

Solving the Revenue Deficit Problem in Cottonil Branch

Tools Used: Python, Pandas, NumPy, Natsort

Date: October 2025

Executive Summary

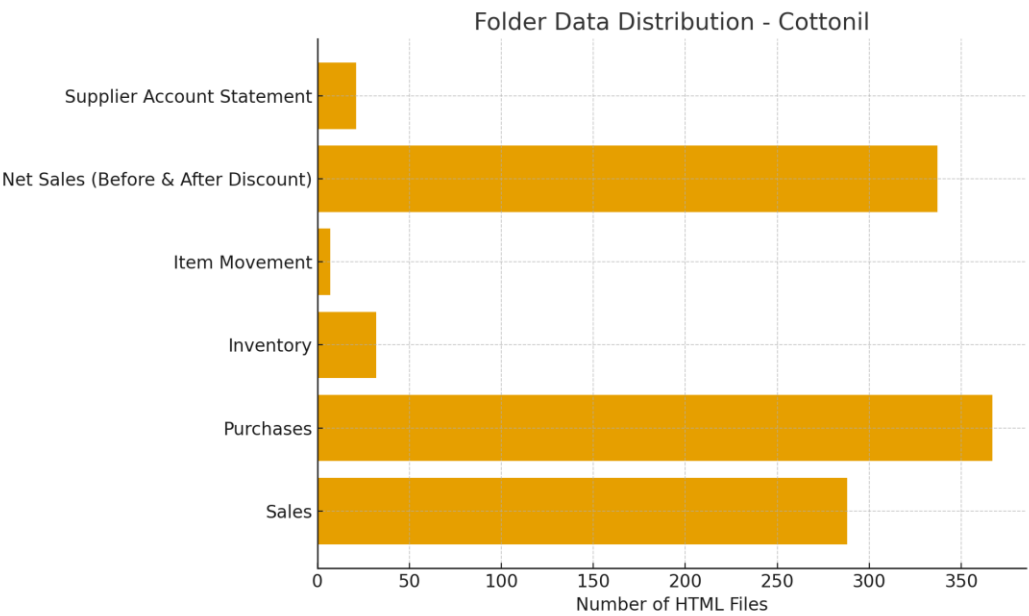
This report presents a detailed overview of the data processing workflow developed to address the revenue deficit problem in a Cottonil branch.

The project involved extracting, cleaning, and merging multiple HTML data files across several folders (Sales, Purchases, Inventory, Item Movement, Net Sales Before & After Discount, Supplier Accounts).

Using Python and data analysis libraries, the data was successfully transformed into a structured Excel format for further analysis and visualization.

Data Overview

Folder Name	File Count
Sales	288
Purchases	367
Inventory	32
Item Movement	7
Net Sales (Before & After Discount)	337
Supplier Account Statement	21



Tools and Libraries

- Python – Core scripting and data transformation
- Pandas – Data manipulation and HTML parsing
- NumPy – Numerical operations
- Natsort – Sorting file names naturally (e.g., file1, file2, ... file10)

Implementation Steps

1. Data Extraction – Read multiple HTML tables using Pandas.
2. Data Cleaning – Removed unwanted header and footer rows, standardized columns.
3. Data Merging – Concatenated tables and added consistent headers.
4. Exporting – Saved cleaned data into structured Excel files.

Python Code Sample

```
import os
import pandas as pd
import numpy as np
from natsort import natsorted

folder = r'C:\Users\HP ZBOOK G3\Downloads\project cotonel\Cottonil_Data-main\المبيعات'

# Sort files naturally
DB = os.listdir(folder)
sorted_files = natsorted(DB)
print(sorted_files)

data_headers = ["NAME", "القيمة", "الكمية", "السعر", "الصنف", "الكود", "الباركود", "رقم الاذن", "التاريخ"]
df_list = []

for num, file_name in enumerate(sorted_files, start=1):
    file_path = fr'{folder}\{file_name}'
    html_tables = pd.read_html(file_path)

    # Select appropriate table
    if num == 1:
        df = html_tables[3]
        df = df.iloc[1:-1]
    elif num == 222:
```

```
        df = html_tables[1]
    else:
        df = html_tables[1]

    df_list.append(df)

# Merge all data
combined_df = pd.concat(df_list, ignore_index=True)
combined_df.columns = data_headers

# Export to Excel
output_path = r'C:\Users\HP ZBOOK G3\Downloads\cotton\cotton.xlsx'
combined_df.to_excel(output_path, index=False)
print(f' File saved successfully at: {output_path}')
```

Results

- All HTML files were successfully read and merged.
- Data headers unified across all datasets.
- Output Excel file generated at: C:\Users\HP ZBOOK G3\Downloads\cotton\cotton.xlsx

Next Steps

- Build Power BI or Excel dashboards to analyze sales and purchases.
- Analyze supplier accounts and discount impact.
- Monthly comparison for branch revenue.

Conclusion

The data integration and cleaning pipeline successfully transformed hundreds of unstructured HTML reports into a consistent Excel dataset.

The resulting data will enable Cottonil to perform accurate revenue analysis and identify causes of revenue deficit at the branch level.