**Tech trend product analysis COMPREHENSIVE PROJECT PLAN**

To conduct a thorough analysis of the relationship between product quality and return rates for TechTrend Pro, we will follow these steps:

**Initial Setup**

1. Load the Data: Import the provided datasets (`product\_data.csv`, `sales\_data.csv`, and `feedback\_data.csv`) into myMYSQL for preliminary examination.

2. Assess Data Quality: Perform an initial assessment of the datasets to identify missing values, data inconsistencies, or anomalies.

**Solution**

**Here are the some observations gotten from doing the initial setup**

*1).* ***Product Data****- 200 rows*

*Columns: Product ID, Product Category, Product Attributes*

*First Few Rows:*

*Contains information about products, their categories, and specific attributes.*

***Sales Data****- 2000 rows*

*Columns: Sale ID, Product ID, Sales Date, Sales Volume, Revenue Generated*

*First Few Rows:*

*Details the sales records, including the volume and revenue generated.*

***Feedback Data****-1500 rows*

*Columns: Feedback ID, Product ID, Return Date, Return Reason, Customer Feedback*

*First Few Rows:*

*Provides data on customer feedback and reasons for product returns.*

2). ***Data Quality Assessment Results***

***Missing Values***

*Product Data: No missing values.*

*Sales Data: No missing values.*

*Feedback Data: No missing values.*

***Data Types***

***Product Data:***

*Product ID is an integer.*

*Product Category and Product Attributes are objects (strings).*

***Sales Data:***

*Sale ID and Product ID are integers.*

*Sales Date is an object (string) and might need conversion to a date format.*

*Sales Volume is an integer.*

*Revenue Generated is a float.*

***Feedback Data:***

*Feedback ID and Product ID are integers.*

*Return Date is an object (string) and might need conversion to a date format.*

*Return Reason and Customer Feedback are objects (strings).*

***Recommendations for Preprocessing***

*Convert Date Columns: In both sales\_data and feedback\_data, the date columns (Sales Date and Return Date) should be converted to datetime format for any time-based analysis.*

*Assessing Data Consistency: We should briefly check for consistency in categorical fields like Product Category and Return Reason.*

**Data Preprocessing and Cleaning (Using MYSQL)**

1. Data Cleaning: Using MYSQL queries, we will clean the data. This includes:

- Handling missing values

- Correcting data types if necessary

- Removing duplicates

2. Data Transformation: Transform the data using MYSQL for easier analysis. This might include:

- Merging tables based on common columns

- Creating new columns if needed (e.g., categorizing products)

- Aggregating data for summary statistics

**Solution**

*The data preprocessing and cleaning using SQL have been successfully completed. Here are the changes made:*

***Conversion to Datetime Format:***

*The Sales Date in the sales data and Return Date in the feedback data have been converted to a standard YYYY-MM-DD format.*

***Detection of Duplicates:***

*Duplicate entries do not exist in the product, sales, and feedback data based on unique identifiers (Product ID, Sale ID, Feedback ID).*

*The datasets are now ready for further analysis. The transformed datasets look as follows:*

*Product Data: Includes Product ID, Product Category, and Product Attributes.*

*Sales Data: Includes Sale ID, Product ID, Sales Date, Sales Volume, and Revenue Generated.*

*Feedback Data: Includes Feedback ID, Product ID, Return Date, Return Reason, and Customer Feedback.*

**Connecting MYSQL with Jupyter Notebook**

We are to establish a connection between Python mysql to proceed with forward analysis with our data.

**Solution**

*Connection has been established, now we can proceed with further analysis.*

**Data Analysis (Using Python Libraries)**

1. Descriptive Analysis:

- Use pandas for data manipulation.

- Generate descriptive statistics to understand the data distribution.

2. Exploratory Data Analysis (EDA):

- Use matplotlib and seaborn for visualizations.

- Investigate trends, patterns, and relationships in the data.

- Specifically focus on return rates in relation to product categories, product quality, and other relevant variables.

3. Deep Dive into Product Quality and Return Rates:

- Analyze the relationship between product quality (as indicated in feedback data) and return rates.

- Use statistical methods to ascertain if there’s a significant correlation.

**Solution:**

#### ***Descriptive Analysis Results***

#### ***Product Data***

*Count: 200 entries.*

*Categories: 3 unique product categories.*

*Attributes: 17 unique product attributes.*

***Sales Data***

*Count: 2000 entries.*

*Date Range: Sales data covers a range of dates.*

*Sales Volume: Varies widely, with an average of around 52 units per sale.*

*Revenue: Average revenue generated per sale is approximately $519.68.*

***Feedback Data***

*Count: 1500 entries.*

*Return Reasons: Various return reasons provided.*

***Exploratory Data Analysis (EDA) Visualizations***

***Univariate Analysis:***

*For the Return Reason, wrong item received is the highest and the least is poor quality*

*While for Product Categories, majority are accessories the least are the Laptops, this helps us to understand the distribution of returns across different product categories*

*For Sales Volume and Revenue Generated proper distribution, with some skewness.*

***Bivariate Analysis- Return Reasons by Product Category:***

*This chart provides insight into the common reasons for returns across different product categories.*

***Deep Dive Analysis: Product Quality and Return Rates***

***Quality Issue Analysis***

***Frequency of Returns Due to Quality Issues:***

*There are 2,717 instances where the customer feedback suggests a quality issue.*

*12,404 instances do not indicate a quality issue in customer feedback.*

***Visualization: Quality Issues by Product Categor****y*

*The chart shows the count of quality-related issues across different product categories.*

***Statistical Analysis: Chi-Square Test of Independence***

*Chi-Square Statistic: 6.559*

*p-value: 0.038*

*Degrees of Freedom: 2*

*Expected Frequencies:*

*Accessories: Expected frequency of returns with and without quality issues - 4543.73 and 995.27, respectively.*

*Smartphones: Expected frequency - 3620.06 and 792.94.*

*Laptops: Expected frequency - 4240.21 and 928.79.*

***Interpretation***

*Statistical Significance: The p-value of 0.038 suggests that there is a statistically significant relationship between product category and the presence of quality issues in the customer feedback.*

***Implications:*** *This implies that the quality issues leading to returns are not uniformly distributed across product categories.*

**Recommendations and Insights**

1. Derive Insights: Based on the analysis, identify key drivers of high return rates.

2. Develop Recommendations: Propose data-driven recommendations for product development and sourcing strategies.

### **Insights**

##### ***Quality-Related Returns Vary Across Categories:***

* A significant proportion of returns are linked to quality issues, especially in certain product categories.
* The chi-square test indicated a statistically significant relationship between product categories and quality issues.

##### ***Specific Categories with Higher Quality Issues:***

* Among the product categories, some show a higher prevalence of quality-related feedback. These categories warrant closer inspection and targeted improvements.

#### **Customer Feedback as a Quality Indicator:**

* Customer feedback frequently mentions quality issues like 'defective products' or 'poor quality', highlighting the importance of addressing these concerns.

### **Recommendations**

##### ***Enhanced Quality Control for Targeted Categories:***

* TechTrend Pro should implement stricter quality control measures, especially for the product categories with a higher incidence of quality-related returns. This could involve more rigorous testing and inspection processes.

##### ***Review and Strengthen Supplier Relationships:***

* Analyze and review supplier performance. For suppliers consistently linked to quality issues, consider renegotiation of terms, additional quality checks, or seeking alternative suppliers.

##### ***Leverage Customer Feedback for Product Improvements:***

* Utilize the insights from customer feedback to inform product development. Identify common complaints and address these in future product designs or updates.

##### ***Enhance Customer Education and Support:***

* Some returns may be due to misunderstandings about product use. Providing better educational materials and proactive customer support could reduce such returns.

##### ***Regular Data Analysis for Continuous Improvement:***

* Continue to analyze sales, feedback, and return data regularly to identify emerging trends or new issues, allowing for timely corrective actions.

##### ***Invest in Predictive Analytics:***

* Implement predictive analytics to identify potential quality issues before products are shipped. This could help in proactively addressing problems and reducing future returns.

### **Conclusion**

By focusing on these areas, TechTrend Pro can potentially reduce return rates, leading to increased customer satisfaction and reduced operational costs associated with handling returns. The key is a combination of proactive quality management, continuous data monitoring, and leveraging customer feedback for ongoing improvements.

**SQL Queries**

-- Preliminaries- creation of Schema/Database

CREATE SCHEMA techtrendpro;

SHOW DATABASES;

USE techtrendpro;

-- UNDERSTANDING DATASETS

-- Load the data into the tables using the table import data wizard.

SHOW TABLES;

-- Table inspection

SELECT COUNT(\*) AS Feedback\_Rows FROM feedback\_data;

SELECT COUNT(\*) AS Product\_Rows FROM product\_data;

SELECT COUNT(\*) AS Sales\_Rows FROM sales\_data;

SELECT \* FROM feedback\_data LIMIT 5;

SELECT \* FROM product\_data LIMIT 5;

SELECT \* FROM sales\_data LIMIT 5;

-- Understanding the structure of the datasets, show datatypes

SHOW COLUMNS FROM feedback\_data;

DESCRIBE product\_data;

DESC sales\_data;

-- Checking for missing values

-- Feedback\_table

SELECT \* FROM feedback\_data

WHERE Feedback\_ID IS NULL

OR Product\_ID IS NULL

OR Return\_Date IS NULL

OR Return\_Reason IS NULL

OR Customer\_Feedback IS NULL;

SELECT

COUNT(\*) AS total\_rows,

SUM(Feedback\_ID IS NULL) AS null\_Feedback\_ID,

SUM(Product\_ID IS NULL) AS null\_Product\_ID,

SUM(Return\_Date IS NULL) AS null\_Return\_Date,

SUM(Return\_Reason IS NULL) AS null\_Return\_Reason,

SUM(Customer\_Feedback IS NULL) AS null\_Customer\_Feedback

FROM feedback\_data;

-- product\_data

SELECT \* FROM product\_data

WHERE Product\_ID IS NULL

OR Product\_Category IS NULL

OR Product\_Attributes IS NULL;

SELECT

COUNT(\*) AS total\_rows,

SUM(Product\_ID IS NULL) AS null\_Product\_ID,

SUM(Product\_Category IS NULL) AS null\_Product\_Category,

SUM(Product\_Attributes IS NULL) AS null\_Product\_Attributes

FROM product\_data;

-- sales\_data

SELECT \* FROM sales\_data

WHERE Sale\_ID IS NULL

OR Product\_ID IS NULL

OR Sales\_Date IS NULL

OR Sales\_Volume IS NULL

OR Revenue\_Generated IS NULL;

SELECT

COUNT(\*) AS total\_rows,

SUM(Sale\_ID IS NULL) AS null\_Sale\_ID,

SUM(Product\_ID IS NULL) AS null\_Product\_ID,

SUM(Sales\_Date IS NULL) AS null\_Sales\_Date,

SUM(Sales\_Volume IS NULL) AS null\_Sales\_Volume,

SUM(Revenue\_Generated IS NULL) AS null\_Revenue\_Generated

FROM sales\_data;

-- DATA CLEANING AND PREPROCESSING

# 1. Convert 'Sales Date' and 'Return Date' to datetime format

-- sales\_data's sales\_date column

SET SQL\_SAFE\_UPDATES = 0; -- turning off safe updates

ALTER TABLE sales\_data MODIFY COLUMN Sales\_Date DATE;

-- feedback\_data's Return\_date column

ALTER TABLE feedback\_data MODIFY COLUMN Return\_Date DATE;

# 2. Check for and remove duplicates (if any)

-- Feedback

SELECT Feedback\_ID, COUNT(\*) AS count\_duplicates

FROM feedback\_data

GROUP BY Feedback\_ID

HAVING COUNT(\*) > 1;

-- Product

SELECT Product\_ID, COUNT(\*) AS count\_duplicates

FROM product\_data

GROUP BY Product\_ID

HAVING COUNT(\*) > 1;

-- Sales

SELECT Sale\_ID, COUNT(\*) AS count\_duplicates

FROM sales\_data

GROUP BY Sale\_ID

HAVING COUNT(\*) > 1;

# Display the first few rows of each dataset after transformation

SELECT \* FROM feedback\_data LIMIT 5;

SELECT \* FROM product\_data LIMIT 5;

SELECT \* FROM sales\_data LIMIT 5;