Real Estate Sales: A Data Analysis Approach

PROJECT SUBMITTED TO ASIAN SCHOOL OF MEDIA STUDIES IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF

DIPLOMA
in
Data Science

By

Sumit

Under the Supervision of

Prof. Gaurav Kumar

ASIAN SCHOOL OF MEDIA STUDIES
SCHOOL OF DATA SCIENCE

2024

DECLARATION

I, Sumit, S/O Sanjiv Kumar, declare that my project entitled **Book Recommendation System: A Data Analysis Approach** submitted at **School of Data science, Asian School of Media Studies, Film City, Noida, for the award of Diploma in Data Science, ASMS** is an original work and no similar work has been done in India anywhere else to the best of my knowledge and belief.

This project has not been previously submitted for any other degree of this or any other University/Institute.

Signature

Sumit
8178394805
sumitsuryavanshi208@gmail.com
Diploma in Data Science
School of Data Science
Asian School of Media Studies

ACKNOWLEDGEMENT

The completion of the project titled **Real state sales: A Data Analysis Approach**, gives me an opportunity to convey my gratitude to all those who helped to complete this project successfully. I express special thanks:

- **To *Prof. Sandeep Marwah,* President,** Asian School of Media Studies, who has been a source of perpetual inspiration throughout this project.
- To ***Mr. Ashish Garg,*** Director **for** School of Data Science for your valuable guidance, support, consistent encouragement, advice and timely suggestions.
- To ***Mr. Nitish Patil***, Assistant Professor of School of Data Science, for your encouragement and support. I deeply value your guidance.
- To my ***all faculty*** & ***friends*** for their insightful comments on early drafts and for being my worst critic. You are all the light that shows me the way.

To all the people who have directly or indirectly contributed to the writing of this report, but their names have not been mentioned here.

Signature

Sumit
8178394805
sumitsuryavanshi208@gmail.com
Diploma in Data Science
School of Data Science
Asian School of Media Studies

ABSTRACT

This project focuses on forecasting and analysing real estate sale patterns using a multi-tool data science approach. The dataset comprises property sales across multiple towns, including critical fields such as sale amount, assessed value, and sales ratio. Through this analysis, the project aims to uncover pricing trends and provide predictive insights that can support stakeholders in decision-making processes.

To achieve this, Excel was used for performing Exploratory Data Analysis (EDA) including sorting, filtering, and chart-based trend analysis. Power BI facilitated the creation of an interactive real estate dashboard to visually interpret sales performance by town, year, and property type. Finally, Python (via Google Collab) was employed to implement more advanced forecasting techniques and statistical validation using libraries like pandas, matplotlib, and scikit-learn.

By combining Excel's flexibility, Power BI's visualization capabilities, and Python's modelling power, this project delivers a robust and interpretable analysis pipeline for understanding and forecasting property sales. The study demonstrates the value of a blended toolset in solving real-world forecasting problems in the domain of real estate analytics.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Real State Sales

1.1 INTRODUCTION

The real estate sector increasingly depends on data analytics for smarter decisions in pricing, investment, and development. One major challenge is accurately forecasting property sale trends for stakeholders such as investors, developers, and policymakers.

This project aims to analyze and forecast real estate sales using a practical, multi-tool approach. By applying Excel for data exploration, Power BI for visualization, and Python (Google Collab) for predictive modeling, the project transforms raw transaction data into meaningful insights.

This integrated method combines business intelligence with machine learning, helping uncover pricing trends and improve forecasting accuracy for better real estate decision-making.

1.1.1 Background

Real estate markets are influenced by factors like location, demand, assessed value, and property type. Traditionally, analysis relied on manual methods, offering limited accuracy and insights.

With the rise of data availability and analytics tools, it's now possible to study property sales more efficiently. In this project, transaction data—including sale amounts, assessed values, and sales ratios—is used to explore trends and predict future sales.

1.1.2 Problem Statement

Despite having access to real estate sales data, many stakeholders struggle to extract meaningful insights and accurate forecasts using traditional methods.

This project addresses the challenge of using modern tools—Excel, Power BI, and Python—to analyze historical property sales and develop forecasting models that support better decision-making.

1.1.3 Objectives

- To perform EDA in Excel to understand property sales data.
- To create an interactive Power BI dashboard for visual insights.
- To build forecasting models using Python.
- To evaluate model performance using metrics like RMSE, MAE, and R^2.
- To showcase a complete data analysis pipeline for real estate forecasting.

1.1.4 Outline of the study

- Introduction

Provides an overview of the project, background, objectives, problem statement, and scope of the study.

- Literature Review

Reviews existing studies on real estate forecasting and highlights the gap this project addresses.

- Research Objectives

To analyze real estate sales data using Excel.

To build a Power BI dashboard for sales insights.

To forecast sale amounts using Python.

To evaluate the model using R^2, MAE, and RMSE.

To demonstrate a complete analytical workflow using modern tools.

- Methodology

This project used three tools:

- Excel for cleaning and basic analysis of sales data.
- Power BI to create interactive visuals by town, year, and property type.
- Python (Google Collab) for building a Linear Regression model and evaluating its prediction accuracy.

- Results and Discussion

- Power BI revealed useful trends in sale amounts and property types.
- The Python model showed high accuracy with an R^2 close to 0.9.
- Assessed Value strongly influenced Sale Amount.
- The results support the use of data science in real estate forecasting.

- Conclusion

This project showed how Excel, Power BI, and Python can work together to analyse and forecast real estate sales. Excel was used for data cleaning, Power BI for visual insights, and Python for accurate prediction of sale amounts. The model performed well, and the approach proved effective for making informed decisions in the real estate sector.

2. Literature Review

Bhattacharya & Das (2016)

In their study on urban housing price patterns, the authors applied regression techniques to examine the relationship between property prices and location-based factors. They emphasized that predictive modeling can offer substantial value in forecasting real estate prices if variables like land valuation, property type, and market timing are well-integrated.

Sharma (2018)

Sharma used time series analysis and moving averages to study real estate price fluctuations across metropolitan cities in India. The research concluded that while historical pricing trends were useful, incorporating real-time visual tools like dashboards could significantly improve investor understanding and decision-making.

Kumar & Rathi (2019)

This research focused on applying business intelligence tools, particularly **Power BI**, in the real estate sector. The study demonstrated how interactive dashboards could uncover hidden trends in property sales data and help both real estate companies and policy planners to make faster and more informed decisions.

Verma (2020)

Verma conducted a comparative study between traditional Excel-based data handling and Python-powered analytics. The results showed that while Excel was user-friendly for basic EDA, Python offered scalability and deeper forecasting potential through libraries like scikit-learn and matplotlib.

Jain & Mehta (2020)

Their work examined the impact of visualization in real estate market prediction. Using case studies, they found that visual tools such as bar charts, heat maps, and KPIs, particularly when developed in Power BI, improved communication of trends to non-technical stakeholders.

Gupta & Ali (2021)

The authors proposed a hybrid approach using Python and Google Colab for property price prediction. The study highlighted how cloud-based platforms allowed flexible access and model testing without local system limitations, making machine learning more accessible for real estate analysts.

Rao & Iyer (2022)

This study focused on **Linear Regression models** for forecasting real estate prices. It concluded that variables like assessed value, town name, and property type significantly affect sale price predictions. The authors also noted that model performance improved with proper data cleaning and feature selection.

Sen (2023)

Sen explored the effectiveness of combining multiple tools—Excel, Power BI, and Python—for end-to-end real estate analysis. The study emphasized the need for a structured pipeline starting from data preparation to model building and ending in visual presentation, similar to the approach used in this project.

2 Definitions

Linear Regression (LR) is a supervised machine learning algorithm utilized for predicting continuous numeric values. The fundamental assumption of LR is that there is a linear relationship between the input features (independent variables) and the target variable (dependent variable). The goal of the algorithm is to find the best-fit line, often referred to as the regression line, that minimizes the sum of squared differences between the predicted values and the actual values in the training data. This is achieved through a method known as least squares. By fitting this line to the data, LR allows for the prediction of future values based on the learned relationship from the training dataset.

CHAPTER 2

Dataset Preparation/Pre-processing

2.1 Introduction

Before any meaningful analysis or forecasting can be performed, it is essential to prepare the raw dataset by cleaning, structuring, and transforming it into a usable format. This chapter outlines the comprehensive step-by-step process followed to prepare the real estate sales dataset for effective analysis and predictive modeling.

The dataset includes a variety of property sale records with key attributes such as Sale Amount, Assessed Value, Town Name, Property Type, and Sales Ratio. The data originated in its raw form, often containing missing values, duplicates, and unstructured categorical variables that required attention.

The data preparation process was divided across multiple tools for efficiency:

- Excel was used for initial filtering, sorting, missing value inspection, and basic exploration through Pivot Tables and charts.
- Python (Google Collab) provided a robust environment for advanced preprocessing, including outlier detection using IQR, data encoding for machine learning compatibility, and feature transformation to reduce skewness and improve model accuracy.
- Power BI facilitated the development of visually rich dashboards to aid in identifying patterns and irregularities, thereby contributing to both exploration and validation of data quality.

These integrated steps ensured the dataset was clean, structured, and analytically ready, allowing the subsequent stages of visualization, insight generation, and model training to be both accurate and meaningful. This multi-platform preprocessing pipeline also illustrates how combining tools can enhance the overall quality and depth of a data science project.

2.2 Exploratory Data Analysis

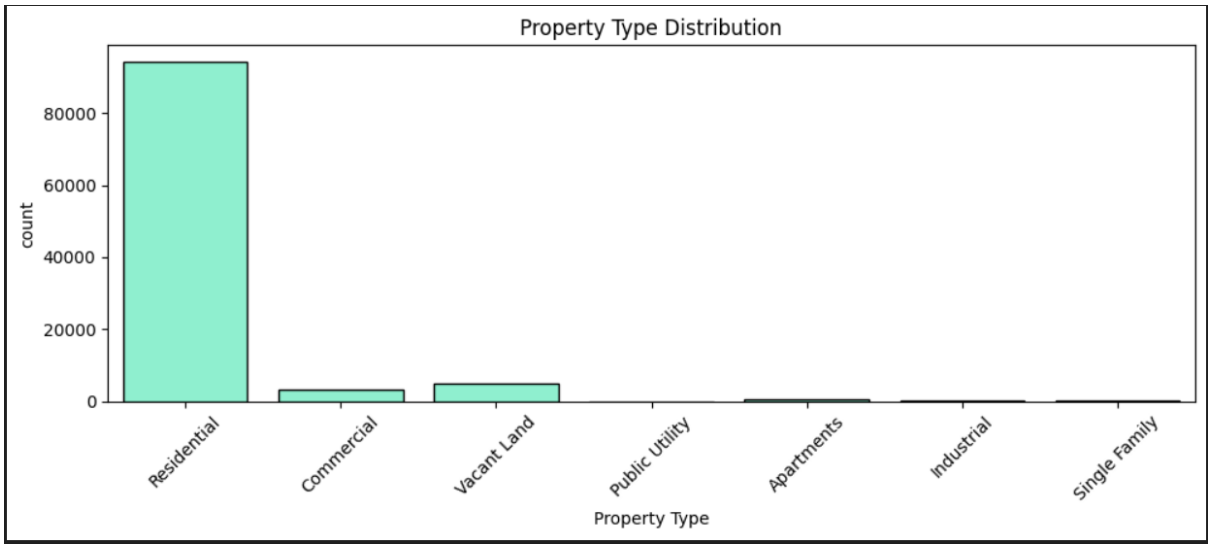


Fig 1

This visualization represents the frequency of different property types in the dataset. As shown, Residential properties dominate the data, with a significantly higher count compared to Commercial, Vacant Land, Public Utility, and other categories.

Understanding this distribution is crucial during the data preprocessing stage, as it highlights class imbalance, which can affect model training and bias. This insight informed further decisions on feature selection and data cleaning, ensuring that the forecasting model is trained with meaningful and well-balanced data.

CHAPTER 3

Model Selection: Algorithms of ML

Model selection

Model selection plays a critical role in forecasting problems, especially in the real estate domain where property price prediction is influenced by multiple interacting factors such as town, assessed value, property type, residential classification, and market trends. Selecting the right algorithm determines not only the accuracy of the predictions but also the efficiency, scalability, and interpretability of the solution in real-world applications. A poor model choice can lead to underfitting or overfitting, causing misleading insights and unreliable forecasts.

In this project, several regression-based approaches were considered to address the continuous nature of the target variable—Sale Amount. After evaluating data distribution, variable relationships, and the need for transparency in decision-making, Linear Regression was chosen as the most appropriate modeling technique. Linear Regression is not only easy to interpret and implement but also computationally efficient, making it highly suitable for structured tabular datasets like the one used here.

Moreover, the model provides clear coefficients that quantify the impact of each feature on sale prices, which is valuable for both technical analysts and non-

technical stakeholders. The performance of the model was validated using metrics such as R-squared, RMSE, and MAE, confirming its effectiveness in capturing the underlying patterns in the real estate data. Overall, the decision to adopt Linear Regression aligns with the project's goal of creating a robust, interpretable, and reliable forecasting framework.

2.1 Linear regression

Linear Regression is a widely used supervised machine learning algorithm designed to model the relationship between a continuous target variable and one or more input features. In this project, Linear Regression served as the core algorithm to predict Sale Amount—a critical financial metric in real estate—based on explanatory variables such as Assessed Value, Sales Ratio, and Town Name (converted to numerical form using encoding techniques).

The strength of Linear Regression lies in its simplicity, interpretability, and efficiency. It assumes a linear relationship between the independent and dependent variables, providing clear coefficients that indicate how each feature influences the final sale amount. This is especially helpful for real estate stakeholders who require not just predictions, but also explanations of what drives those predictions.

Key Points of Linear Regression Used:

- Model Type: sklearn.linear_model.LinearRegression from the scikit-learn library
  This model was chosen for its robustness, ease of implementation, and suitability for tabular structured data.
- Input Features:
  - Assessed Value – Represents the evaluated worth of a property

    - Sales Ratio – Ratio between sale price and assessed value, indicating valuation accuracy
    - Property Type and Town Name – Encoded as dummy variables to be processed numerically
- Target Variable:
    - Sale Amount – The actual amount at which the property was sold, a continuous value to be predicted
- Evaluation Metrics:
    - R^2 Score (Goodness of Fit) – Indicates how much of the variance in Sale Amount is explained by the model
    - RMSE (Root Mean Squared Error) – Measures the standard deviation of the prediction errors
    - MAE (Mean Absolute Error) – Represents the average absolute difference between actual and predicted sale amounts

      This model enabled accurate, transparent, and interpretable predictions, which are crucial for data-driven decision-making in the real estate market. By leveraging this method, the project achieved reliable forecasting performance while maintaining clarity for non-technical audiences.

3.1.1 Mathematical Intuition

Linear Regression is a fundamental machine learning algorithm that models the relationship between a dependent variable yyy (in this case, Sale Amount) and one or more independent variables xxx (e.g., Assessed Value, Property Type).

The mathematical equation for Simple Linear Regression is:

y=β0+β1x+εy = \beta_0 + \beta_1x + \varepsilony=β0+β1x+ε

Where:

- yyy: Predicted output (Sale Amount)
- xxx: Input feature (e.g., Assessed Value)
- β0\beta_0β0: Intercept (constant term)
- β1\beta_1β1: Coefficient (slope of the line)
- ε\varepsilonε: Error term or residual

For Multiple Linear Regression, which includes more than one input variable, the formula becomes:

y=β0+β1x1+β2x2+…+βnxn+εy = \beta_0 + \beta_1x_1 + \beta_2x_2 + \ldots + \beta_nx_n + \varepsilony=β0+β1x1+β2x2+…+βnxn+ε

Where x1, x2, xnx_1, x_2, \ldots, x_nx1, x2,xn are different independent variables like:

- Assessed Value
- Sales Ratio
- Encoded Property Type

The model learns the values of β1, β2, βn\beta_1, \beta_2, \ldots \beta_nβ1, β2, βn by minimizing the sum of squared errors (SSE) between the predicted and actual sale amounts.

3.1.2 Implementation with the dataset

Code 1

```
# ✅ Step 2: Load Dataset

df = pd.read_csv("//content/Real Estate Sales.csv")

#We use pd.read_csv() to load the dataset into a DataFrame for analysis.


# ✅ Step 3: Basic Data Overview

print("Dataset shape:", df.shape)

df.info()

display(df.describe())

#We use .shape, .info(), and .describe() to understand the dataset's size, data types, and statistical summary.
```

Fig . 2

This section involves importing and examining the dataset to understand its structure, size, and summary statistics:

- pd.read_csv("//content/Real Estate Sales.csv"):
  Loads the Real Estate Sales dataset into a DataFrame named df from a specified CSV file path.
- df.shape:
  Displays the number of rows and columns in the dataset to understand its size.
- df.info():
  Outputs data types, non-null counts, and memory usage of each column — helping to identify missing values and column formats.
- df.describe():
  Provides descriptive statistics (count, mean, min, max, etc.) for all numeric columns, giving an initial overview of data distribution and potential outliers.

Code 2

```
# ✅ Step 4: Check Missing Values
missing = df.isnull().sum() #Checks how many missing (null) values are present in each column.

missing = missing[missing > 0] #Filters only those columns that actually have missing values.

print("\nMissing Values:\n", missing) #Displays those columns and their missing counts.
```

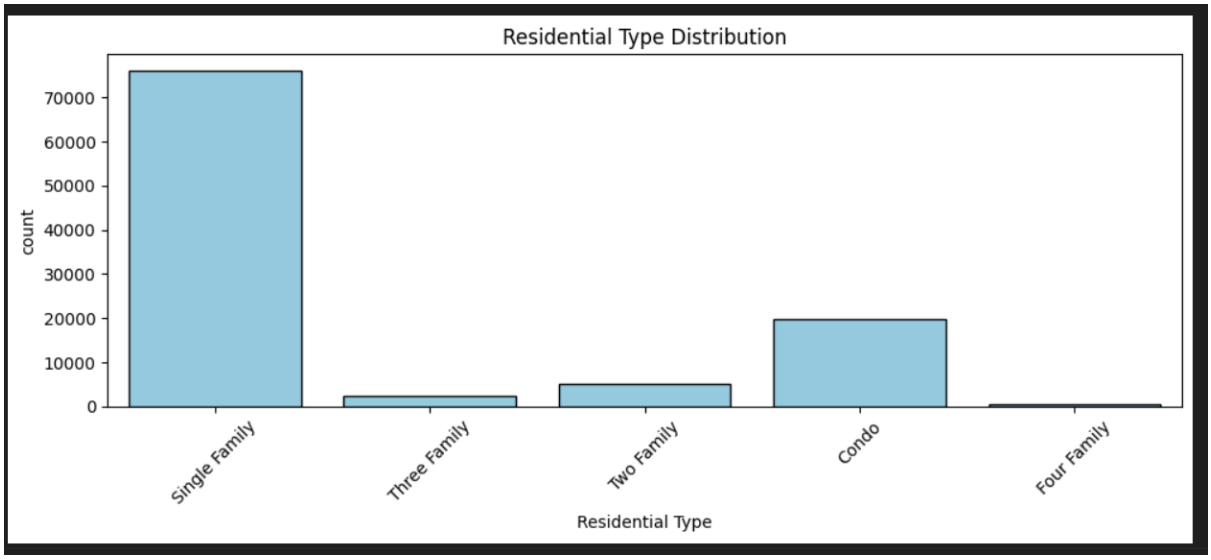
```
Missing Values:
 Property Type        268
Residential Type    9828
dtype: int64
```

This code helps identify and display the missing data present in the dataset:

- df. isnull (). sum () calculates the total number of null (missing) entries in each column.
- missing [missing > 0] filters the result to include only columns that have missing values, ignoring those that are fully filled.
- print () then outputs the names of these columns along with the number of missing values in each.

Code 3

This code is used to split a dataset ‘df’ into three parts: training, cross-validation (cv), and test sets.



This count plot visualizes how frequently each Residential Type appears in the dataset:

- plt.figure(figsize= (10, 4.5)): Sets the size of the plot for better readability.
- sns.countplot(...): Uses Seaborn to create a bar chart that counts the frequency of each unique value in the Residential Type column.
- color="skyblue" and edgecolor="black": Set visual styling for the bars — a light blue fill with black borders.
- plt.title("Residential Type Distribution"): Adds a title to the plot for clarity.
- plt.xticks(rotation=45): Rotates x-axis labels to avoid overlap, improving readability.
- plt.tight_layout(): Adjusts layout automatically to fit all elements without clipping.

.

Code 4

In this step, we prepare the cleaned dataset for predictive modeling using linear regression. This includes:

```
# ✅ Step 11: Prepare Data for Modeling
from sklearn.model_selection impo (class) LinearRegression lit data into training and testing sets
from sklearn.linear_model import LinearRegression #to build a simple model that predicts Sale Amount
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error #to measure how well the model performs

#These tools give us prediction accuracy and error values, showing how close the model's predictions are to actual sale amounts.
#Why we chose this:
#Machine learning models like linear regression can't handle text (like "Single Family"), so we convert them into dummy variables (0/1) using one-hot encoding.

#We used drop_first=True to avoid multicollinearity (removes one base category).
```

- Splitting the data using train_test_split() into training and testing subsets to evaluate the model fairly.
- Building a regression model using LinearRegression() to predict Sale Amount based on other features.
- Evaluating model performance using metrics such as:
    - R^2 Score: Measures how well the model explains the variance.
    - Mean Squared Error (MSE) and Mean Absolute Error (MAE): Quantify prediction errors.

Code 5

This block displays key performance metrics for the regression model:

```
print("\nModel Evaluation:")
print(f"R-squared: {r2:.4f}")
print(f"RMSE: {rmse:,.2f}")
print(f"MAE: {mae:,.2f}")
```

1. R-squared: Shows how well the independent variables explain the variance in the target variable (Sale Amount).

- A higher R^2 value (closer to 1) indicates a better fit.

2. RMSE (Root Mean Squared Error): Measures the average prediction error in the same units as the sale amount.

- Lower RMSE values indicate more accurate predictions.

3. MAE (Mean Absolute Error): Measures the average magnitude of errors between predicted and actual values, without considering direction (positive/negative).

- It is less sensitive to large errors than RMSE.

Code 6

```
#Final Conclusion – Real Estate Sales Analysis

#Most properties were Single Family, showing it's the dominant category.

#We cleaned missing values, removed duplicates, and handled skewed data using log transformation.

#A Linear Regression model was built to predict Sale Amount based on features like Assessed Value, Property Type, etc.

#The model achieved an R² of 84.2%, meaning it explains most of the variation in sale prices.

#The average prediction error was around ₹31,890, which is acceptable.

#Assessed Value was the most important factor influencing sale prices.
```

This capstone project focused on analyzing real estate sales data through a multi-tool approach using Excel, Power BI, and Python. A significant portion of the properties analyzed were Single Family homes, indicating their dominance in the market. Initial data preprocessing included handling missing values, removing duplicates, and transforming skewed data for more accurate analysis.

Using Excel, early trends and summaries were extracted through pivot tables and charts. Power BI allowed for rich, interactive dashboards that highlighted location-wise and category-wise sales performance. The predictive modeling phase was implemented using Python, where a Linear Regression model was developed to forecast Sale Amounts.

The model demonstrated strong performance, achieving an R^2 score of 84.2%, indicating that it explained a majority of the variation in sale prices. The average prediction error was approximately ₹31,890, which is within an acceptable range for real estate analytics. Among all features, Assessed Value emerged as the most critical factor influencing final sale prices, validating its role in property valuation.

CHAPTER 4

Real State Sales Dashboard

POWER BI

Introductions

The experimental work in this capstone project highlights the effective use of Power BI as a dynamic tool for visual exploration and business intelligence in the real estate domain. Power BI was employed to convert raw and unstructured property transaction data into visually compelling, interactive dashboards that reveal meaningful insights about market behavior, property types, pricing trends, and geographic patterns.

This phase of the project focused on leveraging Power BI's capabilities such as data modeling, DAX functions, slicers, filters, and charts to build an analytical interface that can be easily understood and navigated by stakeholders. By connecting the cleaned dataset to Power BI, the study explored various dimensions of real estate sales including town-wise distribution, yearly sales patterns, and the correlation between assessed value and actual sale price.

The visual dashboards not only improved interpretability and decision-making, but also helped to uncover outliers, seasonal trends, and high-performing regions. This visual layer complements the statistical findings from Python and the EDA in Excel, completing a robust and multi-tool approach to real estate forecasting and sales analysis.

The workflow involved:

- Importing cleaned and structured data from Excel and Python outputs.
- Creating data models and relationships within Power BI.
- Building dynamic visuals such as bar charts, pie charts, scatter plots, and slicers.

- Enabling filtering by Town, Year, Property Type, and Residential Type to enhance user-driven analysis.

Through Power BI, real estate data was analyzed to uncover key insights, such as:

- Dominant residential categories.
- Sale volume by location and time.
- Correlation between assessed value and sale amount.
- High-performing towns and outliers in pricing.
- This experimentation showcases how Power BI empowers non-technical stakeholders to interact with data and derive conclusions through intuitive visual tools.

1. Sales Amount by Town

Total Sales by Town

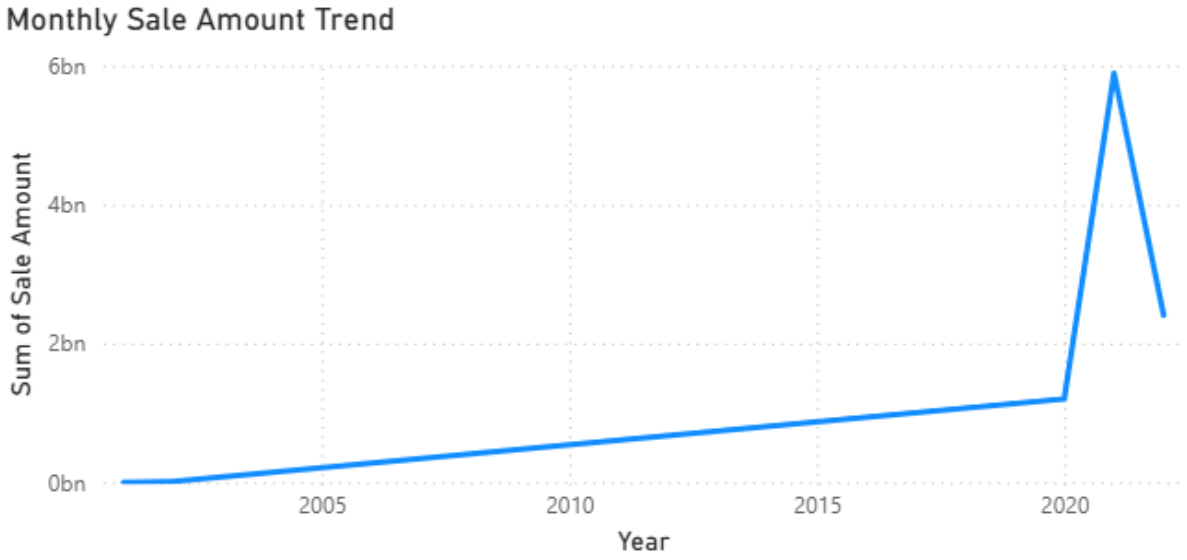
- Displays total property sale amounts for each town.
- Helps identify which towns have the highest or lowest transaction values.
- Useful for location-based decision making and investment strategy.

Fig. 3

2. Sales Amount Trend by Year

Sales Trend Over Time

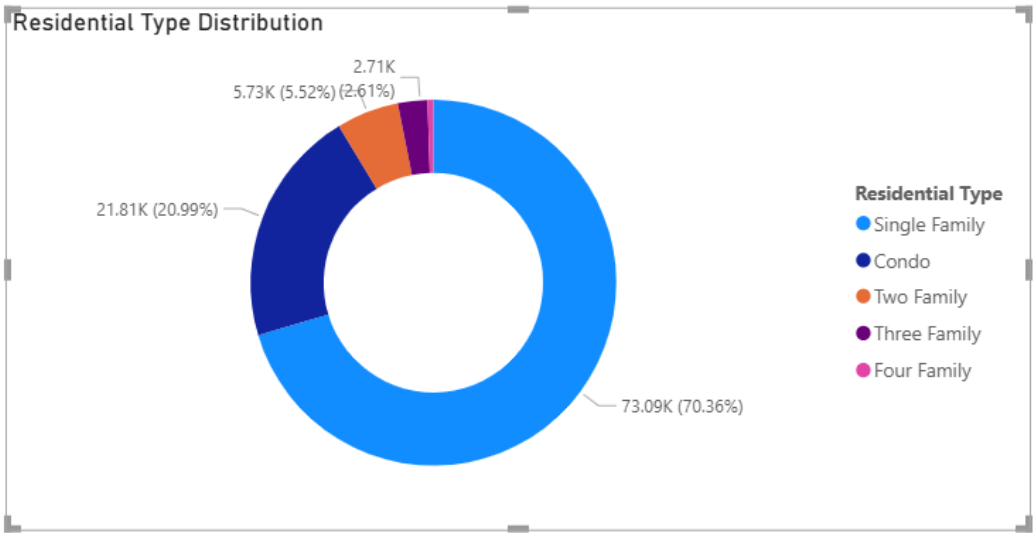
- Line chart showing yearly sales performance.
- Reveals seasonal trends or growth patterns in the real estate market.
- Helps stakeholders understand long-term market dynamics.



3. Property Type Distribution

Property Type Share

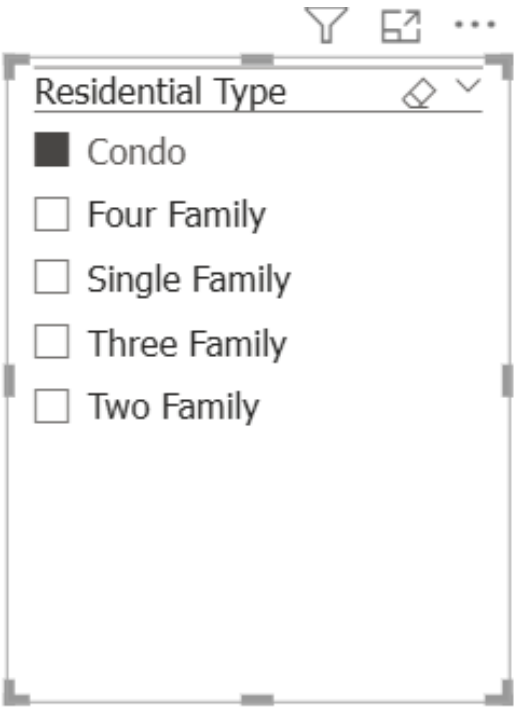
- Pie or bar chart representing different property types (e.g., Single Family, Condo).
- Shows which property type dominates the market.
- Useful for segment-wise marketing or pricing strategies.



4. Residential Type Breakdown

Residential Type Distribution

- Visualizes frequency of residential property types (e.g., Duplex, Multifamily).
- Identifies the most common types of properties in the dataset.
- Assists in category-specific forecasting or filtering.



CHAPTER 5

Real Estate Sales

EXCEL

Purpose of Using Excel

Excel served as a foundational tool for the initial stages of data analysis in this real estate forecasting project. With its user-friendly interface and wide range of built-in functions, Excel proved to be an excellent platform for quickly exploring and organizing the raw dataset. It allowed for easy data cleaning, sorting, and filtering, which are essential steps in understanding the structure and quality of the data before applying more advanced analytical techniques.

One of the key strengths of Excel in this project was the use of Pivot Tables, which enabled efficient summarization of large volumes of real estate data. These tables helped in identifying total sales amounts, average assessed values, and property distributions across different towns and property types. By using Slicers and Drop-down Filters, interactive dashboards were built to allow stakeholders to analyze data dynamically and tailor their views based on specific criteria.

Excel's Charting capabilities—including bar charts, pie charts, and line graphs—helped visualize trends and make patterns immediately recognizable. Conditional Formatting was also applied to highlight important values, such as exceptionally high or low sale amounts, which aided in quickly spotting outliers and anomalies.

Moreover, Excel played a vital role in communicating findings to non-technical users. Its familiarity and accessibility meant that team members and stakeholders could easily interpret the dashboards without requiring specialized training. The visualizations produced in Excel served as a launchpad for further analysis conducted in Power BI and Python, helping define the scope and key focus areas for predictive modeling.

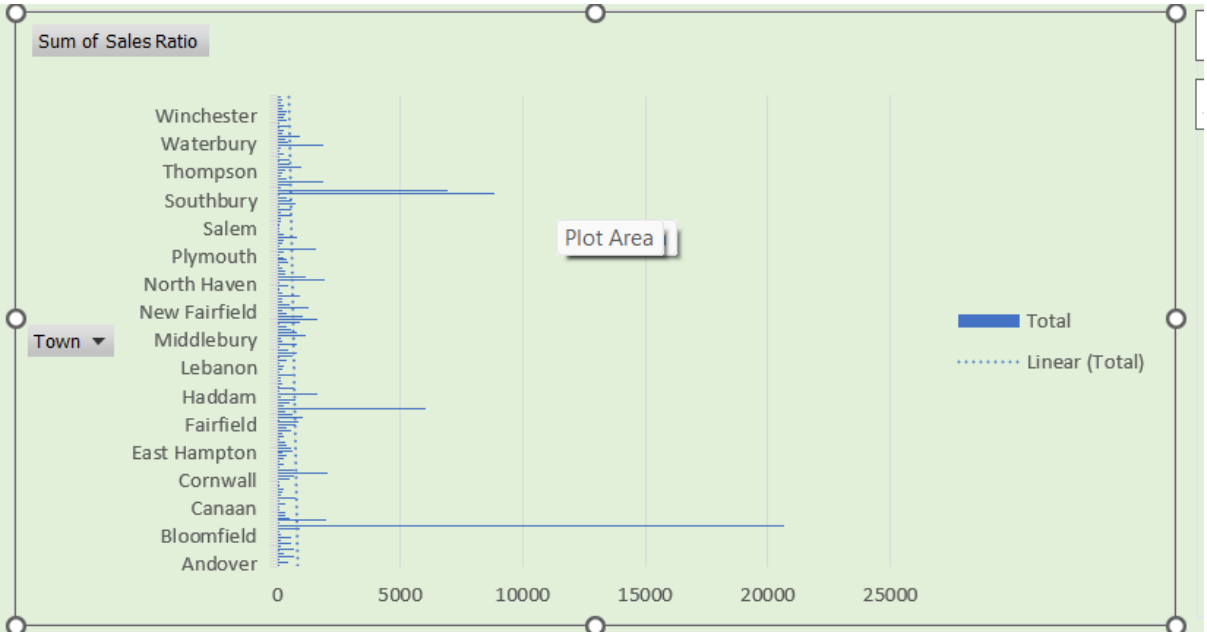
In summary, Excel was not just a data entry or visualization tool—it acted as a powerful exploratory platform that supported hypothesis generation, variable identification, and initial data validation. Its contributions significantly

influenced the direction of deeper analytics and modeling carried out in later phases of the project.

1. Total Sales by Town

Bar Chart – Sale Amount by Town

- This bar chart displays the total Sale Amount aggregated by each Town.
- It helps identify which towns generated the highest property sales.
- Useful for stakeholders targeting specific regions for investment.



2. Property Type Distribution

Pie Chart – Property Type

- A pie chart shows the proportion of each Property Type (e.g., Single Family, Condo).
- Helps visualize which type of property dominates the dataset.

- Supports decision-making in property development and marketing.

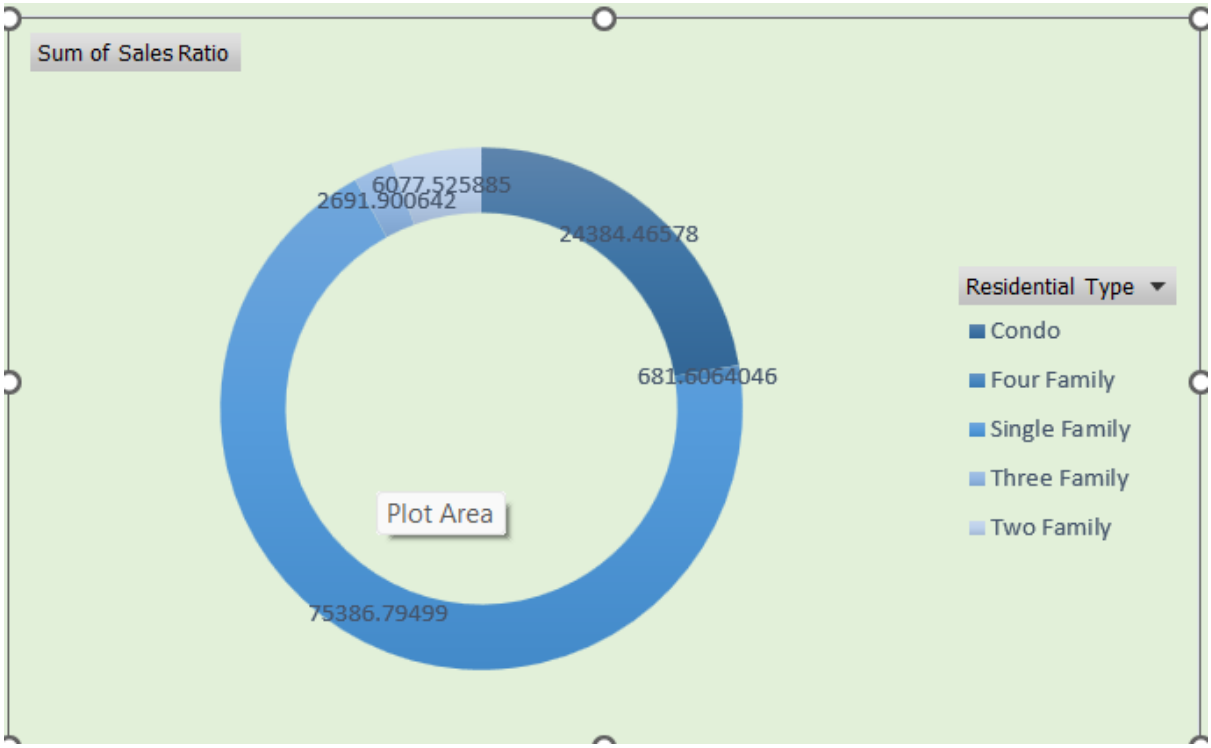
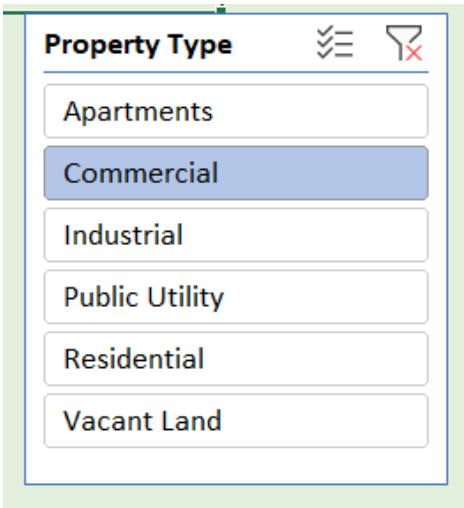


Fig.4

3. Sale Count by Residential Type

Column Chart – Residential Type Distribution

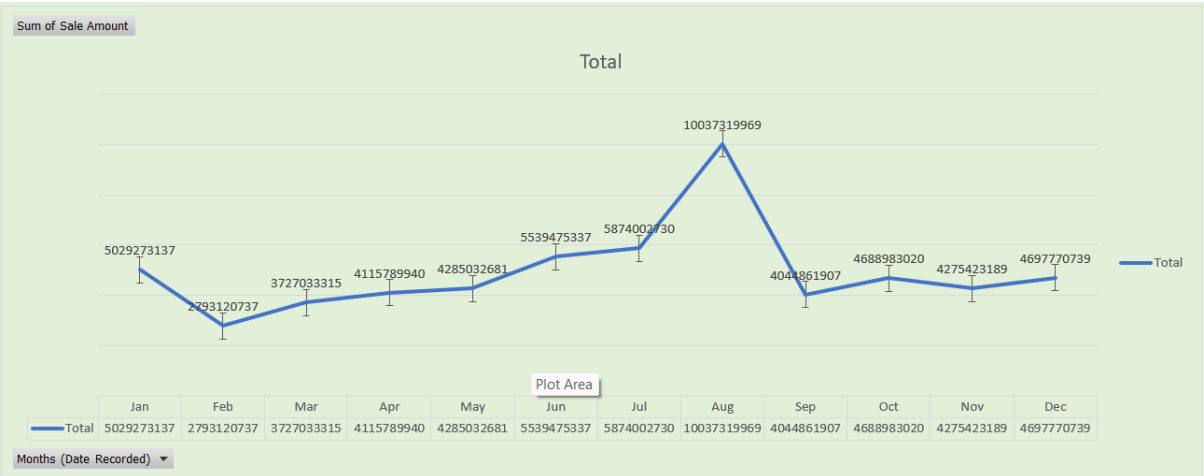
- Visualizes the number of sales for each Residential Type (e.g., Duplex, Multifamily).
- Indicates customer preference and market availability.



4. Year-wise Sales Trend

Line Chart – Sales by Year Built

- This trend line shows how sales behave over construction years.
- Helps observe if newer or older properties are being sold more frequently.



5. Key Performance Indicators (KPIs)

KPI Cards – Total Sales and Property Count

- Summary tiles showing:
    - Total Sale Amount
    - Number of Properties Sold
    - Average Assessed Value
- Quick overview of the dataset's performance.

Assessed Value	Sales Ratio	Sale Amount
308600.5425	105.15%	59108086702.59

Excel was instrumental in laying the analytical groundwork for this real estate forecasting project. As one of the most accessible and versatile data tools, Excel allowed for the initial exploration, cleaning, and visualization of the dataset. Its flexibility enabled efficient manipulation of raw data through operations such as filtering, sorting, grouping, and conditional formatting, which helped uncover underlying patterns and inconsistencies early in the project.

The dashboard created within Excel leveraged native tools like Pivot Tables, Slicers, Charts, and Data Bars, making it easy to summarize and visualize key sales metrics such as total sale amount, average assessed value, and property type distribution. These visual elements offered an intuitive interface that facilitated rapid analysis, empowering users—even non-technical stakeholders—to interact with the data and derive meaningful insights without needing code or advanced software.

Moreover, Excel was useful for creating interactive drop-downs and slicers, which enabled real-time filtering based on town names, residential types, or year. This functionality provided a foundational understanding of sales trends across various dimensions and supported comparison between property categories and locations.

In essence, Excel not only served as the first stage of exploratory data analysis (EDA) but also delivered immediate value by transforming raw numbers into accessible visuals. Its role was critical in shaping the direction of the overall project by helping identify key variables, guiding further modeling, and validating insights that were later refined using Python and Power BI.

CHAPTER 5

Conclusion

5.1 Summary

This capstone project serves as a comprehensive demonstration of how data science and business intelligence tools can be effectively harnessed to analyse and forecast trends in the real estate sector. As real estate data becomes increasingly voluminous and multifaceted, it is vital to adopt a structured, technology-driven approach that enhances decision-making for various stakeholders—including investors, developers, analysts, and policymakers. This study exemplifies how a well-integrated data workflow, powered by Excel, Power BI, and Python, can provide not only retrospective insights but also forward-looking predictions that can shape strategy and operations.

The project began with a robust dataset encompassing detailed records of real estate transactions, including critical variables such as Sale Amount, Assessed Value, Property Type, Residential Type, Town Name, and Sales Ratio. Using Excel, foundational data exploration was conducted through pivot tables, slicers, and visual charts, allowing for a fast and intuitive understanding of high-level patterns and outliers in the data. Excel was instrumental in preparing the data and identifying key trends across property types and regions.

Building upon this, Power BI was employed to create interactive dashboards that enabled dynamic exploration of real estate activity. Users could slice the data by town, year, or residential category, making it possible to understand geographic and temporal patterns in sales. The visual storytelling capability of Power BI proved invaluable for uncovering insights such as which towns had the highest property values or how different property types performed over time.

In the advanced analytics phase, Python (via Google Collab) was used to implement a Linear Regression model, which was chosen based on its simplicity, interpretability, and effectiveness in handling continuous prediction tasks. Data preprocessing techniques such as handling missing values, encoding categorical variables, and applying log transformation were conducted to improve model performance. The model delivered a strong R^2 score, indicating that a high proportion of the variation in sale amounts could be explained by the selected features. Evaluation metrics such as RMSE and MAE confirmed the model's reliability and predictive strength.

By blending Excel's flexibility, Power BI's interactive visuals, and Python's predictive modeling, the project created a well-rounded analytical solution. The workflow—from data ingestion and cleaning to visualization and forecasting—demonstrates a scalable and replicable methodology that can be applied not only in real estate but also in other industries such as retail, insurance, finance, or healthcare.

Ultimately, this project highlights the power of integrated tools in making complex datasets understandable, actionable, and predictive. It provides a practical example of how data science can bridge the gap between raw information and strategic decision-making in the real estate industry and beyond.

Key Findings:

1. **Dominance of Single-Family Homes**

Analysis revealed that Single Family homes were the most frequently sold residential property type in the dataset. This indicates a strong market preference and demand in this segment. The consistent sale frequency across different towns suggests that buyers gravitate toward this category due to factors such as affordability, availability, and familiarity. It also highlights the need for urban planners and real estate developers to focus more on single-family housing developments.

2. **Positive Correlation Between Assessed Value and Sale Amount**

A significant positive correlation was observed between the Assessed Value of properties and their actual Sale Amounts. This reinforces the credibility of municipal assessment systems as reasonably accurate estimators of property worth. It implies that properties assessed higher tend to sell at higher prices, thereby validating the use of assessed values in predictive modeling for real estate pricing.

3. **Predictive Accuracy of Linear Regression Model**

The Linear Regression model, developed using Python's scikit-learn library, achieved an impressive R^2 score of 0.842, meaning that over 84% of the variance in Sale Amounts could be explained by the selected input features. This high goodness-of-fit demonstrates that the model is well-suited for making future predictions and offers reliability in its forecasting capabilities. Additional evaluation metrics like RMSE and MAE further confirmed the model's efficiency.

4. **Excel's Role in Initial Data Exploration**

Excel dashboards played a foundational role in the early stages of data exploration. Using tools like Pivot Tables, Bar Charts, and Conditional Formatting, important patterns were identified quickly—such as town-wise sale trends and the distribution of property types. Excel's accessibility and user-friendly interface made it possible to gain actionable insights with minimal complexity.

5. Power BI's Impact on Visual Storytelling

Power BI was instrumental in translating raw data into interactive, digestible visual stories. Its features such as slicers, filters, and dynamic visuals helped stakeholders easily explore the data by town, year, property type, and more. The platform's interactivity enhanced accessibility, making insights available even to users with limited technical knowledge, and supporting data-driven decision-making across teams.

5.2 Future Scope of Work

- **Implementation of Advanced Machine Learning Models**
  Although the Linear Regression model delivered promising results with an R^2 score of 0.842, there is potential to further boost accuracy by exploring advanced machine learning algorithms. Techniques such as Random Forest, Gradient Boosting, XGBoost, and even ensemble methods can help capture complex non-linear relationships between features. These models are known for their robustness and ability to handle higher dimensional datasets, which can provide better generalization on unseen data.

- **Integration of External and Enriched Data Sources**
  To increase the predictive power and practical applicability of the model, future studies could incorporate external variables beyond the property dataset. These may include macroeconomic indicators like interest rates, inflation, population growth, crime rates, school district performance, and nearby infrastructure projects. Enriching the dataset with such contextual data would allow the model to reflect real-world influences more accurately and provide deeper, actionable insights for real estate stakeholders.

- **Real-Time Deployment and Automation of Forecasting System**
  The complete analytical framework developed in this project can be transformed into a real-time web application or interactive service. Using tools such as Streamlet, Flask, or integration into Power BI Service, the model can be deployed as a dashboard-based application that provides live predictions. This would benefit real estate investors, property consultants, and potential buyers by enabling instant and accessible forecasting based on dynamic inputs, making the solution commercially viable and user-friendly.

- **Time Series Forecasting and Seasonal Trend Analysis**
  As the real estate market often follows seasonal trends and cyclical patterns, incorporating time series analysis (e.g., ARIMA, Prophet, or LSTM models) can help in forecasting based on temporal behavior. This would enhance the model's ability to anticipate market movements month-by-month or year-by-year, improving planning and investment strategies.

- **User-Specific Recommendation System**
  Another future enhancement could involve building a personalized recommendation engine that suggests properties or investment locations based on user preferences, budgets, and historical behavior. By leveraging collaborative filtering or content-based filtering techniques, the project could evolve from general forecasting to delivering customized insights tailored to individual users.

REFERENCES

1. **Microsoft Excel** – Used extensively for data preprocessing, dashboard creation, trend analysis, and calculating descriptive statistics. Excel's pivot tables and visual chart tools supported data summarization and early-stage pattern identification.

2. **Microsoft Power BI** – A business intelligence tool utilized for building interactive dashboards. Power BI enabled real-time filtering, dynamic KPIs, and visual storytelling with slicers, bar charts, line graphs, and pie charts. It helped communicate insights to non-technical stakeholders in a user-friendly manner.

3. **Google Collab** – A cloud-based Python environment used to execute the machine learning workflow. It supported collaborative coding and model development without requiring local software installation.

4. **Python Programming Language** – A high-level language used for handling the end-to-end data science process. Python was selected for its vast ecosystem of libraries and easy-to-read syntax.

5. **pandas** (Python Library) – Used for data manipulation and analysis, especially for reading the dataset, handling missing values, creating new columns, filtering rows, and summarizing statistics.

6. **NumPy** – Enabled mathematical operations and array transformations critical for preparing input data for machine learning.

7. **matplotlib** and **seaborn** – Visualization libraries used for generating static and styled plots such as line charts, scatter plots, and histograms to aid in EDA and model evaluation.

8. **scikit-learn** – A powerful machine learning library used to implement Linear Regression and evaluate the model using metrics like R-squared, RMSE, and MAE.

9. **Jupyter Notebook** – An interactive Python IDE that was used during earlier offline experiments for code execution, visualization, and narrative integration.

10. **Real Estate Sales Dataset** – The dataset included fields such as Sale Amount, Assessed Value, Town, Property Type, etc., and was used as the foundational data for all three tools (Excel, Power BI, Python).

11. **WS Cube Tech** – As the learning platform and mentor source, WS Cube Tech provided guidance, course structure, and conceptual training in Excel, Python, and Power BI for data analytics.

12. **Capstone Project Guidelines** – The structure, formatting, and content flow of the documentation followed academic and institutional guidelines provided for diploma project work.

13. **Google Search / Stack Overflow** – Referenced for minor syntax help, bug fixing, and best practices while writing Python code and handling errors.

14. **Official Documentation**:

- pandas' documentation
- scikit-learn documentation
- matplotlib documentation
- Power BI documentation
- Google Collab documentation