

PLAN DE TRABAJO DEL ESTUDIANTE

1. INFORMACIÓN GENERAL

Apellidos y Nombres:	Juan Piero Vincha Loza	ID:	1406507
Dirección Zonal/CFP:	tacna_moquegua		
Carrera:	Ingeniería de software con inteligencia artificial	Semestre:	04
Curso/ Mód. Formativo	202420-PIAD-425-TEC-NRC_45233		
Tema del Trabajo:	Optimización del Sistema de Recomendación de Productos en un E- commerce		

2. PLANIFICACIÓN DEL TRABAJO

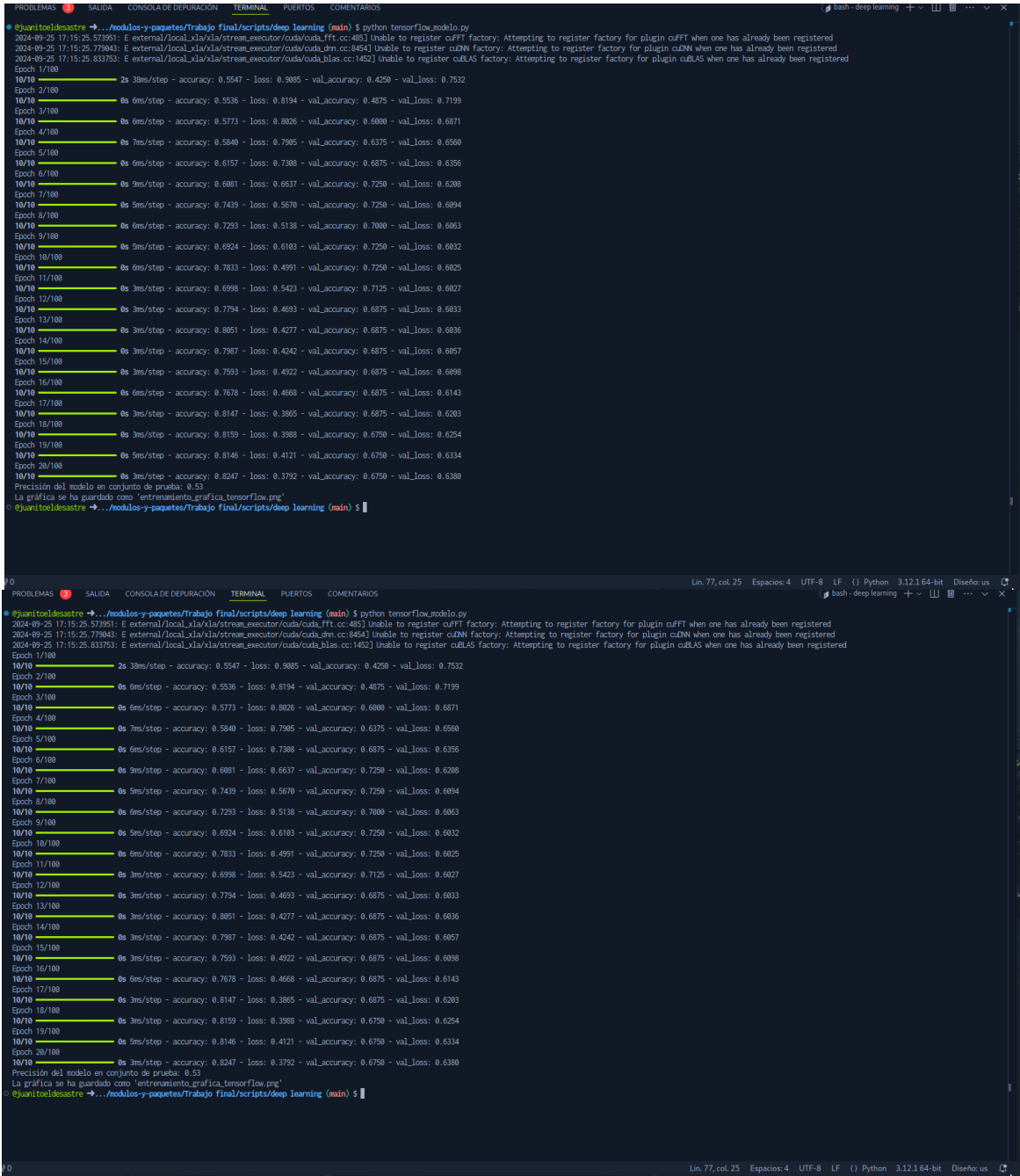
[illegible]

3. ENTREGABLES:

Durante la investigación de estudio, deberán de dar solución a los planteamientos de cada entregable:

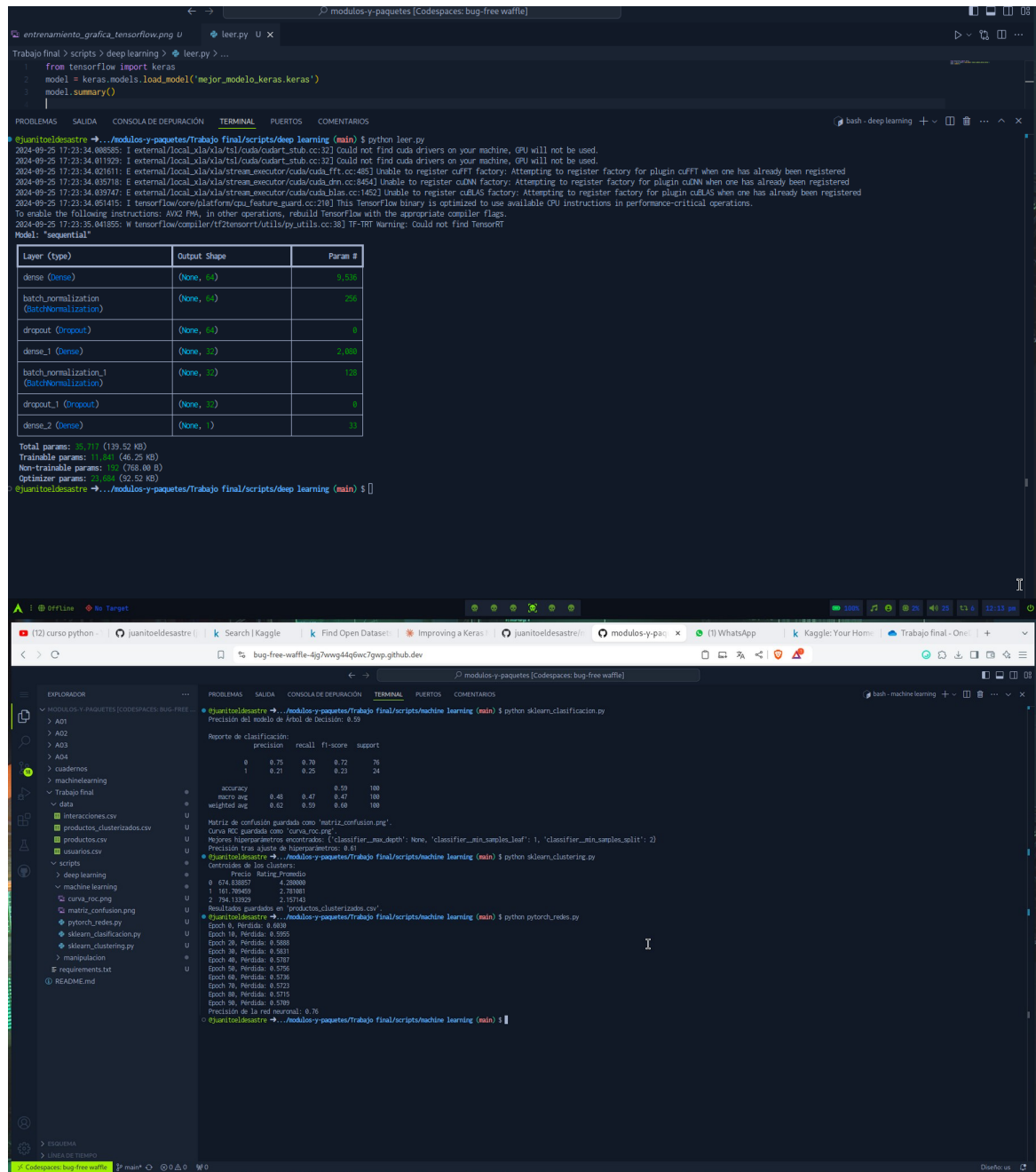
Nº	ENTREGABLE 1
1	<p>En el entregable uno di solución a Maria mediante:</p> <ul style="list-style-type: none"> Lectura y acceso a datos: Los scripts cargan los datos clave de productos, usuarios e interacciones, necesarios para entrenar el modelo de recomendación. Análisis exploratorio: Se analizaron tendencias en productos y comportamiento de usuarios, lo que facilita la personalización de las recomendaciones. Visualización: Le ofrece una vista intuitiva y clara de las distribuciones y patrones dentro de los datos, lo que puede ayudar en la toma de decisiones y en la justificación del diseño del sistema de recomendación

<https://github.com/juanitoeldesastre/modulos-y-paquetes>

Nº	ENTREGABLE 2
2	 <p> PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS COMENTARIOS </p> <pre> @juaritoledesastre → .../modulos-y-paquetes/Trabajo final/scripts/deep_learning (main) \$ python tensorflow_modelo.py 2024-09-25 17:15:25.573951: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:485] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered 2024-09-25 17:15:25.779043: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:8454] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered 2024-09-25 17:15:25.833753: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1462] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered Epoch 1/100 10/10 2s 38ms/step - accuracy: 0.5547 - loss: 0.9085 - val_accuracy: 0.4250 - val_loss: 0.7532 Epoch 2/100 10/10 0s 6ms/step - accuracy: 0.5536 - loss: 0.8194 - val_accuracy: 0.4875 - val_loss: 0.7199 Epoch 3/100 10/10 0s 6ms/step - accuracy: 0.5773 - loss: 0.8026 - val_accuracy: 0.6000 - val_loss: 0.6871 Epoch 4/100 10/10 0s 7ms/step - accuracy: 0.5840 - loss: 0.7905 - val_accuracy: 0.6375 - val_loss: 0.6560 Epoch 5/100 10/10 0s 6ms/step - accuracy: 0.6157 - loss: 0.7308 - val_accuracy: 0.6875 - val_loss: 0.6356 Epoch 6/100 10/10 0s 9ms/step - accuracy: 0.6081 - loss: 0.6637 - val_accuracy: 0.7250 - val_loss: 0.6208 Epoch 7/100 10/10 0s 5ms/step - accuracy: 0.7439 - loss: 0.5670 - val_accuracy: 0.7250 - val_loss: 0.6094 Epoch 8/100 10/10 0s 6ms/step - accuracy: 0.7293 - loss: 0.5138 - val_accuracy: 0.7000 - val_loss: 0.6063 Epoch 9/100 10/10 0s 5ms/step - accuracy: 0.6924 - loss: 0.6103 - val_accuracy: 0.7250 - val_loss: 0.6032 Epoch 10/100 10/10 0s 6ms/step - accuracy: 0.7833 - loss: 0.4991 - val_accuracy: 0.7250 - val_loss: 0.6025 Epoch 11/100 10/10 0s 3ms/step - accuracy: 0.6998 - loss: 0.5423 - val_accuracy: 0.7125 - val_loss: 0.6027 Epoch 12/100 10/10 0s 3ms/step - accuracy: 0.7794 - loss: 0.4693 - val_accuracy: 0.6875 - val_loss: 0.6033 Epoch 13/100 10/10 0s 3ms/step - accuracy: 0.8051 - loss: 0.4277 - val_accuracy: 0.6875 - val_loss: 0.6036 Epoch 14/100 10/10 0s 3ms/step - accuracy: 0.7987 - loss: 0.4242 - val_accuracy: 0.6875 - val_loss: 0.6057 Epoch 15/100 10/10 0s 3ms/step - accuracy: 0.7593 - loss: 0.4922 - val_accuracy: 0.6875 - val_loss: 0.6098 Epoch 16/100 10/10 0s 6ms/step - accuracy: 0.7678 - loss: 0.4668 - val_accuracy: 0.6875 - val_loss: 0.6143 Epoch 17/100 10/10 0s 3ms/step - accuracy: 0.8147 - loss: 0.3865 - val_accuracy: 0.6875 - val_loss: 0.6203 Epoch 18/100 10/10 0s 3ms/step - accuracy: 0.8159 - loss: 0.3988 - val_accuracy: 0.6750 - val_loss: 0.6254 Epoch 19/100 10/10 0s 5ms/step - accuracy: 0.8146 - loss: 0.4121 - val_accuracy: 0.6750 - val_loss: 0.6334 Epoch 20/100 10/10 0s 3ms/step - accuracy: 0.8247 - loss: 0.3792 - val_accuracy: 0.6750 - val_loss: 0.6380 Precisión del modelo en conjunto de prueba: 0.53 La gráfica se ha guardado como 'entrenamiento_grafica_tensorflow.png' @juaritoledesastre → .../modulos-y-paquetes/Trabajo final/scripts/deep_learning (main) \$ </pre>

Ambos scripts usan redes neuronales profundas para predecir si un usuario realizará una compra basada en sus interacciones previas con productos. Al ser más sofisticadas que los modelos tradicionales, las redes neuronales son capaces de identificar patrones complejos en los datos que otros métodos no pueden detectar fácilmente.

Estos scripts permiten a María crear un sistema de recomendación más personalizado, prediciendo productos de interés para los usuarios de forma más precisa y mejorando la experiencia y ventas.



The screenshot displays a Jupyter Notebook environment with a terminal window open. The terminal shows the execution of a Python script for deep learning, followed by a machine learning script. The deep learning script defines a model with layers: dense, batch_normalization, dropout, dense_1, batch_normalization_1, dropout_1, and dense_2. The machine learning script uses sklearn for classification and clustering tasks.

```

from tensorflow import keras
model = keras.models.load_model('mejor_modelo_keras.keras')
model.summary()

@juanitoledastre → .../modulos-y-paquetes/Trabajo Final/scripts/deep learning (main) $ python leer.py
2024-09-25 17:23:34.088585: I external/local_xla/tsl/cuda/cudart_stub.cc:32] Could not find cuda drivers on your machine, GPU will not be used.
2024-09-25 17:23:34.011929: I external/local_xla/tsl/cuda/cudart_stub.cc:32] Could not find cuda drivers on your machine, GPU will not be used.
2024-09-25 17:23:34.021611: E external/local_xla/stream_executor/cuda/cuda_fft.cc:482] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered
2024-09-25 17:23:34.035718: E external/local_xla/stream_executor/cuda/cuda_dnn.cc:8454] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered
2024-09-25 17:23:34.039747: E external/local_xla/stream_executor/cuda/cuda_blas.cc:1452] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered
2024-09-25 17:23:34.051415: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-09-25 17:23:35.041855: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT warning: Could not find TensorRT
Model: "sequential"

Layer (type)                 Output Shape                 Param #
-----
dense (Dense)                 (None, 64)                   5,136
batch_normalization (Batch Normalization) (None, 64)                   256
dropout (Dropout)             (None, 64)                   0
dense_1 (Dense)               (None, 32)                   2,080
batch_normalization_1 (Batch Normalization) (None, 32)                   128
dropout_1 (Dropout)           (None, 32)                   0
dense_2 (Dense)               (None, 1)                    33

Total params: 35,717 (139.52 KB)
Trainable params: 1,441 (46.25 KB)
Non-trainable params: 14 (438.00 B)
Optimizer params: 25,688 (92.52 KB)
@juanitoledastre → .../modulos-y-paquetes/Trabajo Final/scripts/deep learning (main) $

@juanitoledastre → .../modulos-y-paquetes/Trabajo Final/scripts/machine learning (main) $ python sklearn_classification.py
Precisión del modelo de Árbol de Decisión: 0.59

Reporte de clasificación:
precision    recall  f1-score   support
0         0.75    0.70    0.72      76
1         0.21    0.25    0.23      24

accuracy: 0.48    0.47    0.47    100
macro avg: 0.62    0.59    0.60    100
weighted avg: 0.62    0.59    0.60    100

Matriz de confusión guardada como 'matriz_confusion.png'.
Curva ROC guardada como 'curva_roc.png'.
@juanitoledastre → .../modulos-y-paquetes/Trabajo Final/scripts/machine learning (main) $ python sklearn_clustering.py
Contrastes de los clusters
Preciso   Rating_Promedio
0    674.82657    2.20880
1    167.76949    2.76181
2    794.11329    2.15143
@juanitoledastre → .../modulos-y-paquetes/Trabajo Final/scripts/machine learning (main) $ python pytorch_redes.py
Epoch 0, Pérdida: 0.6140
Epoch 10, Pérdida: 0.5955
Epoch 20, Pérdida: 0.5888
Epoch 30, Pérdida: 0.5811
Epoch 40, Pérdida: 0.5787
Epoch 50, Pérdida: 0.5756
Epoch 60, Pérdida: 0.5736
Epoch 70, Pérdida: 0.5723
Epoch 80, Pérdida: 0.5715
Epoch 90, Pérdida: 0.5709
Precisión de la red neuronal: 0.76
@juanitoledastre → .../modulos-y-paquetes/Trabajo Final/scripts/machine learning (main) $

```

En el entregable uno di solución a Maria mediante:

- Mejora en la precisión: Al utilizar árboles de decisión y redes neuronales, el sistema de recomendaciones puede ser mucho más preciso que los métodos basados en reglas simples.
- Personalización: Los modelos pueden adaptarse a los comportamientos y preferencias individuales de los usuarios, ofreciendo recomendaciones más personalizadas.
- Optimización: Con GridSearchCV, María puede ajustar los modelos para obtener el mejor rendimiento posible.

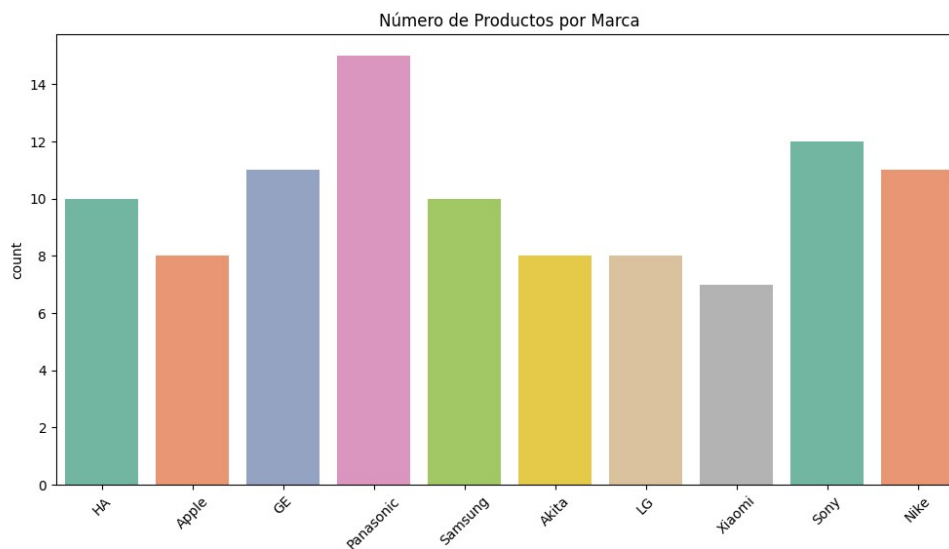
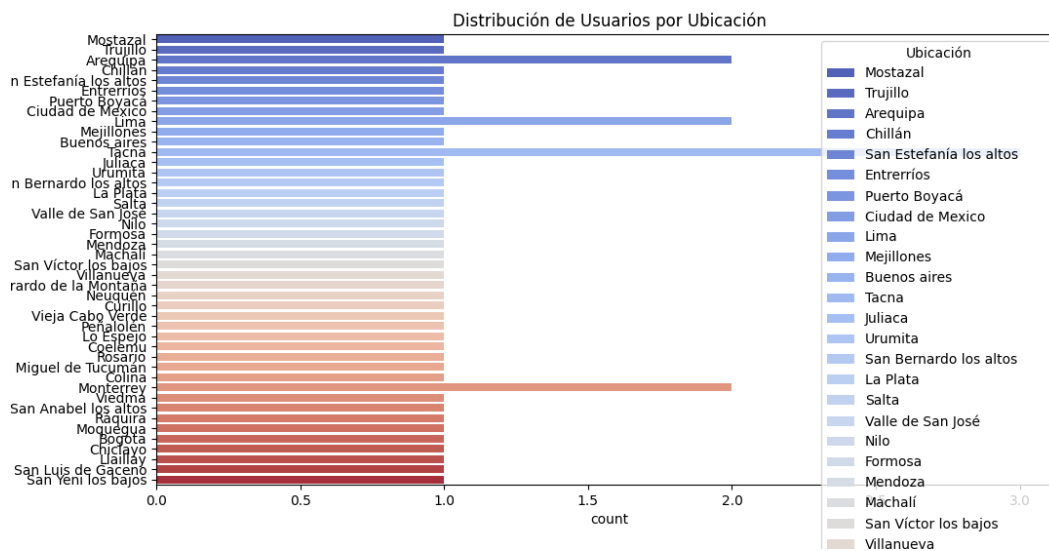
(1,8) -,Cluster 1						
Categoría	Precio	Moneda	Marca	Descripción	Rating_Promedio	Cluster
Tecnología	194.9	COP	HA	sinergia didáctica _	3.4	1
Electrónica	38.8	MXN	Apple	algoritmo modular e_	4.4	1
Electrohogar	807.27	MXN	GE	Reduced grid-enable_	3.3	0
Electrónica	596.6	COP	Panasonic	Multi-layered well_	3.8	0
Tecnología	525.04	CLP	Panasonic	Implemented leading_	4.2	0
Ropa	855.84	MXN	Samsung	infraestructura ter_	2.7	2
Electrohogar	180.28	COP	Akita	medición 24 horas f_	2.9	1
Tecnología	952.35	MXN	L6	encriptar orientada_	1.4	2
Gaming	279.23	COP	Panasonic	colaboración bifurc_	1.6	1
Electrónica	718.92	COP	Akita	Vision-oriented con_	3.3	0
Ropa	824.49	CLP	Xiaomi	migración homogénea_	2.8	2
Gaming	812.49	CLP	Samsung	política basado en _	3.6	0
Gaming	70.99	MXN	Panasonic	Reactive eco-centri_	2.3	1
Electrónica	529.09	ARS	Panasonic	base del conocimien_	2.7	2
Electrónica	745.3	ARS	Apple	paradigma 4ta gener_	3.1	2
Tecnología	625.92	CLP	Panasonic	Proactive cohesive _	2.3	2
Tecnología	225.62	COP	Sony	Secured scalable el_	4.0	1
Tecnología	800.39	CLP	Panasonic	Universal tertiary _	4.6	0
Gaming	240.75	COP	Sony	Exclusive 3dgenera_	4.9	0
Electrohogar	290.48	COP	Sony	Cloned empowering f_	4.7	0
Ropa	680.86	ARS	Nike	Universal national _	5.0	0
Electrónica	743.51	CLP	Panasonic	Innovative directio_	2.4	2
Electrónica	715.22	COP	Sony	data-warehouse mult_	4.2	0
Gaming	956.96	COP	Akita	encriptar increment_	4.3	0
Electrónica	845.13	COP	Apple	mediante ejecutiva _	1.9	2
Tecnología	570.44	COP	Xiaomi	Open-source mission_	2.1	2
Tecnología	963.75	MXN	Akita	Optimized didactic _	1.1	2
Ropa	916.79	MXN	Xiaomi	Focused global prod_	4.9	0
Tecnología	671.1	ARS	Panasonic	Vision-oriented dis_	2.5	2
Gaming	324.6	CLP	Samsung	Streamlined dynamic_	3.3	1
Electrohogar	782.11	COP	Akita	Front-line radical _	2.8	2
Tecnología	866.8	ARS	Apple	Realigned user-faci_	4.7	0
Electrónica	549.66	CLP	Apple	conjunto de instruc_	3.2	0

scripts
<ul style="list-style-type: none"> deep learning <ul style="list-style-type: none"> entrenamiento_grafica_keras.png entrenamiento_grafica_tensorflow.png keras_entrenar.py leer.py mejor_modelo_keras.keras mejor_modelo_tensorflow.keras tensorflow_modelo.py machine learning <ul style="list-style-type: none"> curva_roc.png matriz_confusion.png pytorch_redes.py sklearn_clasificacion.py sklearn_clustering.py manipulacion <ul style="list-style-type: none"> __pycache__ <ul style="list-style-type: none"> analizar_datos.py distribucion_usuarios_ubicacion.png leer_datos.py productos_por_marca.png resumen_datos.py generate.py requirements.txt

Estructura del proyecto

Desarrollo del Entregable N°1- DIBUJO / ESQUEMA / DIAGRAMA
(Adicionar páginas que sean necesarias)

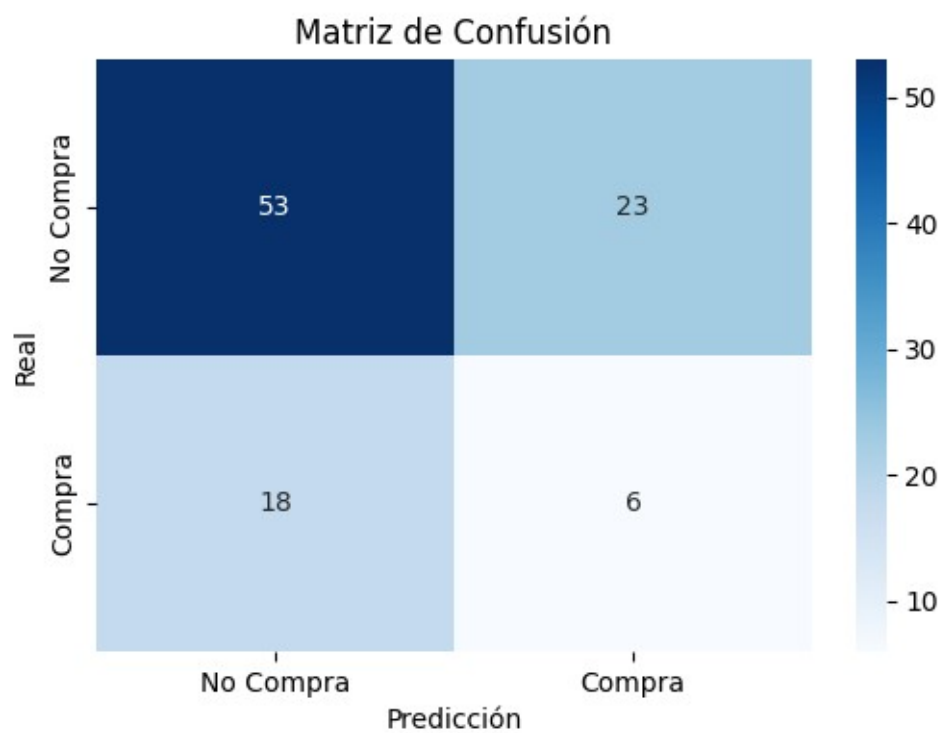
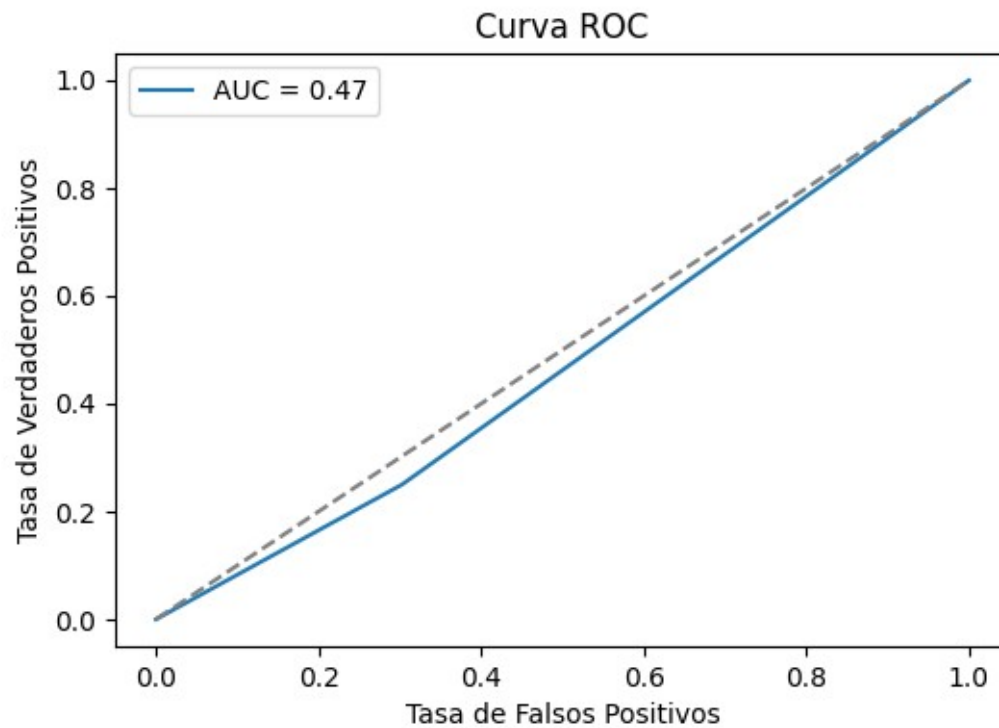
Manipulacion de datos



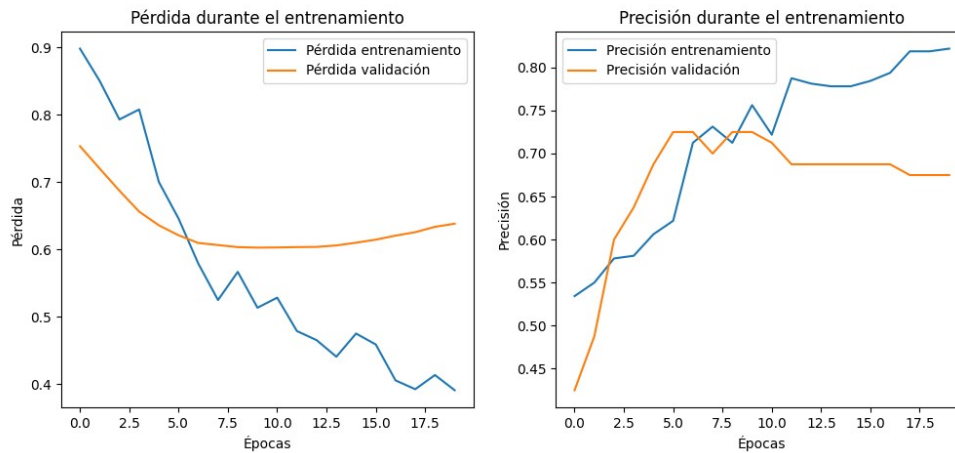
dataset generado artificialmente para el uso del los scripts



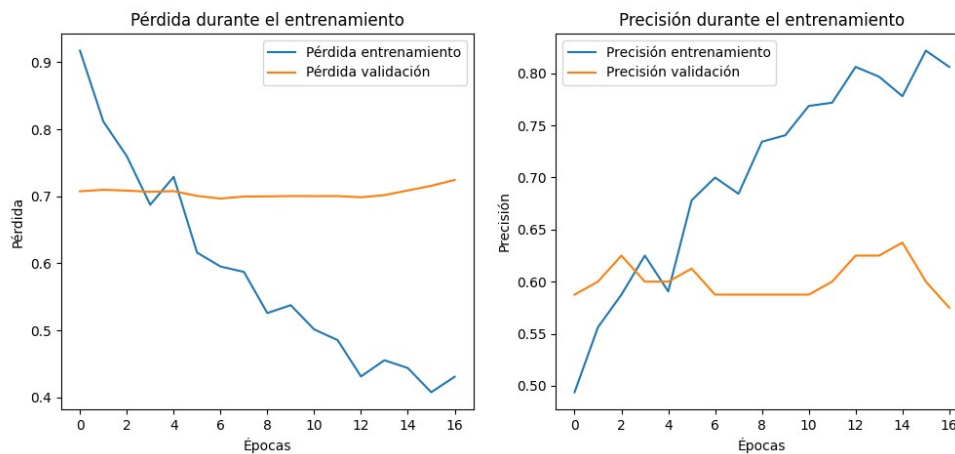
Machine learning



Gráfica entrenamiento de tensorflow



Gráfica entrenamiento de keras



	[Graficos del Trabajo Final]	
	[Juan Piero Vincha Loza]	[6]



PROCESO DE EJECUCIÓN

OPERACIONES / PASOS /SUBPASOS	SEGURIDAD / MEDIO AMBIENTE / NORMAS -ESTANDARES
1. Definir el sistema de recomendación	Python para el bien: Fomentemos un ambiente inclusivo y colaborativo en el uso de Python, respetando las contribuciones de todos y utilizando las bibliotecas de manera responsable. Sigamos las mejores prácticas de codificación, implementemos pruebas para garantizar la calidad del código y seamos transparentes sobre las limitaciones de nuestros modelos. Promovamos el aprendizaje continuo y apoyemos a nuestros compañeros, asegurando el uso ético de los datos y la reproducibilidad de los resultados.
1.1 Investigar las limitaciones del sistema actual.	
1.2 Identificar los objetivos del nuevo sistema (aumentar ventas, personalizar recomendaciones).	
2. Preparar el entorno de desarrollo	
2.1. Instalar Python y crear un entorno virtual (venv).	
2.2 Instalar las librerías necesarias (Scikit-Learn, PyTorch, TensorFlow, Keras) y crear un archivo requirements.txt.	
3. Cargar y explorar los datos	
3.1 Leer los datasets de usuarios, productos e interacciones usando Pandas.	
3.2 Realizar un script en python para crear los csv.	
4. Preprocesar los datos	
5. Dividir los datos en entrenamiento y prueba	
5.1 Asegurarse de que las proporciones de las clases sean equilibradas en ambos conjuntos.	
6. Implementar modelos de machine learning	
7. Crear una red neuronal básica con PyTorch	
8. Implementar un modelo de deep learning con Keras	
8.1. Implementar un modelo de deep learning con Keras	
9. Evaluar los modelos	
10. Documentar el proceso	

INSTRUCCIONES: debes ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. No olvides los aspectos de calidad, medio ambiente y SHI.



<https://www.kaggle.com/datasets/juanitoeldevastador/dataset-trabajofinal-modulosypaquetes>

dataset-trabajofinal-modulosypaquetes

0

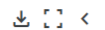
New Notebook

Download (10 kB)



Data Card Code (0) Discussion (0) Suggestions (0) Settings

interacciones.csv (17.32 kB)






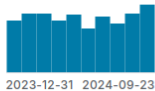

Detail Compact Column

6 of 6 columns

About this file



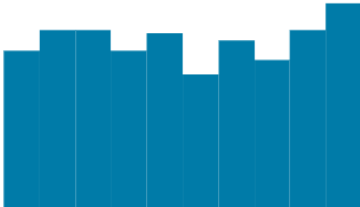
csv artificialmente creado para el trabajo final del curso modulos y paquetes

# ID_Interacción	# ID_Usuario	# ID_Producto	Acción	Fecha	# Calificación
			<div>Añadir a lista de ... 26%</div> <div>Compra 25%</div> <div>Other (241) 48%</div>		
1	500	1	50	100	5
1	22	12	Vista	2024-06-25	4
2	12	1	Carrito	2024-04-18	5
3	42	4	Compra	2024-02-14	0
4	17	72	Vista	2024-08-03	0
5	42	49	Añadir a lista de deseos	2024-04-18	3
6	2	81	Vista	2024-08-17	0
7	36	69	Carrito	2024-08-09	1
8	15	42	Carrito	2024-04-27	1
9	2	79	Carrito	2024-04-03	0
10	20	5	Carrito	2024-04-07	0

Acción

Añadir a lista de deseos	26%	Valid	500	100%
		Mismatched	0	0%
Compra	25%	Missing	0	0%
		Unique	4	
Other (241)	48%	Most Common	Añadir a lis...	26%

Fecha

	Valid	500	100%
	Mismatched	0	0%
	Missing	0	0%
	Minimum	31Dec23	
	Mean	15May24	
	Maximum	23Sep24	

productos.csv (6.9 kB)



Detail Compact Column

7 of 7 columns ▾

About this file

CSV de productos creado a traves de un script en python para el Trabajo Final del curso modulos y paquetes con python 🐍

# ID_Producto	Δ Categoría	# Precio	Δ Moneda	Δ Marca	Δ Descripción	# Rating_Promedio
1	Electrónica	25%	COP	33%	Panasonic	15%
1	Tecnología	23%	MXN	29%	Sony	12%
1	Other (52)	52%	Other (38)	38%	Other (73)	73%
100		990			100 unique values	
1	Tecnología	104.9	COP	HA	sinergia didáctica reactivo	3.4
2	Electrónica	38.8	MXN	Apple	algoritmo modular extendido	4.4
3	Electrohogar	807.27	MXN	GE	Reduced grid-enabled architecture	3.3
4	Electrónica	506.6	COP	Panasonic	Multi-layered well-modulated task-force	3.8
5	Tecnología	525.04	CLP	Panasonic	Implemented leadingedge solution	4.2
6	Ropa	855.84	MXN	Samsung	infraestructura terciaria administrado	2.7
7	Electrohogar	180.28	COP	Akita	medición 24 horas fundamental	2.9
8	Tecnología	952.35	MXN	LG	encriptar orientada a soluciones recontextualizado	1.4
9	Gaming	279.23	COP	Panasonic	colaboración bifurcada personalizable	1.6

Δ Categoría

Electrónica	25%	Valid	100	100%
		Mismatched	0	0%
Tecnología	23%	Missing	0	0%
		Unique	5	
Other (52)	52%	Most Common	Electrónica	25%

Precio

Valid	100	100%
Mismatched	0	0%
Missing	0	0%
Mean	518	
Std. Deviation	321	
Quantiles		
	22.8	Min
	209	25%
	568	50%
	812	75%
	990	Max

Δ Moneda

COP	33%	Valid	100	100%
		Mismatched	0	0%
MXN	29%	Missing	0	0%
		Unique	4	
Other (38)	38%	Most Common	COP	33%

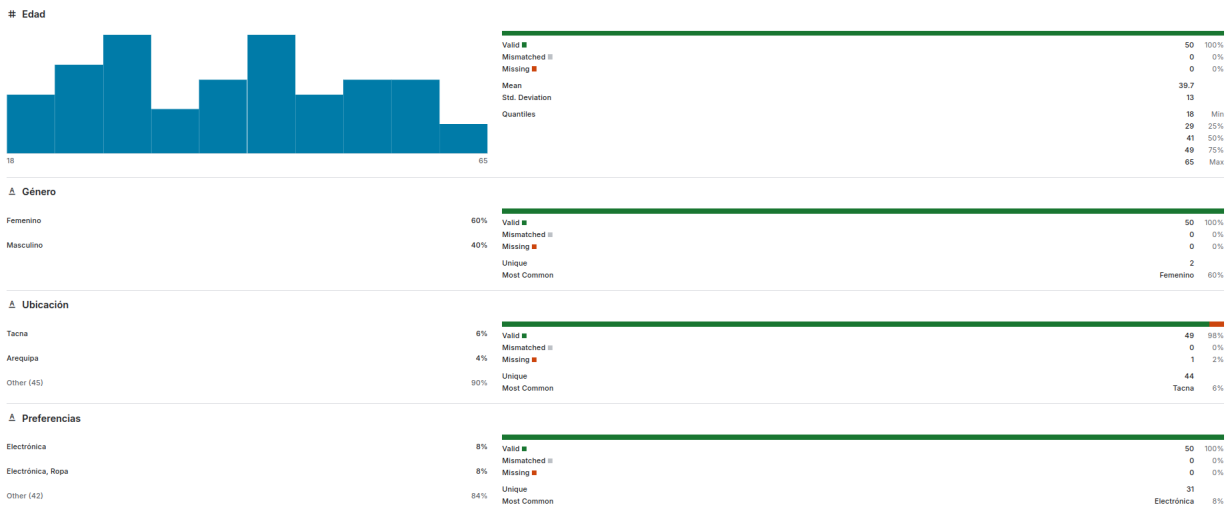
usuarios.csv (4.42 kB)

Detail Compact Column

About this file

CSV de usuarios creado a traves de un script en python para el Trabajo Final del curso modulos y paquetes con python 🐍

# ID_Usuario	Nombre	# Edad	Género	Ubicación	Preferencias	Likes
1	50 unique values	18 65	Femenino 60% Masculino 40%	Tacna 6% Arequipa 4% Other (45) 90%	Electrónica 8% Electrónica, Ropa 8% Other (42) 84%	50 unique values
1	Raquel Mercedes Arguello Tijerina	19	Femenino	Mostazal	Electrónica	[95, 100]
2	Juan Agüero Garcia	29	Masculino	Trujillo	Tecnología, Ropa	[19, 61, 56, 1]
3	Bernardino Bravo	36	Femenino	Arequipa	Tecnología, Electrónica, Ropa	[38, 44, 96, 20, 89]
4	Santino Benjamin Luisana Bravo Maidana	39	Masculino	Chillán	Electrónica	[98, 1, 77, 83, 19]
5	Maria Mayte Rangel	52	Femenino	San Estefania los altos	Gaming, Electrohogar	[46, 93]
6	César Luis Mendoza	35	Femenino	Entrerrios	Electrónica, Ropa	[91, 69, 1, 78]
7	Olivia Santiago Gonzalez	25	Masculino	Puerto Boyacá	Electrónica, Ropa	[80, 72, 41, 63, 4]
8	Maria Emilia Benjamin Nuñez Juarez	48	Femenino	Ciudad de Mexico	Gaming, Tecnología, Ropa	[55, 6, 14]
9	Elvira Avilés	28	Femenino	Lima	Tecnología, Ropa	[43, 45, 59, 75]
10	Sr(a). Josefina Ramirez	63	Femenino	Mejillones	Gaming, Ropa	[44, 99, 21, 54]



OPERACIONES / PASOS /SUBPASOS	SEGURIDAD / MEDIO AMBIENTE / NORMAS -ESTANDARES
1. Definir el sistema de recomendación	Python para el bien: Fomentemos un ambiente inclusivo y colaborativo en el uso de Python, respetando las contribuciones de todos y utilizando las bibliotecas de manera responsable. Sigamos las mejores prácticas de codificación, implementemos pruebas para garantizar la calidad del código y seamos transparentes sobre las limitaciones de nuestros modelos. Promovamos el aprendizaje continuo y apoyemos a nuestros compañeros, asegurando el uso ético de los datos y la reproducibilidad de los resultados.
1.1 Investigar las limitaciones del sistema actual.	
1.2 Identificar los objetivos del nuevo sistema (aumentar ventas, personalizar recomendaciones).	
2. Preparar el entorno de desarrollo	
2.1. Instalar Python y crear un entorno virtual (venv).	
2.2 Instalar las librerías necesarias (Scikit-Learn, PyTorch, TensorFlow, Keras) y crear un archivo requirements.txt.	
3. Cargar y explorar los datos	
3.1 Leer los datasets de usuarios, productos e interacciones usando Pandas.	
3.2 Realizar un script en python para crear los csv.	
4. Preprocesar los datos	
5. Dividir los datos en entrenamiento y prueba	
5.1 Asegurarse de que las proporciones de las clases sean equilibradas en ambos conjuntos.	
6. Implementar modelos de machine learning	
7. Crear una red neuronal básica con PyTorch	
8. Implementar un modelo de deep learning con Keras	
8.1. Implementar un modelo de deep learning con Keras	
9. Evaluar los modelos	
10. Documentar el proceso	

INSTRUCCIONES: debes ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. No olvides los aspectos de calidad, medio ambiente y SHI.



1. MÁQUINAS Y EQUIPOS

PC

Wifi

Teclado y mouse

Escritorio

3. HERRAMIENTAS E INSTRUMENTOS

Vscode, codespace, kaggle , jupyterlab, spider

Python librerías, pip, python3x

Arch linux, tabview, alacritty, nano

Google colab

Youtube, stackoverflow, docs.python, freecode.camp

5. MATERIALES E INSUMOS

Manzana

[illegible]

