

Universidad de los Andes

Facultad de Ingeniería
Departamento de Ingeniería de Sistemas y Computación
Inteligencia de Negocios
2024-10



Proyecto 1 Etapa 2

Analítica de Texto

Grupo 28

Gabriel Felipe Dicelis Ramos - 201920847
Juan Pablo Martínez Pineda - 202012623
Laura María Restrepo Palomino - 202013289

Abril 20, 2024
Bogotá D.C.

Tabla de contenido

Introducción.....	3
Proceso de automatización del procesamiento de datos.....	3
Construcción del modelo.....	3
Persistencia del modelo.....	5
Desarrollo de la aplicación y justificación.....	5
Usuario.....	5
Desarrollo.....	6
Resultados.....	7
Trabajo en equipo.....	7
Anexos.....	8

Introducción

En el sector hotelero, la capacidad para comprender y anticipar las preferencias y experiencias de los clientes es fundamental para mantener un servicio de calidad y competitivo. En este contexto, surge la necesidad de una herramienta que permita analizar eficazmente las opiniones y reseñas de los usuarios, para identificar elementos diferenciadores en el servicio de hospedaje y turismo. Por tanto, se plantea el desarrollo de una aplicación web accesible que, mediante modelos entrenados con un enfoque analítico de datos, pueda proporcionar predicciones precisas sobre las calificaciones basadas en las reseñas de los clientes. Además, se busca realizar una caracterización de palabras significativas que facilite la toma de decisiones informadas en la gestión y operación de los establecimientos hoteleros. Este informe presenta el diseño y desarrollo de dicha aplicación, así como los resultados obtenidos a través de su implementación.

Proceso de automatización del procesamiento de datos

Para el procedimiento de automatización del procesamiento de los datos de las reservas se optó por la implementación de un API que cumpla con estas tareas de procesamiento de datos.¹

La tarea de procesamiento automático de reseñas en la aplicación web mediante el API implica una serie de pasos para preparar los datos antes de su análisis. Esto incluye la normalización del texto para estandarizarlo, la lematización para reducir las palabras a su forma base y otros procesamientos de texto como la eliminación de palabras vacías y la tokenización. Estos procesos se llevan a cabo de manera automatizada al ingresar una reseña en la aplicación, lo que garantiza que los datos estén listos y optimizados para su posterior análisis.

```
def preprocessing(words):  
    words = fix_contractions(words)  
    words = remove_non_ascii(words)  
    words = to_lowercase(words)  
    words = remove_punctuation(words)  
    words = replace_numbers(words)  
    words = remove_stopwords(words)  
    return words
```

Preprocesamiento

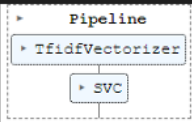
Construcción del modelo

El modelo utilizado para analizar las reseñas previamente procesadas es el algoritmo SVC seleccionado en la etapa 1, ya que demostró mejores resultados en términos de métricas de evaluación. Para facilitar su implementación en la

¹El enlace al repositorio con el código fuente del API puede ser encontrado en los anexos.

aplicación web, se exporta un pipeline en formato .joblib, el cual se carga en el API para realizar predicciones sobre las reseñas que ingresan a la plataforma. Este pipeline encapsula tanto el preprocesamiento de texto como el modelo de clasificación, lo que permite una integración fluida y eficiente en el flujo de trabajo de la aplicación.

```
pipeline = Pipeline(  
    [  
        #('preprocessing', FunctionTransformer(full_preprocessing)),  
        #('tokenize', FunctionTransformer(spacy_full)),  
        ('tfidf', TfidfVectorizer()),  
        ('svm_model', SVC(kernel='linear', random_state=42))  
    ],  
    verbose=1  
)  
  
pipeline.fit(X_train, y_train)  
  
[Pipeline] ..... (step 1 of 2) Processing tfidf, total= 0.5s  
[Pipeline] ..... (step 2 of 2) Processing svm_model, total= 18.3s
```



```
graph TD  
    Pipeline --> TfidfVectorizer  
    TfidfVectorizer --> SVC
```

Creación del pipeline

```
1 from joblib import load  
2  
3 class Model:  
4  
5     def __init__(self):  
6         self.model = load("svcpipeline_v2.joblib")  
7  
8     def make_predictions(self, data):  
9         result = self.model.predict(data)  
10        return result
```

Carga del pipeline en el API

Nótese que además de cargar el pipeline, existe la función `make_prediction`, la cual utiliza el pipeline previamente cargado para hacer una reducción de calificación en la nueva reseña.

Además de esto el modelo cuenta con dos endpoints con respecto a la aplicación:

- **Predicción de calificación:** Este endpoint permite enviar una reseña al modelo para predecir su calificación. A la reseña se le realiza su procesamiento y luego el modelo predice una calificación para asignarse en un rango de 1 a 5.
- **Extracción de palabras importantes:** Aparte de realizar predicciones, el modelo también selecciona las 300 palabras más relevantes en función de su importancia para la calificación de las reseñas. Entonces, cuando se envía una reseña al API a través de este endpoint, se verifica si en ella hay palabras en este top 300 y lo devuelve como resultado.

Por último, el API fue desplegado haciendo uso del servicio Render mediante la siguiente URL <https://fastapi-reviews-app.onrender.com>.

Persistencia del modelo

El modelo es persistente debido a que se guarda en un archivo .joblib en disco. Esto implica que tanto el modelo como sus características, como el preprocesamiento, se almacenan en un archivo que puede ser cargado nuevamente en memoria cuando sea necesario. Esta persistencia del modelo permite su reutilización en diferentes momentos sin necesidad de volver a entrenarlo desde cero, lo que lo hace útil para la aplicación web mencionada posteriormente en el informe.

Desarrollo de la aplicación y justificación

Para hacer uso de la API construida, es necesario crear una aplicación que permita la interacción del usuario y el modelo. Antes, sin embargo, se debe definir el rol de usuario específico de quien usará la aplicación. Asimismo, se debe hacer una validación con los expertos para confirmar su utilidad y que se conforme con los objetivos definidos.

Usuario

Vale la pena recordar que la organización definida en la primera etapa del proyecto fueron las cadenas hoteleras. También dentro de los roles dentro de la organización que se pueden beneficiar del modelo se definieron los sitios turísticos, relacionados mediante un convenio, donde los hoteles le indican a los encargados de los sitios las características que los hacen o no buenos. Siendo así, el usuario final de la aplicación será el mediador entre estas dos entidades. Los beneficios que esto puede traer son los siguientes:

- Por parte de los sitios turísticos, al tener una aplicación que muestre de manera clara las características de las reseñas que son importantes para los turistas, se pueden identificar aspectos de mejora que, en turno, podrán atraer más clientes y, así, ingresos.
- Por parte de las cadenas hoteleras, si se comunican correctamente los aspectos de mejora, y el sitio turístico los tiene en cuenta, se pueden tener convenios en donde los visitantes a los sitios turísticos se hospeden en los hoteles y tengan descuentos. Así, se puede generar una mayor cantidad de huéspedes e ingresos.
- Por parte del mediador, la aplicación justificaría la existencia de su trabajo y lo volvería indispensable a la hora de comunicar a los sitios y hoteles.



Comentarios de los expertos:

“Los objetivos definidos son claros y tienen sentido con el proceso que llevan. Justifican correctamente una relación entre las cadenas hoteleras y los sitios turísticos”.

Desarrollo

Teniendo en cuenta estos objetivos y beneficios, y los comentarios de los expertos, se procede al desarrollo de la aplicación. Como se sabe, se tienen dos endpoints, uno para predecir la calificación y otro para encontrar las palabras clave dentro de la reseña, por lo que se define que una vez el usuario ingrese una reseña, se debe informar al usuario de ambos aspectos.

Para este proceso, se hace uso de la biblioteca React de Javascript para facilitar el proceso. Primero, se estableció un diseño inicial que demostrara el uso de ambos endpoints y su sencillez de interacción, para ser validado por los expertos. Este fue el diseño:

Turismo de los Alpes

Con esta aplicación podrás predecir la calificación dada por usuarios en sus reseñas.

Pasos:

1. Ingresa una reseña en el campo de texto.
2. Da click en el botón **Predecir**.
3. ¡Listo! Observa tus resultados a la derecha.

Reseña:

Predecir

Resultados:

Palabras clave:
lorem ipsum

Calificación:
★ ★ ★ ★ ★ 1/5

Diseño inicial de la aplicación

Como se puede observar, se brinda al usuario una serie de pasos a seguir para hacer uso del modelo de predicción, facilitando el entendimiento de la aplicación. Asimismo, a la derecha, se observa que los resultados muestran las palabras clave y la calificación. Por parte de las palabras clave, originalmente se planeó mostrar un simple listado de las palabras. Sin embargo, se obtuvieron los siguientes comentarios:

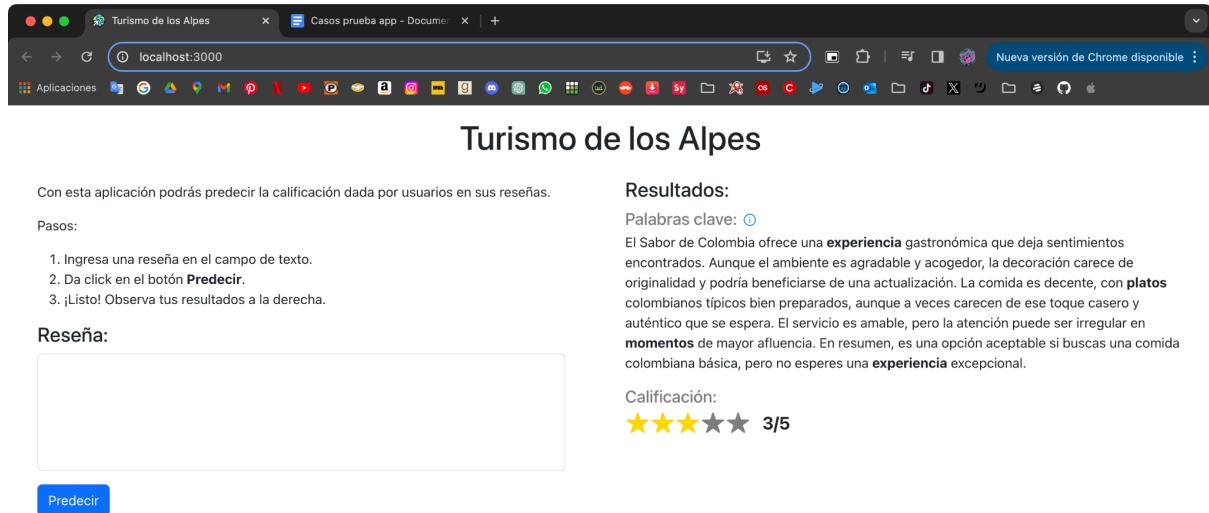


Comentarios de los expertos:

“Como la reseña desaparece después de hacer click en el botón, en vez de mostrar solo un listado, es mejor devolver la misma reseña con las palabras resaltadas, así pueden proporcionar contexto. No solo eso, sino que es mejor que le comuniquen al usuario que las palabras clave pertenecen al top 300”.

Resultados

Habiendo tenido en cuenta las recomendaciones dadas, la aplicación final tiene el siguiente aspecto:²



Aplicación final

Adicionalmente, el ícono de información muestra el siguiente texto:

Resultados:

Palabras clave: ⓘ

lugar bonito

Las palabras resaltadas hacen referencia al top 300 de palabras clave del modelo de entrenamiento

Información adicional



Comentarios de los expertos:

“La aplicación final cuenta con las recomendaciones dadas, es sencilla de usar y se alinea con sus objetivos”.

Trabajo en equipo

Para este trabajo, los roles definidos entre los miembros fueron:

- Gabriel Dicelis - Líder del proyecto e ingeniero de datos.
- Juan Pablo Martínez - Ingeniero de datos.

² El enlace al repositorio con el código fuente de la aplicación puede ser encontrado en los anexos.

- Laura Restrepo - Ingeniera de software responsable del diseño de la aplicación y resultados, e ingeniera de software responsable de desarrollar la aplicación final.

También se definió el siguiente cronograma:

Fecha	Responsabilidad	Tarea	Tiempo dedicado
Abril 12	Grupal	Revisar el enunciado	0.5 horas
Abril 13	Grupal	Iniciar construcción de API y definición de endpoints	2 horas
Abril 14	Grupal	Empezar documento	1 hora
Abril 15	Grupal	Validación de objetivos con expertos	0.5 horas
Abril 15	Grupal	Avanzar construcción de API	2 horas
Abril 15	Grupal	Iniciar desarrollo de la aplicación	2 horas
Abril 16	Grupal	Validación de diseño inicial de la aplicación con expertos	0.5 horas
Abril 17	Grupal	Terminar construcción de API	3 horas
Abril 18	Grupal	Terminar desarrollo de aplicación	1 hora
Abril 19	Grupal	Validación de aplicación final con expertos	0.5 horas
Abril 19	Grupal	Terminar documento	1 hora
Abril 20	Individual (Laura)	Grabar video	0.5 horas

Por último, estas fueron los retos identificados y las estrategias usadas para mejorarlos:

- Conectar el API con el front. Para ello se comunicó efectivamente el error mostrado por la aplicación y se exploraron soluciones.

Anexos

- Repositorio GitHub API: <https://github.com/JP-514/fastapi-reviews-app>
- Repositorio GitHub aplicación: <https://github.com/Laurarestrepo03/Proyecto-1-BI>

- Video de resultados: <https://youtu.be/jRnlrz010V4>