

# **Administración de Bases de Datos**

Arenas Deseado Luis Eduardo



## **Menú con Programa de Regresión Lineal**

Diego Pacheco Valdez

8 de Junio del 2024

## Introducción

La tarea consiste en la realización de un menú dentro de terminal que implemente los conceptos de regresión lineal aprendidos durante clase para la gestión de productos y la predicción de ventas.

## Desarrollo

Aplicando la regresión lineal se deberá crear un programa en python con las siguientes funcionalidades

- Sistema de menú: Implementar un sistema de menú amigable para el usuario que permita:

<pre>----- Menu de Opciones  (1) Crear Producto (2) Mostrar Productos (3) Actualizar Producto (4) Eliminar Producto (5) Realizar Predicción (6) Cerrar sesión</pre>	<pre>def Menu():     print(" ----- Menu de     Opciones ----- \n\n")     print("(1) Crear     Producto\n")     print("(2) Mostrar     Productos\n")     print("(3) Actualizar     Producto\n")     print("(4) Eliminar     Producto\n")     print("(5) Realizar     Predicción\n")     print("(6) Cerrar     sesión\n")</pre>
---	---

- Agregar producto: Facilitar la entrada de información del producto, incluyendo nombre, precio, presupuesto de publicidad y ventas estimadas.

<pre>Introduzca su opción: 1 Introduzca el nombre del producto: Papas Introduzca el precio del producto: 26.00 Introduzca el presupuesto: 3500.00 Introduzca las ventas del producto: 1200 Producto agregado exitosamente. ----- Menu de Opciones</pre>	<pre># - C - Función para agregar un nuevo producto def agregarProducto(nombre, precio, presupuesto, ventas):     query = "INSERT INTO     Productos (nombre, precio,     presu, ventas) VALUES (%s, %s,     %s, %s)"     cur.execute(query,     (nombre, precio, presupuesto,     ventas))     conn.commit()     print("Producto agregado     exitosamente.")</pre>
---	--

- Ver productos: Mostrar una lista de todos los productos existentes con sus detalles.

```

Introduzca su opción: 2
(1, Decimal('26.00'), Decimal('3500.00'), 1200, 'Papas')

# - R - Función para leer todos los productos
def leerVentas():
    cur.execute("SELECT * FROM Productos")
    rows = cur.fetchall()
    for row in rows:
        print(row)

```

- Modificar producto: Permitir la selección de un producto y la modificación de su información, como precio, presupuesto de publicidad o ventas estimadas.

```

Introduzca su opción: 3
Introduzca el id del producto a modificar:1
Introduzca el nuevo nombre del producto (o deje en blanco para no cambiar): Taki
s
Introduzca el nuevo precio del producto (o deje en blanco para no cambiar):
Introduzca el nuevo presupuesto del producto (o deje en blanco para no cambiar):
1 import Decimal
Introduzca las ventas del producto (o deje en blanco para no cambiar):
Producto con ID 1 actualizado

# - U - Función para actualizar un producto
def actualizarProducto(id, nombre = None, precio = None,
presupuesto = None, ventas = None):
    dictionary = {'nombre': nombre, 'precio': precio,
'presu': presupuesto, 'ventas': ventas}
    for key in dictionary:
        if dictionary[key] is not None:
            query = f"UPDATE Productos SET {key} = %s WHERE
id = %s"
            cur.execute(query, (dictionary[key], id))
    conn.commit()
    print(f"Producto con ID {id} actualizado")

```

- Eliminar producto: Habilitar la eliminación de un producto específico de la base de datos.

```
Introduzca su opción: 4
Introduzca el ID del producto a eliminar: 1
Producto eliminado exitosamente.
----- Menu de Opciones -----
```

```
# - D - Función para borrar un producto
def eliminarProducto(id):
    cur.execute("DELETE FROM Productos WHERE id = %s", (id,))
    conn.commit()
    print("Producto eliminado exitosamente.")
```

- Hacer predicción: Esta opción debe solicitar al usuario que ingrese el precio y el presupuesto de publicidad para un nuevo producto y generar un valor de ventas predicho en base al modelo de regresión lineal entrenado.

```
Introduzca su opción: 5
Introduzca el precio del producto: 13.00
Introduzca el presupuesto: 3500
Ventas predichas para el nuevo producto: [1200.]
----- Menu de Opciones -----
```

```
# - P - Función para realizar producción
def realizarPrediccion(NPrecio, NPresu):
    engine = sa.create_engine(
        "postgresql://postgres:1234@localhost:5432/ProductoPrediccion"
    )
```

```
# lectura de datos de la tabla "Productos" en un Dataframe de pandas.
df = pd.read_sql_table("productos", engine)
```

```
# Selección de las columnas "precio" y "publicidad" como
características
X = df[["precio", "presu"]]
```

```
# Selección de la columna "ventas" como variable objetivo
# la dependiente son las ventas
y = df["ventas"]
```

```
# Creación y Entrenamiento del Modelo de regresión lineal
```

```
model = LinearRegression() #instancia de la clase linearRegression
model.fit(X, y)
```

```
# Realizar predicción
nuevo_precio = NPrecio
nueva_publicidad = NPresu # presupuesto de publicidad del nuevo
producto
```

```
prediction = model.predict([[nuevo_precio, nueva_publicidad]])
```

```
# Impresión de la Predicción de venta para el nuevo producto
print("Ventas predichas para el nuevo producto: ", prediction)
```

- Salir: Terminar el programa.

```
Introduzca su opción: 6  
(venv) energizedtea@MacBook-Air-de-Diego actclases %
```

## Conclusiones

Se logro aprendizaje sobre el uso de la regresión lineal, permitiéndonos crear predicciones con datos ya preexistentes.