DATA CLEANING AND ANALYSIS WITH PYTHON AND MYSQL

By: Surbhi Yadav

Introduction

Project Overview:

This project focused on leveraging Python and MySQL to clean, transform, and analyze data effectively.

Key Objectives:

- Prepare and transform raw data in Python.
- Connect the cleaned data to a MySQL database.
- Perform queries to extract valuable insights.

Tools and Technologies

Python Libraries Used:

- Pandas: For data manipulation and cleaning.
- NumPy: For numerical operations.
- SQLAlchemy or MySQL Connector: For database connection.
-

MySQL:

Used for structured data storage and querying.

Additional Tools:

- Jupyter Notebook for coding and visualization.
- MySQL Workbench for database management.

Data Cleaning and Transformation

Example Transformation:

	invoice_id	Branch	City	category	unit_price	quantity	date	time	payment_method	rating	profit_margin
0	1	WALM003	San Antonio	Health and beauty	$74.69	7.0	05/01/19	13:08:00	Ewallet	9.1	0.48
1	2	WALM048	Harlingen	Electronic accessories	$15.28	5.0	08/03/19	10:29:00	Cash	9.6	0.48
2	3	WALM067	Haltom City	Home and lifestyle	$46.33	7.0	03/03/19	13:23:00	Credit card	7.4	0.33
3	4	WALM064	Bedford	Health and beauty	$58.22	8.0	27/01/19	20:33:00	Ewallet	8.4	0.33
4	5	WALM013	Irving	Sports and travel	$86.31	7.0	08/02/19	10:37:00	Ewallet	5.3	0.48

	invoice_id	Branch	City	category	unit_price	quantity	date	time	payment_method	rating	profit_margin	total
0	1	WALM003	San Antonio	Health and beauty	74.69	7.0	05/01/19	13:08:00	Ewallet	9.1	0.48	522.83
1	2	WALM048	Harlingen	Electronic accessories	15.28	5.0	08/03/19	10:29:00	Cash	9.6	0.48	76.40
2	3	WALM067	Haltom City	Home and lifestyle	46.33	7.0	03/03/19	13:23:00	Credit card	7.4	0.33	324.31
3	4	WALM064	Bedford	Health and beauty	58.22	8.0	27/01/19	20:33:00	Ewallet	8.4	0.33	465.76
4	5	WALM013	Irving	Sports and travel	86.31	7.0	08/02/19	10:37:00	Ewallet	5.3	0.48	604.17

Process Highlights:

- Identified and handled missing values
- Standardized data types (e.g., converting strings to dates).
- Removed duplicate entries.

Query and Insights

```sql
-- Business Problem Q1: Find different payment methods, number of transactions, and quantity sold by payment method

SELECT payment_method , count(*) as no_payment , SUM(quantity) as quantity_sold
FROM walmart
GROUP BY payment_method;

-- Project Question #2: Identify the highest-rated category in each branch
-- Display the branch, category, and avg rating

SELECT branch , category , avg_rating
FROM(
  SELECT branch, category , AVG(rating) as avg_rating ,
  RANK() over (PARTITION BY branch ORDER BY AVG(rating) DESC) AS ranks
  FROM walmart
  GROUP BY branch,category
 ) as ranked
 where ranks=1;
```

Result Grid | Filter Rows:

payment_method	no_payment	quantity_sold
Ewallet	3881	8932
Cash	1832	4984
Credit card	4256	9567

branch	category	avg_rating
WALM001	Electronic accessories	7.45
WALM002	Food and beverages	8.25
WALM003	Sports and travel	7.5
WALM004	Food and beverages	9.3
WALM005	Health and beauty	8.366666666666667
WALM006	Fashion accessories	6.797058823529412
WALM007	Food and beverages	7.55
WALM008	Food and beverages	7.4
WALM009	Sports and travel	9.6
WALM010	Electronic accessories	9
WALM011	Food and beverages	7
WALM012	Health and beauty	7.45
WALM013	Health and beauty	7.6

- Most transactions are done through Credit Card.

Query and Insights

```sql
-- Q3: Identify the busiest day for each branch based on the number of transactions
SELECT branch , day_name , no_transactions
FROM(
     SELECT
     branch,
     dayname(STR_TO_DATE(date, '%d/%m/%Y')) AS day_name,
     count(*) as no_transactions,
     RANK() OVER(PARTITION BY branch ORDER BY COUNT(*) DESC) AS ranks
     from walmart
     GROUP BY branch , day_name
     ) AS ranked
 WHERE ranks =1;
```

branch	day_name	no_transactions
WALM001	Thursday	16
WALM002	Thursday	15
WALM003	Tuesday	33
WALM004	Sunday	14
WALM005	Wednesday	19
WALM006	Thursday	15
WALM007	Friday	12
WALM007	Sunday	12
WALM008	Tuesday	17
WALM009	Sunday	42

Query and Insights

```sql
 -- Q4: Calculate the total quantity of items sold per payment method
SELECT payment_method, sum(quantity)
FROM walmart
GROUP BY payment_method;

-- Q5: Determine the average, minimum, and maximum rating of categories for each city

SELECT city , category ,
     MIN(rating), MAX(rating),AVG(rating)
FROM walmart
GROUP BY city,category;
```

payment_method	sum(quantity)
Ewallet	8932
Cash	4984
Credit card	9567

- Most quantity of items sold are done by Credit Card.

city	category	MIN(rating)	MAX(rating)	AVG(rating)
San Antonio	Health and beauty	5	9.1	7.05
Harlingen	Electronic accessories	9.6	9.6	9.6
Haltom City	Home and lifestyle	3	9.5	6.227777777777778
Bedford	Health and beauty	6.1	9.3	8.15
Irving	Sports and travel	5.3	5.3	5.3
Denton	Electronic accessories	4.1	9	6.7
Cleburne	Electronic accessories	5.8	7.8	7.25
Canyon	Home and lifestyle	3	9	6.25

Query and Insights

```sql
-- Q6: Calculate the total profit for each category

SELECT category , sum(total * profit_margin) as profit
FROM walmart
GROUP BY category
ORDER BY profit DESC;
```

category	profit
Fashion accessories	192314
Home and lifestyle	192213
Electronic accessories	30772
Food and beverages	21552
Sports and travel	20613
Health and beauty	18671

20613

- Fashion accessories and Home and lifestyle category has largest profit.

Query and Insights

```sql
-- Q7: Determine the most common payment method for each branch

SELECT branch , payment_method
FROM(
  SELECT branch, payment_method ,
  RANK() over (PARTITION BY branch ORDER BY COUNT(*) DESC) AS ranks
  FROM walmart
  GROUP BY branch,payment_method
 ) as ranked
 where ranks=1;
```

branch	payment_method
WALM001	Ewallet
WALM002	Ewallet
WALM003	Credit card
WALM004	WALM003
WALM005	Ewallet
WALM006	Ewallet
WALM007	Ewallet
WALM008	Ewallet

```sql
-- Q8: Categorize sales into Morning, Afternoon, and Evening shifts

SELECT
    branch,
    CASE
        WHEN HOUR(TIME(time)) <12 THEN 'Morning'
        WHEN HOUR(TIME(time)) BETWEEN 12 AND 17 THEN 'Afternoon'
        ELSE 'Evening'
    END as shifts,
    count(*) as num_invoices
FROM walmart
GROUP BY branch,shifts
ORDER BY branch , num_invoices desc;
```

branch	shifts	num_invoices
WALM001	Afternoon	36
WALM001	Evening	30
WALM001	Morning	8
WALM002	Afternoon	29
WALM002	Evening	21
WALM002	Morning	15
WALM003	Afternoon	95
WALM003	Morning	50

- Most sales are done in Afternoon in almost every branch.

Query and Insights

```sql
-- Q9: Identify the 5 branches with the highest revenue decrease ratio from last year to current year (e.g., 2022 to 2023)

WITH revenue2022 AS (
        SELECT branch ,
            SUM(total) as revenue
        FROM walmart
        WHERE YEAR(STR_TO_DATE(date, '%d/%m/%Y')) = 2022
        GROUP BY branch
),
revenue2023 AS (
        SELECT branch ,
            SUM(total) as revenue
        FROM walmart
        WHERE YEAR(STR_TO_DATE(date, '%d/%m/%Y')) = 2023
        GROUP BY branch
)

SELECT
    r2022.branch,
    r2022.revenue AS last_year_revenue,
    r2023.revenue AS current_year_revenue,
    ROUND(((r2022.revenue - r2023.revenue) / r2022.revenue) * 100, 2) AS revenue_decrease_ratio
FROM revenue2022 AS r2022
JOIN revenue2023 AS r2023 ON r2022.branch = r2023.branch
WHERE r2022.revenue > r2023.revenue
ORDER BY revenue_decrease_ratio DESC
LIMIT 5;
```

branch	last_year_revenue	current_year_revenue	revenue_decrease_ratio
WALM045	1731	647	62.62
WALM047	2581	1069	58.58
WALM098	2446	1030	57.89
WALM033	2099	931	55.65
WALM081	1723	850	50.67

Query and Insights

```sql
-- Q10: Average rating per payment method
-- checks average rating for each payment method
SELECT
    payment_method,
    AVG(rating) AS avg_rating
FROM walmart
GROUP BY payment_method
ORDER BY avg_rating DESC;
```

payment_method	avg_rating
Ewallet	6
Cash	5
Credit card	5

```sql
-- Q 11: Correlation Between Rating and Profit Margin
-- Check if higher ratings correlate with higher profit margins by grouping ratings into ra
SELECT
    CASE
        WHEN rating< 4 THEN 'Poor'
        WHEN rating BETWEEN 4 AND 6 THEN 'Bad'
        WHEN rating BETWEEN 7 AND 8 THEN 'Good'
        ELSE 'Excellent'
    END as ranking_category,
    ROUND(AVG(profit_margin),2) as avg_profit_margin
FROM walmart
GROUP BY ranking_category
ORDER BY avg_profit_margin DESC;
```

ranking_category	avg_profit_margin
Excellent	0.4
Good	0.4
Bad	0.39
Poor	0.38

- Higher ranking have higher profit margin.

Conclusion

Project Outcomes:

- Successfully cleaned and transformed raw data in Python.
- Efficiently stored and analyzed data using MySQL.
- Derived actionable insights through SQL queries.

Skills Learned:

- Python-MySQL integration.
- Advanced data cleaning techniques.
- Writing optimized SQL queries.

.