Proyecto 1 - Etapa 2 - Yelp

al.bayona - sn.rodriguezc - ed.diaz11

Universidad de los Andes
Facultad de Ingeniería

Samuel Nicolas Rodriguez Celis – Edward David Diaz Fontecha – Andrés Leonardo Bayona Latorre

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

Para automatizar la preparación de los datos decidimos crear un pipeline que englobe todos los pasos del preprocesamiento, incluida la vectorización. Tenemos entonces 2 “pasos” generales.

- Procesamiento de texto

  En esta primera parte tenemos un archivo TextProcessor.py en el que tenemos una función que aplica las transformaciones que consideramos adecuadas para este caso.

  En cuanto al procesamiento “básico” hicimos lo siguiente: eliminar caracteres que no fueran ASCII, convertir todas las letras a minúsculas, remover la puntuación y las ‘stop words’, las cuales son palabras muy comunes que son propias del lenguaje y que no aportan nada en contenido.

  Lo segundo fue normalizar, con base en stemming y lemmatization, logramos que palabras con raíces etimológicas similares compartieran el mismo token o fueran consideradas como la misma palabra. Esto con el objetivo de reducir la complejidad y mejorar la interpretación del modelo.

- Vectorización

  En este punto probamos con varias representaciones (BoW, Tfidf, …) y la que nos arrojaba mejores resultados era BagofWords, en el que cada valor corresponde a la cantidad de veces que un determinado token aparece en una reseña en específico, modificando los atributos de tal manera que los tokens que aparecían con mucha o muy poca frecuencia sobre el total de los datos, fueran ignorados (no aportan y solo generan más complejidad).

En la etapa anterior gracias al uso de GridSearch logramos determinar los parámetros que permitían construir el modelo de regresión logística más preciso, los cuales son:

Solver	newton-cg
C	50
Penalty	L2
PCA components	100

Hasta este punto ya tenemos dos archivos '.joblib', el primero con el pipeline que contiene los pasos de procesamiento de texto y vectorización, y el segundo con el modelo que fue escogido con anterioridad. Así, al momento de que un usuario ingrese una reseña, el texto podrá ser transformado para que el modelo pueda realizar la predicción.

2. Desarrollo de la aplicación y justificación:

Como rol o usuario principal encontramos que pueden ser los administradores de los servicios que son ofrecidos en Yelp, los cuales están interesados en hacer un seguimiento constante a la calidad de los servicios en función a los comentarios que los clientes hagan después de usarlos.

En esta tarea no solo es importante evaluar el grado de satisfacción general que tienen los clientes, sino también encontrar los aspectos más influyentes al momento de evaluar la calidad de un servicio.

Con nuestra aplicación tratamos de cubrir ambas necesidades, ya que podemos predecir qué tan positivo o negativo es un comentario y a la vez mostrarle al usuario las palabras usadas que más relevancia tuvieron al momento de generar la predicción, y así un administrador de un servicio (restaurante, empresa de mantenimiento, …) podrá interpretar qué partes de su negocio pueden estar fallando o cuáles debe seguir implementando en caso de que los comentarios sean positivos.

El usuario ingresará uno o más textos que correspondan a un comentario o reseña, la aplicación generará las predicciones y las mostrará en pantalla junto a las palabras más relevantes y una serie de gráficos con los valores de las métricas de precisión, recall y F1-score para el modelo utilizado, de manera que el usuario tenga suficiente información como para determinar qué tanto confía en la predicción.

3. Resultados:

Link video: https://youtu.be/fjoWNtMqNPc

Al entrar en la aplicación el usuario tendrá la posibilidad de introducir un comentario cualquiera y realizar una predicción sobre nuestro modelo. La cual se mostrará en pantalla (estrellas 1-5) junto dos "mapas" con las palabras que son más influyentes al momento de predecir de manera positiva o negativa. Además, se tienen 3 gráficos que muestran los valores de las métricas para el modelo que se esté usando en el momento, dado que el modelo puede ser reentrenado ingresando un archivo .json con datos nuevos para realizar las siguientes predicciones.

- Una forma de mejorar el funcionamiento de la aplicación sería la habilitación de un socket para poder notificar al usuario cuando el modelo termina de

reentrenarse, debido a que para un gran volumen de datos el proceso puede tardar bastante.

- Una “mejora” que se podría realizar en un futuro consiste en reducir las etiquetas a dos, una positiva y una negativa, de manera que reseñas con etiquetas de 1-3 sean clasificadas como negativas y las de 4-5 como positivas. Esta implementación necesitaría una definición más completa de los objetivos de negocio, ya que puede proveer mejores resultados al momento de predecir, pero también puede ser menos intuitiva o darle menos información al usuario final.

4. Trabajo en equipo:

Integrante	**Rol(es)**	**Puntos/100**
Nicolás Rodríguez	Líder de proyecto, Ingeniero de datos	**30**
David Diaz	Ingeniero de software responsable del diseño de la aplicación y resultados	**35**
Andrés Bayona	Ingeniero de software responsable de desarrollar la aplicación final	**35**