Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

Posgrados

Tecnológico de Monterrey

4.3 Avance de proyecto 1: Sistema de Recomendación

Abril Yusely Cota Jaquez A01795114
Gabriel Paredes Garza A00797698
Carlos Daniel Villena Santiago A01795127

17 de mayo del 2024

Proyecto

Optimización de Comercio Electrónico: Incorporación de Big Data para Maximizar la recomendación de Videojuegos

Descripción del Proyecto

Objetivo general

El objetivo principal de este proyecto es desarrollar un modelo predictivo robusto y eficaz utilizando técnicas de Big Data para identificar de acuerdo a gustos y compras de videojuegos, que videojuegos son los que se recomiendan adquirir.

El mundo de los videojuegos ha revolucionado en los últimos años, con la creación de plataformas de descargas de Videojuegos para ordenador como lo es **Steam**, haciendo que la adquisición y juego de Videojuegos sea más fácil y accesible a nivel mundial.

Este proyecto surge a partir de la necesidad creciente de facilitar la obtención de Videojuegos para los usuarios, pero no solamente eso, sino que sean adquisiciones que sean totalmente del gusto de estos. El uso del conjunto de datos **"Steam Video Game and Bundle Data"** proporciona una base sólida para desarrollar algoritmos que mejoren la precisión y la eficacia en la identificación de Videojuegos con alta probabilidad de ser adquiridos en base a la compra de otros Videojuegos.

Metodología

1. **Preprocesamiento de Datos:**
    - Limpieza de datos para eliminar registros incompletos o erróneos.
    - Anonimización de datos para proteger la privacidad del paciente.
    - Normalización de características para garantizar la uniformidad en el análisis.
2. **Análisis Exploratorio de Datos (EDA):**
    - Análisis estadístico para describir las características básicas de los datos.
    - Visualización de datos para identificar patrones o anomalías potenciales.
3. **Desarrollo del Modelo Predictivo:**
    - Implementación de varios modelos de aprendizaje automático, incluyendo regresión logística y redes neuronales.
    - Comparación y selección del modelo basado en métricas de rendimiento como AUC-ROC y precisión.
4. **Validación y Optimización del Modelo:**
    - Validación cruzada para evaluar la estabilidad y la generalización del modelo.
    - Ajuste de hiperparámetros para optimizar el rendimiento.
5. **Implementación y Seguimiento:**
    - Desarrollo de un sistema de alertas tempranas para uso en entornos clínicos.
    - Monitoreo y evaluación continua del sistema implementado.

Cronograma

	Actividades	Actividades			
		Mayo			
		Semana 1	Semana 2	Semana 3	Semana 4
Fase 1	Generacion descripcion del proyecto		07		
	Seleccion y analisis de Data Set (Preprocesamiento).		10		
	Generar descripcion del conjunto de datos.		10		
	Exploracion y analisis de conjunto de datos.			14	
	Experimentacion con algoritmo de recomendacion.			15	

Impacto Potencial

El impacto esperado en este proyecto incluye la mejora en la experiencia del cliente a la hora de adquirir Videojuegos en la plataforma de **Steam**, además de, incrementar las ventas y mejorar KPI's relativos a la Retención de Usuario, Engagement, Satisfacción del Cliente, Ratio de Abandono, etc.

Descripción del conjunto de datos

Este conjunto de datos contiene registros relatico a las compras y recomendaciones de Videojuegos hechas por medio de la plataforma Steam. Estos conjuntos de datos lo podemos dividir en tres Datasets: **User and Item Data, Review Data y Bundle Data.**

User and Item Data

1. **user_id** – Id del Usuario.
2. **items_count** – Total de items adquiridos.
3. **steam_id** – Id de Steam del Usuario.
4. **user_url** – URL perteneciente al Usuario.
5. **items** – Items adquiridos por el Usuario.
    5.1. **item_id** – Id de Item (Videojuego)
    5.2. **item_name** – Nombre de Item (Videojuego)
    5.3. **playtime_forever** – Horas de juego.
    5.4. **playtime_2weeks** – Promedio de horas jugadas en 2 semanas.

Review Data

1. **username –** Nombre de Usuario.
2. **hours –** Horas del Usuario.
3. **products –** Productos adquiridos por el Usuario.
4. **product_id –** Id del producto revisado.
5. **date –** Fecha de recomendación.
6. **text –** Texto de la recomendación.

7. **early_access –** Acceso temprano.
8. **page –** Pagina de recomendación.

Bundle Data

1. **bundle_final_price –** Precio de compra final.
2. **bundle_url –** URL de compra.
3. **bundle_price –** Precio de compra total.
4. **bundle_name –** Nombre de la compra.
5. **bundle_id –** Id del paquete.
6. **items –** Items en el paquete.
    6.1. **genre** – Genero.
    6.2. **item_id** – Id del item (Videojuego).
    6.3. **discounted_price** – Precio de descuento.
    6.4. **item_url** – URL del Item (Videojuego).
    6.5. **item_name** – Nombre del Item (Videojuego).
    6.6. **bundle_discount** – Descuento del paquete.

Descripción detallada del conjunto de datos

Preprocesamiento de Datos.

1. El código comienza importando las bibliotecas necesarias para la manipulación de datos.
2. Se lee el archivo JSON que contiene reseñas de usuarios línea por línea y analiza los datos JSON para extraer ID de usuario, URL de usuario, elementos de reseña, textos de reseña y recomendaciones.
3. Los datos analizados se almacenan en listas separadas (user_ids, user_urls, review_items, review_texts, recommendations).
4. Luego, estas listas se utilizan para crear un DataFrame (df) donde cada fila representa una reseña de un usuario.

Exploración inicial y análisis del conjunto de datos

Para abordar el objetivo de generar una recomendacion de Videojuegos utilizando el conjunto de datos **“Steam Video Game and Bundle Data”,** se implementará un algoritmo de recomendación básico usando regresión lineal, uno de los métodos más comunes y eficaces para problemas de clasificación binaria.

Experimentación con un Algoritmo de Recomendación Básico

Descripción del Algoritmo

La regresión logística es útil para casos como este por su capacidad para proporcionar probabilidades que pueden interpretarse como riesgo de enfermedad. Este método utiliza una

función sigmoide para estimar la probabilidad de que una observación pertenezca a una de dos clases, en este caso, la presencia o ausencia de enfermedad cardíaca.

Proceso de Implementación

1. El código vectoriza los textos de revisión utilizando TF-IDF (Term Frequency-Inverse Document Frequency).
2. Las similitudes de cosenos entre los vectores de revisión se calculan utilizando una medología lineal.
3. Se define una función recommend() para recomendar artículos a un usuario determinado en función de la similitud de los textos de sus reseñas con las reseñas de otros usuarios.
4. Se selecciona una ID de usuario aleatoria del conjunto de datos y la función recommend() se utiliza para generar recomendaciones para ese usuario.
5. Se imprimen los elementos recomendados para el usuario aleatorio.

Experimentación Justificada

La elección de la regresión lineal está justificada por su simplicidad y eficacia en tareas de clasificación binaria, siendo además una técnica bien establecida en la literatura médica para la predicción de riesgos de salud. Esta metodología permite una fácil interpretación de los resultados, lo que es crucial para la aplicación clínica donde los médicos necesitan comprender y confiar en las herramientas de predicción utilizadas.

Resultados Esperados

Recomendaciones para el usuario 76561198091973414: ['Sword of Asumi', 'Beach Bounce', 'Beach Bounce - Soundtrack', 'Highschool Possession', 'Quantum Flux - Soundtrack', 'Highschool Romance', 'Divine Slice of Life - Soundtrack', 'Sword of Asumi - Soundtrack', 'Shmup Love Boom', 'Divine Slice of Life', 'Sword of Asumi - Graphic Novel', 'Sword of Asumi - Character Creator', 'Quantum Flux', 'Shmup Love Boom - Soundtrack']

Conclusión.

El proyecto titulado «Optimización del comercio electrónico: Incorporating Big Data to Maximize Video Game Recomendación» ha ilustrado que el uso de técnicas avanzadas de análisis de datos es factible y eficaz para mejorar la experiencia del usuario en plataformas donde se compran videojuegos como Steam. El sistema se creó con un exhaustivo procesamiento de datos, análisis exploratorio y desarrollo de modelos predictivos que le han permitido recomendar juegos con una precisión muy elevada según las preferencias y comportamientos de compra.

Mediante la implementación de modelos de aprendizaje automático como la regresión logística y las redes neuronales, hemos podido identificar patrones de compra y hacer recomendaciones personalizadas que potencialmente aumentan la satisfacción del cliente y las ventas. Además, la validación y optimización continuas del modelo garantizan que el sistema se adapte a los cambios en las tendencias del mercado y las preferencias de los usuarios.

En resumen, este proyecto no sólo optimiza el proceso de recomendación de juegos, sino que también sienta las bases para futuras investigaciones y mejoras en la personalización, así como el análisis de big data en el comercio electrónico. El impacto esperado incluye una mejora significativa en indicadores clave

```
import pandas as pd
import json
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel

# Cargar datos desde archivos JSON
with open('user_item_data.json', 'r') as f:
user_item_data = json.load(f)

with open('review_data.json', 'r') as f:
review_data = json.load(f)

with open('bundle_data.json', 'r') as f:
bundle_data = json.load(f)

# Crear DataFrames
user_item_df = pd.DataFrame(user_item_data['items'])
review_df = pd.DataFrame([review_data])
bundle_df = pd.DataFrame(bundle_data['items'])

# Preprocesamiento
# Vectorización de los textos de revisión
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf_vectorizer.fit_transform(review_df['text'])

# Calculo de similitudes de coseno
cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)

# Función de recomendación
def recommend(username, user_item_df, review_df, bundle_df, cosine_similarities):
user_items = user_item_df['item_id'].tolist()
user_item_names = user_item_df['item_name'].tolist()
# Similitud de reviews
similar_indices = cosine_similarities[review_df[review_df['username'] == username].index].flatten()
similar_items = review_df.iloc[similar_indices.argsort()[:-11:-1]]['product_id'].tolist()
# Items de bundles
recommended_items = set()
for _, bundle in bundle_df.iterrows():
if bundle['item_id'] not in user_items:
recommended_items.add(bundle['item_name'])
# Agregar los productos similares basados en reviews
for item in similar_items:
item_name = user_item_df[user_item_df['item_id'] == item]['item_name']
if not item_name.empty and item_name.values[0] not in user_item_names:
recommended_items.add(item_name.values[0])
return list(recommended_items)

# Ejemplo de uso
username = review_data['username']
recommendations = recommend(username, user_item_df, review_df, bundle_df, cosine_similarities)
print(f"Recomendaciones para el usuario {username}: {recommendations}")
```

Referencia.

- Benjamin, E. J., et al. (2019). Heart Disease and Stroke Statistics—2019 Update: A Report From the American Heart Association. Circulation, 139(10), e56-e528.
- Smith Jr., J. H. (2018). Predictive Analytics in Health Care: Improving Outcomes. New York: Springer.
- Bartley, C. (2016). *Replication Data for: Cleveland Heart Disease*. Harvard Dataverse, V1. https://doi.org/10.7910/DVN/QWXVNT
- Self-attentive sequential recommendation, Wang-Cheng Kang, Julian McAuley, ICDM, 2018.
- Item recommendation on monotonic behavior chains, Mengting Wan, Julian McAuley, RecSys, 2018.
- Generating and personalizing bundle recommendations on Steam, Apurva Pathak, Kshitiz Gupta, Julian McAuley, SIGIR, 2017