

Documentation

Vanilla Steel Interview Solutions

Name: Shreyas Sandbhor

E-mail: shreyassandbhor301@gmail.com

Github link: github.com/Shreyas301

1. Task 1:

Instruction on how to run Task 1 (Data Cleaning):

- It is an .ipynb file so open google colab.
- Import all three databases into the notebook page (supplier_data_1, supplier_data_2, supplier_data2)
- Go to runtime and click Run All.

Result of Task 1:

- **Inventory_dataset.csv** is created.

2. Task 2:

Link to dashboard: <https://lookerstudio.google.com/s/gmHqePNdXSQ>

Key Insights:

This Looker dashboard provides key insights into sales, revenue, cost breakdown, and transportation efficiency for Vanilla Steel.

- **Strong Performance Metrics:** High conversion rate (100%) and total revenue of \$53.2M indicate successful deal closures.
- **Buyer and Supplier Trends:** Germany dominates as a key buyer region, with certain companies driving most of the revenue.
- **Operational Efficiency:** Transportation costs and booked truck prices show a declining trend, suggesting cost optimizations over time.
- **Account Manager Performance:** Revenue distribution among managers and companies highlights areas for growth and improvement.

- Cost Breakdown: Material, logistics, and financing costs vary significantly across companies, influencing profit margins.

Overall, the data reflects a well-performing business with opportunities for refining cost structures and optimizing account management efficiency.

Code used to calculate Account Manager Share

```
CASE
```

```
WHEN buyer_am_id IS NOT NULL AND supplier_am_id IS NOT NULL THEN 1
```

```
WHEN buyer_am_id IS NOT NULL OR supplier_am_id IS NOT NULL THEN 0.5
```

```
ELSE 0
```

```
END
```

Code used to calculate Total Revenue from Buyer

```
SUM(booked_gross_revenue) + SUM(confirmed_gross_revenue)
```

3. Task 3:

Instruction on how to run Task 3 (Data Pipeline):

- It is an .ipynb file so open google colab.
- Import all three databases into the notebook page (supplier_data1, supplier_data2, supplier_data2, buyer_preferences)
- Go to runtime and click Run All.

Working of Data Pipeline:

1. Data Pipeline Steps: Data Ingestion (Extract):

This is the first step where data is brought into the pipeline. In the code, data is ingested by reading two different Excel files (supplier_data_1.xlsx and supplier_data_2.xlsx) for the supplier data and one Excel file (buyer_preferences.xlsx) for the buyer data. This is done using `pd.read_excel()`, which loads the databases.

2. Data Cleaning (Transform):

Data is often "dirty" (e.g., missing values, wrong data types, or irrelevant columns) and needs to be cleaned to ensure proper analysis or processing. In your code, several cleaning steps are carried out: **Selecting Relevant Columns:** After loading the datasets, unnecessary columns are dropped, and only the relevant columns are kept (Grade, Finish, Thickness (mm), Width (mm), Gross weight (kg), and Quantity). **Handling Missing Values:** Missing values are dropped using `dropna()`, which removes any rows with NaN values. **Converting Data Types:** The relevant columns are converted to numeric types to ensure consistency and facilitate proper merging. The conversion is done using `pd.to_numeric()`. This ensures that all data can be properly used in further steps, such as merging and filtering.

3. Data Integration (Merge):

In this step, data from different sources is combined. Here, you're merging the two supplier datasets (supplier1 and supplier2) and the buyer preferences dataset into a single data structure. Merging Data: The merge() function is used to combine the supplier data and buyer preferences based on matching columns (like Grade, Finish, Thickness, Width). This step aligns the data according to the criteria that are relevant for further processing.

4. Data Transformation (Apply Filters/Business Logic):

After the data is merged, additional transformations are applied. Applying Constraints: The business rules (buyer constraints on weight and quantity) are applied to the merged data: Strict Constraints: Initially, the code applies strict buyer constraints by filtering data that meets the buyer's maximum weight and minimum quantity requirements. Relaxed Constraints: If no strict matches are found, the code relaxes the constraints by increasing the weight tolerance and allowing for a slight decrease in the quantity. This transformation is done by using conditional filters on the columns.

5. Data Output (Load):

The final step in a data pipeline is often storing or loading the transformed data into an output location. In your code, this is done by saving the filtered data (the matched recommendations) into an Excel file (recommendation_table.xlsx). This file is ready to be used by stakeholders or further processed in subsequent pipelines.

Result obtained in Task 3:

As there were no matching records from buyer preferences, an empty recommendation_table.xlsx file will be generated.