

Índice del TFG. la aplicada al ecommerce.

Portada

- Título del trabajo
- Autor
- Tutor
- Departamento y universidad
- Fecha
- Logotipos institucionales

Agradecimientos

Resumen

- Breve resumen del trabajo en español (150-250 palabras).

Abstract

- Resumen del trabajo en inglés.

Índice de contenidos

Índice de figuras

Índice de tablas

Glosario de términos y acrónimos

1. Introducción

1.1. Motivación del trabajo

- Justificación de la elección del tema.
- Problemas actuales que aborda el proyecto.

1.2. Objetivos del proyecto

- Objetivo general.
- Objetivos específicos (detallados y medibles).

1.3. Metodología de trabajo

- Enfoque de desarrollo (ágil, cascada, etc.).
 - Herramientas y tecnologías utilizadas.
 - Planificación y fases del proyecto.
-

2. Estado del Arte

2.1. Inteligencia Artificial Aplicada al eCommerce

- Definición y evolución de la IA en el comercio electrónico.
- Aplicaciones más comunes de la ia en ecommerce:
 - Chatbots y asistentes virtuales.
 - Análisis de sentimientos.
 - Sistemas recomendación.

2.2. Sistemas de Recomendación

2.2.1. Filtrado Colaborativo

- Conceptos clave.
- Métodos basados en usuarios y en ítems.
- Ventajas y desventajas.

2.2.2. Sistemas Basados en Contenido

- Análisis de características y atributos de los ítems.
- Técnicas de procesamiento de lenguaje natural.
- Limitaciones.

2.2.3. Enfoques híbridos

- Combinación de métodos colaborativos y basados en contenido.
- Casos de éxito y aplicaciones prácticas.

2.3. Limitaciones y desafíos actuales

- Problemas de escalabilidad y rendimiento.
- Manejo de datos escasos o dispersos (sparsity).
- Sesgos y problemas éticos en las recomendaciones.

2.4. Casos de uso en grandes plataformas

- **Amazon.**
 - **Netflix.**
 - **Spotify.**
 - **Alibaba.**
 - Análisis comparativo y conclusiones generales.
-

3. Análisis de requisitos

3.1. Requisitos funcionales

3.2. Requisitos no funcionales

- **Rendimiento.**
- **Escalabilidad.**
- **Seguridad.**
- **Usabilidad.**
- **Compatibilidad.**

3.3. Requisitos de información

- **Datos de usuario.**
- **Datos de productos.**
- **Datos de interacción.**

3.4. Requisitos legales y éticos

- Cumplimiento con el RGPD.
- Políticas de privacidad y manejo de cookies.
- Consentimiento informado para la recopilación de datos.

4. Estimación de Costos

4.1. Infraestructura

- Servidores y almacenamiento.
- Bases de datos.
- Redes y seguridad.

4.2. Desarrollo

- Horas hombre estimadas.
- Costos de licencias y herramientas de desarrollo.

4.3. Mantenimiento

- Actualizaciones de software.
- Soporte técnico y corrección de errores.
- Escalabilidad futura.

4.4. Despliegue en plataformas Cloud

- **AWS:** Servicios como EC2, S3, RDS.
 - **Comparativa:** Ventajas, desventajas y costos asociados.
-

5. Diseño del Sistema

5.1. Arquitectura General

- Diagrama de arquitectura del sistema.
- Interacción entre frontend, backend y base de datos.
- Uso de APIs y servicios externos.

5.2. Diseño del sistema de recomendaciones

- **Elección del modelo:** Justificación (colaborativo, basado en contenido, híbrido).
- **Algoritmo de machine learning:** Detalle técnico y matemático.
- **Flujo de datos:**
 - Preprocesamiento.
 - Entrenamiento y actualización del modelo.
 - Generación y entrega de recomendaciones.

5.3. Gestión de cookies y seguimiento

- Implementación técnica del seguimiento.
- Diferenciación entre cookies propias y de terceros.
- Mecanismos para garantizar la privacidad del usuario.

5.4. Modelado de datos

- Diseño de esquemas de bases de datos.
 - Normalización y optimización de consultas.
-

6. Desarrollo del backend

6.1. Desarrollo con Django

- Configuración inicial del proyecto.
- Organización de aplicaciones y módulos.
- Implementación de vistas y controladores.

6.2. Implementación de la API REST

- Uso de Django REST Framework.
- Definición de endpoints y serializadores.
- Gestión de autenticación y permisos.

6.3. Algoritmos de machine learning

- Desarrollo de modelos con librerías como Scikit-learn o TensorFlow.
- Entrenamiento, validación y ajuste de hiperparámetros.
- Integración del modelo en el flujo de datos del backend.

6.4. Gestión de datos

- Manipulación con Pandas y NumPy.
 - Pipelines de procesamiento de datos.
 - Estrategias para manejar grandes volúmenes de información.
-

7. Desarrollo del frontend

7.1. Estructura del frontend

- Diseño de la interfaz con HTML5 y CSS3.
- Uso de **bootstrap** para estilos y componentes.
- Implementación de interactividad con **HTMX**.

7.2. Integración con el backend

- Consumo de la API REST mediante AJAX.
 - Actualización dinámica de contenidos.
 - Manejo de estados y sesiones del usuario.
-

8. Pruebas y validación

8.1. Pruebas unitarias y de integración

- Desarrollo de pruebas con frameworks como PyTest.
- Cobertura de código y análisis de resultados.
- Integración continua.

8.2. Evaluación del sistema de recomendación

- **Métricas de Precisión:**
 - RMSE (Root Mean Square Error).
 - MAE (Mean Absolute Error).
 - **Métricas de Clasificación:**
 - Precisión.
 - Recall.
 - F1-Score.
 - **Análisis Cualitativo:**
 - Feedback de usuarios.
 - Estudio de casos.
-

9. Despliegue

9.1. Contenedorización con Docker

- Creación de dockerfiles para backend y frontend.
- Configuración de Docker Compose para orquestación.
- Pruebas locales de los contenedores.

9.2. Despliegue

- Despliegue en AWS Elastic Beanstalk o EC2.
- Uso de plataformas como kubernetes para orquestación.
- Costos y escalabilidad.

9.3. Mantenimiento y Actualizaciones

- Estrategias de despliegue continuo (CI/CD).
 - Gestión de versiones y rollbacks.
 - Plan de contingencia ante fallos.
-

10. Conclusiones

10.1. Resumen de Logros del Proyecto

- Evaluación del cumplimiento de objetivos.
- Impacto obtenido en la experiencia de usuario.

10.2. Aprendizajes

- Lecciones aprendidas durante el desarrollo.

10.3. Limitaciones del sistema

- Aspectos que podrían mejorarse.
- Restricciones técnicas y operativas.

10.4. Líneas de trabajo futuro

- Integración de algoritmos avanzados (deep learning).
- Personalización y adaptabilidad del sistema.

Bibliografía

- Formato según normas APA, IEEE u otro estándar requerido.
 - Referencias a libros, artículos científicos, sitios web y otros recursos utilizados.
-

Apéndices

A. Código Fuente

- Enlace al repositorio (GitHub, GitLab).
- Estructura del proyecto y guía de navegación.
- Licencia de uso.

B. Documentación Técnica

- Manual de instalación y configuración.
- Documentación de la API (Swagger, OpenAPI).
- Especificaciones técnicas detalladas.

C. Resultados de pruebas

- Informes completos de pruebas unitarias e integración.
- Gráficos y tablas de métricas de evaluación.

D. Plan de Proyecto

- Cronograma detallado.
- Diagramas de Gantt.
- Gestión de riesgos y mitigaciones.