DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS

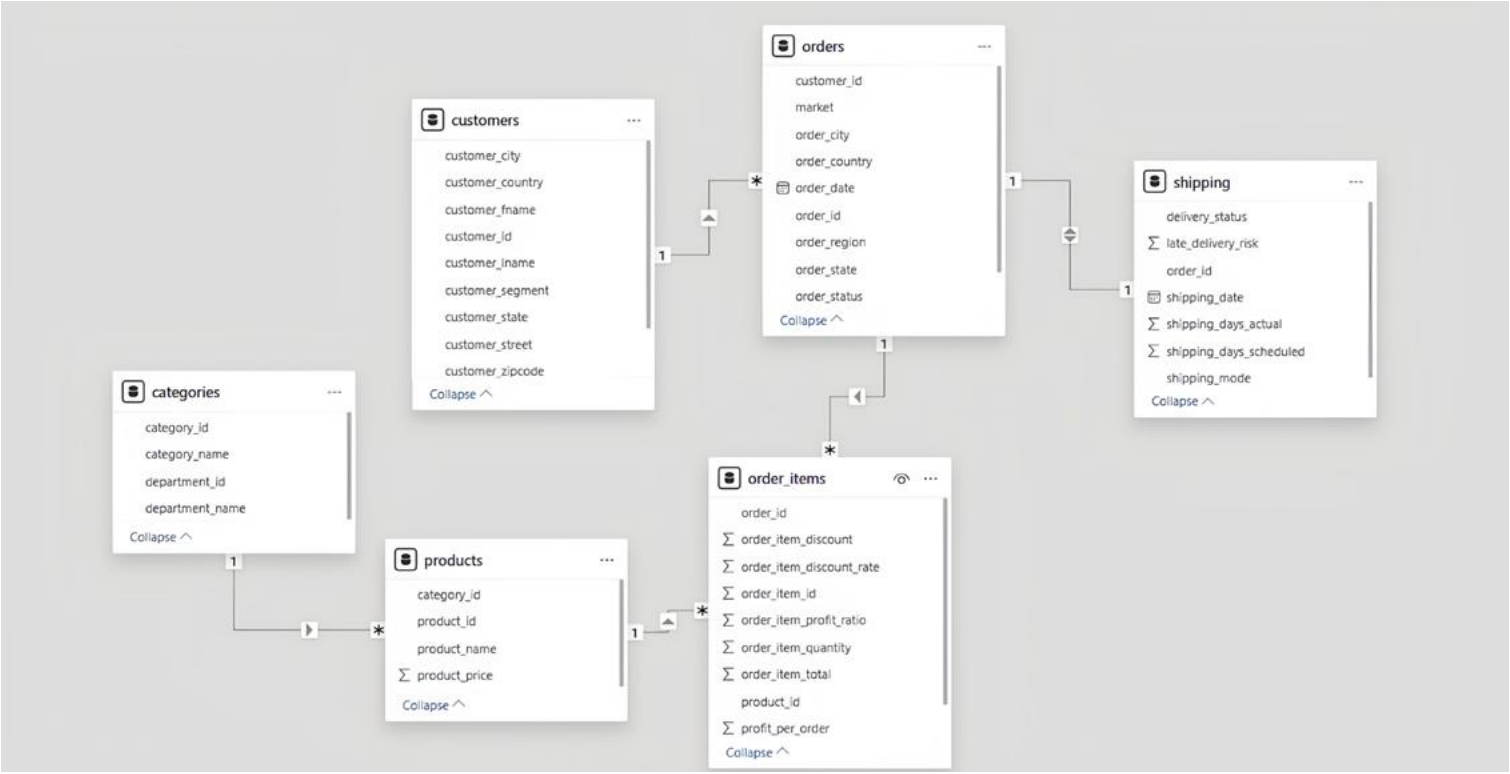
Data Source
DataCo Global Company

Data Analysis Team

- Eslam Mohamed
- Ebrahim Elnaqer
- Mahmoud Elafandy
- Abdullah Elbelawy
- Radwan Mohamed

- One of the most in-demand fields
- Big Data (180519 Raw, 53 Columns)
- A lot of challenges

WHAT IS THIS DATA ABOUT?



01

DATA CLEANING

Tools used
Python & Excel

Type	Days for shipping (real)	Days for shipment (scheduled)	Benefit per order	Sales per customer	Delivery Status

type	shipping_days_actual	shipping_days_scheduled	profit_per_order	delivery_status

```python
df[['product_status', 'product_image', 'customer_password', 'customer_email']].head(5)
```

[56] ✓ 0.0s

	product_status	product_image	customer_password	customer_email
0	0	http://images.acmesports.sports/Smart+watch	XXXXXXXXX	XXXXXXXXX
1	0	http://images.acmesports.sports/Smart+watch	XXXXXXXXX	XXXXXXXXX
2	0	http://images.acmesports.sports/Smart+watch	XXXXXXXXX	XXXXXXXXX
3	0	http://images.acmesports.sports/Smart+watch	XXXXXXXXX	XXXXXXXXX
4	0	http://images.acmesports.sports/Smart+watch	XXXXXXXXX	XXXXXXXXX

```python
# Redundancy and useless columns cleaning
df.drop(columns=['product_status', 'product_image', 'customer_password', 'customer_email',
                 'order_customer_id', 'order_item_product_price', 'order_profit_per_order',
                 'sales_per_customer', 'product_category_id', 'order_product_id'], inplace=True)
```

[57] ✓ 0.0s

```python
df.shape
```

[58] ✓ 0.0s

```
(180519, 43)
```

BEFORE

```
df.dtypes[df.dtypes == 'object']
# to date 'order_date', 'shipping_date'
```

[97] ✓ 0.0s

```
type                 object
delivery_status      object
category_name        object
customer_city        object
customer_country     object
customer_fname       object
customer_lname       object
customer_segment     object
customer_state       object
customer_street      object
department_name      object
market               object
order_city           object
order_country        object
order_date           object
order_region         object
order_state          object
order_status         object
product_name         object
shipping_date        object
shipping_mode        object
dtype: object
```

```
df.dtypes[df.dtypes == 'int64']
```

[96] ✓ 0.0s

```
shipping_days_actual       int64
shipping_days_scheduled    int64
late_delivery_risk         int64
category_id                int64
customer_id                int64
department_id              int64
order_id                   int64
order_item_id              int64
order_item_quantity        int64
product_id                 int64
dtype: object
```

```
df.dtypes[df.dtypes == 'float64']
```

[95] ✓ 0.0s

```
profit_per_order           float64
customer_zipcode           float64
latitude                   float64
longitude                  float64
order_item_discount        float64
order_item_discount_rate   float64
order_item_profit_ratio    float64
sales                      float64
order_item_total           float64
order_zipcode              float64
product_description        float64
product_price              float64
dtype: object
```

AFTER

```
df.dtypes[df.dtypes == 'int64']
```

[191] ✓ 0.0s

```
shipping_days_actual       int64
shipping_days_scheduled    int64
late_delivery_risk         int64
order_item_quantity        int64
dtype: object
```

```
df.dtypes[df.dtypes == 'datetime64[ns]']
```

[168] ✓ 0.0s

```
order_date       datetime64[ns]
shipping_date    datetime64[ns]
dtype: object
```

```
df.dtypes[df.dtypes == 'float64']
```

[165] ✓ 0.0s

```
profit_per_order           float64
latitude                   float64
longitude                  float64
order_item_discount        float64
order_item_discount_rate   float64
order_item_profit_ratio    float64
sales                      float64
order_item_total           float64
product_description        float64
product_price              float64
dtype: object
```

```
df.dtypes[df.dtypes == 'object']
# to date 'order_date', 'shipping_date'
```

[167] ✓ 0.0s

```
type                 object
delivery_status      object
category_id          object
category_name        object
customer_city        object
customer_country     object
customer_fname       object
customer_id          object
customer_lname       object
customer_segment     object
customer_state       object
customer_street      object
customer_zipcode     object
department_id        object
department_name      object
market               object
order_city           object
order_country        object
order_id             object
order_item_id        object
order_region         object
order_state          object
order_status         object
order_zipcode        object
product_id           object
product_name         object
shipping_mode        object
dtype: object
```

NAN HANDLING

```python
df.isnull().sum().sort_values(ascending=False).head()
```

[242] ✓ 0.2s

```
product_description    180519
order_zipcode          155679
customer_lname              8
customer_zipcode            3
delivery_status             0
dtype: int64
```

```python
# product_description, order_zipcode
df.drop(columns=['product_description', 'order_zipcode'], inplace=True)
```

[243] ✓ 0.0s

Customer_lname column

```python
# customer_lname
mask = df['customer_lname'].isna()
lname_most = df.groupby('customer_city')['customer_lname'].transform(lambda x : x.mode()[0] if not x.mode().empty else np.nan)
df['customer_lname'] = df['customer_lname'].fillna(lname_most)
df.loc[mask, 'customer_lname']
```

[244] ✓ 0.3s

```
41422     Smith
50447     Smith
86260     Smith
92621     Smith
97530     Smith
120404    Smith
166129    Smith
167119    Brooks
Name: customer_lname, dtype: object
```

Customer_zipcode column

```python
# customer_zipcode
df['customer_zipcode'] = df['customer_zipcode'].replace('nan', np.nan)
mask = df['customer_zipcode'].isna()
cols_target = ['customer_zipcode', 'customer_state', 'customer_city', 'customer_street']
cols_source = ['customer_state', 'customer_city', 'customer_street']
zip_nan = df.loc[mask]
zip_nan[cols_target]
```

[280] ✓ 0.0s

	customer_zipcode	customer_state	customer_city	customer_street
35704	NaN	95758	CA	Elk Grove
46440	NaN	95758	CA	Elk Grove
82511	NaN	91732	CA	El Monte

```python
df.loc[mask, cols_target] = df.loc[mask, cols_source].assign(street_nan=np.nan).values
df.loc[mask, cols_target]
```

[281] ✓ 0.0s

	customer_zipcode	customer_state	customer_city	customer_street
35704	95758	CA	Elk Grove	NaN
46440	95758	CA	Elk Grove	NaN
82511	91732	CA	El Monte	NaN

```python
street_most = df.groupby('customer_city')['customer_street'].transform(
    lambda x : x.mode()[0] if not x.mode().empty else np.nan)

df['customer_street'] = df['customer_street'].fillna(street_most)
df.loc[mask, cols_target]
```

[77] ✓ 0.3s

	customer_zipcode	customer_state	customer_city	customer_street
35704	95758	CA	Elk Grove	4096 Amber Bend
46440	95758	CA	Elk Grove	4096 Amber Bend
82511	91732	CA	El Monte	2845 Jagged Oak Vale

RECATEGORIZATION

```python
df.select_dtypes(include='object').nunique().sort_values()
```

[283] ✓ 0.5s

```
customer_country         2
customer_segment         3
delivery_status          4
type                     4
shipping_mode            4
market                   5
order_status             9
department_id           11
department_name         11
order_region            23
customer_state          44
category_name           50
category_id             51
product_id             118
product_name           118
order_country          164
customer_city          562
customer_fname         782
customer_zipcode       997
order_state           1089
customer_lname        1109
order_city            3597
customer_street       6951
customer_id          20652
order_id             65752
order_item_id       180519
dtype: int64
```

```python
df[df['category_name'] == 'Electronics']['product_name'].unique()
```

✓ 0.0s

```
array(["Under Armour Men's Compression EV SL Slide",
       'Bridgestone e6 Straight Distance NFL San Dieg',
       'Titleist Pro V1x Golf Balls',
       'Bridgestone e6 Straight Distance NFL Tennesse',
       "Under Armour Kids' Mercenary Slide",
       "Under Armour Women's Ignite PIP VI Slide",
       'Titleist Pro V1x High Numbers Golf Balls',
       'Bridgestone e6 Straight Distance NFL Carolina',
       "Under Armour Women's Ignite Slide",
       'Titleist Pro V1x High Numbers Personalized Go',
       'Titleist Pro V1 High Numbers Personalized Gol'], dtype=object)
```

RECTEGORIZATION

1	category	product_name
2	Baby	Baby sweater
3	Basketball	Nike Kids' Grade School KD VI Basketball Shoe
4	Basketball	Nike Men's Kobe IX Elite Low Basketball Shoe
5	Books	Sports Books
6	Cameras	Web Camera
7	Cameras	GoPro HERO3+ Black Edition Camera
8	Camping & Hiking	ENO Atlas Hammock Straps
9	Camping & Hiking	Diamondback Women's Serene Classic Comfort Bi
10	Camping & Hiking	Diamondback Boys' Insight 24 Performance Hybr
11	Camping & Hiking	The North Face Women's Recon Backpack
12	Camping & Hiking	insta-bed Neverflat Air Mattress
13	Camping & Hiking	Diamondback Girls' Clarity 24 Hybrid Bike 201
14	Camping & Hiking	Yakima DoubleDown Ace Hitch Mount 4-Bike Rack
15	Camping & Hiking	Merrell Men's All Out Flash Trail Running Sho
16	Camping & Hiking	Merrell Women's Grassbow Sport Waterproof Hik
17	Camping & Hiking	Merrell Women's Siren Mid Waterproof Hiking B
18	Camping & Hiking	Merrell Women's Grassbow Sport Hiking Shoe
19	Cardio Equipment	Smart watch
20	Cardio Equipment	Polar FT4 Heart Rate Monitor
21	Cardio Equipment	SOLE E25 Elliptical
22	Cardio Equipment	Mio ALPHA Heart Rate Monitor/Sport Watch
23	Cardio Equipment	Polar Loop Activity Tracker
24	Cardio Equipment	Garmin Forerunner 910XT GPS Watch
25	Cardio Equipment	Fitbit The One Wireless Activity & Sleep Trac
26	Cardio Equipment	GolfBuddy VT3 GPS Watch
27	Cardio Equipment	Bushnell Pro X7 Jolt Slope Rangefinder
28	Cardio Equipment	SOLE E35 Elliptical
29	CDs	CDs of rock
30	Children's Clothing	Children's heaters
31	Cleats	adidas Men's F10 Messi TRX FG Soccer Cleat
32	Cleats	Nike Men's CJ Elite 2 TD Football Cleat
33	Cleats	adidas Kids' F5 Messi FG Soccer Cleat
34	Computers	Dell Laptop
35	Consumer Electronics	Industrial consumer electronics

```python
mapping_df = pd.read_excel('products_list.xlsx')

unique_cats = mapping_df['new_category'].unique()
cat_to_id = {name: i+1 for i, name in enumerate(unique_cats)}
mapping_df['new_category_id'] = mapping_df['new_category'].map(cat_to_id)

df = df.merge(mapping_df, on='product_name', how='left')

df = df.drop(['category_id', 'category_name'], axis=1)
```

[284] ✓ 0.3s

DATA SPLITTING

```python
def create_modular_tables(df):
    customers = df[[
        'customer_id', 'customer_fname', 'customer_lname', 'customer_segment',
        'customer_street', 'customer_zipcode', 'customer_city', 'customer_state', 'customer_country', 'latitude', 'longitude'
    ]].drop_duplicates(subset=['customer_id'])

    products = df[[
        'product_id', 'product_name', 'product_price', 'category_id'
    ]].drop_duplicates(subset=['product_id'])

    categories = df[[
        'category_id', 'category_name', 'department_id', 'department_name'
    ]].drop_duplicates(subset=['category_id'])

    orders = df[[
        'order_id', 'customer_id', 'order_date', 'order_status',
        'type', 'market', 'order_region', 'order_city', 'order_state', 'order_country'
    ]].drop_duplicates(subset=['order_id'])

    items = df[[
        'order_item_id', 'order_id', 'product_id', 'order_item_quantity',
        'order_item_discount', 'order_item_discount_rate', 'sales',
        'order_item_total', 'profit_per_order', 'order_item_profit_ratio'
    ]]

    shipping = df[[
        'order_id', 'shipping_date', 'shipping_days_actual',
        'shipping_days_scheduled', 'shipping_mode', 'delivery_status', 'late_delivery_risk'
    ]].drop_duplicates(subset=['order_id'])

    return customers, products, categories, orders, items, shipping

customers, products, categories, orders, items, shipping = create_modular_tables(df)
```

```python
output_dir = 'processed_data'
if not os.path.exists(output_dir):
    os.makedirs(output_dir)

customers.to_csv(f'{output_dir}/customers.csv', index=False)
products.to_csv(f'{output_dir}/products.csv', index=False)
categories.to_csv(f'{output_dir}/categories.csv', index=False)
orders.to_csv(f'{output_dir}/orders.csv', index=False)
items.to_csv(f'{output_dir}/order_items.csv', index=False)
shipping.to_csv(f'{output_dir}/shipping.csv', index=False)
```

✓ 0.0s

02

DATA ANALYSIS

Tools used
Python, SQL, Excel, Power Bi

2.1 SUPPLY CHAIN AND SHIPPING EFFICIENCY

2.2 SALES

2.3 MARKETS AND CUSTOMERS BEHAVIOR

2.4 PRODUCTS AND CATEGORIES

01

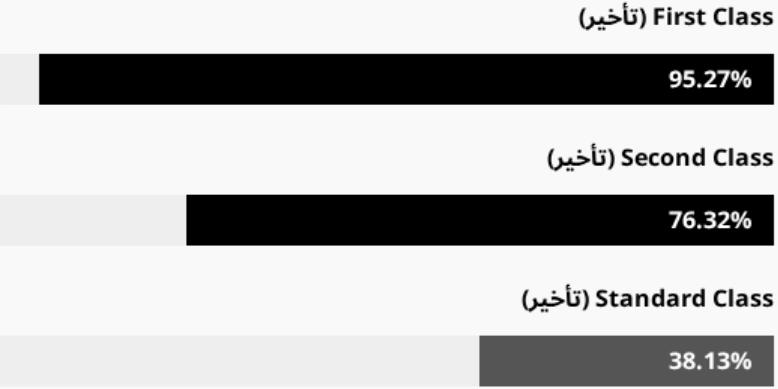
SHIPPING PROBLEMS

STRATEGIC ANALYSIS & SOLUTIONS

مشكلة (1): أزمة الوعود بالشحن السريع

أزمة "First Class"

نسبة التأخير الفعلي تصل إلى **95.27%** العميل يدفع مقابل سرعة لا يحصل عليها.



الحل المقترح (Action Plan)

- → **تعديل خوارزمية المواعيد :**إضافة يوم وقائي (Buffer Day)لمطابقة الواقع.
- → **نظام الإنذار المبكر** Dashboard **:**لمراقبة Late Riskلحظياً.
- → **تعويضات استباقية :**إرسال كود خصم آلي فور رصد التأخير لامتصاص غضب العميل.

الأثر المالي المباشر للإلغاءات

2,855

إجمالي عدد الإلغاءات

1.57M

القيمة الإجمالية (دولار)

58%

إلغاء بسبب التأخير

الاستنتاج :التأخير هو المحرك الرئيسي لضياع **898 ألف دولار** من المبيعات .معالجة التأخير تعني استرداد هذه السيولة فوراً.

التوصيات النهائية (Strategic Roadmap)

تقنياً

تعديل خوارزمية الجدولة فوراً للفئات الحرجة (Buffer Day) لضمان صدق الوعود.

تشغيلياً

إعادة تقييم شركات شحن First Class في الأقاليم المتضررة (Central Asia) أو تغيير المورد.

تسويقياً

تفعيل نظام "الاعتذار الرقمي" بكوبونات خصم 10% قبل أن يشتكي العميل.

أثر تطبيق الحلول (محاكاة النتائج)

استعادة الكفاءة التشغيلية

- **تحويل الحالات :**تحويل 21,130 طلب من "متأخر " إلى "في الموعد."
- **حماية الإيرادات :** تأمين مبيعات بقيمة 10.6مليون دولار من خطر الإلغاء.
- **الولاء :** زيادة احتمالية عودة العميل بنسبة 40% عند الاعتذار الاستباقي.

الوضع الحالي

الوضع المستقبلي

مشكلة (2): فشل الموردين في فئات محددة

أزمة فئات الجولف والكاميرات

فئات مثل **Golf Shoes** تسجل نسبة تأخير 68.85%
المشكلة تكمن في المورد (Lead Time).

الحل: فرض اتفاقيات مستوى خدمة (SLA) صارمة، وتفعيل "القائمة السوداء" للموردين غير الملتزمين.

عدد الطلبات الفعلي وعدد الطلبات المتأخرة

```sql
SELECT
    COUNT(*) AS Total_Orders,
    SUM(CASE WHEN shipping_days_actual > shipping_days_scheduled THEN 1 ELSE 0 END) AS Total_Late_Orders,
    (SUM(CASE WHEN shipping_days_actual > shipping_days_scheduled THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS Late_Percentage
FROM shipping;
```

Results | Messages

	Total_Orders	Total_Late_Orders	Late_Percentage
1	65749	37697	57.334712315016

تحليل كفاءة التوصيل حسب نوع الشحن ونسبة التأخير

```sql
SELECT
    s.shipping_mode AS [Shipping Mode],
    -- إجمالي عدد الطلبات في كل فئة
    COUNT(s.order_id) AS [Total Orders],
    -- عدد الطلبات التي سُجلت حالتها "Late delivery" فعلياً
    SUM(CASE WHEN s.delivery_status = 'Late delivery' THEN 1 ELSE 0 END) AS [Actual Late Orders],
    -- حساب نسبة التأخير الفعلية لتوضيح حجم المشكلة
    ROUND(CAST(SUM(CASE WHEN s.delivery_status = 'Late delivery' THEN 1 ELSE 0 END) AS FLOAT) / COUNT(s.order_id) * 100, 2) AS [Late Percentage %]
FROM shipping s
GROUP BY s.shipping_mode
ORDER BY [Late Percentage %] DESC;
```

	Shipping Mode	Total Orders	Actual Late Orders	Late Percentage %
1	First Class	10078	9601	95.27
2	Second Class	12778	9803	76.72
3	Same Day	3571	1648	46.15
4	Standard Class	39322	14995	38.13

تحليل كفاءة الشحن حسب الأقاليم الجغرافية

```sql
SELECT
    o.order_region AS [Region],
    COUNT(o.order_id) AS [Total Orders],
    -- متوسط أيام التأخير (الفعلي - المخطط)
    ROUND(AVG(CAST(s.shipping_days_actual AS FLOAT) - s.shipping_days_scheduled), 2) AS [Avg Delay Days],
    -- نسبة خطر التأخير (Late Risk Rate)
    ROUND(AVG(CAST(s.late_delivery_risk AS FLOAT)) * 100, 2) AS [Late Risk %],
    -- عدد الحالات التي سُجلت فعلياً "Late delivery"
    SUM(CASE WHEN s.delivery_status = 'Late delivery' THEN 1 ELSE 0 END) AS [Actual Late Count]
FROM orders o
JOIN shipping s ON o.order_id = s.order_id
GROUP BY o.order_region
HAVING AVG(CAST(s.shipping_days_actual AS FLOAT) - s.shipping_days_scheduled) > 0.5
ORDER BY [Avg Delay Days] DESC;
```

	Region	Total Orders	Avg Delay Days	Late Risk %	Actual Late Count
1	Central Asia	184	0.65	55.43	102
2	Central Africa	556	0.64	57.55	320
3	South of USA	1345	0.6	55.99	753
4	South Asia	3335	0.6	55.92	1865
5	Eastern Europe	1292	0.59	55.96	723
6	East Africa	613	0.59	56.77	348
7	Western Europe	10009	0.59	55.79	5584
8	West Asia	2022	0.58	55.98	1132
9	Southeast Asia	4355	0.58	55.59	2421
10	East of USA	2323	0.58	54.93	1276
11	Eastern Asia	3318	0.57	54.79	1818
12	Oceania	4362	0.57	54.61	2382
13	Central America	9396	0.57	54.87	5156
14	US Center	1935	0.57	55.35	1071
15	West of USA	2667	0.56	54.41	1451
16	South America	4979	0.55	54.15	2696

عدد الأيام المجدولة للشحن وعدد الأيام الفعلي

```sql
SELECT
    shipping_mode,
    AVG(shipping_days_actual) as Actual,
    AVG(shipping_days_scheduled) as Scheduled
FROM shipping
GROUP BY shipping_mode;
```

	shipping_mode	Actual	Scheduled
1	First Class	2	1
2	Same Day	0	0
3	Standard Class	3	4
4	Second Class	4	2

عرض مشكلة تأخر categories من قبل الموردين

```sql
SELECT top 15
    c.category_name,
    COUNT(oi.order_id) AS Total_Orders,
    SUM(CASE WHEN s.shipping_days_actual > s.shipping_days_scheduled THEN 1 ELSE 0 END) AS Delayed_Orders,
    ROUND(CAST(SUM(CASE WHEN s.shipping_days_actual > s.shipping_days_scheduled THEN 1 ELSE 0 END) AS FLOAT) / COUNT(oi.order_id) * 100, 2) AS Delay_Rate
FROM order_items oi
JOIN products p ON oi.product_id = p.product_id
JOIN categories c ON p.category_id = c.category_id
JOIN shipping s ON oi.order_id = s.order_id
GROUP BY c.category_name
HAVING COUNT(oi.order_id) > 50
ORDER BY Delay_Rate_Percentage DESC;
```

	category_name	Total_Orders	Delayed_Orders	Delay_Rate_Percentage
1	Golf Shoes	61	42	68.85
2	Cameras	624	388	62.18
3	Pet Supplies	492	302	61.38
4	Men's Clothing	1075	657	61.12
5	Crafts	484	288	59.5
6	Golf Accessories	1893	1126	59.48
7	Music	434	258	59.45
8	Women's Apparel	423	251	59.34
9	Consumer Electronics	430	254	59.07
10	Women's Clothing	1247	732	58.7
11	Health and Beauty	362	212	58.56
12	Golf Bags & Carts	1365	799	58.53
13	Cardio Equipment	730	427	58.49
14	Slides	1455	851	58.49
15	Golf Balls	2029	1186	58.45

02

SALES PERFORMANCE

AND STRATEGIC GROWTH

1. تحليل أوقات الذروة والركود

فجوة الـ 9 أضعاف

المبيعات تنهار في يونيو ($3,017) وتقفز في أكتوبر ($27,731). المشكلة في تثبيت ميزانية التسويق طوال العام.

شهر أكتوبر (الذروة)

$27,731

شهر يونيو (الركود)

$3,017

الحل والأثر

إعادة الهيكلة: ترحيل ميزانية أضعف 3 شهور لدعم مخزون الذروة.

المال: توفير 30% من المصاريف الإعلانية المهدرة.

الأداء: رفع مبيعات الذروة بنسبة 30% إضافية.

2. مصفوفة الخسائر وتآكل الأرباح

$1.3M

نزيف خصومات في فئتين فقط

15-20%

تحسن متوقع في هامش الربح

15%

توفير في مساحة المخازن

المشكلة :

فئات (Hunting & Shooting) و (Water Sports) تستنزف السيولة بخصومات هائلة لمجرد محاولة بيع منتجات غير مطلوبة.

القرار : فرض سقف خصم صارم (Discount Cap) وإيقاف التعاقد الفوري مع موردي هذه الفئات.

3. متوسط الطلب وطرق الدفع

فجوة القيمة الشرائية ($70)

المتجر يفقد إيرادات محتملة بسبب عملاء الدفع النقدي الذين يشترون سلالاً أرخص.

الدفع الإلكتروني AOV - (TRANSFER/DEBIT)

الدفع النقدي AOV - (CASH)

استراتيجية الحوافز الذكية

- تقديم شحن مجاني أو خصم 5% للدفع المسبق بالبطاقة.
- **الأثر:** زيادة إيرادات شريحة الكاش بنسبة 40% فورياً.

4. الربحية الوهمية والثغرة الأمنية

ثغرة التحويل البنكي (TRANSFER)

اكتشفنا أن 100% من محاولات النصب (Fraud) تتم عبر هذه الطريقة لمنتجات معينة مثل Nike و Perfect Fitness.

+$20,000

إيرادات وهمية (Fake Revenue)

20%

خفض مستهدف لمعدل الاحتيال

DEBIT

الحل :فرض المصادقة الفورية

5. الارتباط الشرائي واستراتيجية "الطُعم"

تحليل سلة المشتريات

الأوردرات التي تتجاوز 1,400$ تحتوي دائماً على منتج مخفض يعمل كـ "طُعم " لجذب العميل لشراء منتجات أخرى عالية الربحية.

توصية الـ Bundles

منع بيع "المنتج الخاسر" بمفرده؛ يجب ربطه بمنتجات رابحة ضمن حزم (Bundles) لتعويض الخصم وتحقيق صافي مكسب.

6. الجدوى الحقيقية للخصومات

فخ الـ 25%

ندفع أكثر من ضعف التكلفة عند نسبة 25% لنحصل على نفس المبيعات تقريباً عند نسبة 10%.

تكلفة خصم 25% ($511k)

21,355 قطعة

تكلفة خصم 10% ($204k)

21,338 قطعة

$300,000

توفير مباشر عند اعتماد "النسبة الذهبية (13% - 10%) " كمعيار للحملات.

7. حساسية السعر جغرافياً

أسواق النخبة (Premium)

شرق آسيا وشمال أوروبا: إيرادات مليونية بخصومات 4% فقط .

القرار :البيع بالسعر الكامل.

صائدي الخصومات (Hunters)

آسيا الوسطى :خصومات 18% مقابل إيرادات ضعيفة.

القرار :خفض ميزانية التسويق.

الأثر المالي

تعظيم هوامش الربح الصافي عبر توجيه الـ ROI للأسواق الأكثر ولاءً للجودة وليس السعر.

8. مصفوفة التقسيم التجاري للمنتجات

محركات السيولة

MASS
MARKET

منتجات (Nike, Perfect Fitness): مبيعات ضخمة تضمن تدفق الكاش .تستخدم كأدوات جذب (Bundles).

صناع الأرباح

TOP
LEADERS

منتجات (Smart Watches): هوامش استثنائية . تتطلب برامج ولاء (VIP) وتأمين مخزون استباقي.

STRATEGIC ROADMAP

الخلاصة :المال، الأداء، الوقت

$1.3M

توفير من الفئات الخاسرة

$300K

توفير من عقلنة الخصومات

+40%

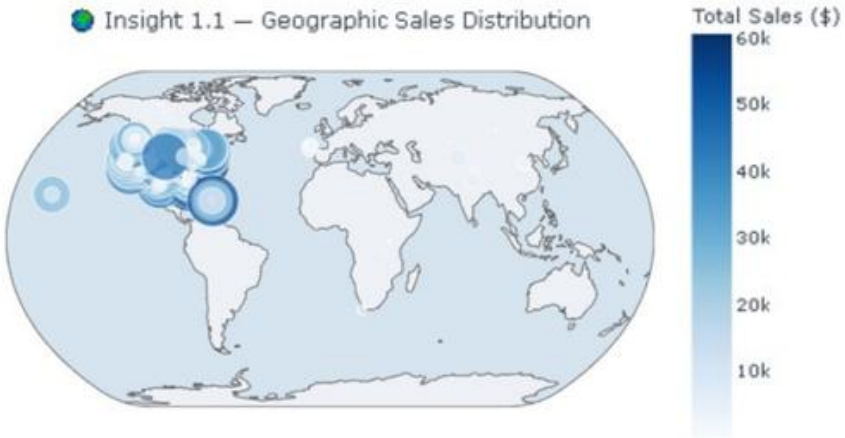
نمو مبيعات الذروة

03

MARKETS AND

CUSTOMERS BEHAVIOR

1.1التوزيع الجغرافي للمبيعات



هيمنة الأسواق الرئيسية

هناك 5 أسواق رئيسية تسيطر على إجمالي المبيعات، مع تركز كبير في **بورتو ريكو، الولايات المتحدة، وأمريكا اللاتينية.**

تكرر اسم الولايات المتحدة 4 مرات في قائمة الأعلى مبيعاً مما يعكس تنوعاً جغرافياً داخل الدولة الواحدة.

توصية: التركيز على النمو الآمن

15%

ميزانية تسويق رقمي مقترحة

3

أهم أسواق مستهدفة

12%

نمو متوقع في المبيعات

التركيز الإعلاني على المناطق الأعلى شراءً هو قرار استراتيجي منخفض المخاطر لضمان التوسع المستدام.

1.2تحليل شرائح العملاء

التوصية :زيادة معدل تكرار الشراء لشريحة **Corporate** بنسبة 20% عبر برنامج ولاء مخصص، لتعظيم العائد من الـ AOV المرتفع.

1.3 الموسمية: تحليل الذروة والركود

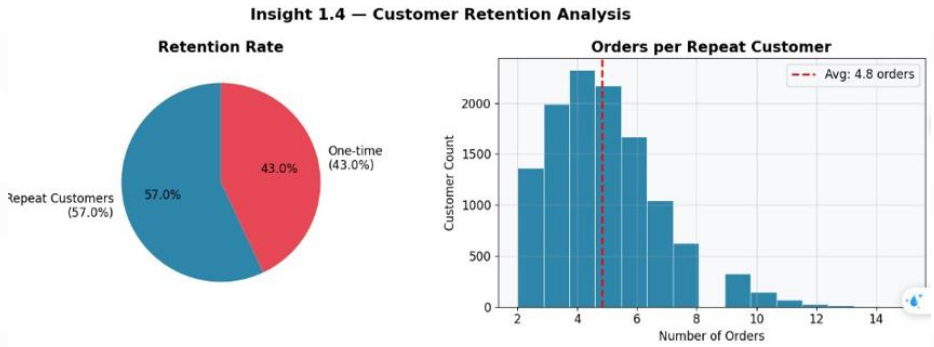
يناير (الذروة): ضرورة زيادة المخزون بنسبة 20% وتكثيف الإعلانات لتقليل نفاذ البضاعة.

ديسمبر (الركود): تطبيق خصومات ترويجية (10-15%) لتحفيز المبيعات وتحريك الراكد.

معدل الاحتفاظ بالعملاء 1.4

قوة الاحتفاظ بالعملاء

الشركة تمتلك قاعدة قوية حيث أن أكثر من نصف العملاء يعودون للشراء مرة أخرى بمتوسط 4.8 أوردر لكل عميل.

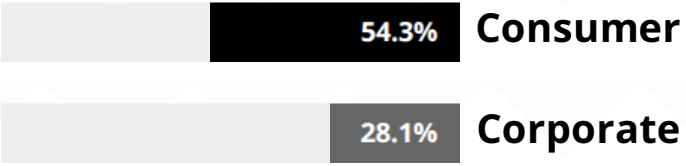


→ إطلاق حملة Email Retention تستهدف العملاء غير النشطين (90) يوم.

→ **الهدف:** رفع معدل الاحتفاظ بنسبة 15%.

1.5 تحليل أنماط الاحتيال

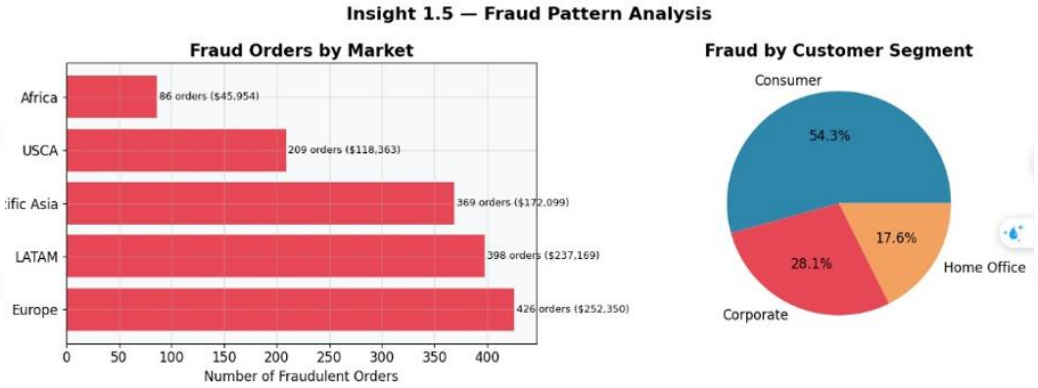
الاحتيال حسب الشريحة



$825K

إجمالي خسارة الاحتيال المشتبه به

أوروبا وأمريكا اللاتينية هما الأكثر تضرراً



الحل : تفعيل نظام Fraud Scoring تلقائي

لخفض الخسائر بنسبة 40%

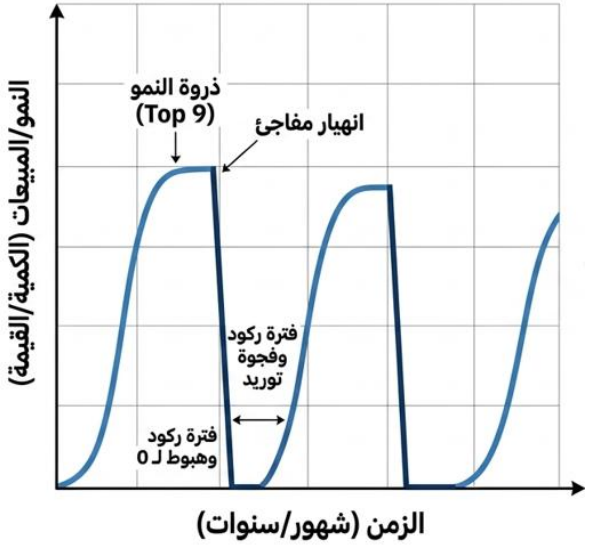
ملخص التوصيات الاستراتيجية

- → **تحسين الأسواق:** تخصيص 15% من الميزانية لأقوى 3 أسواق (USA, PR, LATAM).
- → **برامج الولاء:** استهداف عملاء Corporate ببرامج مخصصة لرفع المبيعات بنسبة 20%.
- → **إدارة المخزون:** زيادة المخزون قبل شهر يناير لتفادي الـ Stockouts.
- → **الأمان المالي:** تطبيق أنظمة تقييم الاحتيال التلقائية في الأسواق عالية الخطورة.
- → **تنشيط العملاء:** حملات إعادة استهداف بريدية للعملاء المنقطعين منذ 3 أشهر.

04

PRODUCTS AND CATEGORIES

أزمة "الترينـد" وسقوط الـ Top 9

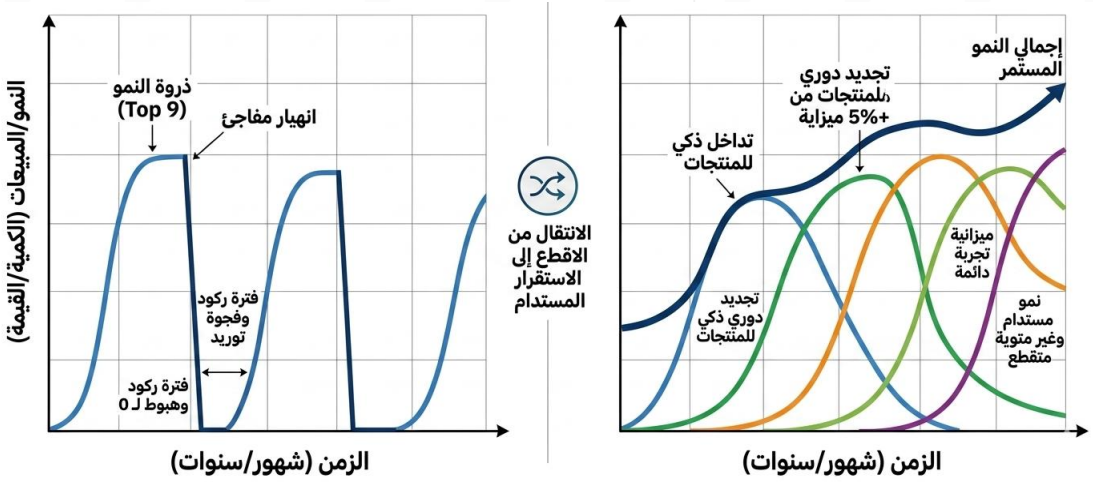


مؤشر الخطر (2017 - 2018)

مبيعات أفضل 9 منتجات هبطت للصفر في أكتوبر 2017 بعد هبوط مستمر منذ أغسطس. استمر عدم الاستقرار لمدة 5 أشهر، وهو ما يمثل تهديد وجودي للشركة اذا استمر الوضع لفترات أطول.

التوصية :نموذج "الهايبرد"

بدلاً من انتظار هبوط الكيرف، نقوم ببناء "كيرف "جديد موازي عبر تجربة المنتجات باستمرار.



5%

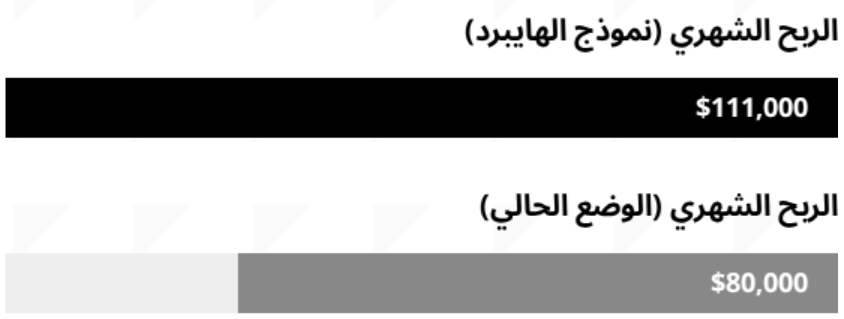
تخصيص ميزانية
للتجارب الجديدة

السيناريو التحفظي (تجنب الانهيار)

حماية قاع الأرباح

يهدف هذا السيناريو إلى الحفاظ على استقرار المتجر ومنع الانهيار المالي عند سقوط المنتجات القديمة من التريند.

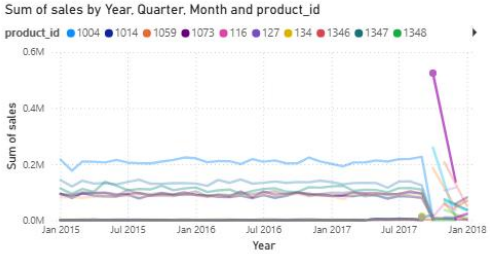
النتيجة :ضمان استمرار الأرباح عند مستوى آمن يمنع الخسائر التشغيلية.



28%

رفع متوقع للربحية
وقت الركود

السيناريو المتفائل (اقتناص القمم)



اكتشاف "المنتجات الخارقة" مثل منتج Dell Laptop

اكتشاف المنتج قبل وصوله للذروة بـ 5 أشهر يغير المعادلة المالية بالكامل.

$54.9K

صافي ربح شهري للمنتج الواحد

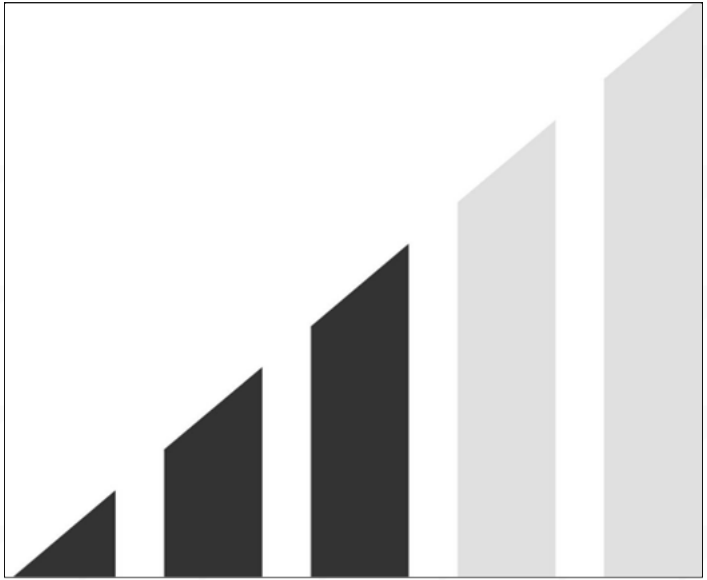
$274.5K

إجمالي أرباح مضافة في 5 شهور

10.98%

هامش الربح المستهدف للمنتج

القوة المالية :سد فجوة الركود



معالجة العجز المالي

الأرباح الناتجة من منتج "خارق "واحد كفيلة بتغطية فجوة أرباح الركود بالكامل ($156,451).

$118,049 فائض ربح فوق المعتاد بعد سد الفجوة.

المقارنة بين فترة الاستقرار والركود ونظام الهايبرد

```python
df['order_date'] = pd.to_datetime(df['order_date'])

stability_start = '2015-01-01'
stability_end = '2017-09-01'
recession_end = '2018-01-30'

stability_df = df[(df['order_date'] >= stability_start) & (df['order_date'] < stability_end)]
recession_df = df[(df['order_date'] >= stability_end) & (df['order_date'] < recession_end)]

monthly_sales_stb = stability_df['sales'].sum() / 32
monthly_sales_rec = recession_df['sales'].sum() / 5
monthly_cost_stb = stability_df['product_price'].sum() / 32
monthly_cost_rec = recession_df['product_price'].sum() / 5
monthly_profit_stb = stability_df['profit_per_order'].sum() / 32
monthly_profit_rec = recession_df['profit_per_order'].sum() / 5

sales_drop_pct = ((monthly_sales_stb - monthly_sales_rec) / monthly_sales_stb) * 100
cost_drop_pct = ((monthly_cost_stb - monthly_cost_rec) / monthly_cost_stb) * 100
profit_drop_pct = ((monthly_profit_stb - monthly_profit_rec) / monthly_profit_stb) * 100

margin_stb = monthly_profit_stb / monthly_sales_stb
margin_rec = monthly_profit_rec / monthly_sales_rec
target_sales = monthly_sales_stb

profit_from_stb_part = (target_sales * 0.95) * margin_stb
profit_from_rec_part = (target_sales * 0.05) * margin_rec
hybrid_monthly_profit = profit_from_stb_part + profit_from_rec_part
```

```
--- فترة الاستقرار (شهرياً) ---
المبيعات: 1,034,515.33$
التكلفة الحقيقية: 692,042.60$

--- فترة الركود (شهرياً) ---
المبيعات: 730,189.13$
التكلفة الحقيقية: 664,099.33$

---معدلات الهبوط ---
نسبة الهبوط في الأداء المالي: 29.42%
نسبة الهبوط في التكلفة: 4.04%
نسبة الهبوط في الربح: 28.00%
------------------------------
نسبة ربح الاستقرار الفعلية: 10.76%
نسبة ربح الركود الفعلية: 10.98%
------------------------------
الربح الشهري الصافي فترة الاستقرار: 111,360.08$
الربح الشهري الصافي فترة الركود: 80,181.76$
الربح الشهري المتوقع (نموذج 95/5): 111,472.06$
```

CONCLUSION

الخلاصة والتوصيات الاستراتيجية

سلاسل الإمداد والشحن

معالجة فشل الشحن السريع (First Class) عبر "اليوم الوقائي" تحمي **10.6مليون دولار** من الإيرادات المعرضة للخطر وتخفض مخاطر التأخير بنسبة 50%

الأداء المالي والربحية

إيقاف نزيف الفئات الخاسرة ($1.3M) وعقلنة الخصومات عند "النقطة الذهبية (10-13%) "يوفر للشركة **1.6مليون دولار** سنوياً تضاف لصافي الربح.

سلوك الأسواق والعملاء

تحويل 30% من عملاء الكاش إلى الدفع الإلكتروني يرفع قيمة الطلب (AOV) بنسبة 40%، مع ضرورة غلق ثغرة التحويل البنكي (Transfer) لوقف الاحتيال.

استراتيجية النمو (Hybrid)

تخصيص 5% لمشاريع الابتكار يضمن استدامة منحنى النمو (S-Curve) وسد فجوة أرباح الركود عبر اكتشاف "المنتجات الخارقة "قبل المنافسين.

THANKS!

Inspirational words