
Hands-On Exercises

1. DataFrame Creation and Inspection

- Load the CSV using Pandas, PySpark, and Dask.
- Display the first 5 and last 5 records.
- Print schema and check data types.

---

2. Selection, Renaming, and Filtering

- Select only `OrderID`, `CustomerName`, and `Amount`.
- Rename `Amount` to `OrderAmount`.
- Filter orders where `Amount > 500`.
- Filter orders from a specific city using `.query()` or `.filter()`.

---

3. Data Manipulation

- Drop `CustomerSince` column.
- Add a new column `FinalAmount = Amount - (Amount * Discount)`.
- Sort by `FinalAmount` descending.
- Replace all “Cancelled” status with “Order Cancelled”.

---

4. Aggregations and GroupBy

- Count of orders by `DeliveryStatus`.
- Average `Amount` by `ProductCategory`.
- Group by `City` and show total sales.

---

5. Null Handling & Update

- Intentionally inject nulls in `City` column and handle them using `fillna()`, `dropna()`.
- Use `.when().otherwise()` in PySpark to tag high-value customers (`Amount > 800`).

---

6. Date & Time Functions

- Extract year and month from `OrderDate`.
- Calculate customer loyalty in years = `today - CustomerSince`.

---

7. Joins and Unions

- Create a second DataFrame with city-wise region mapping.
- Perform inner and left joins with the main dataset.
- Union two datasets: e.g., orders from 2023 and 2024.

---

8. Complex JSON Simulation (Advanced)

- Convert each order to a JSON string and load it back into a DataFrame.
- Access nested fields using `explode()` and `get_json_object()`.

---

9. Applying Functions

- Create a function to tag orders: “Big”, “Medium”, “Small” based on Amount.
- Apply it using `.apply()` in Pandas, and UDF in PySpark.
