1. **Identificación de roles y tareas**

Los roles dentro del proyecto son:

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Encargado** | **Tareas** |
| Líder de proyecto | María José Cely | 1.Definir las fechas de reuniones, pre-entregables del grupo  2. Verificar las asignaciones de tareas para que la carga sea equitativa  3. Subir el trabajo |
| Ingeniero de software del diseño de la aplicación y resultados | Alejandro González | 1. Liderar el diseño de la aplicación  2. Grabar el video con los resultados obtenidos. |
| Ingeniero de datos | Javier Serrano | 1. Velar por la calidad del proceso de automatización |
| Ingeniero de software responsable de desarrollar la aplicación final | Alejandro González | 1. Gestionar el proceso de construcción de la aplicación |

Además, se realizaron las siguientes tareas:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tarea** | **Encargado** | **Número de horas dedicadas** | **Puntaje** |
| Automatización de la preparación de datos, construcción del modelo, persistencia del modelo y acceso por medio de API | María José Cely | 8 | 33.33 |
| Desarrollo de aplicación y justificación | Alejandro González | 8 | 33.33 |
| Resultados | Javier serrano | 8 | 33.33 |

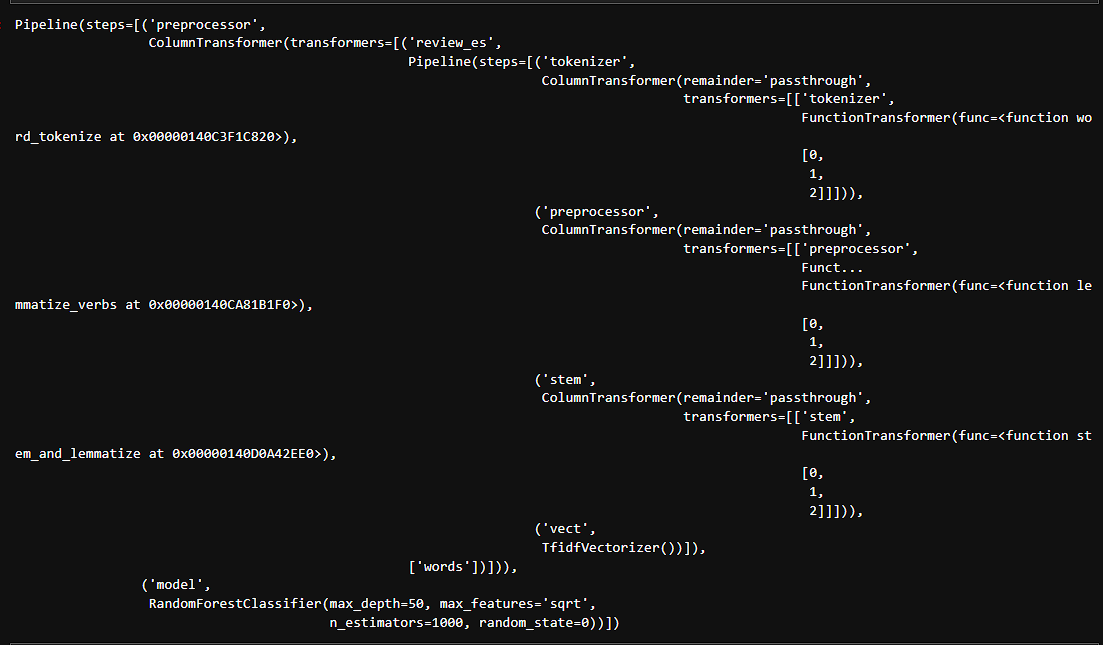
|  |  |  |
| --- | --- | --- |
| **Actividad** | **Encargado** | **Tareas** |
| Reunión de lanzamiento y planeación: | María José Cely | 1. Definición de roles 2. Pactos de grupo 3. Definición de tiempos 4. Ideas para el trabajo |
| Reunión de ideación: | María José Cely | 1. Revisión de tareas asignadas 2. Asignación de nuevos entregables 3. Comunicación con el chico de Estadística 4. Corrección de errores |
| Reuniones de seguimiento | María José Cely | 1. Revisión de cumplimiento de asignaciones 2. Revisión de faltantes |
| Reunión de finalización | María José Cely | * 1. Revisión del proyecto completo   2. Revisar que se puede mejorar |

**2. Automatización de la preparación de los datos, construcción del modelo, persistencia del modelo y acceso por medio de la API**

Para esta segunda etapa se utilizó el proceso de preparación de datos realizado en la anterior etapa. Es decir, se siguieron los mismos procesos: 1) tokenización, 2) preprocesamiento (caracteres ASCII, minúsculas, no signos de puntuación, número en palabras, se removieron stopwords), 3) lematización y stemming y 4) vectorización.

Se tomó el mejor de los cuatro modelos construidos en la etapa anterior: el Random Forest. Se seleccionó este modelo debido a que presentaba mejores métricas

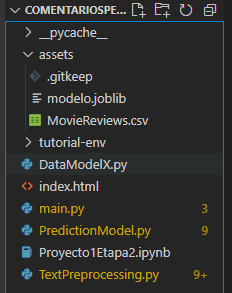
Para esta etapa precisamente se adaptaron dichas transformaciones y modelos para construir un pipeline serializable que pueda ser persistido. Se utilizaron las librerías ColumnTransformer, FunctionTransformer y Pipeline para realizar esto. Luego de construir el Pipeline asociado se obtuvo el siguiente diagrama:



Como se puede observar, el Pipeline se compone de cinco transformadores: tokenizer, preprocessor (explicado anteriormente), lemm, stem y el vectorizer, y además de un modelo de regresión logística.

Este modelo se serializó usando la función dump() mediante el nombre modelo.joblib.

Para la construcción de la API se utilizó el framework FastAPI(). Para esto se creó un proyecto en python con el siguiente directorio:



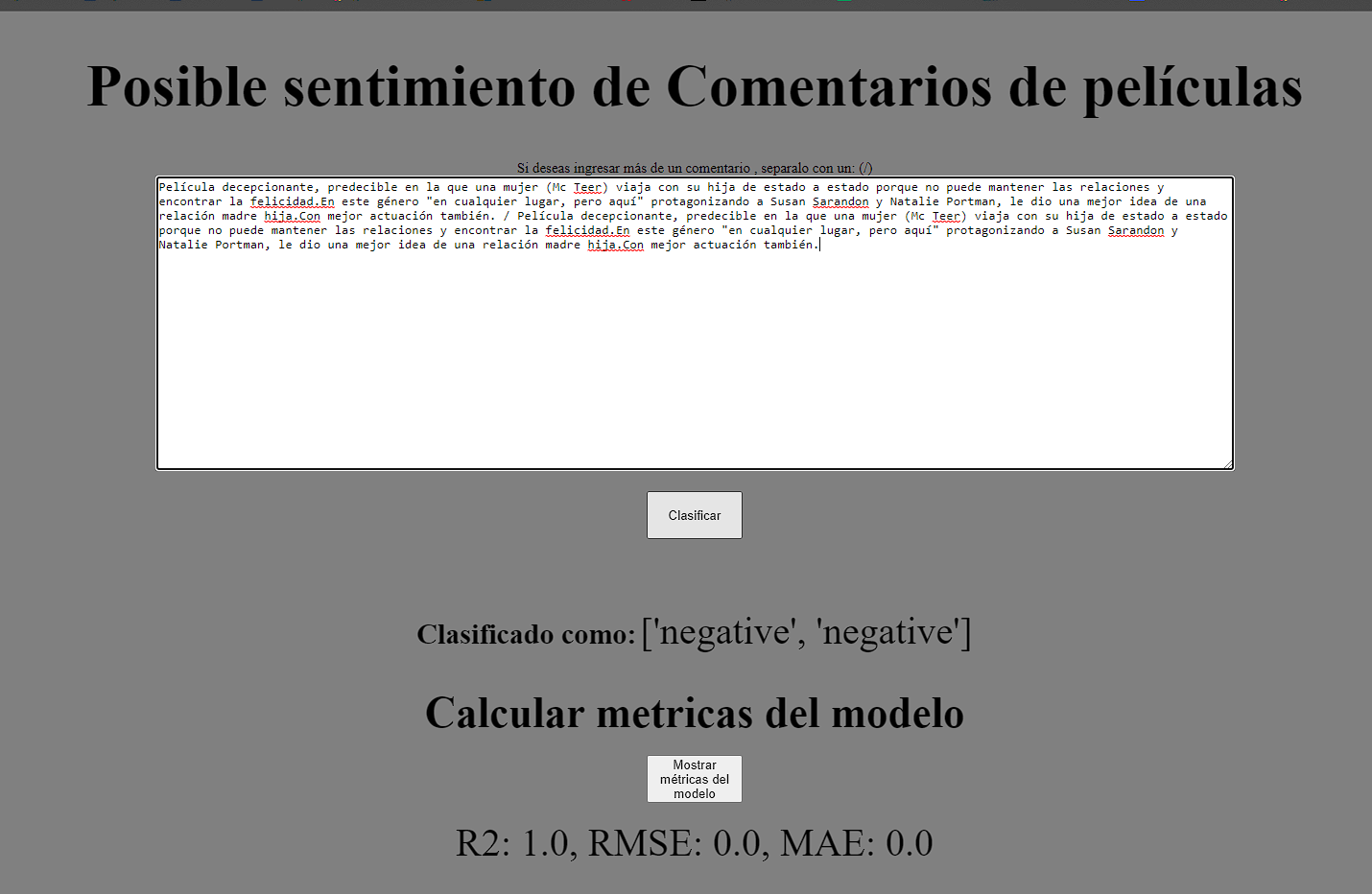
Como archivos específicos de la aplicación está el main.py, encargado de crear la API y definir los endpoints para cada una de los dos servicios o acciones definidas y la dirección de la ruta para la aplicación web creada que se explicará más adelante.

Se crearon dos servicios específicos como funcionalidades de la aplicación. El primer endpoint (definido con la ruta /endpoint1) se encarga de recibir uno o más textos ingresados por el usuario, meterlos en el modelo y predecir su clasificación entre las clases “positivo” o “negativo” y devolver ese resultado. El segundo endpoint (definido con la ruta /endpoint2) se encarga de calcular las métricas de desempeño como el R2 principalmente.

Siguiendo en la descripción de los archivos importantes de la aplicación, el archivo DataModelX.py define el tipo de entrada que va a tener el endpoint1, que es simplemente un texto o más bien su contenido. Por otro lado, el archivo PredictionModel.py es el encargado de toda la lógica. Este tiene el pipeline, los datos, y demás librerías y herramientas necesarias para transformar los datos, ingresarlos, predecir y calcular las métricas.

1. **Desarrollo de la aplicación y justificación**

Utilizando el modelo y la API creada se creó también una aplicación web como herramienta para que un usuario potencial pueda utilizar los servicios definidos. Se creó una página web utilizando lenguaje HTML y JavaScript. La aplicación web tiene un diseño sencillo e intuitivo para que el potencial usuario pueda usarla de la mejor manera. Para acceder a ella simplemente se debe iniciar el servidor uvicorn y acceder en el navegador a la siguiente dirección: <http://127.0.0.1:8000/>. A continuación, se muestra su diseño:



Para utilizar el endpoint1 el usuario simplemente debe insertar el texto a analizar en el cuadro de texto y presionar el botón “Clasificar”. Ahí mismo aparecerá un aviso diciendo que el modelo estará procesando la información y después de unos segunos devolverá en el label “Clasificado como:” si lo clasifica como una reseña positiva o negativa. Para ingresar más de un texto el usuario debe ingresar los textos en el cuadro separados por un slash (/).

Para utilizar el endpoint2 el usuario solo tiene que presionar el botón “Mostrar métricas del modelo” y el programa botará un aviso indicando que las métricas se están calculando. Después de calcularlas aparecerán abajo del botón las tres métricas escogidas, para medir la calidad del modelo.

El usuario potencial de la aplicación puede ser alguien que necesite verificar el sentimiento de varias reviews de películas de su sitio web y así poder automatizar ciertos procesos para la identificación de sentimiento de estas reviews y ver cuál es la proporción de los usuarios , así facilitando esta tarea a largo plazo , generando oportunidades del negocio para saber por ejemplo que tipo de películas o series gustan más entre los espectadores.

1. **Resultado**

Se realizó un video explicando el uso de la aplicación, este, así como todos los archivos de desarrollo se pueden encontrar en el GitHub del proyecto

Las 2 acciones que puede realizar como resultado de la interacción son :

Acción 1: Evaluar la opinión del público

Las empresas de películas pueden aprovechar la clasificación de las reseñas brindada por la herramienta para evaluar cómo sus películas son recibidas por el público. Al introducir reseñas en la plataforma, obtendrán una rápida clasificación que les permitirá comprender si la reacción del público es mayormente positiva o negativa. Esta información puede ser valiosa para adaptar sus estrategias de marketing, considerando el sentimiento de los espectadores, y optimizar su enfoque en la promoción y la comunicación.

Acción 2: Detectar oportunidades de mejora en sus producciones

Gracias a la clasificación proporcionada por la herramienta, las compañías pueden identificar tendencias en las reseñas, tanto positivas como negativas, y descubrir oportunidades de mejora en sus películas. Al examinar las reseñas negativas, las empresas pueden encontrar elementos recurrentes en las críticas y concentrarse en mejorar esos aspectos en futuras producciones. De igual manera, al analizar las reseñas positivas, pueden reconocer los aspectos sobresalientes de sus películas y seguir sacando provecho de esos elementos en futuros proyectos.

Link video: https://uniandes.padlet.org/mavillam/exposici-n-proyecto-anal-tica-de-texto-de-bi-202310-44pma9b354htyill/wish/2573014178

**Puntos para mejorar para próximos trabajos:**

1. Organizar sesiones de estudio en grupo: Establecer horarios regulares para que los estudiantes se reúnan y trabajen juntos en tareas o proyectos, compartiendo conocimientos y habilidades.
2. Establecer normas de comunicación claras: Acordar reglas básicas sobre cómo y cuándo comunicarse, por ejemplo, responder a los mensajes en un plazo determinado o evitar discutir temas no relacionados en los chats de estudio.
3. Fomentar la retroalimentación abierta y honesta: Crear un ambiente en el que los estudiantes se sientan cómodos compartiendo sus opiniones y preocupaciones sobre el trabajo en equipo o el curso en general.
4. Aprovechar las habilidades y conocimientos de cada miembro: Identificar y aprovechar las habilidades y conocimientos individuales de cada miembro del grupo, asignando tareas y responsabilidades según sus fortalezas.
5. Designar un líder o coordinador de grupo: Elegir a un miembro del grupo para que actúe como líder o coordinador, facilitando la comunicación y ayudando a mantener a todos enfocados y organizados.
6. Establecer objetivos y plazos claros: Asegurarse de que todos los miembros del equipo conozcan las expectativas y plazos para las tareas asignadas. Esto permitirá que cada miembro planifique adecuadamente su tiempo y priorice las tareas.
7. Programar reuniones de seguimiento regulares: Realizar reuniones de equipo regulares para discutir el progreso, abordar problemas o inquietudes y mantener a todos en la misma sintonía.

**Retos y soluciones presentados en el proyecto:**

**Reto 1:** Desbalance de clases en los datos de entrenamiento

Solución: Utilizar técnicas de remuestreo, como sobremuestreo de la clase minoritaria o submuestreo de la clase mayoritaria, para equilibrar las clases en los datos de entrenamiento. También se puede aplicar el aprendizaje con ponderación de clases para que el modelo preste más atención a la clase menos representada.

**Reto 2:** Procesamiento y limpieza de texto

Solución: Aplicar técnicas de preprocesamiento de texto, como la eliminación de caracteres especiales, puntuación, tokenización, eliminación de palabras vacías (stopwords) y stemming o lematización, para mejorar la calidad de los datos y aumentar la eficacia del modelo.

**Reto 3:** Elección del algoritmo de Machine Learning adecuado

Solución: Probar diferentes algoritmos de clasificación, como regresión logística, Naive Bayes, máquinas de vectores de soporte (SVM) y redes neuronales, para determinar cuál funciona mejor para el problema de clasificación de comentarios. Realizar validación cruzada y ajuste de hiperparámetros para optimizar el rendimiento del modelo.

**Reto 4:** Evaluación y mejora del rendimiento del modelo

Solución: Utilizar métricas como precisión, recall, F1-score y área bajo la curva ROC (AUC-ROC) para evaluar el rendimiento del modelo. Investigar enfoques para mejorar el rendimiento, como el ajuste de hiperparámetros, la incorporación de características adicionales o el uso de modelos de lenguaje pre-entrenados, como BERT.

**Reto 5:** Creación de una API fácil de usar y escalable

Solución: Utilizar frameworks como Flask o FastAPI para desarrollar una API REST que permita a los usuarios enviar comentarios y recibir predicciones del modelo. Asegurarse de que la API sea escalable y pueda manejar múltiples solicitudes simultáneas.