**INFORME TÉCNICO: ANÁLISIS DE DATOS Y MODELADO CON PYTHON**

Nombre del Proyecto: Análisis de Productos Amazon

Alumno: Fanny Merly Paucar Carrasco

ID: 2595077

SEMESTRE: IV

CARRERA: ING. Software con IA

Fecha: 27 Agosto 2025

**1. INTRODUCCIÓN**

El presente informe describe los procedimientos realizados para el análisis y modelado de datos de productos Amazon, utilizando Python y sus librerías especializadas en análisis de datos, machine learning y deep learning. El objetivo principal fue realizar un análisis exploratorio, entrenar modelos predictivos y generar visualizaciones que permitan comprender mejor las características de los productos y su relación con la calificación recibida por los usuarios.  
  
Las herramientas principales utilizadas fueron: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, TensorFlow, Keras y NLTK.

**2. CONFIGURACIÓN DEL ENTORNO**

Se configuró el entorno de trabajo instalando las bibliotecas necesarias:  
- Pandas y NumPy: manipulación y cálculo con datos.  
- Matplotlib y Seaborn: visualización de información.  
- Scikit-learn: entrenamiento de modelos de machine learning.  
- TensorFlow y Keras: creación de modelos de deep learning.  
- NLTK: procesamiento de lenguaje natural.  
  
Comando de instalación:  
!pip install pandas numpy scikit-learn tensorflow keras matplotlib seaborn nltk  
  
Se descargaron recursos necesarios para NLP con NLTK:  
nltk.download('punkt')  
nltk.download('vader\_lexicon')  
nltk.download('punkt\_tab')

**3. CARGA DE DATOS**

Se montó Google Drive y se cargó el archivo CSV con los datos de productos:  
from google.colab import drive  
drive.mount('/content/drive')  
file\_path = '/content/drive/MyDrive/kaggle.json/Amazon\_bestsellers\_items\_2025.csv'  
data = pd.read\_csv(file\_path)  
Se visualizaron las primeras filas del conjunto de datos para confirmar su correcta carga.

**4. PREPROCESAMIENTO DE DATOS**

Conversión de la columna product\_star\_rating a valores numéricos.  
Manejo de valores faltantes mediante reemplazo con la media o asignación de la categoría 'Unknown'.  
data['product\_star\_rating'] = pd.to\_numeric(data['product\_star\_rating'], errors='coerce')  
data.fillna({'product\_star\_rating': data['product\_star\_rating'].mean(), 'rank\_change\_label': 'Unknown'}, inplace=True)  
Se aseguró la integridad de los datos antes de realizar análisis o entrenamiento de modelos.

**5. ANÁLISIS EXPLORATORIO DE DATOS (EDA)**

Se analizaron los datos mediante visualizaciones.  
Distribución de Ratings:  
plt.figure(figsize=(12, 6))  
sns.countplot(data=data, x='product\_star\_rating')  
plt.title('Distribución de Ratings')  
plt.xlabel('Rating')  
plt.ylabel('Cantidad de Reseñas')  
plt.savefig('distribution\_of\_ratings.png')  
plt.show()

**6. MODELADO DE MACHINE LEARNING**

Preparación de Datos:  
- Variable predictora (X): títulos de productos.  
- Variable objetivo (y): calificación de estrellas discretizada en 5 categorías.  
bins = [0, 1, 2, 3, 4, 5]  
labels = [1, 2, 3, 4, 5]  
y\_discrete = pd.cut(y, bins=bins, labels=labels, include\_lowest=True, right=True)  
Conversión de texto a características numéricas con CountVectorizer.  
División en conjuntos de entrenamiento y prueba (80%-20%).  
  
Entrenamiento y Evaluación:  
Modelo utilizado: RandomForestClassifier.  
Evaluación con classification\_report.

**7. PROCESAMIENTO DE LENGUAJE NATURAL (NLP)**

Tokenización de textos:  
text = 'Este es un reporte financiero positivo.'  
tokens = word\_tokenize(text)  
  
Análisis de sentimiento:  
sia = SentimentIntensityAnalyzer()  
sentiment = sia.polarity\_scores(text)  
Clasificación de texto como positivo, negativo o neutral

**8. MODELOS DE DEEP LEARNING**

Creación de un modelo secuencial de capas densas:  
model\_dl = keras.Sequential([...])  
Compilación con categorical\_crossentropy y entrenamiento durante 10 épocas.  
Evaluación en conjunto de prueba mostrando precisión final.

**9. VISUALIZACIÓN DE RESULTADOS**

Relación entre Ratings y Precio:  
data['product\_price\_numeric'] = data['product\_price'].replace('[^0-9.]', '', regex=True).replace('', np.nan).astype(float)  
plt.figure(figsize=(10, 6))  
sns.scatterplot(data=data, x='product\_star\_rating', y='product\_price\_numeric')  
plt.title('Gráfico de dispersión de Ratings y Precio')  
plt.xlabel('Rating')  
plt.ylabel('Precio')  
plt.show()  
  
Histograma de Ratings:  
plt.figure(figsize=(10, 6))  
sns.histplot(data['product\_star\_rating'], bins=5)  
plt.title('Histograma de Ratings')  
plt.xlabel('Rating')  
plt.ylabel('Frecuencia')  
plt.show()

**10. CONCLUSIÓN**

Se logró un análisis completo del conjunto de datos, incluyendo:  
- Limpieza y preprocesamiento de datos.  
- Análisis exploratorio con gráficos de distribución y dispersión.  
- Entrenamiento y evaluación de modelos de machine learning y deep learning.  
- Procesamiento de lenguaje natural y análisis de sentimientos.  
- Visualización clara de los resultados utilizando Matplotlib y Seaborn.  
El informe evidencia la correcta integración de herramientas de Python para análisis de datos y modelado predictivo.