



Universidad Peruana de Ciencias Aplicadas

Inteligencia Artificial

Ciclo: 2024-2

Sección: CC63

Docente: Flor Cagniy Cardenas Mariño

Informe Trabajo Parcial

Integrantes:

- Jhonny Elias Ruiz Santos - U201910128
- Hector Jesus Quintana Robatti - U20201B280

2024

ÍNDICE

1. Introducción.....	3
2. Descripción del problema.....	3
3. Descripción y visualización del dataset.....	4
4. Propuesta.....	8
5. Diseño de aplicativo.....	8
6. Referencias Bibliográficas.....	11

Introducción

El presente trabajo se basa en la aplicación de diferentes técnicas avanzadas de inteligencia artificial con el objetivo de anticipar el desempeño futuro de una tienda y evaluar su viabilidad a largo plazo. A través del análisis de datos y la implementación de algoritmos predictivos, se busca determinar la probabilidad de que la tienda mantenga su viabilidad en el mercado. Para lograr esto se creará un sistema de predicción de rentabilidad. La solución hará uso de una interfaz web para mayor comodidad del usuario.

El objetivo principal es ofrecer una perspectiva más precisa del posible éxito o fracaso del negocio en un determinado tiempo, proporcionando así información valiosa para la toma de decisiones estratégicas que puedan asegurar su continuidad.

Descripción del problema

Actualmente, los supermercados se enfrentan a varios problemas, uno de ellos es la rentabilidad del negocio. Este puede ser afectado por diversos factores externos, como mala atención al cliente, poco stock, etc. Todo esto puede llevar a que un negocio termine con pocas ventas y clientes, dándole así una rentabilidad baja o nula.

Pero, y si se pudiera predecir la rentabilidad próxima de una tienda?. Con esta propuesta, buscamos mejorar la viabilidad de la tienda, haciéndole saber al encargado cuál será el futuro próximo de la tienda para que así, pueda tomar las medidas necesarias y evitar un posible cierre.

Descripción y visualización del dataset

Metodología

Para el desarrollo del trabajo, se seguirá una metodología estructurada en varias etapas:

1. Obtención de los datos: El dataset a utilizar en este proyecto fue extraído de la plataforma Kaggle, bajo el nombre **Supermarket store branches sales analysis**. Este conjunto de datos contiene información sobre 896 tiendas, incluyendo variables como el área de la tienda, la cantidad de clientes diarios, los artículos disponibles y las ventas totales. Este dataset es adecuado para el análisis de la viabilidad y rentabilidad de tiendas comerciales.
2. Preparación de los datos: Antes de realizar cualquier análisis, será necesario limpiar los datos. Se verificarán y eliminarán valores faltantes, se revisará la coherencia de los datos y se va a estandarizar el formato de las variables para asegurar su correcto uso en el modelo de regresión lineal.
3. División de los datos: Para entrenar y probar el modelo, los datos serán divididos en dos conjuntos: el 80% fue utilizado para el entrenamiento y el 20% restante para la validación del modelo. Esto permitirá evaluar el desempeño del modelo en datos no vistos previamente.
4. Selección del modelo: Se optó por la regresión lineal debido a la naturaleza del problema, donde la relación entre las variables independientes (número de clientes, área, artículos disponibles) y la variable dependiente (ventas) puede ser modelada linealmente.

Estructura del dataset

<u>variable</u>	<u>tipo</u>	<u>descripción</u>
Store_ID	int64 (Enteros de 64 bits)	Identificador único de cada tienda
Store_Area	int64 (Enteros de 64 bits)	Espacio total de la tienda
Items_Available	int64 (Enteros de 64 bits)	Cantidad de items disponibles en la tienda
DailyCustomerCount	int64 (Enteros de 64 bits)	Cantidad diaria de clientes que visitan la tienda
Store_Sales	int64 (Enteros de 64 bits)	Cantidad de ventas totales de la tienda

Analisis Exploratoria de Datos (EDA)

*Verificar valores nulos

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df=pd.read_csv("/content/Stores.csv")

df.isnull().sum()
```

```
0
Store ID      0
Store_Area    0
Items_Available 0
Daily_Customer_Count 0
Store_Sales   0
dtype: int64
```

*Gráfico de dispersión e histograma de ventas

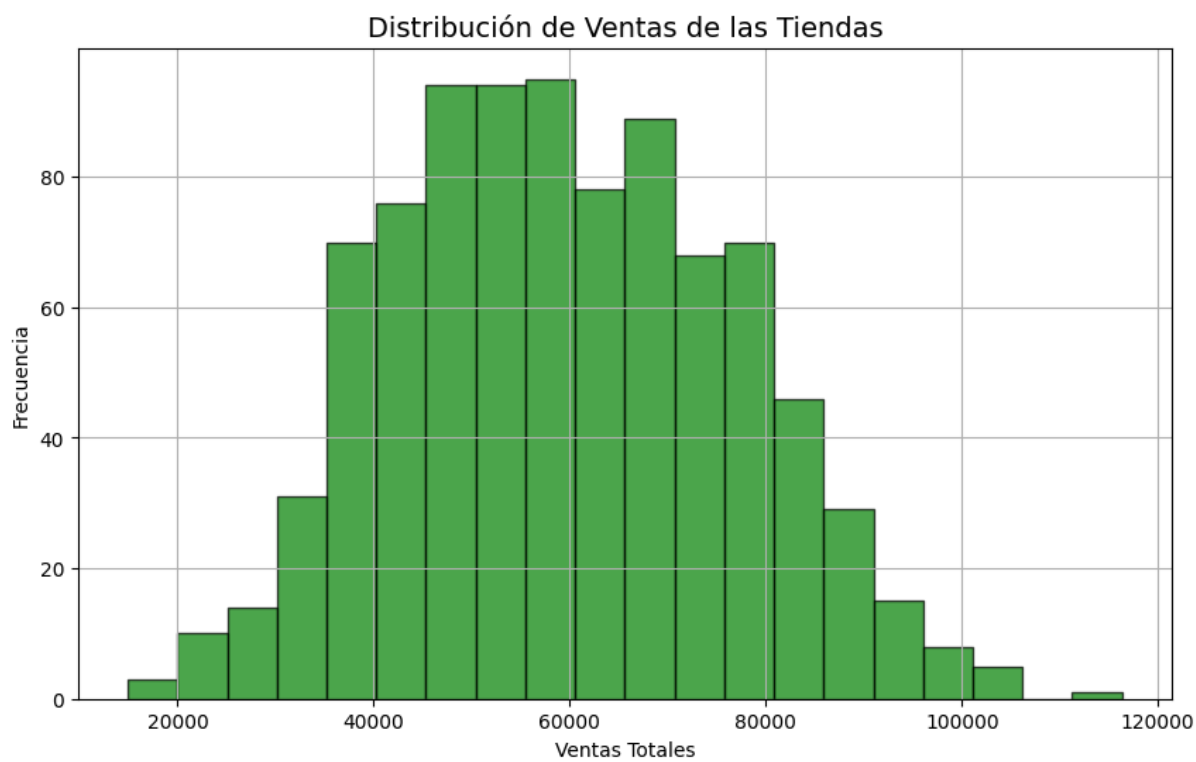
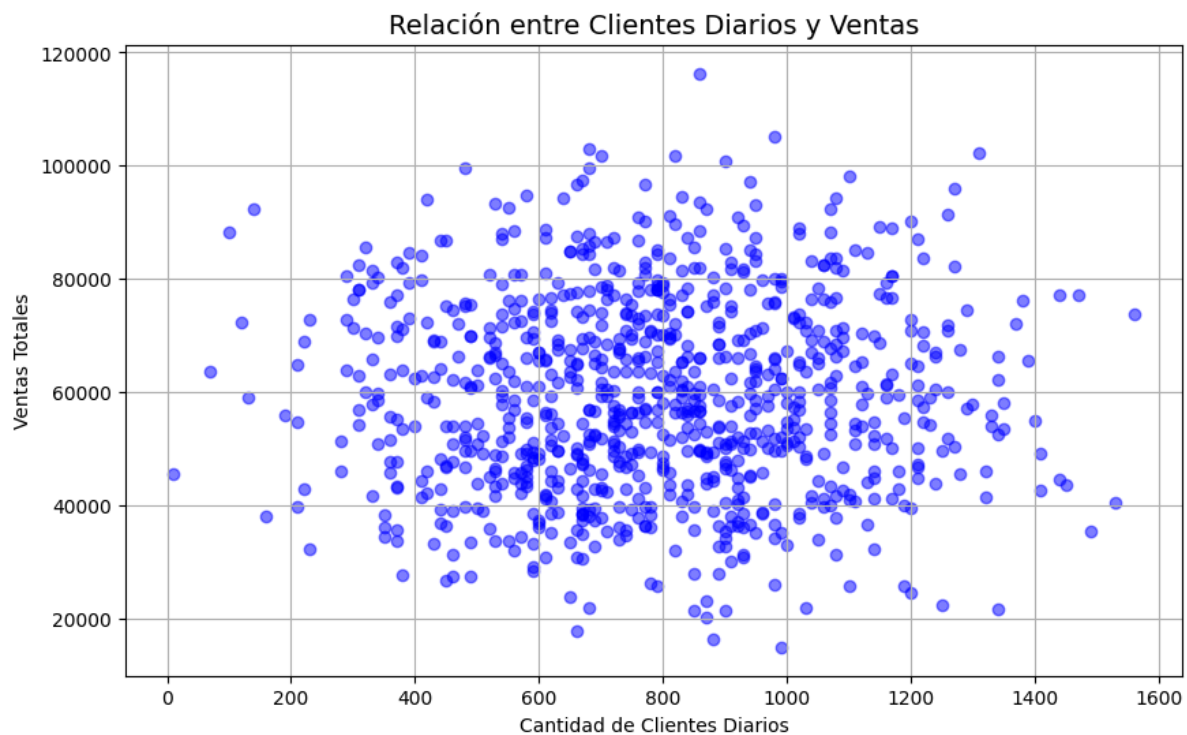
```
print(df.head())

df.columns = ['Store_ID', 'Store_Area', 'Items_Available', 'Daily_Customer_Count', 'Store_Sales']

# Gráfico de dispersión entre Clientes Diarios y Ventas
plt.figure(figsize=(10, 6))
plt.scatter(df['Daily_Customer_Count'], df['Store_Sales'], color='blue', alpha=0.5)
plt.title('Relación entre Clientes Diarios y Ventas', fontsize=14)
plt.xlabel('Cantidad de Clientes Diarios')
plt.ylabel('Ventas Totales')
plt.grid(True)
plt.show()

# Histograma de las Ventas de las tiendas
plt.figure(figsize=(10, 6))
plt.hist(df['Store_Sales'], bins=20, color='green', edgecolor='black', alpha=0.7)
plt.title('Distribución de Ventas de las Tiendas', fontsize=14)
plt.xlabel('Ventas Totales')
plt.ylabel('Frecuencia')
plt.grid(True)
plt.show()
```

	Store ID	Store_Area	Items_Available	Daily_Customer_Count	Store_Sales
0	1	1659	1961	530	66490
1	2	1461	1752	210	39820
2	3	1340	1609	720	54010
3	4	1451	1748	620	53730
4	5	1770	2111	450	46620



*Detectar outliers entre las columnas de interés

```
#detectar outliers usando el rango intercuartílico (IQR)
def detectar_outliers(df, columna):
    Q1 = df[columna].quantile(0.25)
    Q3 = df[columna].quantile(0.75)
    IQR = Q3 - Q1
    limite_inferior = Q1 - 1.5 * IQR
    limite_superior = Q3 + 1.5 * IQR
    outliers = df[(df[columna] < limite_inferior) | (df[columna] > limite_superior)]
    return outliers

# detectar outliers en las columnas numéricas de interés
outliers_ventas = detectar_outliers(df, 'Store_Sales')
outliers_clientes = detectar_outliers(df, 'Daily_Customer_Count')

print("Outliers en ventas:")
print(outliers_ventas)

print("\nOutliers en número de clientes diarios:")
print(outliers_clientes)
```

```
Outliers en ventas:
   Store_ID  Store_Area  Items_Available  Daily_Customer_Count  Store_Sales
649      650      1989             2414                860      116320

Outliers en número de clientes diarios:
   Store_ID  Store_Area  Items_Available  Daily_Customer_Count  Store_Sales
39         40      1270             1516                10      45480
349        350      1886             2228               1530      40350
848        849       919             1099               1560      73810
```

Propuesta

Para la propuesta de solución es la creación de un sistema de predicción basado en algoritmos de **regresión lineal** para estimar la viabilidad y rentabilidad de una futura tienda comercial. El sistema aprovechará datos históricos disponibles, como el número de artículos disponibles, cantidad de clientes diarios y ventas realizadas, con el fin de predecir el desempeño futuro.

El modelo utilizado permitirá identificar las relaciones entre las variables y prever cómo influyen en la rentabilidad a largo plazo. La complicitad y efectividad de la regresión lineal se ajustan bien a este tipo de análisis, porque ofrece una forma clara de interpretar los factores que afectan.

Diseño de aplicativo

- **Comprensión del negocio:**

Nuestro objetivo es crear una aplicación web que permita subir a los usuario un csv con los datos de su tienda, estos serán: Area, stock, clientes diarios, ventas mensuales. Una vez cargado este archivo, se procede a aplicar un algoritmo de regresión lineal que nos permitirá predecir si el negocio está siendo o no rentable. Para mayor confiabilidad, se decidió crear un login, asegurando que los datos de cada tienda sean únicos por cada usuario, a la vez, se podría implementar un historial de predicciones, para que cada usuario vea si el negocio está o bajando su rentabilidad.

- **Comprensión de datos:**

Una vez tengamos los datos necesarios, se procede a hacer una validación, para ello, el formulario solo permitirá subir archivos en formato csv.

- **Preparación de datos:**

En esta etapa, el aplicativo podrá hacer la etapa de limpieza de datos de manera automática. Debido a que no se procesa una cantidad excesiva de datos en el caso de una tienda independiente, solo se incluye este pre-procesamiento de datos.

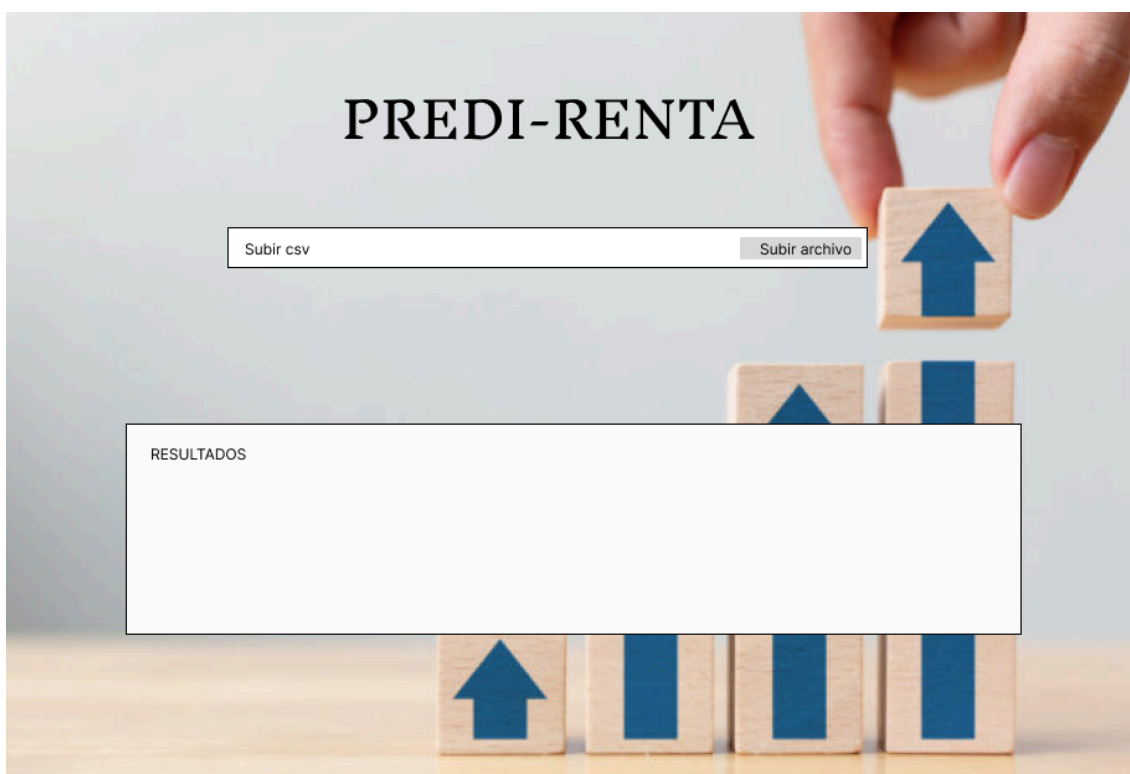
- **Prototipo:**

A continuación, mostramos el prototipo de nuestra aplicación, la cual tendrá por nombre “PREDI-RENTA”.

-Login:



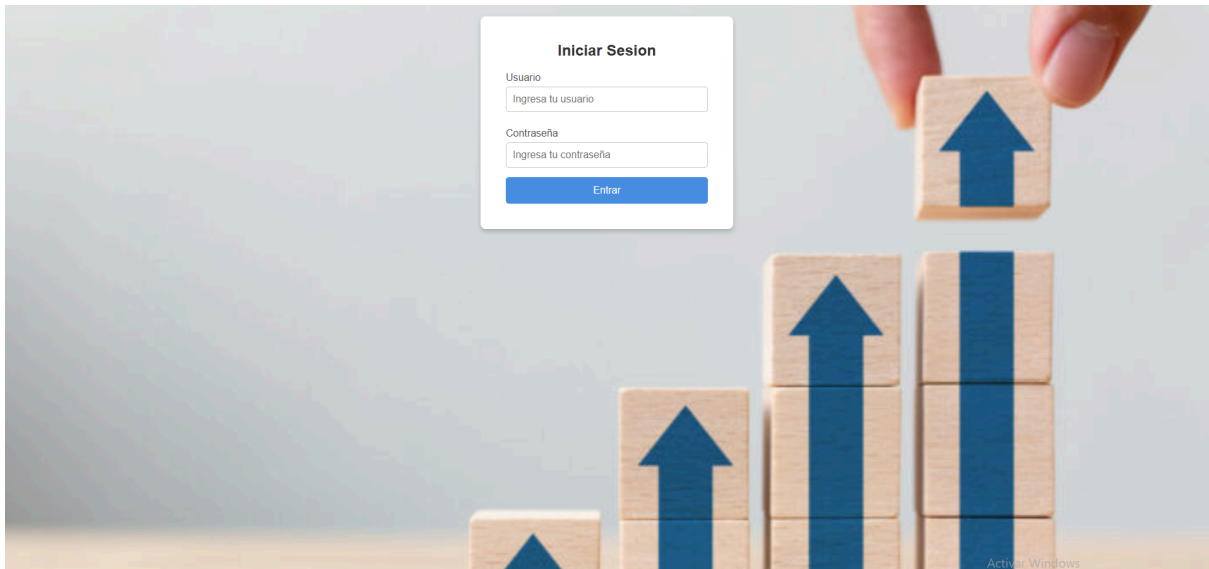
-Aplicación principal



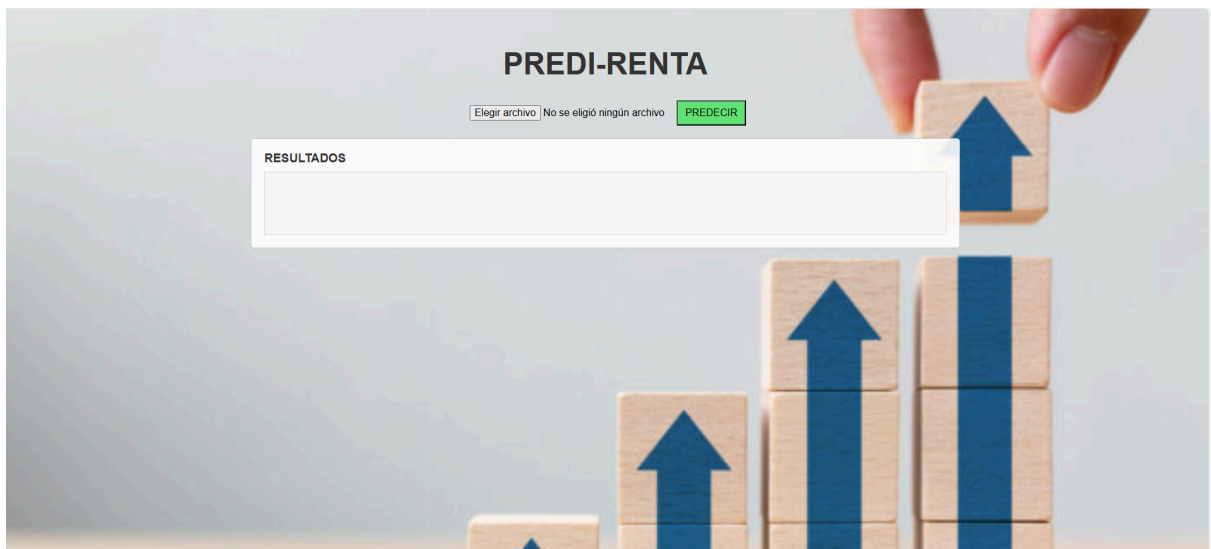
- **Desarrollo:**

En esta sección, mostraremos el diseño de la aplicación ya desarrollado:

-Login:



-Aplicación principal.



Referencias bibliográficas

- JS. (2021 - 2022). *Supermarket store branches sales analysis* [Dataset]. Kaggle. <https://www.kaggle.com/datasets/surajjha101/stores-area-and-sales-data>
- McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython* (2da ed.). O'Reilly Media. <https://www.oreilly.com/library/view/python-for-data/9781491957653/>
- Geron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2da ed.). O'Reilly Media. <https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>