palabras como “ciclismo”. Por último, la dirección del hotel podría tomar estrategias con respecto a expansión e implementación de servicios atractivos como lo es el desplazamiento entre terminales de transporte y sitios turísticos, o también, implementar más representantes y concierges que apoyen a los visitantes.

Habiendo definido tanto la aplicación, los roles, la utilidad y los usos, se tomaron decisiones con respecto a la implementación y presentación de la información.

Backend:

Operaciones y sus endpoints:

Para el desarrollo del backend se tomaron las operaciones fundamentales que muestran el funcionamiento y desempeño de la pipeline para traducirlas a endpoint del backend que sería consumido por el frontend. Para estas funcionalidades se definieron endpoints para re-entrenamiento, predicción, y métricas.

Endpoint: POST “/retrain”

Este endpoint recibe una lista de objetos json con los cuales reentrena el modelo y lo guarda para futuras peticiones y calculo de métricas sobre este. Este endpoint solamente retorna un objeto json indicando el éxito de la operación. No obstante, vale la pena resaltar que la cantidad de datos proporcionados debe ser significativa, o sino la operación fallara al realizar evaluaciones por falta de información. Con el re-entrenamiento, será posible usar datos customizados o propios de un negocio específico para sacar observaciones interesantes.

Vale la pena resaltar que el modelo es persistente y guarda sus parámetros del último entrenamiento dado. Esto se hizo mediante la librería joblib que crea un archivo que permite cargar y guardar el modelo analítico.

Endpoint: GET “/predict”

Este endpoint obtiene la predicción generada por el modelo en base a datos pasados en el cuerpo de la petición. Esta recibe un arreglo de objetos json con un atributo de nombre “Review” el cual contiene la reseña a evaluar. Este retorna un arreglo del tamaño de la entrada y para cada entrada devuelve un objeto con la predicción y las probabilidades de que pertenezca a cada clase. Con este endpoint será posible cumplir con el objetivo de predecir reseñas.

Endpoint: GET “/metrics”

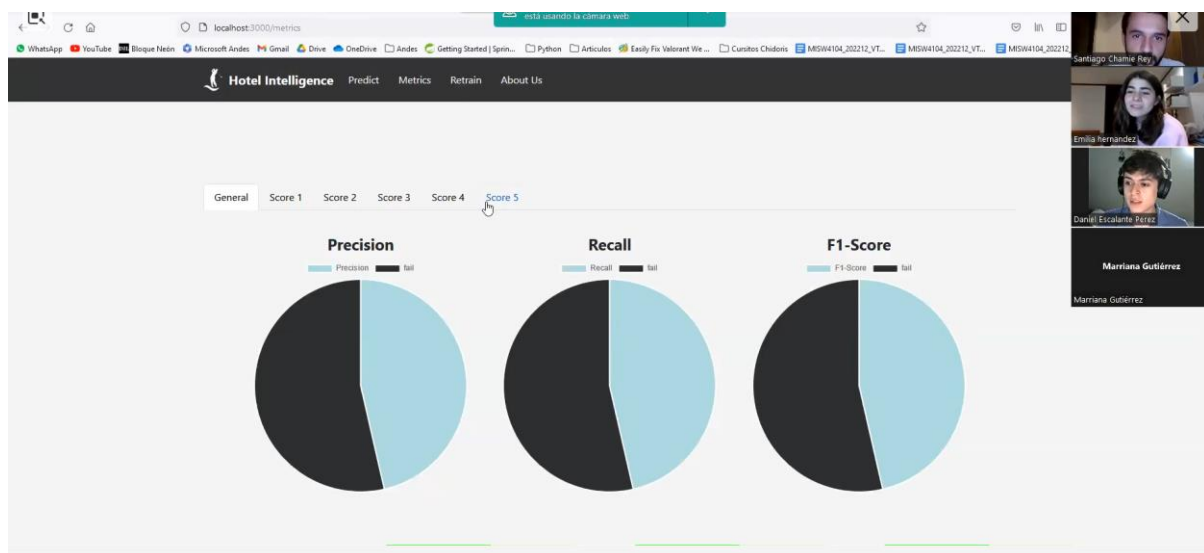
Este endpoint obtiene las métricas del modelo con el ultimo entrenamiento dado. Este endpoint no recibe nada pero devuelve un arreglo con 6 objetos. El primer objeto son las métricas promedio del modelo incluyendo precisión, recall y f1-score. Los demás

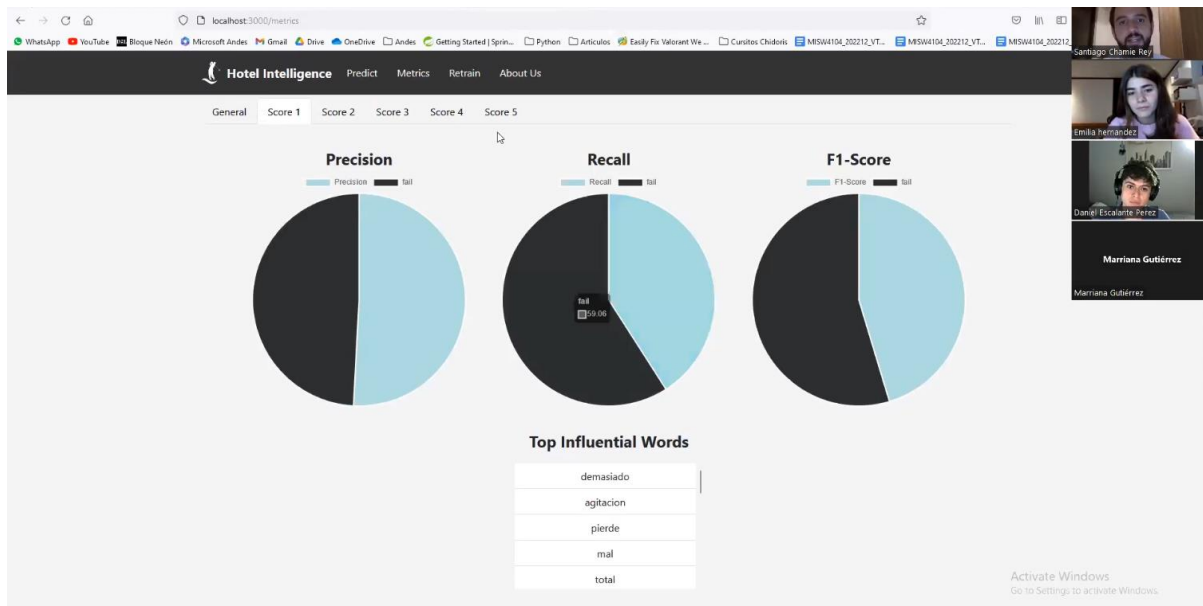
objetos son iguales pero con las métricas de cada clasificador individual (para cada una de las 5 calificaciones) del modelo OVR (One vs Rest) entrenado en la primera etapa. Adicional a las métricas, estos objetos contienen un arreglo con las palabras más relevantes y que indican con mayor fuerza que la reseña pertenece a esa calificación. Esto será útil para poder ver los aspectos más relevantes de una calificación para poder sacar observaciones de que mejorar o que implementar en el hotel y ser más populares. Adicionalmente, se está cumpliendo con el objetivo de caracterizar los sitios turísticos atractivos y no atractivos.

Frontend:

Presentación de métricas:

En la pantalla de métricas podrá ver un menú de pestillas que permite navegar entre las métricas generales o las de cada calificación. Para presentar las métricas del modelo, se pensó en la forma más simple de mostrar un porcentaje aparte de mostrar el porcentaje textualmente. Por eso se decidió usar diagramas de pie para mostrar los valores de precisión, recall y f1-score que son porcentajes. Adicionalmente, debajo de estos diagramas y para las métricas de cada calificación y no del modelo general, se decidió mostrar un pequeño listado con las 20 palabras más relevantes que acercan la reseña a ser calificada con ese valor.

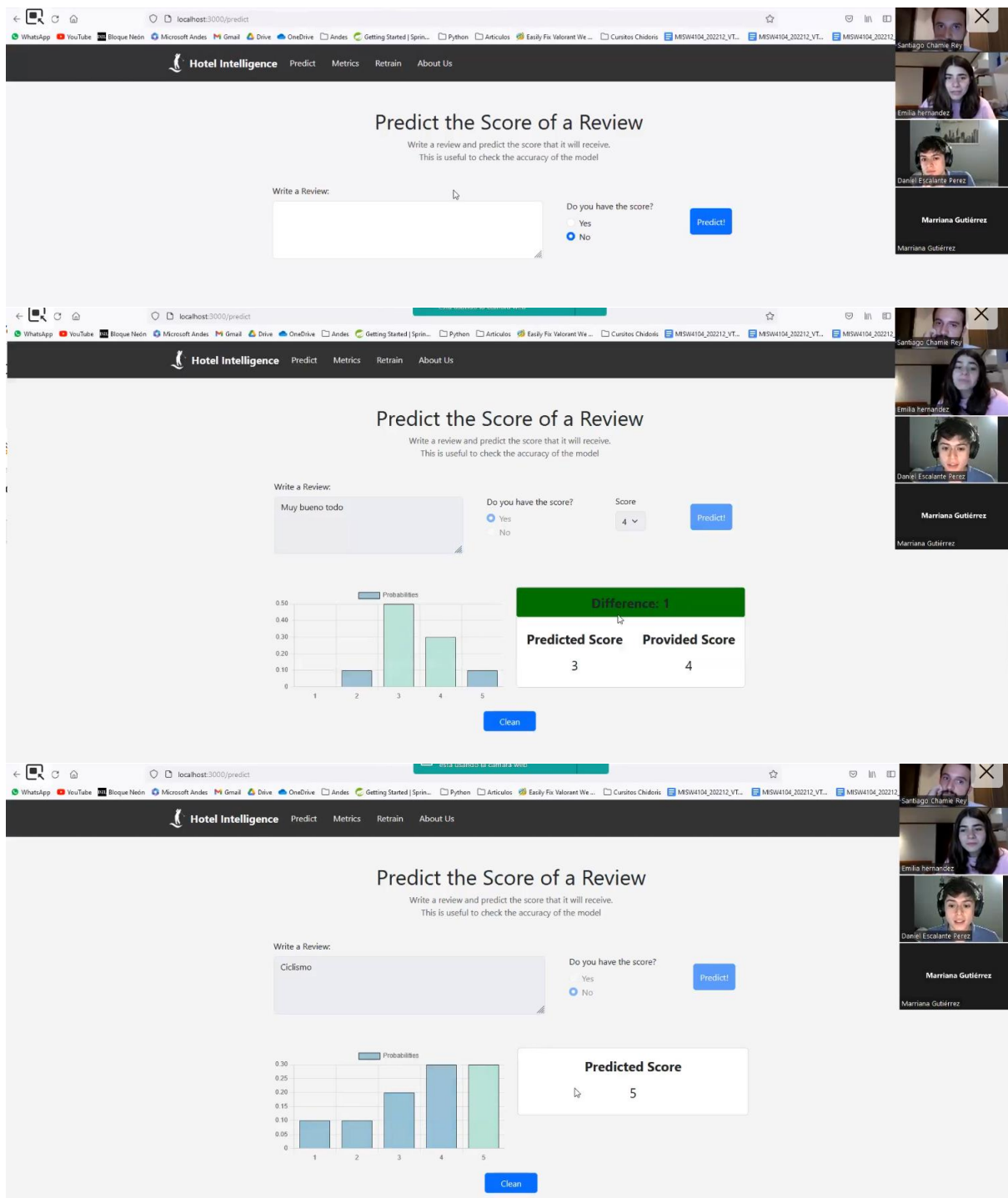




Predicciones:

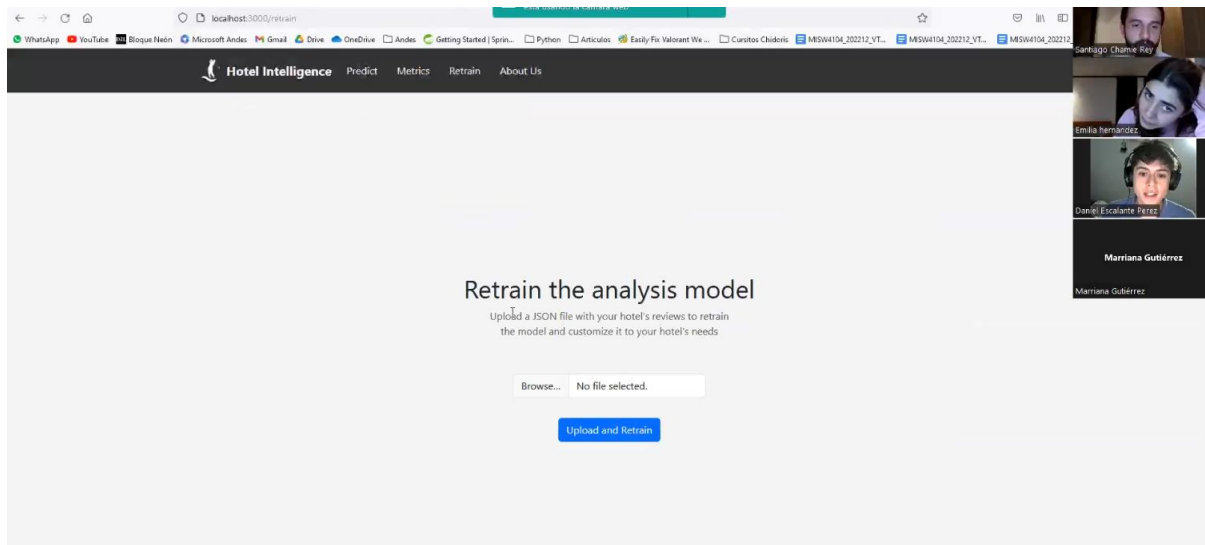
Respecto a las predicciones en el frontend, se decidió crear un formulario simple con una caja de texto que permita escribir tu reseña. Posteriormente, puedes o no indicar la calificación real de la reseña para hacer la comparación con el resultado del modelo. Finalmente, se puede oprimir el botón de predecir y esto dará la clase predecida y mostrará una gráfica que compara los valores de probabilidad de cada calificación dados por el modelo. Así se podrá ver si la respuesta coincide con la reseña y se verá si hubo disputa entre 2 clases y que tan cercanas estuvieron en términos de probabilidad.

Adicionalmente, para realizar predicción sobre múltiples valores, existe la opción de subir archivos tipo json con los objetos representando las reseñas. De esta forma se pueden recibir múltiples respuestas a múltiples peticiones con facilidad.



Re-entrenado:

Para el reentrenado se tomó una aproximación similar a la de la predicción multiple donde se sube un archivo con múltiples registros, solamente que en este caso los registros deben contener las etiquetas con la calificación real. De este archivo se saca la lista de objetos json y se manda al backend. Posteriormente, se tendrá el resultado de éxito y se podrán ver las nuevas métricas en esa sección de la aplicación. Esto facilita el proceso de re-entrenamiento y hace una experiencia más cómoda para el usuario al no tener que estar copiando y pegando registros.



Evaluación del Cliente:

Ya con la aplicación definida y desarrollada se realizó una reunión de seguimiento donde se mostró el producto logrado a las estudiantes de estadística. Durante la sesión se les permitió interactuar con la página desarrollada y que probaran las diferentes funcionalidades. Las estudiantes, como parte del proceso de desarrollo y como se menciona previamente en las alternativas de aplicación, aportaron multiples ideas, en esta reunión manifestaron sus satisfacción respecto a las ideas aportadas. Adicionalmente, señalaron que la aplicación era altamente intuitiva y facil de usar. Tambien, despues de un pequeño tutorial y explicación, resaltaron que era bastante util y completa, otra observación fue que las utilidades planteadas previamente para los diferentes roles si se ven posibles lograrlas con el producto alcanzado.

Si desea probar el producto final del proyecto, puede dirigirse al repositorio:

<https://github.com/Dani32002/ISIS3301-proyecto1-grupo2>

Después de clonar el proyecto, siguiendo las instrucciones de uso podrá usar la aplicación.

Trabajo en equipo

Integrantes

Para el desarrollo de nuestro proyecto el grupo consistió en las siguiente personas y roles:

Grupo de ingeniería

- Daniel Felipe Vargas Ulloa - d.vargasu@uniandes.edu.co
 - Rol: Ingeniero de software responsable del diseño de la aplicación y resultados
- Daniel Escalante Perez - d.escalante@uniandes.edu.co
 - Roles: Ingeniero de software responsable de desarrollar la aplicación final
- Santiago Chamie Rey - s.chamie@uniandes.edu.co
 - Rol: Líder de proyecto e Ingeniero de datos

Grupo de estadística

- Mariana Gutierrez - m.gutierrez11234@uniandes.edu.co
- Emilia Hernández - e.hernandez112@uniandes.edu.co

Reuniones

Nos reunimos en tres ocasiones con todos los miembros con los siguientes propósitos:

- Reunión de lanzamiento y planeación [7:00 pm | abril 10, 2024] [1h]
- Reunión de seguimiento [7:00 pm | abril 17, 2024] [30 min]
- Reunión de finalización [7:00 pm | abril 20, 2024] [30 min]

Tareas

Cada integrante del grupo de sistemas tuvo asignado las siguientes tareas con su duración

Daniel Felipe Vargas Ulloa

- Tareas
 - Diseño backend aplicación
 - Trabajo en el Documento
 - Edición y organización del video
 - Grabación de video

Daniel Escalante Perez

- Tareas
 - Diseño frontend aplicación [3.5h]
 - Implementación del backend [2h]
 - Trabajo en el Documento [1.5h]
 - Grabación de video [1h]

Santiago Chamie Rey

- Tareas:
 - Diseño pipeline [5h]
 - Trabajo en el Documento [1h]
 - Grabación de video [1h]
 - Organizador del documento [1h]

Reflexión grupal

Durante el desarrollo de esta entrega del proyecto nuestra dinámica de trabajo en equipo fue excelente. Logramos comunicar nuestras ideas y distribuir las tareas de tal manera que cada integrante fue responsable de una cantidad equitativa de trabajo. Por esto, si fuéramos a repartir 100 puntos entre los integrantes del grupo los repartiríamos equitativamente, dándole 33.3 puntos a cada uno de los miembros del grupo de ingeniería. Además, el grupo de estadística fue de gran ayuda, ya que nos dio retroalimentación oportuna y brindó ideas para desarrollar con respecto al enfoque del proyecto. Sin embargo, siempre existe campo para mejorar, por lo que para un futuro proyecto podríamos intentar establecer unos roles más claros y asignar tareas

alrededor de estos y de las fortalezas y debilidades de los integrantes. En el caso de este proyecto, si asignamos roles, mas, las tareas que asignamos no siempre estuvieron de la mano a los roles indicados.