PROYECTO 1: ANALITICA DE TEXTO

Etapa 2

Contenido

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

En base a la etapa 1 se eligió el modelo mas certero, que corresponde a Support Vector Machine(Esto corresponde a la etapa 1)

Ya para la etapa 2, exportamos el modelo.

```
#exportamos el modelo de SVM en joblib
dump(svm_pipeline, 'modelo_svm.joblib')
```

Utilizamos el framework Flask para el desarrollo de las APIs y de la página web. Por temas de sencillez y congruencia con la etapa 1, se hizo el preprocesamiento de datos desde la aplicación para que entre al pipeline del modelo como tal posteriormente.

```
# Función para procesar los datos

def process_data(text):
    return text.apply(porter_stemmer_spanish).apply(join_words)

def limpiar_texto(texto):
    tokenizado = word_tokenize(texto)
    procesado = procesar_texto(tokenizado)
    stemmizado = porter_stemmer_spanish(procesado)
    completado = join_words(stemmizado)
    return completado
```

Después de cargar el modelo que se exportó. Definimos las siguientes 3 plantillas para cargar las diferentes páginas que soporta la aplicación web

```
# Ruta principal para cargar la página HTML
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/csv')
def csv():
    # Lógica para renderizar la plantilla HTML del formulario de carga de CSV
    return render_template('csv.html')

@app.route('/review')
def review():
    # Lógica para renderizar la plantilla HTML del formulario de carga de texto
    return render_template('review.html')
```

Se definieron 2 APIs.Una para predecir una entrada de texto individual/reseña y otra para cargar entradas de texto en masa por formato CSV.Las dos tienen métodos POST y GET.

El primer api tiene una función llamada predecir_text. En el primer bloque if, se verifica si la solicitud es del tipo POST. Esto significa que el cliente está enviando datos al servidor, como el contenido de un formulario. Si es así, se espera que los datos se encuentren en el cuerpo de la solicitud (request body) y se accede a ellos utilizando request.json, que interpreta el contenido como JSON y lo convierte en un diccionario de Python. Luego, se extrae el texto ingresado por el usuario utilizando la clave 'text'.

En el bloque elif, se verifica si la solicitud es del tipo GET. Esto significa que el cliente está solicitando información del servidor, como una página web o datos específicos. Si es así, se espera que los datos estén en la URL de la solicitud y se pueden acceder a través de los argumentos de la URL utilizando request.args.get('text', "). Esto obtendrá el valor del parámetro llamado 'text' en la URL, o un valor predeterminado vacío si no se proporciona ese parámetro. Ya hechas las dos solicitudes, se limpia el texto y se envia al modelo para predecir.

```
# Rutas para la predicción de texto
@app.route('/predict_text', methods=['POST', 'GET'])
def predict_text():
    if request.method == 'POST':
        data = request.json
        text = data['text']
    elif request.method == 'GET':
        text = request.args.get('text', '')

    text = tr.limpiar_texto(text)
    prediction = svm_model.predict([text])[0]
    prediction = int(prediction) # Convertir la predicción a tipo int
    return jsonify({'prediction': prediction})
```

Para el lado de la función predict_csv tiene un funcionamiento muy similar a la función de predecir reseñas individuales, la diferencia principal radica en que esta función tiene incluida una limpieza básica del CSV antes de aplicar el preprocesamiento definido en tratado.py. En las dos funciones se usa la función *jsonify* con el objetivo de que lo retornado se devuelva como un json y se pueda manejar con el javascript de la lógica de la aplicación.

2.Desarrollo de la aplicación y justificación. Descripción del usuario/rol de la organización que va a utilizar la aplicación, la conexión entre esa aplicación y el proceso de negocio que va a apoyar (si aplica), y la importancia que tiene para ese rol la existencia de esta aplicación.

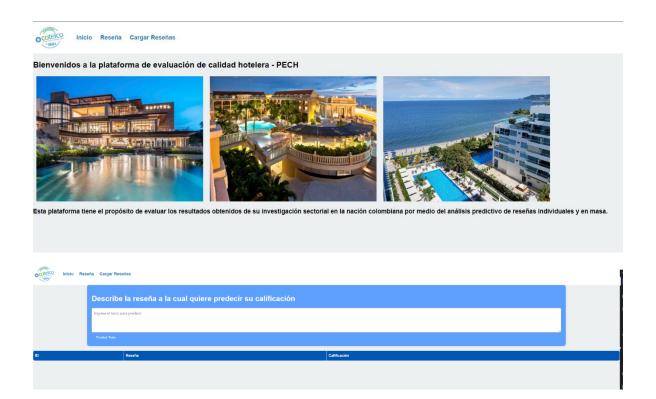
Dentro de la organización de COTELCO la aplicación está destinada principalmente a las subdivisiones/Capítulos de COTELCO y en menor medida a los hoteles de esa área en específico. El proceso al que podría estar relacionado son las investigaciones sectoriales que hace COTELCO a través del CPTUR, que consisten en análisis y seguimiento del turismo en ciertos sectores. La aplicación

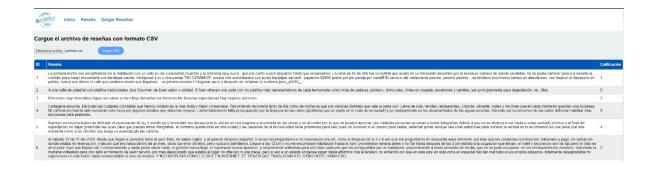
valdría para soportar los reportes de las investigaciones de factores agravantes o beneficiosos en las áreas hoteleras con análisis predictivo.

Esta aplicación es importante para el CPTUR pues les permite validar la información que ellos encuentren al momento de hacer los procesos asociados a sus investigaciones de sector. Por ejemplo, si COTELCO y el CPTUR deciden hacer entrevistas a habitantes de cierta zona preguntándoles su opinión de la seguridad del área, se puede validar dichas entrevistas por medio de reseñas ,sean fabricadas o ya existentes, que contengan lo que se socializó en dichas entrevistas y corroborar resultados.

Ya definido el actor de la organización con los estudiantes de estadística, orientamos el diseño estético de la aplicación hacia COTELCO. Las funcionalidades de la aplicación se limitaron a predecir reseñas individuales y en masa por medio de carga de archivos. Se decidió implementar solo estás funciones debido a que es lo más pertinente en términos de como se entreno el modelo y del entendimiento del negocio. Aunque se pensó en la predicción de reseñas incluir los sectores asociados a esa reseña, no se implementó debido a la falta de datos para implementar esto de manera apropiada por parte del negocio.

Este es el resultado final de la aplicación:





3.Trabajo en Equipo

Se asignaron los siguientes roles:

Líder de Proyecto:Santiago Rodriguez Cruz

Ingeniero de software responsable de desarrollar la aplicación final:Sebastián Umaña Peinado

Ingeniero de datos:Sebastián Umaña Peinado

Ingeniero de software responsable del diseño de la aplicación y resultados:Santiago Rodriguez Cruz, Sebastián Umaña Peinado

Reunión de Lanzamiento y Planeación:8/04/2024 duración 20m

Reunión de Seguimiento:12/04/2024 duración 1h

Reunión De Finalización:19/04/2024 duración 1h30m

Tareas Realizadas:

El desarrollo de cada tarea se hizo en conjunto entre los dos.La ultima tarea se hizo con el apoyo del grupo de estadística.

Exportar Modelo y realizar funciones externas de tratamiento de texto:1h Desarrollo de la API:1h Desarrollo de requerimientos de la aplicación web:3h