Smartphone Retail Sales Insights Report

Prepared for: Retail Product Manager and Inventory/Supply Chain Manager

Purpose: The purpose of this project is to analyse smartphone retail sales data to uncover trends in product performance, customer behaviour, and seasonal patterns to guide smarter inventory and marketing decisions.

Prepared By: Chinmaya Sri Rama Seshu Pasupuleti

Executive Summary

This report reviews over five years of transaction data from the smartphone retail outlet, aiming to uncover trends in customer behaviour, product performance, and payment preferences. Through an analysis of total revenue, product categories, seasonal shifts, and profit margins, it provides clear guidance for enhancing inventory decisions, sales campaigns, and pricing strategies.

Key Findings & Strategic Takeaways

1. Mobiles Dominate Sales & Revenue

- **Total revenue recorded:** $123.64 million
- **Mobile phones alone contributed:** $115.17 million
- **Total products sold:** 6,795 units
- **Most sold product type:** Mobile (5,308 units), followed by Accessories (1,307) and Tablets (180)

Insight: Mobiles are the primary revenue drivers.
Action: Focus stock and promotional efforts on flagship and midrange phones. Bundle them with accessories to raise overall cart value.

---

2. Seasonal Spikes in Q2

- **Quarterly revenue breakdown:**
  - Q1: $31.68M
  - **Q2: $48.06M** (highest)
  - Q3: $26.47M
  - Q4: $17.43M

Insight: The second quarter (April to June) brings peak sales, likely due to summer promotions, salary cycles, or gifting seasons.
Action: Increase stock levels, advertise early, and launch limited-time offers ahead of Q2.

---

3. Cash Still Leads, But Card Use Rising

While cash remains the dominant payment method, a noticeable shift toward card and mobile payments is underway — especially in higher-value transactions.

Insight: Card and digital payments are linked with larger purchases.
Action: Promote card usage through cashback deals or loyalty programs. Consider small discounts for digital payers.

---

4. Accessories Are Fast Sellers

Although they contribute less to total revenue, accessories such as chargers, wireless earphones, and covers are high in demand and sell quickly.

Insight: Accessories boost volume and can lead to easy upsells.
Action: Ensure accessories are always stocked. Bundle them with phones or place them near checkout counters.

---

5. Some Products Offer High Profit Margins

Items like data cables, batteries, and wireless headsets deliver strong profit margins — some exceeding 50% based on price vs. estimated cost.

Insight: High-margin products offer outsized returns despite lower price tags.
Action: Prioritize and promote these products both in-store and online. Train staff to recommend them actively during purchases.

---

6. A Handful of SKUs Drive the Bulk of Sales

Analysis shows that just five product codes generate a disproportionate share of revenue.

Insight: These top products are customer favourites and deserve special focus.
Action: Ensure these SKUs are always in stock and feature prominently in marketing, website banners, and store displays.

Conclusion

The analysis clearly shows that smartphones continue to dominate both in sales volume and revenue, with the second quarter emerging as the peak sales period. Accessories, though lower in price, play a critical role as fast-moving, high-margin products that contribute meaningfully to overall profitability. While cash remains the most used payment method, digital transactions are steadily increasing, particularly in higher-value purchases.

These insights reinforce the importance of strategic planning especially around product assortment, payment options, and seasonal demand. By aligning our inventory, sales tactics, and marketing campaigns with these patterns, we can increase profit margins, enhance the customer experience, and stay ahead of demand surges.

Maintaining focus on core smartphone offerings, bundling them with high-margin accessories, and preparing in advance for peak periods will be key to sustaining strong performance in the months ahead.

Technical Appendix

Data Set –

I explored a detailed dataset from a fictional smartphone retail outlet that provides a comprehensive view of sales across an entire fiscal year. It includes product codes, product types (mobiles and accessories), quantities, prices, total amounts, and payment methods. [Dataset link.]

Column Name	**Description**
Date	The date of the transaction (format: dd-mm-yyyy)
F.Y	Fiscal year during which the transaction occurred
QUARTER	The quarter of the fiscal year (e.g., 1, 2, 3, or 4)
P_NO	Product number or product code
PAYMENT TYPE	Method used for payment (e.g., CASH, CARD, ONLINE)
TYPE OF PRODUCT	General category of the product (e.g., ACCESSORY, MOBILE)
Quantity	Number of units purchased
Price	Price per unit of the product
Amount	Total amount of the transaction (Quantity × Price)
TYPE OF ACCESSORY/MOBILE	Specific details of the product (e.g., COVER, WIRELESS HEADSET, SMARTPHONE)

Data Cleaning Process

Before normalization, the raw dataset was carefully cleaned to ensure high data quality. The following steps were performed programmatically using Python (pandas), replicating and expanding on the manual Excel cleaning you initially did.

1. **Date Standardization**
   The Date column had inconsistent formats (some with hyphens, some with slashes). We replaced hyphens (-) with slashes (/) and converted the column to a consistent date format using pd.to_datetime(). This ensured all date values were recognized properly as datetime objects.

2. **Blank Values**
   We checked for missing values using df.isnull().sum(). The only blank cell was found in the Product No (P_NO) column. We filled this blank with 'UNKNOWN' as a placeholder, or you can optionally fill it using domain knowledge if available.

3. **Numeric Cleanup**
   Columns Price, Quantity, and Amount were forced to numeric types using pd.to_numeric(errors='coerce') to handle any accidental text entries or corrupt data.

4. **Text Column Standardization**
   We stripped extra spaces and converted values in PAYMENT TYPE, TYPE OF PRODUCT, and TYPE OF ACCESSORY/MOBILE to uppercase for consistent grouping and reporting.

5. **Final Check**
   A final missing values check confirmed no remaining nulls, ensuring the dataset was clean and ready for normalization.

Data Normalization Process

We started with the **Smartphone Retail Outlet Sales** dataset, which originally included the following columns:
Date, Financial Year (F.Y), Quarter, Product No (P_NO), Payment Type, Type of Product, Type of Accessory/Mobile, Quantity, Price, Amount

To achieve better structure and eliminate redundancy, we normalized the dataset into four tables, targeting **Third Normal Form (3NF)**.

Splitting the tables

Table	**Description**	**Primary Key (PK)**	**Foreign Keys (FKs)**
DimDate	Contains unique dates, financial years, and quarters.	date_id	—
DimProduct	Contains unique product numbers, product types, and accessory/mobile types.	product_id	—
DimPayment	Contains unique payment types (Cash, Credit, Debit).	payment_id	—

Table	Description	Primary Key (PK)	Foreign Keys (FKs)
FactSales	Stores all transactions: sales ID, quantity, price, amount, and foreign keys.	sales_id	date_id, product_id, payment_id

Ensuring 3NF (Third Normal Form)

- **1NF**: All tables have atomic columns with no repeating groups.
- **2NF**: All non-key columns depend fully on the primary key (for example, product details depend only on product_id).
- **3NF**: No transitive dependencies; for example, payment_id determines only payment type, and product details are separated from sales facts.

This design prevents:

- Redundant storage of product and payment details
- Update anomalies
- Inconsistent reporting across time and product lines

SQL Queries Explanation

1. **Total sales amount by product type**

   ```
   SELECT dp."TYPE OF PRODUCT", SUM(fs."Amount") AS total_sales
   FROM FactSales fs
   JOIN DimProduct dp ON fs.product_id = dp.product_id
   GROUP BY dp."TYPE OF PRODUCT"
   ORDER BY total_sales DESC;
   ```

   Groups sales by product type and calculates total revenue.

2. **Payment type with highest revenue**

   ```
   SELECT dpay."PAYMENT TYPE", SUM(fs."Amount") AS total_sales
   FROM FactSales fs
   JOIN DimPayment dpay ON fs.payment_id = dpay.payment_id
   GROUP BY dpay."PAYMENT TYPE"
   ORDER BY total_sales DESC;
   ```

   Shows revenue contribution of each payment method.

3. **Monthly sales trends across financial years**

   ```
   SELECT strftime('%Y-%m', dd.Date) AS month, dd."F.Y", SUM(fs."Amount") AS
   total_sales
   ```

```
FROM FactSales fs
JOIN DimDate dd ON fs.date_id = dd.date_id
GROUP BY month, dd."F.Y"
ORDER BY month;
```

Aggregates sales amounts by month and year for trend analysis.

4. **Accessory/mobile type with highest sales quantity**

```
SELECT dp."TYPE OF ACCESSORY/MOBILE", SUM(fs.Quantity) AS total_quantity
FROM FactSales fs
JOIN DimProduct dp ON fs.product_id = dp.product_id
GROUP BY dp."TYPE OF ACCESSORY/MOBILE"
ORDER BY total_quantity DESC;
```

Identifies the most sold accessory or mobile type by quantity.

5. **Top 5 product codes by revenue**

```
SELECT dp."P_NO", SUM(fs."Amount") AS total_sales
FROM FactSales fs
JOIN DimProduct dp ON fs.product_id = dp.product_id
GROUP BY dp."P_NO"
ORDER BY total_sales DESC
LIMIT 5;
```

Ranks the top five products by total sales revenue.

Pandas Queries Explanation

1. **Total sales amount by product type**

```
fact_sales.merge(dim_product, on='product_id') \
    .groupby('TYPE OF PRODUCT')['Amount'].sum() \
    .sort_values(ascending=False)
```

Merges fact_sales with dim_product, groups by product type, and sums amounts.

2. **Payment type with highest revenue**

```
fact_sales.merge(dim_payment, on='payment_id') \
    .groupby('PAYMENT TYPE')['Amount'].sum() \
    .sort_values(ascending=False)
```

Merges fact_sales with dim_payment, groups by payment type, and sums amounts.

3. **Monthly sales trends across financial years**

```
fact_date = fact_sales.merge(dim_date, on='date_id')
fact_date['month'] = fact_date['Date'].dt.to_period('M').astype(str)
fact_date.groupby(['month', 'F.Y'])['Amount'].sum()
```

Merges fact_sales with dim_date, extracts the month, groups by month and financial year, and sums amounts.

4. **Accessory/mobile type with highest sales quantity**

```
fact_sales.merge(dim_product, on='product_id') \
  .groupby('TYPE OF ACCESSORY/MOBILE')['Quantity'].sum() \
  .sort_values(ascending=False)
```

Merges fact_sales with dim_product, groups by accessory/mobile type, and sums quantities.

5. **Top 5 product codes by revenue**

```
fact_sales.merge(dim_product, on='product_id') \
  .groupby('P_NO')['Amount'].sum() \
  .sort_values(ascending=False).head(5)
```

Merges fact_sales with dim_product, groups by product number, sums amounts, and selects the top five.

Summary

This technical appendix explains:

- How the dataset was normalized into dimension and fact tables
- How we ensured the design fits Third Normal Form (3NF)
- What each SQL and pandas query does to deliver meaningful business insights

With this foundation, the stakeholder report and public-facing story are built on clean data, a robust schema, and reproducible analysis.