



DATATHON 2024

RETO VIVA AEROBUS

Equipo no. 38 Datahack

Integrantes:

Alberto Tienda

Elías Valencia

Ana Sarmiento

Diego Garza



RETO

Objetivo:

Crear un modelo predictivo (machine learning) para optimizar el abastecimiento de los productos de venta a bordo durante el vuelo con el fin de evitar la pérdida de ventas por poco inventario y disminuir las pérdidas de productos en excedente.

Entregables:

- **Principales insights** de la información
- Modelo de predicción de volumen de pasajeros en vuelos de **Enero 2024**
- Modelo de predicción de venta de productos en vuelos de **Enero 2024**

Data sources:

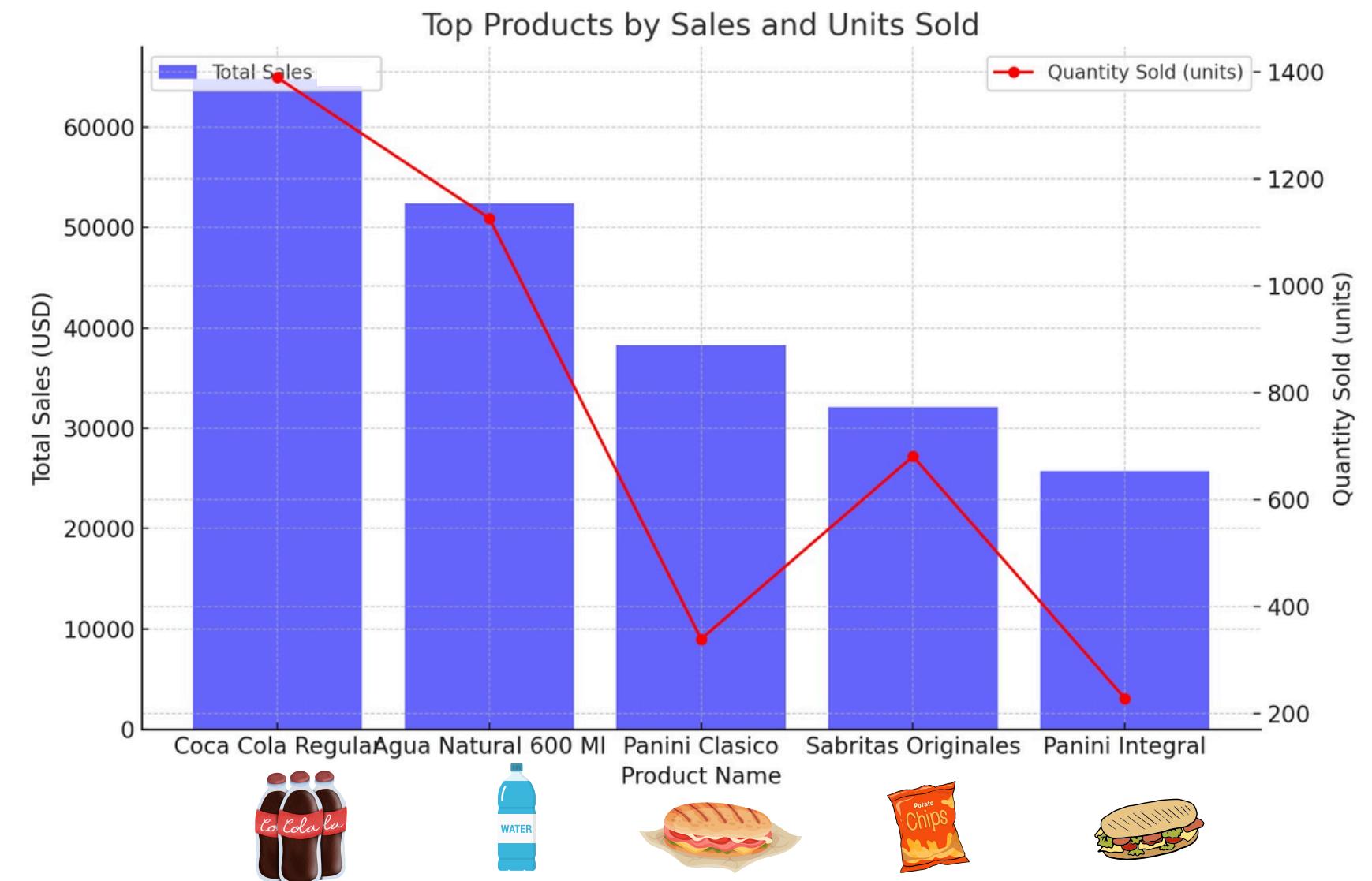
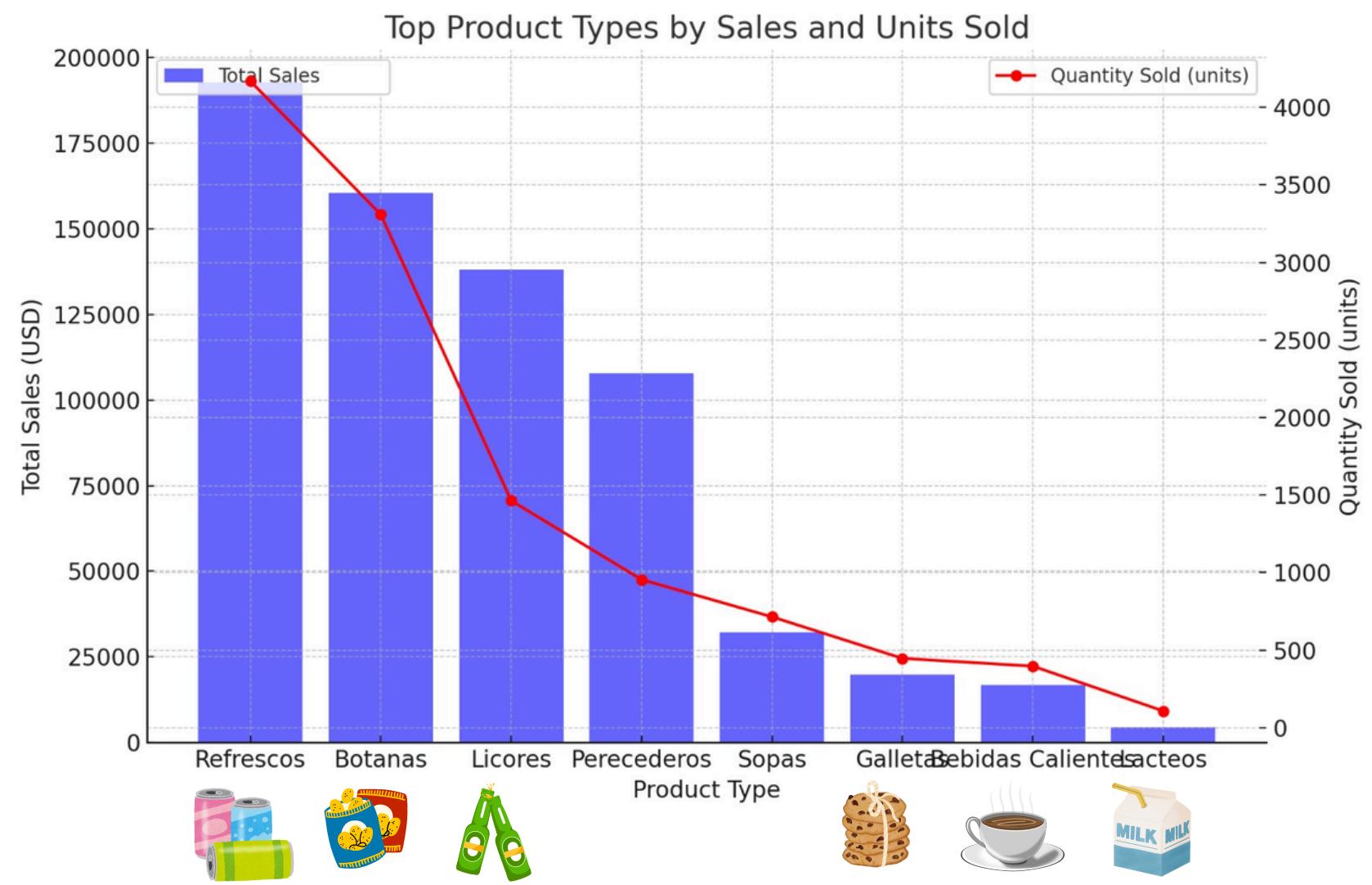
- Flights TEC_Valid
- Sales TEC_Valid



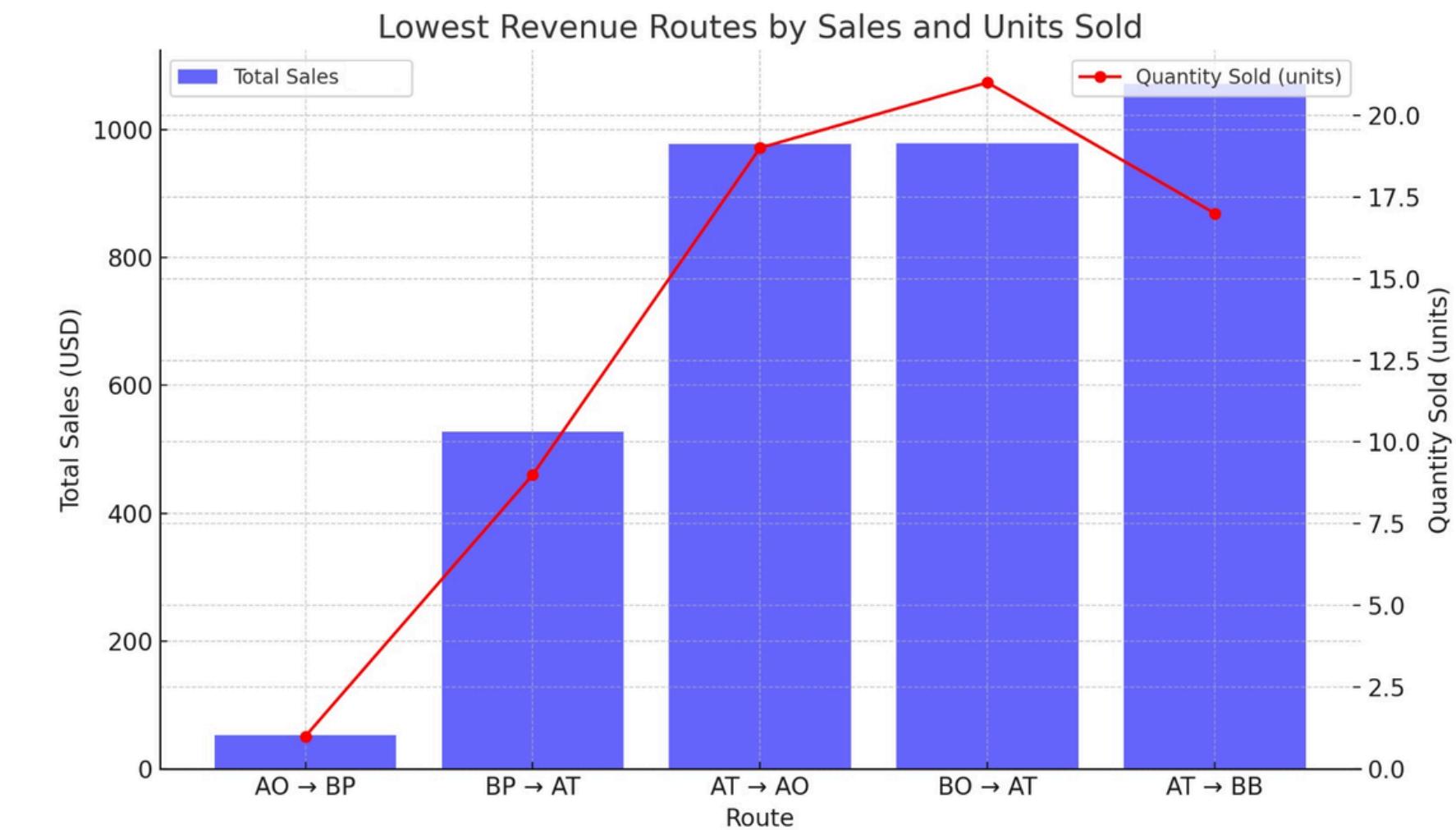
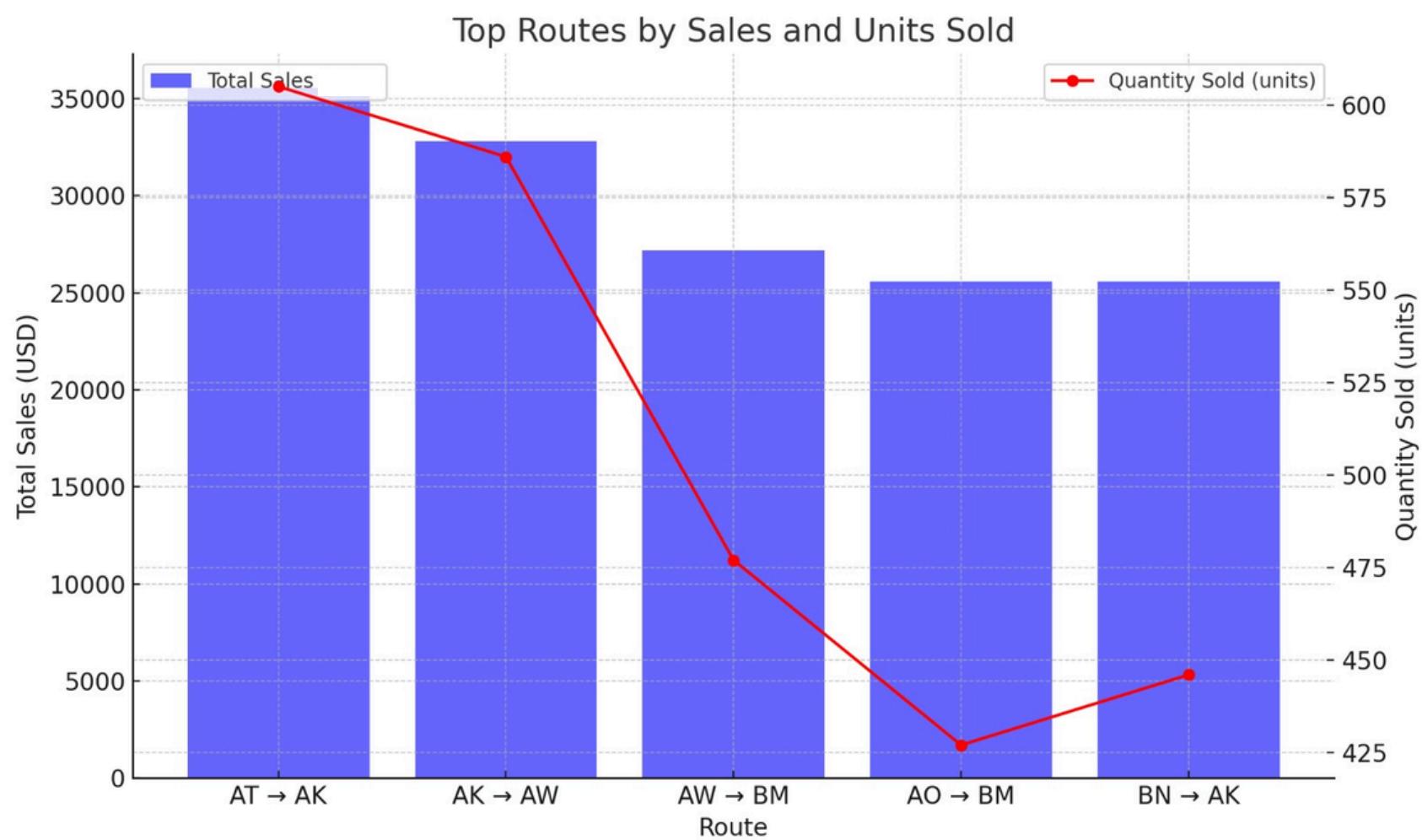
METODOLOGIA



PRINCIPALES INSIGHTS



PRINCIPALES INSIGHTS

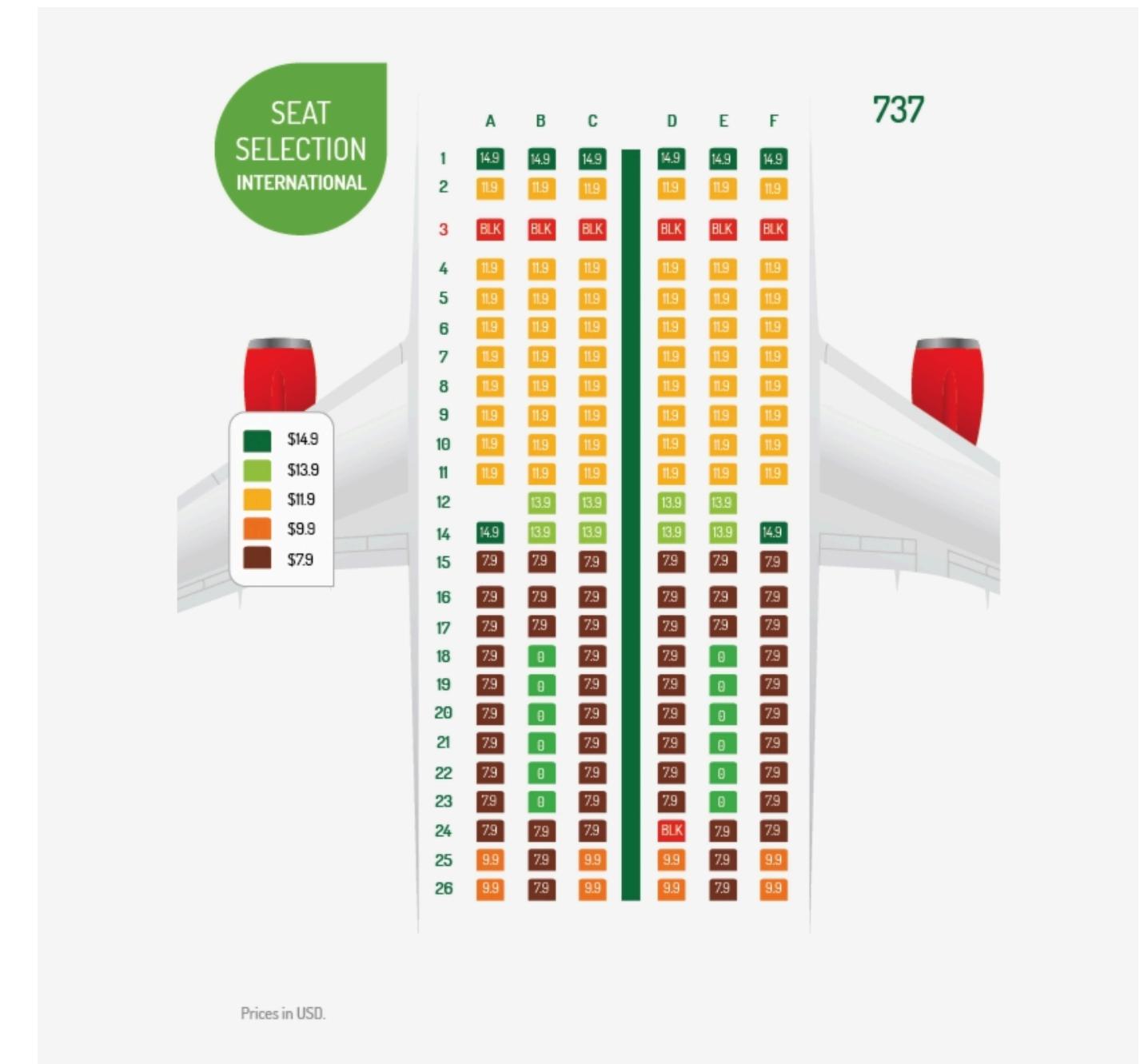


MODELO DE PREDICCIÓN VOLUMEN DE PASAJEROS EN VUELOS DE ENERO 2024

El modelo para realizar predicciones del mes enero del 2024 es el **Gradient Boosting Regressor**.

Métricas Utilizadas:

- Departamentos de origen
- Departamentos de destino
- Tipo de destino
- Tipo de temporada
- Horario de vuelo



RESULTADOS

	Flight_ID	PredictedPassengers
18624	b5b6665fdc1048a63fd7e2f224a51b6c	164.222885
32670	2385ceef5123f0cdf95994f9978d216	193.554184
77528	c073f59d3bea0a295cc8e466265c28c1	185.164856
102760	4d95c80c854b637e6fdf951e85f1a05b	182.540955
128230	c073f59d3bea0a295cc8e466265c28c1	190.146240
224693	b5b6665fdc1048a63fd7e2f224a51b6c	177.299225
226285	35b62c4a09ba0a69e0c9b9a006d83f18	177.299225

MODELO DE PREDICCIÓN VENTA DE PRODUCTOS DE ENERO 2024

El modelo para realizar predicciones del mes enero del 2024 es el **Gradient Boosting Regressor**.

Métricas Utilizadas:

- Departamentos de origen
- Departamentos de destino
- Tipo de destino
- Tipo de temporada
- Horario de vuelo
- Ocupación de vuelo
- Tipo de producto
- Nombre de producto



PLAN ABASTECIMIENTO OPTIMIZADO

```
import math

# Lista de aeropuertos disponibles
aeropuertos_disponibles = ["AO", "AU", "AW", "BA", "BM", "AT", "AK"]

# Función para calcular el inventario mínimo
def calcular_inventario_minimo(quantity_list):
    return math.ceil(sum(quantity_list[:2]) * 1.05)

# Función para seleccionar aeropuertos y determinar cuánto recargar de cada producto
def seleccionar_aeropuertos_con_productos(inventario, inventario_maximo, quantities, ruta):
    aeropuertos_reabastecer = []

    for idx, aeropuerto in enumerate(ruta):
        recarga = {}

        for producto, inventario_actual in inventario.items():
            inventario_minimo = calcular_inventario_minimo(quantities[producto][idx:idx+2])

            # Mostrar inventario antes de reabastecer
            print(f"Inventario de {producto} al aterrizar en {aeropuerto}: {inventario_actual}\n")

            if inventario_actual < inventario_minimo and aeropuerto in aeropuertos_disponibles:
                cantidad_a_recargar = min(inventario_minimo - inventario_actual, inventario_maximo[producto] - inventario_actual)
                recarga[producto] = cantidad_a_recargar
                inventario[producto] = inventario_actual + cantidad_a_recargar

            # Reducir el inventario en función del consumo del vuelo actual
            if idx < len(quantities[producto]):
                inventario[producto] -= quantities[producto][idx]

        if recarga:
            print(f"Recarga en {aeropuerto}: {recarga}\n")

    return aeropuertos_reabastecer

# Ejemplo de uso
inventario_actual = {"refrescos": 55, "papas": 30}
inventario_maximo = {"refrescos": 55, "papas": 30}
quantities = {
    "refrescos": [20, 15, 10, 40, 40, 5], # Consumo de refrescos por vuelo
    "papas": [25, 20, 5, 5, 1, 0]         # Consumo de papas por vuelo
}

# Ejemplo de uso
ruta_vuelo = ["BM", "AU", "AX", "AW", "XA", "AK"]

aeropuertos_para_reabastecer = seleccionar_aeropuertos_con_productos(inventario_actual, inventario_maximo, quantities, ruta_vuelo)
```

IMPACTO Y TRABAJO FUTURO



- 1 Mejora de la experiencia del cliente
- 2 Incremento de ingresos
- 3 Reducción del desperdicio de alimentos
- 4 Uso eficiente de recursos