Forecasting the Return on Investment (ROI) for a New House in Melbourne: A Five-year Sales Regression Analysis

BDA & DaViz Group Member:

2540135473 – Dora Kalifa Dharmawan – 30%

2501983105 – Pristian Budi Dharmawan – 40%

2540131310 – Pirelli Rahelya Piri – 30%

INTRODUCTION

A real estate company just got an investment with a total of $100 million. This investment is aimed at constructing several new home clusters with a target Return on Investment (ROI) of around $200 thousand for each unit sold. As we know, the cost of a property is increasing every year by around 7.9% per year in Melbourne (propertyupdate.com).

This company has difficulties developing home clusters for certain areas in Melbourne. Their target market is the new family or a family with a medium- to high-class monetary level. They wanted to open these home clusters with a low population density, a good environment, and the nearest to the centre of the city. They took a dataset from 2017 to 2018 to see which region has the highest ROI in Melbourne. However, there are several limitations to building a home cluster, such as the law, construction costs, land availability, etc.

With those criteria, the company assumes that they will be able to overcome their limitations in this project. To increase their ROI, they could build certain facilities in their area, bundle packages, sell furnished or unfurnished houses, etc. Other than that, to make their forecasting of ROI more valid, they wanted to use the dataset that had already been received.

However, they are unclear about the dataset they received. Thus, this company hired a data scientist to process the data and give them the best advice on which region they should construct to have the highest ROI within five years of analysis. The data scientist suggested that to build a forecasting model, he wanted to conduct regression analysis by leveraging several regression models, such as Multiple Linear Regression (MLR), Lasso Regression (LR), and Random Forest Regression (RFR).

METHODOLOGY

The dataset of this project can be accessed through the link below:

[Melbourne Housing Market]

This dataset contains the following attributes:

- Suburb
- Address
- Rooms
- Price
- Method
- Type
- SellerG
- Date
- Distance
- Postcode
- Bedroom2
- Bathroom
- Car
- Landsize
- BuildingArea
- YearBuild
- CouncilArea
- Lattitude
- Longitude
- Regionname
- Propertycount

However, there are two datasets, which are:

1. MELBOURNE_HOUSE_PRICES_LESS.csv
2. Melbourne_housing_FULL.csv

The only difference between these two datasets is the number of attributes. Thus, we will forecast it using these two datasets and two results. Below are the summary statistics for dataset number 1:

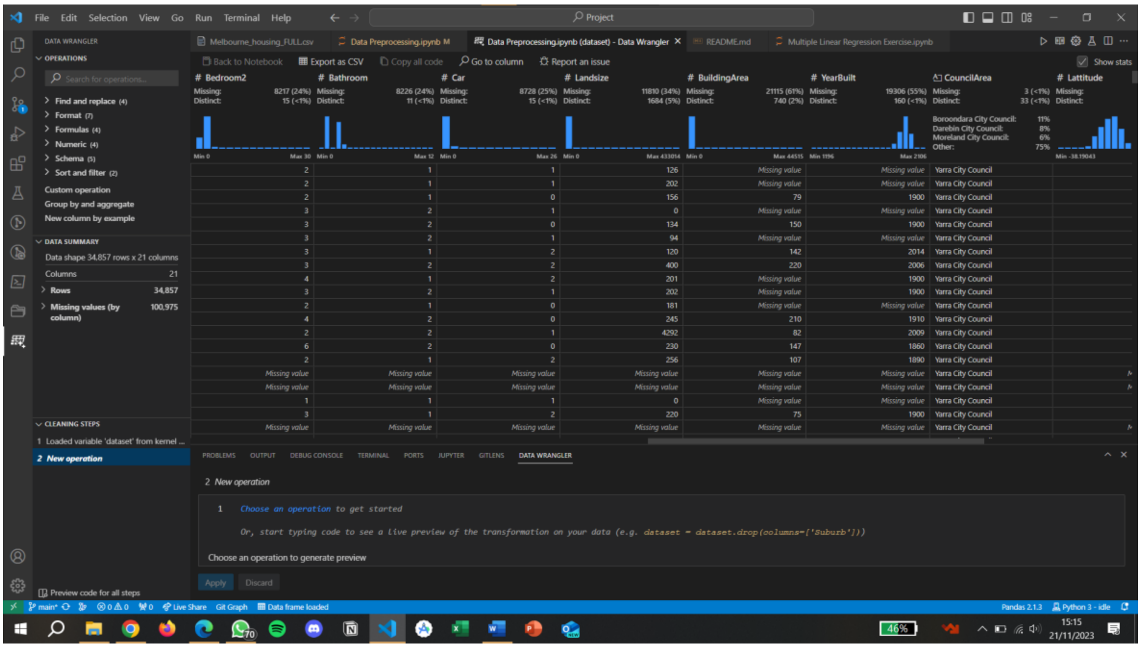
	Rooms	Price	Postcode	Propertycount	Distance
count	63023.000000	4.843300e+04	63023.000000	63023.000000	63023.000000
mean	3.110595	9.978982e+05	3125.673897	7617.728131	12.684829
std	0.957551	5.934989e+05	125.626877	4424.423167	7.592015
min	1.000000	8.500000e+04	3000.000000	39.000000	0.000000
25%	3.000000	6.200000e+05	3056.000000	4380.000000	7.000000
50%	3.000000	8.300000e+05	3107.000000	6795.000000	11.400000
75%	4.000000	1.220000e+06	3163.000000	10412.000000	16.700000
max	31.000000	1.120000e+07	3980.000000	21650.000000	64.100000

```
RangeIndex: 63023 entries, 0 to 63022
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Suburb         63023 non-null  object
 1   Address        63023 non-null  object
 2   Rooms          63023 non-null  int64
 3   Type           63023 non-null  object
 4   Price          48433 non-null  float64
 5   Method         63023 non-null  object
 6   SellerG        63023 non-null  object
 7   Date           63023 non-null  object
 8   Postcode       63023 non-null  int64
 9   Regionname     63023 non-null  object
 10  Propertycount  63023 non-null  int64
 11  Distance       63023 non-null  float64
 12  CouncilArea    63023 non-null  object
dtypes: float64(2), int64(3), object(8)
memory usage: 6.3+ MB
```

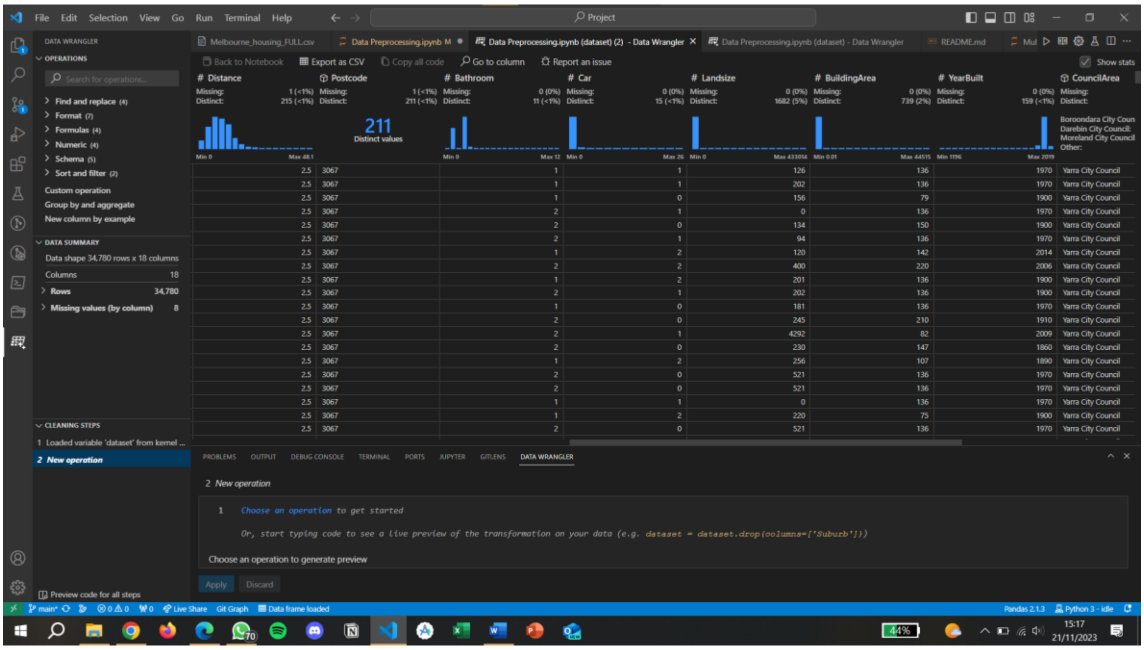
DATA PREPROCESSING

We decided to choose the “Melbourne_housing_FULL.csv” dataset to be analyzed using Multiple Linear Regression and we have already cleaned the dataset. However, the difference between the uncleaned dataset and the cleaned dataset can be seen below:

- Uncleaned Dataset



- Cleaned Dataset



It might be hard to see the differences, thus we made the full documentation of the data preprocessing in our GitHub below:

[Full Documentation of the Melbourne Housing Market Regression Analysis]

Data Cleaning

1. Handling the Duplicate or Ambiguous Columns

   In the original dataset, we find that two main columns lead to ambiguity, which is 'Rooms' and 'Bedroom2'. Thus, we checked it and found there is no significant difference between them, and we decided to remove the 'Bedroom2' column from this analysis since it is not unnecessary.

```
dataset['Duplicate Columns'] = dataset['Rooms'] - dataset['Bedroom2']
print("Rooms VS Bedroom2 (Average): ", dataset['Duplicate Columns'].mean())
print("Rooms VS Bedroom2 (Median): ", dataset['Duplicate Columns'].median())
dataset.head(10)
[249] ✓ 0.0s

Rooms VS Bedroom2 (Average):  0.016253753753753753
Rooms VS Bedroom2 (Median):  0.0
```

2. Handling Missing Values

   Other than that, we found that there are several missing values in this analysis which you can see as follows:

As you can see in the figure, there are a lot of missing values within each column.

To handle these missing values, we chose to use the measure of Central Tendency (e.g., Median), with exceptions for 'Car', 'Latitude', and 'Longitude' will be replaced with '0'

```
# Summary of missing values
dataset.isnull().sum()
✓ 0.0s

Suburb              0
Address             0
Rooms               0
Type                0
Price            7610
Method              0
SellerG             0
Date                0
Distance            1
Postcode            1
Bathroom         8226
Car              8728
Landsize        11810
BuildingArea    21115
YearBuilt       19306
CouncilArea         3
Lattitude        7976
Longtitude       7976
Regionname          3
Propertycount       3
dtype: int64
```

Missing Values Summary

```
# Average of missing values
dataset.isnull().sum()/len(dataset)*100
✓ 0.0s

Suburb           0.000000
Address          0.000000
Rooms            0.000000
Type             0.000000
Price           21.832057
Method           0.000000
SellerG          0.000000
Date             0.000000
Distance         0.002869
Postcode         0.002869
Bathroom        23.599277
Car             25.039447
Landsize        33.881286
BuildingArea    60.576068
YearBuilt       55.386293
CouncilArea      0.008607
Lattitude       22.882061
Longtitude      22.882061
Regionname       0.008607
Propertycount    0.008607
dtype: float64
```

Missing Values Average

3. Handling the Outliers

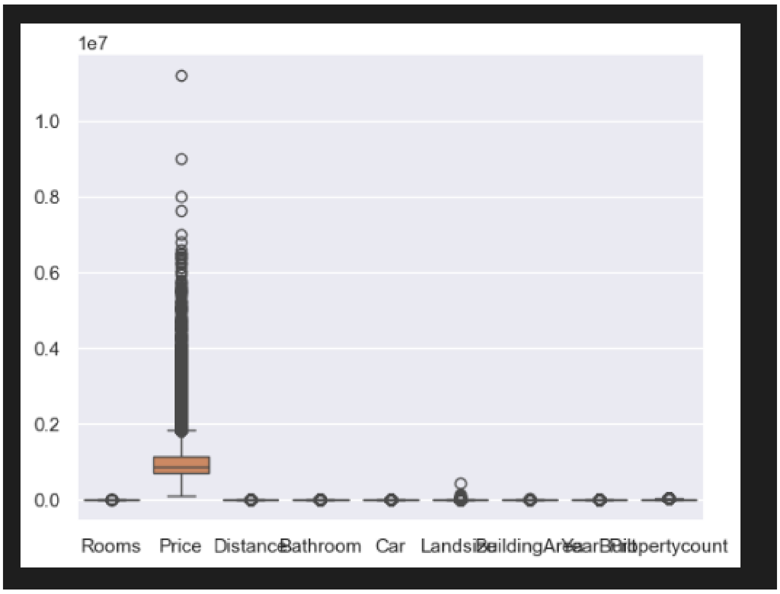
To find the outlier, we could generate the simple descriptive statistics below to obtain some oddities of the data, which can be seen as follows:

```
dataset.describe().transpose()
```

✓ 0.0s

	count	mean	std	min	25%	50%	75%	max
Rooms	34857.0	3.031012e+00	0.969933	1.00000	2.00000	3.00000	4.000000e+00	1.600000e+01
Price	34857.0	1.010838e+06	571999.150635	85000.00000	695000.00000	870000.00000	1.150000e+06	1.120000e+07
Distance	34856.0	1.118493e+01	6.788892	0.00000	6.40000	10.30000	1.400000e+01	4.810000e+01
Bathroom	34857.0	1.713343e+00	0.652754	0.00000	1.00000	2.00000	2.000000e+00	1.200000e+01
Car	34857.0	1.295952e+00	1.151893	0.00000	0.00000	1.00000	2.000000e+00	2.600000e+01
Landsize	34857.0	5.690015e+02	2763.907731	0.00000	357.00000	521.00000	5.980000e+02	4.330140e+05
BuildingArea	34857.0	1.455628e+02	252.222644	0.00000	136.00000	136.00000	1.360000e+02	4.451500e+04
YearBuilt	34857.0	1.967899e+03	25.042048	1196.00000	1970.00000	1970.00000	1.970000e+03	2.106000e+03
Lattitude	34857.0	-2.915878e+01	15.883671	-38.19043	-37.84690	-37.77659	-3.763499e+01	0.000000e+00
Longtitude	34857.0	1.118224e+02	60.912399	0.00000	144.72629	144.97410	1.450517e+02	1.455264e+02
Propertycount	34854.0	7.572888e+03	4428.090313	83.00000	4385.00000	6763.00000	1.041200e+04	2.165000e+04

After several research, we find that it will not make any sense if the minimum ‘BuildingArea’ is ‘0’ and the maximum ‘YearBuilt’ is ‘2106’. This research is also supported by the boxplot of these columns, as follows:



To handle these outliers, we decided to exclude any observations that have the minimum ‘BuildingArea’ == 0 and the maximum ‘YearBuilt’ >= 2023.

4. Feature Selection

In feature selection, we would like to remove these columns since it is not too necessary in this analysis by dropping these columns; ‘Address’, ‘Propertycount’, and ‘house_age’.

Thus, to store the cleaned dataset, we have stored it in this file “MELBOURNE_CLEANED_DATASET.csv” dataset.