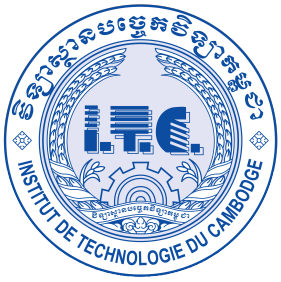

Land Price Analysis in Cambodia

Programming For Data Science

2022 - 2023

3rd year Engineering's Degree in Data Science
Department of Applied Mathematics and Statistics
Institute of Technology of Cambodia

Lecturer: Mr. CHAN Sophal(Course and TP)

Members of Group 03	Student ID
1. KOH Tito	e20200353
2. CHEA Makara	e20201131
3. AOV Keatmeng	e20201812
4. BUT Cheableng	e20200861
5. HOK Kimleang	e20200637

Contents

1 Abstract

There are many kinds of information on website which are the keys or references for people in vary ways. Some of those are about social, E-commerce, academic and financial. Most people search through website for educational purposes, and some are for business purposes. The information in each category is written in different formats and in different websites.In this project, our team focuses on data that we collected from other websites in order to analyze it on the website using specific library. Our main goal is to make a use from the data that we have into something useful. Land price analysis is a very good project for people who are interested in Real Estate and want to invest but they have no experience or idea about how it works. Here, we have included from the very start of the process of Land Price Analysis to the last past by showing graphs, details and some important libraries that we have used.

2 Introduction

This project can occur by having a dataset. Data is very important for all data analysists. It is like getting advantages from the data that they have and turn it into something that is useful for them in many ways. For this project, it is about Land Price Analysis, so the purpose of this might be about investment or know how investing in real estate works. There are many ways that people can tell about the price of the land. Some people can tell by asking owners and take note by themselves. Some are using technology as smart phone, laptop or other devices to seek on website to know the detail. In this project, we will try to use technology and mathematics to interpret the way data analysis works with data from the start to the end. Additionally, we will show the result of Land Price Analysis on the website which is way easy to see. Data is the main object that creates these kinds of techniques. Web scraping is the powerful method for data analytic to extract data from website and make it more useful. However, before scraping data from any websites, the knowledge for scrapping data step-by-step of manuscript is really importance. Knowing the title of the project gives the importance information to do web scraping. It can be for data analysis on some businesses or for study purpose. Web scraping can be easy or difficult depends on the structure or preparation that web developer or content management team created . Its can be more efficient since the technology has become more advanced and the self-learning is more popular for the new generation in order to obtain new skill by just learning from the internet. Web scraping has been used in various fields such as social science, business, and healthcare. However, despite its popularity, it can lead to the error or lack of a standard for data analysis. This can appear if the quality of the collected data is bad. Making a manuscript for web scraping is the best way and a practical guide to web scraping that will help researchers navigate the challenges and opportunities presented by this powerful tool.

3 Data Collection

Data collection refers to the process of gathering information or data from various sources and storing it for analysis, interpretation, and decision-making purposes. It involves systematically

collecting relevant and accurate data that is necessary for a specific purpose or research objective.

For the project in this report, it is about Land Price Analysis. For Data Collection, Land Price Analysis needs to have to some important information in order to analyze the Land Price. We chose "Real estate Cambodia" and "Khmer24" websites to scrap those data.

3.1 Methodology

Web scraping is the process of extracting data from website. There are some tools which help scraper to successfully obtain the data they want. There are several methodologies for web scraping such as Manual scraping, Automated scraping, API scraping and Hybrid scraping. For the main purpose in this report, Automated scraping is chosen to collect data. The language that is popular for all data scientist and also learner in Data science is Python. People can choose IDE or platform for writing a code as they want, but for the easiest way is to use Jupyter Notebook. Jupyter Notebook has a built-in support for popular data science libraries such as NumPy, Pandas, and Matplotlib, making it easier for data scientists to manipulate and visualize data and the code is written in cell which is convenient for coder to code and execute the code. Jupyter notebook also provides a lot of useful libraries such as Beautiful Soup and Requests, which provide easy-to-use tools for downloading and parsing HTML documents. These libraries allow scrapers to extract data from websites, navigate through web pages, and save scraped data in various formats such as csv file, excel file... etc. We are going to scrap data from realestate.com.khand khmer24. Before the particular data of each land, we have to get the specific URL or link of each land for the purpose of obtaining more information of the land. After getting those URL, we continue on scraping data by using loop in the code.

3.2 Data Extraction

The data set contains 6536 rows and 12 columns. Include the 12 features importance with meaning as following :

STT	Attribute	Meaning
1.	Price	The total price of land
2.	Size	The size of land area.
3.	Location	The location of land in particular area.
4.	Type	The property of land for sale or for rent.
5.	Title	Refers to hard title or soft title
6.	property ID	The identify the land code after set in the system.
7.	Original ID	The identify the land code before set in the system.
8.	Listed	The date of land sale
9.	Updated	The day update of land sale
10.	Road	The type of road that belong to the land sale
11.	Latitude	The location on the google map
12.	Longitude	The location measuring on the google map

Figure 1: The Attribute Meaning

We have enough dataset to analysis the price of land sale. The figure 2 is describe the value of the first 5 rows of dataset that have scrapped on the 2 websites.

	Price	Size	Location	Type	Title	Property ID	Original ID	Listed	Updated	Road	Latitude	Longitude
0	2624100.0	17494	Phnom Penh	Land for Sale	Hard Title	9168610	NaN	14-Jun-23	NaN	Ring Road	10.374895	112.344616
1	30990000.0	134727	Kandal	Land for Sale	Hard Title	5871599	NaN	12-Jun-23	NaN	Ring Road	10.374895	112.344616
2	206550.0	4590	Kampong Cham	Land for Sale	Soft Title	9185107	NaN	12-Jun-23	NaN	National Road	12.079261	104.716659
3	203100.0	4062	Siem Reap	Land for Sale	Soft Title	8571644	NaN	12-Jun-23	NaN	On Road	11.570873	104.833842
4	3096250.0	2477	Phnom Penh	Land for Sale	Hard Title	9168667	NaN	12-Jun-23	NaN	On Road	11.570873	104.833842

Figure 2: The data set of first 5 rows

4 Exploring Data Analysis

Exploratory Data Analysis (EDA)is a crucial step in the data analysis process, where analysts examine and explore the data to gain insights, identify patterns, detect anomalies, and formulate hypotheses. EDA involves a variety of techniques and methods to understand the underlying structure and characteristics of the data before applying more complex analysis or modeling techniques.

4.1 Data Cleaning

Data cleaning is the process of identifying and correcting or removing errors, inconsistencies, inaccuracies, or discrepancies in a dataset. It is a crucial step in the data preprocessing pipeline before conducting any analysis or modeling.
Data cleaning aims to ensure that the dataset is accurate, reliable, and suitable for analysis by addressing various issues that can arise in real-world data.

4.1.1 Checking Duplicated of Dataset

Duplicates can occur due to various reasons, such as data entry errors, system glitches, or merging data from multiple sources. So, we are going to find the duplicated data and then we will drop it out.

```
In [3]: #check duplicated values
        len(df[df.duplicated()])

Out[3]: 36
```

Figure 3: Checking duplicated values

As the result, We have 36 duplicated data. To avoid the accuracy of model, so we decide to drop.

4.1.2 Checking Missing Values

Missing values refer to the absence or lack of data in one or more variables or attributes of a dataset. Handling missing values is an important part of the data cleaning process as they can affect the accuracy and reliability of data analysis and modeling. Next, we will find the missing value of our dataset.

```
In [9]: #check missing values
        df.isnull().sum()

Out[9]: Price            0
        Size             0
        Location         0
        Type             0
        Title            0
        Property ID      0
        Original ID    3089
        Listed           0
        Updated       3089
        Road             0
        Latitude         0
        Longitude        0
        dtype: int64
```

Figure 4: Checking Missing Values

Because of the missing values are minimal and don't significantly impact the analysis, we may consider deleting the rows or variables with missing values. In addition the " Property ID " and " Original ID" have the lack of effect of the model then we are also drop it.

4.2 Summary Statistic

The summarizing statistics, analysts can gain a comprehensive understanding of the data, detect patterns and anomalies, make data-driven decisions, and effectively communicate insights. These statistics provide a concise summary that enables efficient data exploration, analysis, and decision-making across various domains and disciplines. In summarise statistic we need to some of importance statistics such as mean value, maximum value, minimum value , standard deviation and quartiles. In the dataset now we have only 7 features such as " Price " , " Size " ," Location" , " Road" ," Title" , " Latitude" and " Longitude". The target of us is to know the price per square of the land sale. Therefore, we are going to create one more feature is " Price per m2" which mean price per square is obtained from " Price / Size".

Out[24]:

	Price	Size	Location	Title	Road	Latitude	Longitude	Price per m2
0	2624100.0	17494	Phnom Penh	Hard Title	Ring Road	10.374895	112.344616	150.000000
2	206550.0	4590	Kampong Cham	Soft Title	National Road	12.079261	104.716659	45.000000
3	203100.0	4062	Siem Reap	Soft Title	On Road	11.570873	104.833842	50.000000
5	16999.0	100	Phnom Penh	Soft Title	On Road	11.624164	104.779325	169.990000
6	39999.0	108	Phnom Penh	Soft Title	On Road	11.562108	104.888535	370.361111

Figure 5: Create Price Per Square attribute

In this table is describe the statistic of numerical variable :

In [25]:
```
df.describe()
```

Out[25]:

	Price	Size	Latitude	Longitude	Price per m2
count	3.089000e+03	3.089000e+03	3089.000000	3089.000000	3089.000000
mean	6.229351e+06	3.063248e+05	11.807268	104.652609	1509.551910
std	9.318772e+07	1.595531e+07	1.000973	0.881446	19438.761981
min	1.000000e+00	1.000000e+00	2.839147	100.953227	0.000014
25%	3.000000e+04	2.000000e+02	11.483956	104.336879	38.000000
50%	1.050000e+05	7.770000e+02	11.567225	104.827196	150.000000
75%	5.000000e+05	3.706000e+03	11.819107	104.916489	448.275862
max	3.214227e+09	8.867496e+08	37.203511	127.018152	1000000.000000

Figure 6: Summarise Statistic

4.3 Data Visualization and Feature Selection

Data visualization is the process of representing data in a visual and graphical form to facilitate understanding, analysis, and communication of information. It involves using visual elements such as charts, graphs, maps, and other interactive visual representations to present data patterns, trends, and relationships. The next step we are going to look the bar graph of ” Location” frequency and Figure 10 describe the detail

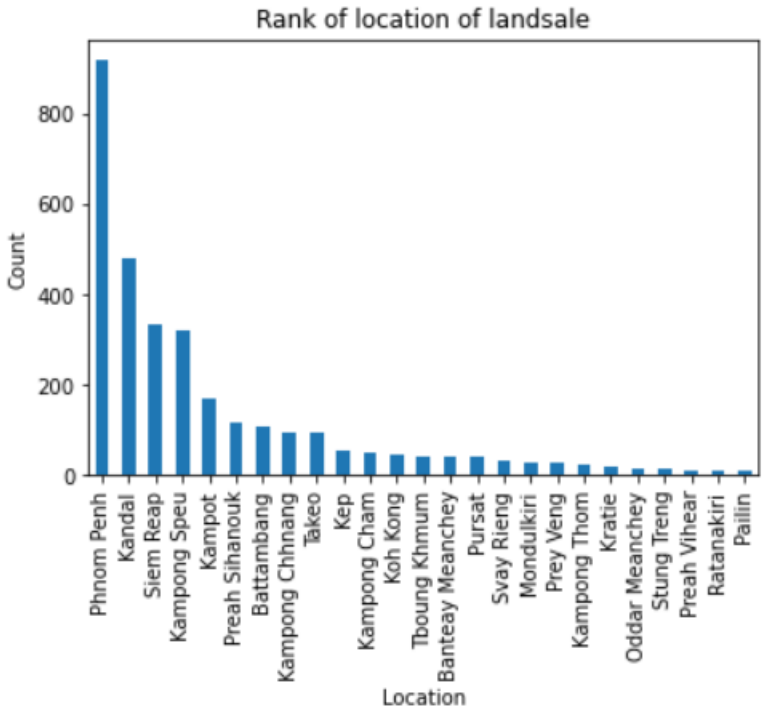

Figure 7: The Frequency of Location of Land Sale

```
In [21]: df['Location'].value_counts()
```

```
Out[21]: Phnom Penh          919
         Kandal              481
         Siem Reap           334
         Kampong Speu        322
         Kampot              170
         Preah Sihanouk      114
         Battambang          108
         Kampong Chhnang      96
         Takeo                95
         Kep                  56
         Kampong Cham         48
         Koh Kong             43
         Tboung Khmum         42
         Banteay Meanchey     42
         Pursat               39
         Svay Rieng           30
         Mondulkiri           28
         Prey Veng            26
         Kampong Thom         25
         Kratie               17
         Oddar Meanchey       13
         Stung Treng          12
         Preah Vihear         10
         Ratanakiri           10
         Pailin                9
         Name: Location, dtype: int64
```

Figure 8: The Details of Location Count

The next step we want to know the how many the Title of dataset

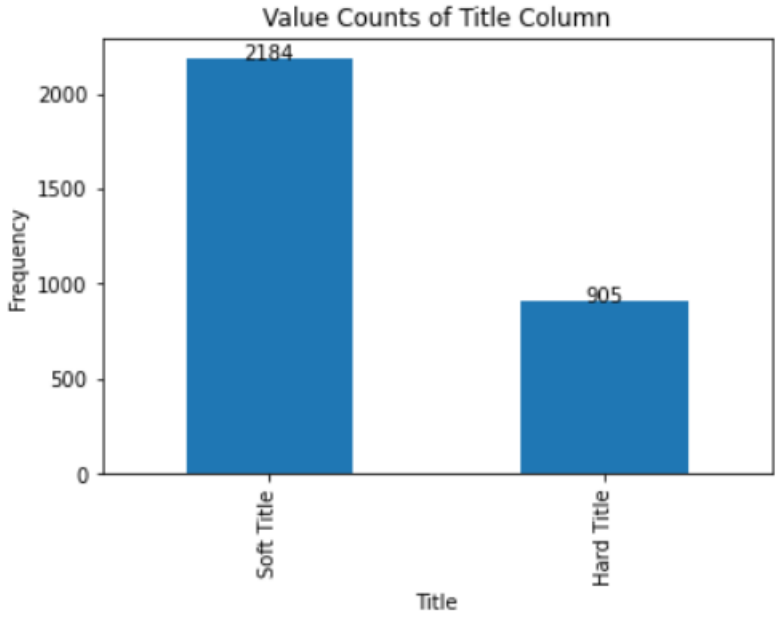

Figure 9: Title count

4.3.1 Data Transformation

We are going to transform the categorical variable into numerical variable to train the dataset. Firstly, we encode the ” Location” group by their average price per square.

Location	After Encoding	Average price per Square
Kep	0	237.337756
Mondulkiri	1	272.256617
Prey Veng	2	408.564680
Koh Kong	3	518.134338
Banteay Meanchey	4	560.496189
Preah Sihanouk	5	599.299841
Battambong	6	649.329908
Takeo	7	710.164795
Kampong Thom	8	903.259803
Kandal	9	907.812419
Pailin	10	955.112481
Preah Vihear	11	975.002589
Svay Rieng	12	1061.636240
Kampong Chhang	13	1272.118403
Kampong Speu	14	1282.157065
Phnom Penh	15	1518.401282
Kampong Cham	16	1703.233593
Pursat	17	1780.320631
Kampot	18	1828.186387
Tboung Khmum	19	2120.939241
Stung Treng	20	2312.193733
Kratie	21	2493.377252
Ratanakiri	22	3507.553122
Siem Reap	23	3680.810758
Oddar Meanchey	24	4173.304473

Figure 10 : Transforming Data for Location

For other categorical variable, we also transform it to numerical variable

```
#transform Road and Title into numerical data
df.loc[(df.Title == 'Hard Title'), 'Title'] = 2
df.loc[(df.Title == 'Soft Title'), 'Title'] = 1
print(df['Title'].unique())

df.loc[(df.Road == 'National Road'), 'Road'] = 3
df.loc[(df.Road == 'Ring Road'), 'Road'] = 2
df.loc[(df.Road == 'On Road'), 'Road'] = 1
print(df['Location'].unique())
```

```
[2 1]
[15 16 23 7 9 19 14 13 6 4 8 18 5 21 3 17 24 10 2 11 0 12 20 22 1]
```

Figure 11: Other Encoding Categorical Variable

4.3.2 Checking Outlier

As we know, an outlier is an observation or data point that significantly deviates from the majority of the data in a dataset. It is an observation that lies an abnormal distance away from other similar observations. Here some coding that we check the outlier and then drop the outlier.

```
#Get rid of outliers
# Loop through each numeric column and handle outliers using the IQR method
for col in df.select_dtypes(include=np.number).columns:
    # Calculate Q1, Q3, and IQR for the current column
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1

    # Determine the acceptable range of values
    lower_range = Q1 - 1.5*IQR
    upper_range = Q3 + 1.5*IQR

    # Identify and handle outliers by replacing them with the maximum non-outlier value
    outliers = df[(df[col] < lower_range) | (df[col] > upper_range)]
    df.loc[(df[col] < lower_range) | (df[col] > upper_range), col] = df[col][~((df[col] < lower_range) | (df[col] > upper_range))].max()

    print("Outliers in Column", col, ":\n", outliers)
```

Figure 12: Checking Outlier

```
Outliers in Column Price :
             Price    Size Location Title Road   Latitude   Longitude  \
0       2624100.0   17494       15     2    2  10.374895  112.344616
15      4840000.0   22000        9     1    3  11.846601  104.956074
29      1280000.0   40000        9     2    1  11.554119  104.910936
49      7382050.0  147641       14     1    1  11.456021  104.953488
55      2000000.0   20000        9     1    1  11.539648  104.929398
...           ...     ...      ...   ...  ...        ...         ...
6451   20592000.0    9360       15     1    1  11.477527  104.910153
6466    3600000.0    2000       15     1    1  11.535358  104.944916
6498    4200000.0    1258       15     1    1  11.583185  104.877603
6512  149745045.0   14997       14     1    1  11.704297  104.700926
6522    2250000.0  150000       13     1    1  11.923620  104.555745

      Price per m2
0        150.00000
15       220.00000
29        32.00000
49        50.00000
55       100.00000
...            ...
6451    2200.00000
6466    1800.00000
6498    3338.63275
6512    9985.00000
...
6512  1200000.0  8833       14     1    1  11.704297  104.700926    9985.00000
6525   700000.0   100       14     1    1  11.667600  104.685062    7000.00000

[456 rows x 8 columns]
```

Figure 13: The results of the outlier that is dropped

4.3.3 Headmap

Heatmaps are frequently used to visualize correlation matrices. A correlation matrix is a square matrix that shows the correlation coefficients between multiple variables. Each cell in the matrix represents the correlation between two variables, and the intensity of the color in the heatmap corresponds to the strength of the correlation.

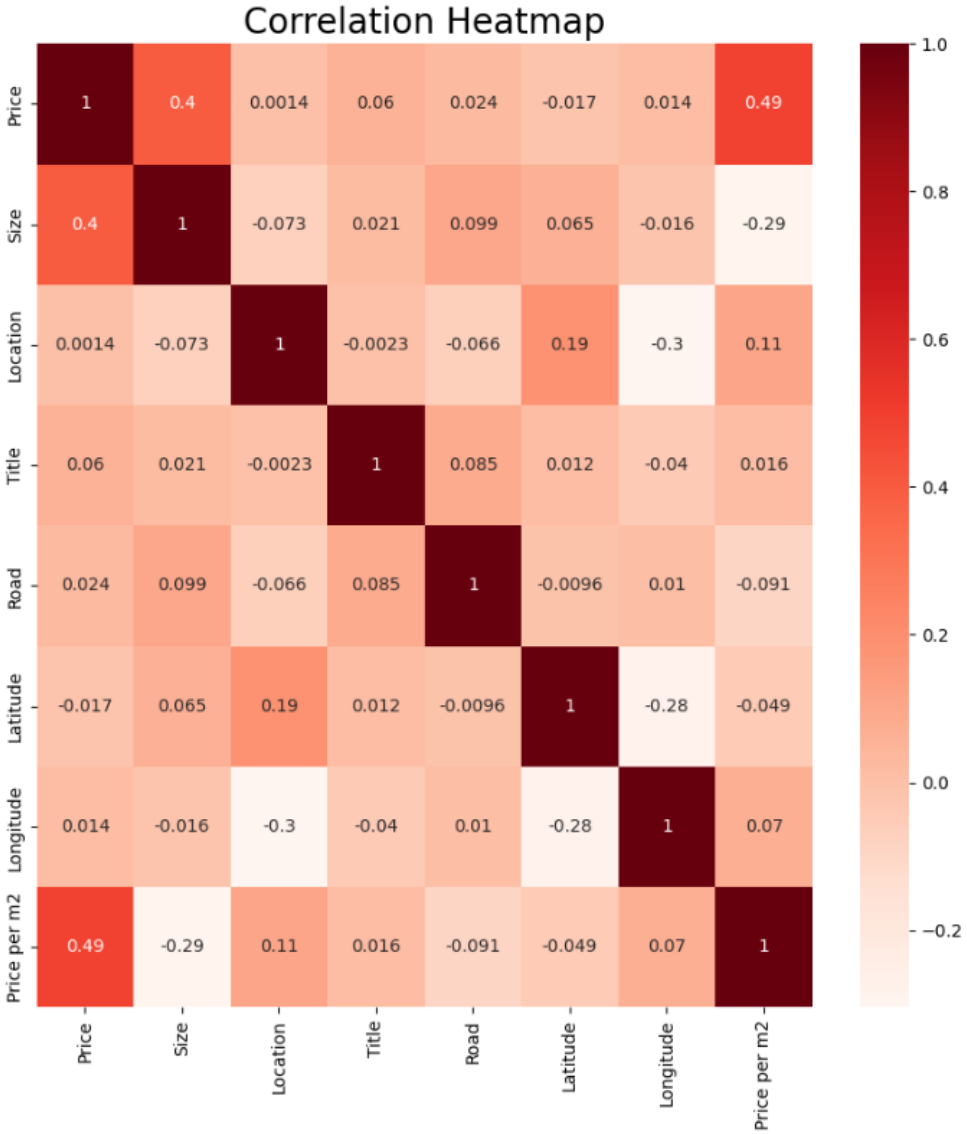

Figure 14: The correlation Heatmap

• In our project we are doing the get the insight of the price per square of the land sale. The graphs below is shown the pair plot graph of price per square and other attributes.

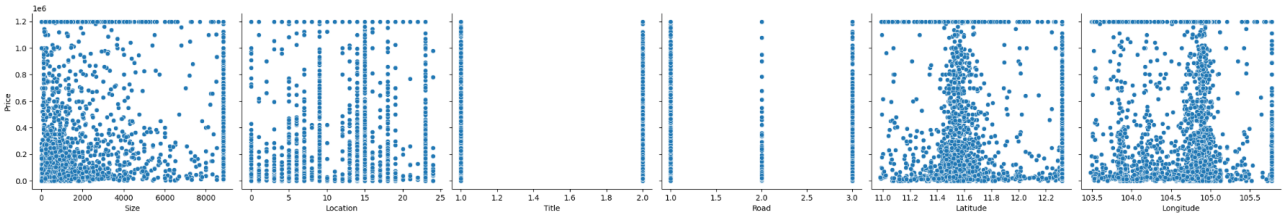

Figure 15: Pair Plot visualization dataset

5 Model Construction

Model selection is a critical step in the machine learning workflow, where the most suitable algorithm or model is chosen for a given task or problem. The goal is to select a model that will generalize well to unseen data and provide accurate predictions or desired outcomes. Model selection are importance as following

1. Problem Understanding: Gain a thorough understanding of the problem you are trying to solve. Define the task, the available data, and the expected output. Determine whether it is a classification, regression, clustering, or other type of problem. Understanding the problem will guide your choice of models.

2. Available Data: Assess the characteristics of your data, including the number of samples, the number of features, and their types (continuous, categorical, text, etc.). Consider whether the data has missing values, outliers, or class imbalance. Some models are better suited for specific types of data, so understanding the data properties helps in selecting appropriate models.

3. Model Complexity: Consider the complexity of the problem and the model's ability to capture the underlying patterns. A simple model may be sufficient for a straightforward problem, while a complex problem may require more sophisticated models with higher capacity to learn intricate relationships.

4. Model Assumptions: Understand the assumptions made by different models. Some models assume linearity, independence, or normality of the data, while others are more flexible and can handle nonlinear relationships or complex interactions. Ensure that the model's assumptions align with your data and problem requirements.

5. Performance Metrics: Determine the evaluation metrics that are relevant to your problem. Accuracy, precision, recall, F1 score, mean squared error (MSE), or area under the receiver operating characteristic curve (AUC-ROC) are examples of common metrics. Different models may perform differently based on these metrics, so choose the ones that align with your goals.

6. Model Interpretability: Consider the interpretability of the model. Some models, such as linear regression or decision trees, provide transparent interpretations of the relationships between features and outcomes. Other models, like deep neural networks, may be more opaque in their decision-making process. Depending on your needs, you may prioritize interpretability or favor predictive performance.

7. Scalability and Efficiency: Assess the computational requirements and scalability of the models. Some models may be computationally expensive or memory-intensive, limiting their use for large datasets or real-time applications. Consider the trade-off between model complexity and computational resources available.
8. Cross-validation and Hyperparameter Tuning: Use techniques like cross-validation to estimate the performance of different models on unseen data. Perform hyperparameter tuning to optimize model performance by adjusting parameters that affect the model's behavior. This helps in fine-tuning the models and selecting the best-performing configuration.
9. Prior Knowledge and Domain Expertise: Leverage any prior knowledge or domain expertise you have about the problem. Some models may be more suitable for specific domains or have been successfully applied in similar contexts. Consulting domain experts can provide valuable insights and guide your model selection process.

5.1 Multiple Linear Regression

Generally, Multiple linear regression is a statistical technique used to model the relationship between a dependent variable and two or more independent variables.

In our dataset we use all the features to fit the multiple linear regression model. For all the model we use the sklearn library in python the build the model. Next, we are going to train the dataset into multiple linear regression.

```
In [63]: #Split the dataset into training and testing sets
         X = df[['Size','Location','Title','Road','Latitude','Longitude','Price per m2']]
         y = df['Price']
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.05, random_state=0)

         #Select a machine learning model and train it on the training set
         from sklearn.linear_model import LinearRegression
         model = LinearRegression()
         model.fit(X_train, y_train)

Out[63]: LinearRegression()
```

Figure 16: Training Dataset into Multiple Linear Regression.

To compare the accuracy of the model. The Mean Square Error (MSE) and R square are things that are really importance. So, We will find MSE and R^2 between the dataset that we predict and dataset that we test.

```python
#Evaluate the model's performance on the testing set
from sklearn.metrics import mean_squared_error, r2_score
y_pred = model.predict(X_test)
print('Mean squared error:', mean_squared_error(y_test, y_pred))
print('R^2 score:', r2_score(y_test, y_pred))
```

```
Mean squared error: 62220948944.69456
R^2 score: 0.6552455889592035
```

Figure 17: The MSE and R^2 for Multiple Linear Regression.

5.2 Polynomial Regression

In this section, we consider the polynomial regression with degree 2. The polynomial model provide the significant model with is non-linear. The Figure below is the training dataset to polynomial model ,MSE and R^2.

In [26]:
```python
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import make_pipeline

X = df[['Size','Location','Title','Road','Latitude','Longitude','Price per m2']]
y = df['Price']

# Generate polynomial features
degree = 2  # Set the degree of the polynomial
poly_features = PolynomialFeatures(degree=degree)
X_poly = poly_features.fit_transform(X)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_poly, y, test_size=0.05, random_state=0)

# Create and train the polynomial regression model
model = make_pipeline(PolynomialFeatures(degree=degree), LinearRegression())
model.fit(X_train, y_train)
```

Out[26]:
```
Pipeline(steps=[('polynomialfeatures', PolynomialFeatures()),
                ('linearregression', LinearRegression())])
```

Figure 18: Training dataset to polynomial model

```
Mean Squared Error (MSE): 24024419932.299725
R-squared (R^2): 0.8668852712015251
```

Figure 19: The MSE and R^2 of Polynomial Model

5.3 Support Vector Machine (Linear SVM)

For obtain the powerful model we try to find the other model. The part we cover on the Support vector Machine (SVM) model for this dataset. The figure 20 is shown the training dataset and its MSE , also R squre

In [65]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

data = df

# Separate independent and dependent variables
X = data[['Size', 'Price per m2', 'Latitude', 'Longitude', 'Location', 'Title', 'Road']]
y = data['Price']

# Convert categorical variables to numerical representation (one-hot encoding)
X_encoded = pd.get_dummies(X, columns=['Location', 'Title', 'Road'])

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.2, random_state=42)

# Perform feature scaling on numerical variables
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train[['Size', 'Price per m2', 'Latitude', 'Longitude']])
X_test_scaled = scaler.transform(X_test[['Size', 'Price per m2', 'Latitude', 'Longitude']])

# Create an instance of the SVM model
svm_model = SVR(kernel='linear', C=1.0)

# Fit the model to the training data
svm_model.fit(X_train_scaled, y_train)

# Use the trained model to make predictions on the test set
y_pred = svm_model.predict(X_test_scaled)

price_bins = [0, 500000, 1000000, 1500000, 2000000, 3000000, float('inf')]
labels = [0, 1, 2, 3, 4, 5]

y_pred_class = pd.cut(y_pred, bins=price_bins, labels=labels)
y_test_class = pd.cut(y_test, bins=price_bins, labels=labels)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")

# Calculate accuracy score for classification
accuracy = accuracy_score(y_test_class, y_pred_class)
print(f"Accuracy Score: {accuracy}")
```

```
Mean Squared Error: 241390587919.41534
Accuracy Score: 0.7411003236245954
```

Figure 20: Training dataset , MSE and R square

5.4 Decision tree

As we know, a Decision Tree model is a machine learning algorithm used for both classification and regression tasks. It builds a tree-like model of decisions and their possible consequences based on the features or independent variables in the dataset. Now we are doing with it.

In [67]:
```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

data = df

# Preprocessing: Separating features and target variable
X = data[['Size', 'Price per m2', 'Location', 'Title', 'Road', 'Latitude', 'Longitude']]
y = data['Price']

# Splitting the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Decision Tree
decision_tree_model = DecisionTreeRegressor()
decision_tree_model.fit(X_train, y_train)
y_pred_dt = decision_tree_model.predict(X_test)
mse_dt = mean_squared_error(y_test, y_pred_dt)
rmse_dt = mse_dt ** 0.5
print('Decision Tree MSE:', mse_dt)
print('Decision Tree RMSE:', rmse_dt)

# Calculate R-squared (coefficient of determination)
r2 = r2_score(y_test, y_pred_dt)
print(f"R-squared (R2): {r2}")
```

```
Decision Tree MSE: 45938264749.69073
Decision Tree RMSE: 214332.13653041096
R-squared (R2): 0.754456233384935
```

Figure 21: Decision Tree Model and its accurary

5.5 Random Forest

In general, The Random Forest model is an ensemble learning method that combines multiple Decision Trees to create a powerful machine learning algorithm. It is widely used for both classification and regression tasks and offers improved accuracy and robustness compared to individual Decision Trees. The algorithm of the random forest is following:

```
data = df

# Preprocessing: Separating features and target variable
X = data[['Size', 'Price per m2', 'Location', 'Title', 'Road', 'Latitude', 'Longitude']]
y = data['Price']

# Splitting the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

#random forest
random_forest_model = RandomForestRegressor()
random_forest_model.fit(X_train, y_train)
y_pred_rf = random_forest_model.predict(X_test)
mse_rf = mean_squared_error(y_test, y_pred_rf)
rmse_rf = mse_rf ** 0.5
print('Random Forest MSE:', mse_rf)
print('Random Forest RMSE:', rmse_rf)

# Calculate R-squared (coefficient of determination)
r2 = r2_score(y_test, y_pred_rf)
print(f"R-squared (R2): {r2}")
```

```
Random Forest MSE: 27007481283.553116
Random Forest RMSE: 164339.53049571827
R-squared (R2): 0.8556428128645379
```

Figure 22: The Random Forest Model

5.6 Gradient Boosting

Formally, Gradient Boosting is a powerful machine learning technique that combines weak learners (usually decision trees) into an ensemble model. It iteratively builds a strong predictive model by minimizing a loss function through gradient descent. Gradient Boosting is known for its high predictive accuracy and is widely used for both regression and classification tasks.

In [70]:
```
import numpy as np
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_squared_error, r2_score
```

In [71]:
```
# Assuming you have your input features stored in X and target variable stored in y
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [72]:
```
# Initialize the model with desired hyperparameters
model = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42)
```

In [73]:
```
# Fit the model to the training data
model.fit(X_train, y_train)
```

Out[73]:
```
GradientBoostingRegressor(random_state=42)
```

```
Mean Squared Error: 24679674159.752728
R^2 score: 0.8680851315338598
```

Figure 23: The Gradient Boosting

5.7 Model Comparison

In the previous section, we try to find the best model to fit the dataset that we have scraped on the website. As the result, We see the polynomial model is the provide the less mean square error and higher R-square. We can say the form of polynomial of degree to is defined by:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \beta_{n+1} X_1^2 + \beta_{n+2} X_1 X_2 + \ldots + \beta_{2n} X_n^2 + \varepsilon$$

where

- $Y$is the dependent variable or the target variable.
- $X_1, X_2, ..., X_n$are the independent variables or predictors.
- β_0 is the intercept term.
- $\beta_1, \beta_2, \ldots, \beta n$are the coefficients associated with the linear terms of the independent variables.
- $\beta n + 1, \beta_{n+2}, \ldots, \beta_{2n}$ are the coefficients associated with the quadratic terms and cross-product terms of the independent variables.
- $X_1^2, X_1 X_2, ..., X_n^2$represent the squared terms and cross-product terms of the independent variables.
- ε represents the error term or the unexplained portion of the dependent variable.

6 Conclusion

The data analysis is also importance, we gain valuable insights into patterns, trends, and correlations that would otherwise go unnoticed. This information can be used to make informed decisions, identify areas for improvement, and optimize performance. Additionally, data analysis allows us to measure the effectiveness of our strategies and initiatives, enabling us to make data driven decisions that are grounded in evidence. Ultimately, the ability to analyze data is a powerful tool that can help company, organizations and individuals achieve their goals and improve their outcomes. As such, it is essential that we continue to invest in data analysis and leverage its power to drive innovation and progress.y doing this project, we can see that the Price, Size, Location and Title are popular for people. People can see the differences between the land to land. Plus, they can predict the price of the land by trying the model that we have trained. This is very helpful for investor to invest on real estate by seeing the graph with different types of comparison. In the project above the polynomial Model in most powerful for fitting the non-linearity dataset.

7 Reference

1. A Comparative Study on Web Scraping (2015), De Sirisuriya, SCM available at http://ir.kdu.ac.lk/handle/345/1051
2. Land Price Forecasting Research by Macro and Micro Factors and Real Estate Market Utilization Plan Research by Landscape Factors: Big Data Analysis Approach (2021), Sang-Hyang Lee,Jae-Hwan Kim and Jun-Ho Huh available at https://www.mdpi.com/2073-8994/13/4/616
3. Inter-metropolitan land price characteristics and pattern in the Beijing-Tianjin-Hebei urban agglomeration, China(September 2021) , Pengfei An,Can Li ,Yajing Duan,Jingfeng Ge,Xiaomiao Feng available at https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0256710
4. Lessons from web scraping coroners' Prevention of Future Deaths reports(January, 2023), Qingyang Zhang , Georgia C Richards available at https://pubmed.ncbi.nlm.nih.gov/36688377/
5. Evaluating and comparing web scraping tools and techniques for data collection (October 2022), Shqipe Sejdiu , Vesa Morina , available at https://www.researchgate.net/publication/369114323_Evaluating_and_comparing_web_scraping_tools_and_techniques_for_data_collection
6. Web Scrapping - Data Scrapper (March 2020), Rizul Sharma KIIT University, available at https://www.researchgate.net/publication/342011184_Web_Data_Scraping
7. Web Scraping in the Statistics and Data Science Curriculum: Challenges and Opportunities (2021), Mine Dogucuaand , Mine Çetinkaya-Rundel available at https://www.tandfonline.com/doi/epdf/10.1080/10691898.2020.1787116?needAccess=true&role=button
8. Land Price Forecasting Research by Macro and Micro Factors and Real Estate Market Utilization Plan Research by Landscape Factors: Big Data Analysis Approach (2021), Sang-Hyang Lee,Jae-Hwan Kim andJun-Ho Huh available at https://www.mdpi.com/2073-8994/13/4/616
9. A Step-by-Step Guide to Web Scraping with Python and Beautiful Soup(2023) by Aryan Garg avaible at https://www.kdnuggets.com/2023/04/stepbystep-guide-web-scraping-p
html
10. Web Data Extraction Approach for Deep Web using WEIDJ (2019), . Published by Elsevier B.V at https://www.researchgate.net/publication/334784343_ScienceDirect_Web_Data_Extraction_Approach_for_Deep_Web_using_WEIDJ
11. Applied Linear Regression Established by WALTER A.SHEWHART and SAMMUEL S.WILKS, fourth Edition.
12. Regression Modeling Strategies by Frank E.Harrell,Jr.Second Editionhttps://www.springer.com/series/692