

TP03: Full Data Analysis Project using Python & Power BI

kiram mohammed ali

1 Introduction

This report presents the complete workflow of TP03, which includes data loading, cleaning, exploratory data analysis (EDA), machine learning modeling, and dashboard creation using Power BI. Each step is supported with screenshots and code overviews extracted from the Jupyter Notebook.

2 Data Loading

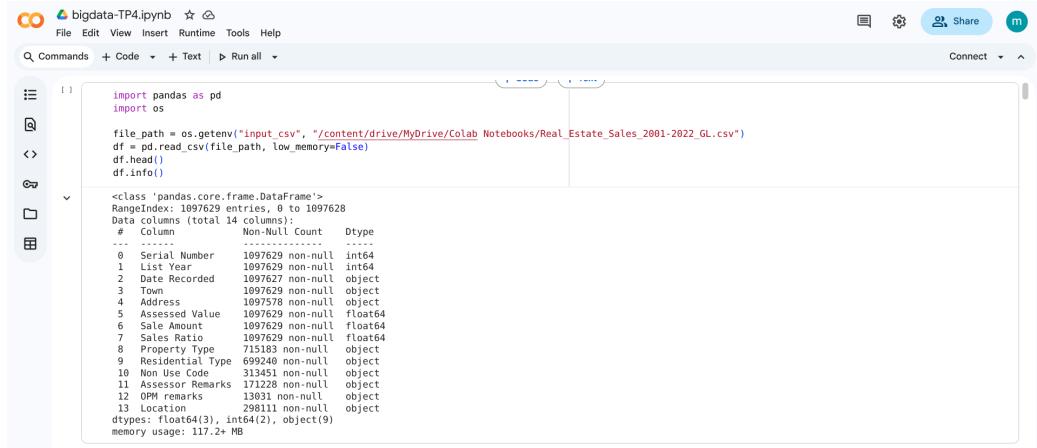
The dataset is loaded using Python as follows:

```
import pandas as pd  
df = pd.read_csv('your_data.csv')
```

2.1 Overview

The data is successfully imported and ready for cleaning.

2.2 Screenshot



The screenshot shows a Jupyter Notebook interface with the following code:

```
import pandas as pd
import os

file_path = os.getenv("input.csv", "/content/drive/MyDrive/Colab Notebooks/Real_Estate_Sales_2001-2022_GL.csv")
df = pd.read_csv(file_path, low_memory=False)
df.head()
df.info()
```

The output of the `df.info()` command is displayed, showing the following information about the DataFrame:

#	Column	Non-Null Count	Dtype
0	Serial Number	1097629	int64
1	List Year	1097629	non-null int64
2	Date Recorded	1097629	non-null object
3	Town	1097629	non-null object
4	Address	1097578	non-null object
5	Assessed Value	1097629	non-null float64
6	Sale Amount	1097629	non-null float64
7	Sale Ratio	1097629	non-null float64
8	Sale Date	715183	non-null object
9	Residential Type	699240	non-null object
10	Non Use Code	313451	non-null object
11	Assessor Remarks	171228	non-null object
12	OPM remarks	13031	non-null object
13	Location	298111	non-null object

dtypes: float64(3), int64(2), object(9)
memory usage: 117.2+ MB

Figure 1: Data Import Preview (Replace with your screenshot)

3 Data Cleaning

The following operations were performed:

- Removing missing values
- Removing duplicates
- Converting date fields
- Handling outliers

3.1 Code Overview

```
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)
df['Date'] = pd.to_datetime(df['Date'])
```

3.2 Screenshot

The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** bigdata-TP4.ipynb, File Edit View Insert Runtime Tools Help, Share, Connect.
- Sidebar:** Commands, Code, Text, Run all, etc.
- Output Cell:** Displays Python code and its execution results.

```
import pandas as pd
import os

file_path = os.getenv("input_csv", "/content/drive/MyDrive/Colab Notebooks/Real_Estate_Sales_2001-2022_GL.csv")
df = pd.read_csv(file_path, low_memory=False)
df.head()
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1097629 entries, 0 to 1097628
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Serial Number    1097629 non-null   int64  
 1   List Year        1097629 non-null   int64  
 2   Date Recorded   1097627 non-null   object  
 3   Town             1097629 non-null   object  
 4   Address          1097578 non-null   object  
 5   Assessed Value   1097629 non-null   float64 
 6   Sale Amount      1097629 non-null   float64 
 7   Sale Date        1097629 non-null   datetime64[ns]
 8   Property Type    715183 non-null   object  
 9   Residential Type 699240 non-null   object  
 10  Non Use Code     313451 non-null   object  
 11  Assessor Remarks 171228 non-null   object  
 12  OPM remarks      13931 non-null   object  
 13  Location         298111 non-null   object  
dtypes: float64(3), int64(2), object(9)
memory usage: 117.2+ MB
```

Figure 2: Cleaning Process Output (Replace with your screenshot)

4 Exploratory Data Analysis (EDA)

4.1 KPI Cards

- Total Sale Amount: 324M
 - Number of Towns: 170
 - Average Sales Ratio: 9.09

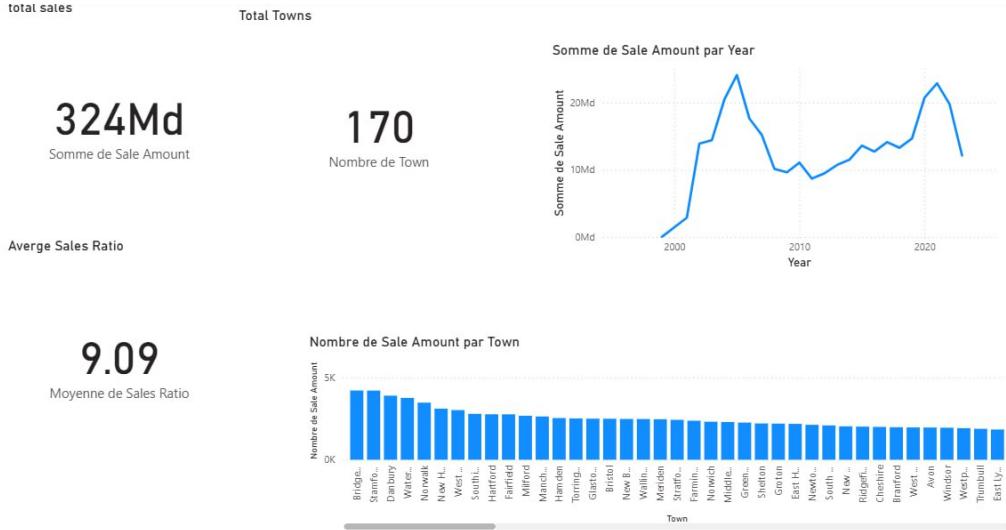


Figure 3: Power BI KPI Overview

The trend shows a rapid increase after 2000, a peak around 2018, and a noticeable decline in the most recent year.

Bridgeport, Stamford, Danbury, Waterbury, and Norwalk appear as the top-performing towns.

Scatter Plot: Assessed Value vs Sale Amount



Figure 4: Assessed Value vs Sale Amount

Overview: A general positive correlation is visible, with significant variance and some extreme outliers.

Town-Level Summary

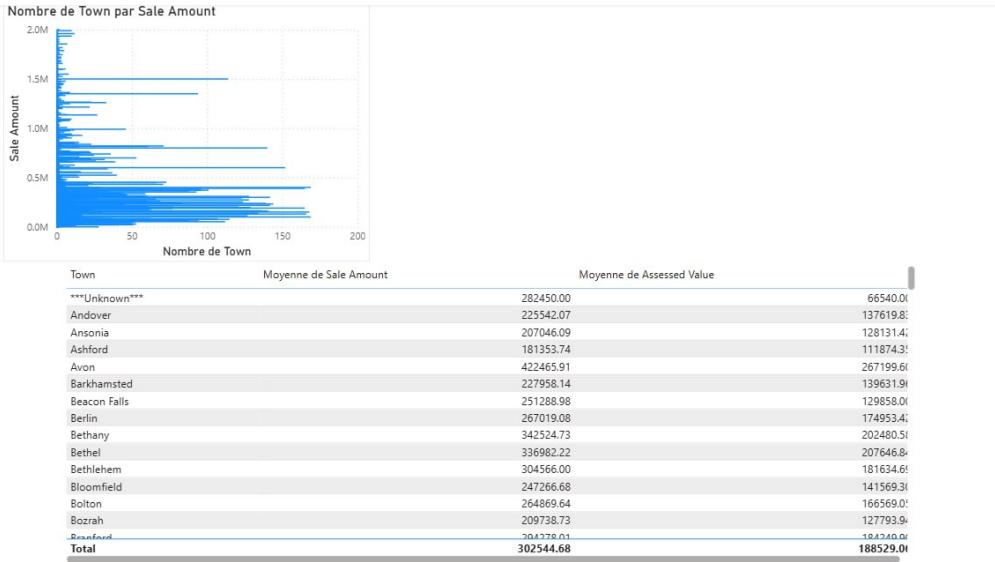


Figure 5: Summary Table of Town Metrics

Overview: Overall average Sale Amount = 302,544. Overall average Assessed Value = 188,529.

5 Machine Learning: Random Forest

5.1 Code Overview

```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()
model.fit(X_train, y_train)
pred = model.predict(X_test)
```

5.2 Error Distribution

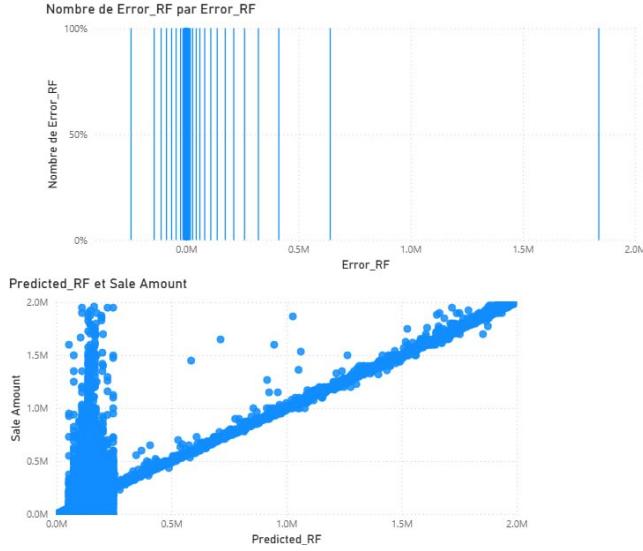


Figure 6: Random Forest Error Distribution

Overview: The model shows consistent errors, especially for high-value properties.

The RF model performs well at mid-range values but clearly underestimates luxury properties.

6 Power BI Dashboard Overview

All Power BI visuals are integrated in the previous sections. This dashboard enables interactive exploration of real estate trends.

7 Conclusion

This TP demonstrates the full analytical workflow: data loading, cleaning, exploration, modeling, and dashboard creation. Power BI and Python together provide a strong analytical stack for business insight generation.