Proyecto 1 - Entrega 2

Universidad de Los Andes

Ingeniería de Sistemas y Computación

ISIS-3301 Inteligencia de negocios

Grupo 5

Natalia Ortega	201814519
David Leon	201615216
Juan Camilo Mercado	202021541

Proyecto 1 – Entrega 2

Inteligencia de Negocios – ISIS3301

Presentado por grupo 5: Natalia Ortega, David Leon y Juan Camilo Mercado

0. Tabla de contenido

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

El proceso realizado por el ingeniero de datos consta de 4 etapas. La primera se trata del proceso de automatización de datos, pues, en el modelo construido en la etapa pasada se usaban 3 pasos principales para la preparación de las reseñas para ser procesadas por el modelo, primero se tokenizaban las palabras, lo que significa dividir cada comentario en sus palabras o tokens para poder tratarlos de manera aislada. Se construye la función tokenize() responsable de hacer este proceso, la cual se muestra a continuación:

Figura 1. Código fuente de tokenize()

```python
def tokenize(data: pd.DataFrame):
    data['Words'] = data['Review'].apply(word_tokenize)
    return data
```

Donde dado un DataFrame como el del negocio, se hace una nueva columna llamada ‘Words’ en donde residen los comentarios separados en sus tokens dentro de una lista.

Luego de esto se hace la limpieza de las palabras identificadas que entorpecen el entrenaiento, como lo son las stopwords, signos de puntuación y el uso de mayúsculas, por lo que se construye un objeto capaz de llevar a cabo este proceso. En este mismo se hace el análisis lexicográfico, que consiste en extraer los lemas de las palabras (formas significativas sin importar su conjugación) y se hace a su vez el proceso de ‘stemming’ que consiste en reducir la palabra a su raíz. Se hacen ambos procesos ya que, al preparar el cuaderno se demuestra una mayor efectividad de los clasificadores con el uso de las técnicas ‘lemmatize’ y ‘stemming’ que usar solo una de forma aislada, pues se sabe que el uso de las técnicas depende del contexto de las reseñas y del ejercicio a realizar. Este proceso se hace mediante un objeto de Python responsable de contener cada paso del proceso acumulado en su método preprocess(), donde aplica cada una de las correcciones mencionadas:

Figura 2. Código de la clase TextProcessor

```python
class TextProcessor:
    def __init__(self):
        self.stop_words = set(stopwords.words('spanish'))
        self.lemmatizer = WordNetLemmatizer()
        self.stemmer = SnowballStemmer('spanish')
        self.punctuation = string.punctuation

    def to_lowercase(self, texto):
        return [word.lower() for word in texto]

    def remove_punctuation(self, texto):
        return [word for word in texto if word not in self.punctuation]

    def remove_triple_punctuation(self, texto):
        return [word for word in texto if word != '...']

    def remove_stopwords(self, texto):
        return [word for word in texto if word not in self.stop_words]

    def lemmatize(self, texto):
        return [self.lemmatizer.lemmatize(word) for word in texto]

    def stem(self, texto):
        return [self.stemmer.stem(word) for word in texto]

    def remove_non_ascii(self, texto):
        return [unicodedata.normalize('NFKD', word).encode('ascii', 'ignore').decode('utf-8', 'ignore') for word in texto]

    def preprocess(self, texto):
        texto = self.to_lowercase(texto)
        texto = self.remove_punctuation(texto)
        texto = self.remove_triple_punctuation(texto)
        texto = self.remove_stopwords(texto)
        texto = self.lemmatize(texto)
        texto = self.stem(texto)
        texto = self.remove_non_ascii(texto)
        return texto
```

Para su uso reutilizable se construye una función responsable de crear un objeto de la clase TextProcessor y utilizar su método preprocess() para aplicar el proceso a cada una de las palabras previamente tokenizadas de cada uno de los comentarios con el código que se muestra a continuación y se guardan en una nueva columna llamada 'Stemmed Words' que corresponde a la lista de palabras preprocesadas:

Figura 3. Código de la función clean_process()

```
def clean_process(data: pd.DataFrame):
    processor = TextProcessor()
    data['Stemmed Words'] = [processor.preprocess(text) for text in data['Words']]
    return data
```

En este punto se tiene un vector de raíces sin conjugación de cada una de las palabras significativas, no stopwords del español. Por lo que, en este punto, para el entendimiento del algoritmo se juntan las palabras procesadas en reseñas procesadas y se vectorizan para que la máquina pueda tener entendimiento de ellas. Esto se hace con el vectorizador elegido y entrenado por el modelo, el cual es tf-idf, y se utiliza en una función donde se reúnen las palabras de los comentarios tokenizados y limpios, se hace la vectorización correspondiente y se eliminan las columnas generadas en el proceso de transformar a vectores las reseñas del dataframe, como se muestra en el código posterior:

Figura 4. Código fuente de la función clean_vectorize()

```
def clean_vectorize(data: pd.DataFrame, vectorizer):
    data['Text'] = [' '.join(text) for text in data['Stemmed words']]
    x_etiquetado = vectorizer.transform(data['Text'])
    data.drop(['Text', 'Words', 'Stemmed words'], axis=1, inplace=True)
    return x_etiquetado
```

Estos 3 pasos de transformación se incluyen en un pipeline de preprocesamiento llamado 'preprocessor', donde dado un DataFrame que contenga comentarios en una columna denominada 'Reviews', su resultado es el conjunto de comentarios tokenizados en una nueva columna llamada 'Text', este pipeline se construye de la siguiente forma:

Figura 5. Código del pipeline 'preproccesor'

```
preproccessor = Pipeline(
    [
        ("tokenize", FunctionTransformer(
            tokenize, validate=False)),
        ("clean", FunctionTransformer(
            clean_process, validate=False)),
        ("vectorize", FunctionTransformer(clean_vectorize, validate=False, kw_args={"vectorizer": vectorizer})),
    ]
)
```

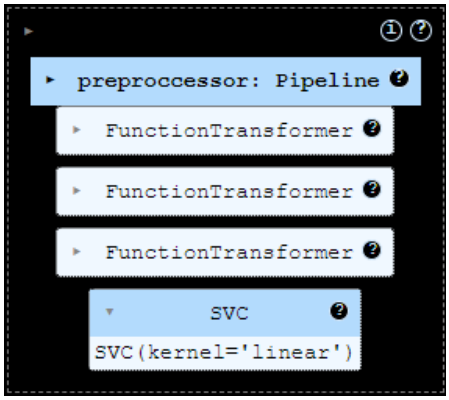
Se puede observar que sigue el proceso de tokenizar, limpiar y vectorizar con las funciones construidas, ya que con este conjunto nuevo se puede proceder a entrenar el mejor modelo hallado en la etapa 1 del proyecto, el cuál es el clasificador SVC con un kernel lineal, el cual demuestra un rendimiento óptimo. Por esto, en un pipeline llamado 'pipeline' se agrega el pipeline de 'preprocessor' y luego el modelo mencionado con los hiperparámetros obtenidos:

Figura 6. Código del pipeline final ‘pipeline’

```
pipeline = Pipeline(
    [
        ("preproccessor", preproccessor),
        ("model", SVC(kernel='linear', probability=False))
    ]
)
```

Siendo el proceso completo el siguiente:

Figura 7. Flujo de ejecución del pipeline final ‘pipeline’



Este pipeline se entrena con el set de datos etiquetados completo para tener la mayor cantidad de datos de entrenamiento posible, el cual, luego de ser entrenado se exporta como un archivo .joblib para su reutilización.

Entonces, para generar una API que dado una reseña le de su calificación, se utilizan dos archivos .py del FrameWork Django, en el cual el primer archivo es responsable de cargar el modelo del joblib y generar una función que, recibiendo una reseña, la etiquete con el modelo y retorne la calificación de esta:

Figura 8. Código fuente de la función predict()

```
def predict(review):
    location = os.path.dirname(os.path.abspath(__file__))
    fullpath = os.path.join(location, 'model.joblib')
    pipeline = load(fullpath)

    new_data = pd.DataFrame({'Review': [review]})

    return pipeline.predict(new_data)[0]
```

Esta función se consume por medio del protocolo HTTP cuando en el front, un usuario hace una petición de predicción de su comentario, lo cual se hace con el siguiente método:

Figura 9. Código fuente de la función review_create()

```python
def review_create(request):

    if request.method == 'POST':
        form = ReviewForm(request.POST)

        if form.is_valid():
            review = form.save(commit=False)
            text = review.review
            review.classification = predict(text)
            review.save()
            messages.add_message(request, messages.SUCCESS,
                                 f'La reseña fue agregada exitosamente con una clasificación de {review.classification}')
            return HttpResponseRedirect(reverse('reviewCreate'))
        else:
            print(form.errors)
    else:
        form = ReviewForm()

    context = {
        'form': form,
    }
    return render(request, 'review_create.html', context)
```

Donde la API se encarga de que, cuando se llame una petición POST dentro de la aplicación se reciba el formulario con la reseña, se procese en el modelo y se guarde en la base de datos de la aplicación, la cual se pasa a un frontend usando Jinja2 por medio del método render.

2. Desarrollo de la aplicación y justificación

Para este apartado se pensó en los usuarios dentro de las organizaciones que podrían usar la aplicación. En primer lugar, hemos diseñado el servicio para servir a varios usuarios dentro del Ministerio de Comercio, Industria y Turismo de Colombia, así como al sector turístico en general. Para el Departamento de Turismo, dentro del ministerio, esta herramienta podría proporcionar una fuente de datos en tiempo real sobre la percepción de los turistas acerca de los destinos en Colombia. Creemos que al recopilar y analizar reseñas, el departamento puede identificar tendencias, áreas de mejora y oportunidades para promover destinos específicos, lo que contribuiría al crecimiento y desarrollo del turismo en el país.

Por otro lado, la Dirección del Presupuesto y Finanzas del Ministerio de Comercio, Industria y Turismo encontraría en la app, una manera de evaluar el impacto financiero de las inversiones en el sector turístico. Esto se puede lograr ya que al analizar las calificaciones y comentarios de los turistas, se puede determinar la efectividad de la asignación de recursos a diferentes destinos y ajustar el presupuesto en consecuencia, maximizando así el impacto de los fondos nacionales destinados.

Las empresas de tecnología y consultoría especializadas encontrarían valor en nuestro servicio, al utilizar los datos recopilados por Reviews app pues pueden ofrecer servicios de consultoría más informados y personalizados a sus clientes del sector turístico. La aplicación les proporcionaría una fuente de datos actualizada y relevante sobre la percepción de los turistas, lo que mejora su capacidad para ofrecer recomendaciones estratégicas.

Finalmente, tanto los ciudadanos como las empresas del sector turístico se benefician de Reviews app al tener acceso a una plataforma donde pueden compartir y consultar opiniones sobre destinos turísticos del país. Esta retroalimentación ayuda a mejorar la experiencia del turista y a mantener altos estándares de calidad en la prestación de servicios turísticos, lo que contribuiría a una industria turística más exitosa y competitiva.

Nos enfocaremos específicamente en las Empresas de Tecnología y Consultoría Especializadas, reconociendo su importancia en el desarrollo y aplicación de soluciones tecnológicas para el sector turístico. Decidimos que la información recopilada se presentará a través de una aplicación, donde las reseñas de los turistas son calificadas mediante un sistema de estrellas. Esta decisión busca facilitar el acceso y comprensión de los datos para estas empresas, permitiéndoles aprovechar eficazmente la información recopilada para ofrecer soluciones más informadas y adaptadas a las necesidades del sector turístico.

Tabla 1. Mapa de beneficiarios de la solución web

Rol dentro del Ministerio	Beneficio
Departamento de Turismo	Mejora en la toma de decisiones relacionadas con la promoción turística, asignación de recursos y políticas públicas para el sector.
Dirección de Presupuesto y Finanzas	Mejora en la eficiencia del gasto público al identificar áreas de inversión más rentables y estratégicas.
Empresas de Tecnología y Consultoría Especializadas	Oportunidad de desarrollar y ofrecer soluciones tecnológicas y consultoría especializada en análisis de datos y modelos predictivos para el sector turístico.
Ciudadanos y Empresas del Sector Turístico	Acceso a información y análisis para tomar decisiones más informadas y aprovechar la oferta turística en Colombia.

Por último, el desarrollo de la aplicación web se realiza por medio de servicios construidos con Django, donde se tienen 4 pestañas principales:

- **Inicio**: página estática donde se puede ver el contexto resumido de la aplicación, el objetivo de negocio y la historia de usuario guía.
- **Contexto**: página estática establece el objetivo de esta etapa, se presenta nuevamente los actores beneficiados y el contexto del negocio.
- **Predicciones**: Página donde se pueden observar las reseñas de la base de datos y hacer predicciones de nuevas reseñas en el endpoint /reviewcreate/, donde introduciendo una reseña la página otorga una calificación al modelo correspondiente (por medio del API) y la persiste.
- **Filtrar reseñas**: Página donde se pueden observar los comentarios por grupos de tipo, como clase 1 y 2 o 4 y 5 juntos, otorgando el valor al negocio de ver cuáles son buenos y cuales son malos (en general). Igualmente se pueden obtener los comentarios de una clase en específico.

El código de la aplicación se encuentra en el repositorio correspondiente al proyecto, pero, además de esto se despliega en un servidor Render de baja capacidad para exponer los servicios al público por medio del siguiente link: **https://reviewsapp.onrender.com/**

3. Resultados

Para esta segunda etapa del proyecto se automatizo el desarrollo de analítica de textos creado en la primera etapa. Para esto se creó una aplicación más orientada al usuario, que en este caso serían las estudiantes de estadística quienes la estarían probando.

En esta podemos encontrar una página de inicio que nos habla del objetivo y algunos aspectos técnicos utilizados como el algoritmo de clasificación, la técnica de vectorización y algunas métricas consideradas, además se presenta el requerimiento a modo de historia de usuario usado para desarrollar este proyecto. Ya que esta sección tiene un lenguaje más técnico, se creó la sección de contexto donde se habla un poco más en palabras cotidianas de lo que trata el proyecto. En este también se habla del objetivo y de los actores involucrados en el proyecto. Y también se presente el beneficio que tendrían estos al usar la aplicación

En la pestaña de predicciones donde encontramos las reseñas hechas junto con su calificación de 1 a 5 estrellas. En esta pestaña también encontramos un botón donde como usuario podemos poner nuestras propias reseñas y el programa podrá decir la calificación que tiene esta. Esta reseña queda en el programa, y aunque se cometan errores de ortografía aún puede clasificar el comentario.

Por último, en la pestaña de filtro de reseñas donde podemos ver a las reseñas clasificadas en malas, regulares y buenas. Donde en las malas se encuentras las reseñas clasificadas entre 1 y 2 estrellas, las regulares corresponden a las reseñas clasificadas con 3 estrellas, y las buenas corresponden a las reseñas con una puntuación de 4 a 5 estrellas. Y tambien podemos filtrarla segun su calificación exacta.

Algunos comentarios generales de la aplicación hechas por las estudiantes de estadística se encuentran que la aplicación fue muy intuitiva y rápida y que la pestaña de contexto fue muy útil para entender mejor el porqué y para quién fue diseñada la aplicación, además de su estética.

4. Elementos del equipo

Tabla 2. Tabla de participación grupo BI

Actividad	Camilo Mercado	Natalia Ortega	David Leon
Implementación de automatización de preparación de datos		**X**	
Construcción de un Pipeline		**X**	

Construcción de la API	**X**		**X**
Validación de proceso y resultados de negocio		**X**	**X**
Desarrollo de aplicación web	**X**		**X**
Realización de video	**X**		
Aporte total	**33,34%**	**33,33%**	**33,33%**

Dedicación horaria: Camilo = 7 horas aprox, Natalia = 6 horas aprox, David = 6 horas aprox

Tabla 3. Tabla de participación grupo Estadística

Actividad	Sofia Botia	Sophia Feghali
Análisis de requerimientos de la app	**X**	**X**
Análisis de tabla de actores	**X**	**X**
Ideación módulos de la app	**X**	
Evaluación de usabilidad de la app	**X**	**X**
Identificación de usuario final	**X**	
Evaluación de disposición de resultados	**X**	**X**
Aporte total	**50%**	**50%**

Tabla 4. Roles desempeñados por cada integrante

Estudiante	Curso	Rol
Sofía Botia	Estadística	Líder del proyecto
Sophia Feghali	Estadística	Líder del proyecto
Natalia Ortega	Inteligencia de negocios	Ingeniera de datos
Camilo Mercado	Inteligencia de negocios	Ingeniero de software responsable del diseño de la aplicación y resultados
David Leon	Inteligencia de negocios	Ingeniero de software responsable de desarrollar la aplicación final

Tabla 5. Planeación de reuniones

Reunión	Fecha	Descripción
Reunión de lanzamiento y planeación	16/04/24	Se establecen objetivos, alcance, roles, requisitos del cliente y plan de acción. Se asignan roles, se delinean hitos clave, se identifican riesgos y se establecen estrategias de mitigación.
Reunión de ideación	17/04/24	Se generan ideas y soluciones para el análisis de emociones, promoviendo la participación abierta y la colaboración entre los asistentes. También se establecen los siguientes pasos para desarrollar y evaluar más adelante.

Reunión de seguimiento	18/04/24	Se realiza el seguimiento del proyecto, revisando el progreso, identificando desafíos y estableciendo medidas correctivas. Se asignan responsabilidades para la etapa final y se programa la próxima reunión de finalización.
Reunión de finalización	19/04/24	Se evalúa la finalización del proyecto, revisando logros y objetivos. Se analizan lecciones aprendidas, se destacan puntos fuertes y áreas de mejora, y se planifica la entrega del resultado.

5. Link del video y del servicio web

- **Link del video**: https://www.youtube.com/watch?v=NiXghKCUQaU
- **Link del servicio web**: https://reviewsapp.onrender.com/

6. Referencias

[1] J. Vinagre Triguero, «CLASIFICACIÓN DE COMENTARIOS,» Universitat de Barcelona, Barcelona, 2023.

[2] M. Campos Mocholí, «Clasificación de textos,» Universitat Politécnica de Valéncia, Valencia, 2021.

[3] J. Amat Rodrigo, «Ciencia de datos,» Abril 2017. [En línea]. Available: https://cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines. [Último acceso: 2 Abril 2024].