

MovIA: Proyecto desarrollo de un Sistema de Recomendación de Películas con IA

Fase 1: Preparación del Entorno y Datos

- Preparamos el entorno de desarrollo en Python, Visual Studio Code, Anaconda y pytorch.
- **Aprendizaje de Python:** Realizamos diferentes ejercicios independientes para demostrar y practicar los conceptos del lenguaje.
- Recopilamos varios dataset que nos ayudarán a desarrollar el proyecto.
- Entendimiento y discusión de la capa de negocio (valor, desafíos, etc.)
- Realizamos algunos ejercicios independientes con numPy y pandas para introducir el uso de estas librerías.

Fase 2: Análisis Exploratorio de Datos (EDA) y Procesamiento del Lenguaje Natural (NLP)

Realizamos un EDA para entender la naturaleza de los datos que hemos recopilado la semana anterior. Implementamos procesamiento básico de texto como tokenización, eliminación de stopwords, y stemming utilizando bibliotecas como Pandas y NLTK. Introducción a spaCy para procesamiento del lenguaje natural en español: tokenización y análisis de dependencias. Comparación con NLTK para destacar las fortalezas de cada librería.

Librerías/Bibliotecas: pandas, NLTK, spaCy, seaborn, matplotlib.

Énfasis en la exploración continua de datos a través de diferentes técnicas, teniendo en cuenta sesgos, irregularidades y otras características de los datos.

Fase 3: Generación y Análisis de Sentimientos

- Implementamos una red Generative Adversarial Networks (GAN) para generar nuevas reviews o contenido basado en los datos recopilados. Además, construiremos un modelo de análisis de sentimientos utilizando una arquitectura como BERT.

Librerías/Bibliotecas: PyTorch, TensorFlow, GANs, Transformers, veremos algunos ejemplos adicionales con stable diffusion y otros modelos.

Fase 4: Sistema de Recomendación

- Diseñamos un sistema de recomendación que, basándose en el perfil del usuario y el análisis de sentimientos del contenido, recomiende nuevos contenidos generados o existentes. Exploramos embeddings de texto para mejorar la calidad de las recomendaciones.

Librerías/Bibliotecas: scikit-learn, gensim (para Word2Vec), transformers (para BERT).

Fase 5: Despliegue y Consumo de Modelos

- Preparamos nuestro modelo para el despliegue en una plataforma en la nube con AWS Sagemaker (a valorar según requisitos). Creamos APIs para servir el modelo y desarrollamos un front-end sencillo para interactuar con el sistema de recomendación.
 - Preparación del modelo para el despliegue utilizando AWS SageMaker:
 - Creación de un modelo SageMaker, configuración de los recursos necesarios, y despliegue del modelo en un endpoint de SageMaker.
 - Desarrollo y prueba de APIs para interactuar con el endpoint de SageMaker y obtener recomendaciones.
 - Desarrollo de un front-end sencillo:
 - Creación de una interfaz de usuario sencilla, como una página web, donde los usuarios puedan introducir sus preferencias y recibir recomendaciones.
 - Integración de la API desarrollada anteriormente en el front-end para obtener recomendaciones en tiempo real desde el modelo desplegado en SageMaker.
- **Técnicas y Herramientas:**
 - AWS SageMaker para desplegar el modelo y gestionar endpoints.
 - AWS Lambda y API Gateway para crear y gestionar APIs.
 - Frameworks de desarrollo web como Flask o Django para construir el front-end.
 - Herramientas de desarrollo front-end como HTML, CSS, y JavaScript.
-

Fase 6: Optimización y Monitoreo

- Implementamos técnicas de optimización para mejorar el rendimiento del sistema. Establece métricas de monitoreo para evaluar la eficacia del sistema en tiempo real. A/B testing sobre la solución implementada.

Librerías/Bibliotecas: scikit-learn, TensorBoard, MLflow

Fase 7: Documentación y Reflexión

- Documentamos todo el proceso, desafíos enfrentados, soluciones implementadas y el rendimiento del sistema. Reflexionamos sobre las implicaciones y eficiencia del sistema desarrollado, especialmente en lo que respecta a la IA generativa y el análisis de sentimientos.

Discusión sobre la interpretabilidad de los modelos y las recomendaciones generadas. Entender cómo el modelo está tomando decisiones puede ayudar a diagnosticar problemas, identificar sesgos y hacer mejoras en el sistema.

- **Librerías/Bibliotecas:** Sphinx, Jupyter Notebook. Breve introducción a Local Interpretable Model-agnostic Explanations y SHapley Additive exPlanations.