REAL-ESTATE INVESTMENT ANALYSIS

Ruchitha Reddy Koluguri
Software Engineering
Computer Engineering
Department San Jose State University *San Jose, California*
ruchithareddy.koluguri@sjsu.edu

Suresh Ravuri
Software Engineering
Computer Engineering
Department San Jose State University *San Jose, California*
suresh.ravuri@sjsu.edu

Harshavardhan Valmiki
Software Engineering
Computer Engineering
Department San Jose State University *San Jose, California*
harshavardhanraghavendra.valmiki @sjsu.edu

Sri Vinay Appari
Software Engineering
Computer Engineering
Department San Jose State University *San Jose, California*
srivinay.appari@sjsu.edu

***Abstract*—This project aims to guide real estate investors towards properties that promise the highest returns on investment (ROI) by ensuring that combined Homeowner Association (HOA) fees and mortgage payments fall below rental income. Employing a strategic analysis, we identify properties across a spectrum of desirability based on current and predicted market conditions. Utilizing advanced data preparation techniques, including feature transformation and amalgamation with enriched datasets, our study meticulously crafts a data narrative that highlights the critical features influencing property values.**

Through the application of machine learning algorithms, we segment properties into categories of desirability, forecast future price movements, and provide actionable insights into the real estate market. Our analysis extends to the exploration of latent variables, such as location-based features, enhancing predictive accuracy and investment strategies. This comprehensive approach allows us to offer tailored advice to investors, retirees, and buyers, focusing on assets that offer both immediate rental yields and long-term capital appreciation.

Our narrative is rooted in a deep understanding of the market dynamics, underscored by rigorous model explainability and validation against industry-standard metrics. By forecasting house price trends and identifying high-potential investment opportunities, we empower stakeholders to make informed decisions, capitalizing on the ever-evolving real estate landscape for optimal financial outcomes.

Keywords—Data Preparation, feature Transformation, Machine Learning Algorithms, Return on Investment, Homeowner Association, Classification, Regression, Data Visualization.

INTRODUCTION

In the fast-changing world of real estate investment, maximizing returns through strategic property selection is key. This paper explores how to identify residential properties that not only generate positive cash flow from rental income surpassing HOA fees and mortgage payments but also have strong potential for value appreciation. Using machine learning and data analytics, we forecast property values and rental trends to aid investor decisions.

We've developed a data-driven method to classify properties by their investment potential into categories ranging from least to most desirable. This involves combining various data sources and applying advanced analytics to uncover the key factors influencing property value and rental yield.

Our approach includes detailed data preparation and the application of machine learning to segment the market and predict price movements. We also examine hidden factors, like location, that significantly affect property valuation.

This study aims to provide a straightforward guide for investors, retirees, and property buyers to navigate real estate investment, focusing on both immediate income and long-term growth. Our research offers insights into making informed investment choices in the competitive real estate market.

DATA NARRATIVE

Dataset:1

For the analysis presented in this study, we utilized a comprehensive dataset capturing a wide range of attributes relevant to residential real estate investments. The dataset comprises the following attributes for each property:

Attributes: Rank, Property ID, Address, Latitude, Longitude, Price, Currency, Bathrooms, Bedrooms, Area, Land Area, Zestimate (Zillow's estimated market value), Rent Zestimate (Zillow's estimated rent price), Days on Zillow, Sold Date, Is Zillow Owned, Image, Listing Type, Status Text, Broker Name, Input, Property URL, Listing URL

These attributes were selected for their potential impact on investment returns, providing a detailed snapshot of each property's characteristics, market positioning, and financial metrics. This rich dataset serves as the foundation for our subsequent data preparation, feature engineering, and machine learning analyses.

Dataset:2

For the secondary dataset integrated into our analysis, we focused on additional attributes that provide deeper insights into the physical characteristics and amenities of each property,

as well as more granular details on location and sale price. The dataset includes the following attributes:

Attributes: Year Built, Stories, Num Bedrooms (Number of Bedrooms), Full Bathrooms, Half Bathrooms, Livable SqFt (Livable Square Feet), Total SqFt (Total Square Feet), Garage Type, Garage SqFt (Garage Square Feet), Carport SqFt (Carport Square Feet), Has Fireplace, Has Pool, Has Central Heating, Has Central Cooling, House Number, Street Name, Unit Number, City, Zip Code, Sale Price

These attributes were carefully chosen to augment the primary dataset by providing a more comprehensive understanding of each property's features, from construction details to specific amenities that could influence investment attractiveness and property valuation.

METHODOLOGY

DATA PREPARATION: Data preparation is a vital stage in data analysis, where raw data is cleaned, transformed, and organized for further analysis or modeling. This process involves correcting errors, handling duplicates and missing values, and dealing with outliers during the cleaning phase. Transformation includes steps like normalization, scaling, and encoding to make the data suitable for analysis. The data is then organized, potentially through aggregation or partitioning, to better suit analytical needs. This thorough preparation ensures the data's quality and relevance, enhancing the accuracy and effectiveness of any subsequent analysis or modeling.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a process used to understand and summarize the main characteristics of a dataset, often visually, before formal modeling or hypothesis testing. It involves examining the distribution of data, identifying patterns, detecting anomalies, and investigating relationships between variables using statistical summaries and visualization techniques. EDA is crucial for gaining insights into the data's underlying structure and informing the choice of appropriate models and analysis techniques.

EDA on Dataset:1

Data Cleaning:

Data cleaning is a critical preprocessing step in the data analysis process, aimed at identifying and correcting (or removing) errors and inconsistencies in the data to improve its quality and reliability for analysis and modeling. This process is essential because the accuracy and outcomes of any data-driven method, such as machine learning models, are heavily dependent on the quality of the data used.

Dropped 'sold_date' and 'currency' columns as they are empty.

Handling Missing Values:

Fill missing values in 'land_area' and 'area' with their respective medians

```
# Fill missing values in 'land_area' and 'area' with their respective medians
data['land_area'] = data['land_area'].fillna(data['land_area'].median())
data['area'] = data['area'].fillna(data['area'].median())
# Save the cleaned dataset
data.to_csv('/content/drive/MyDrive/ML-Spring-2024/team-GeekSquad/Data/midterm-dataset/updated_Midterm-Dataset-Real-estate.csv', index=False)

print("Missing values in 'land_area' and 'area' filled with median values.")
```

```
Missing values in 'land_area' and 'area' filled with median values.
```

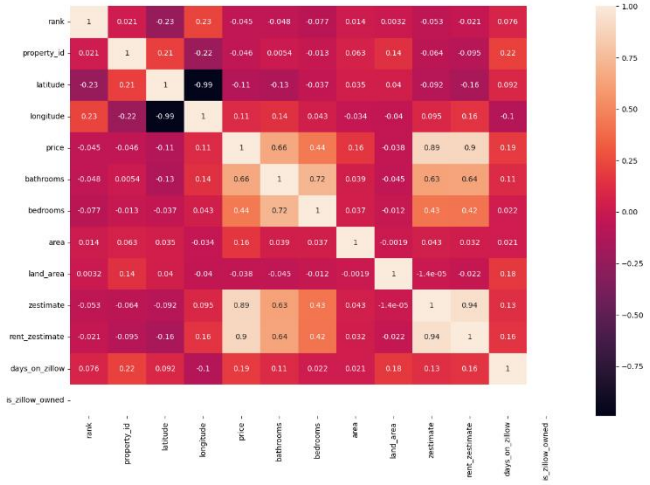
Fill missing values for 'bathrooms' and 'bedrooms' with the median, as they are typically discrete values

Removed all null values:

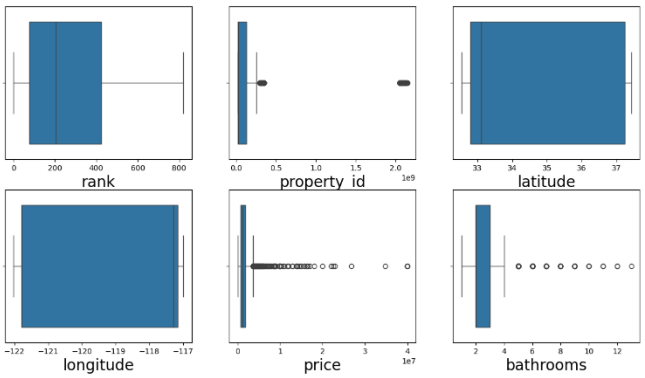
```
data.isnull().sum()
```

```
rank                0
property_id         0
address             0
latitude            0
longitude           0
price               0
bathrooms           0
bedrooms            0
area                0
land_area           0
zestimate           0
rent_zestimate      0
days_on_zillow      0
is_zillow_owned     0
image               0
listing_type        0
status_text         0
broker_name         0
input               0
property_url        0
Street              0
City                0
State               0
ZIP                 0
dtype: int64
```

Plotting and Correlation of Dataset:1



Box Plot of Dataset:1



EDA on Dataset:2

Data Cleaning:

Dropped ‘stories’, ‘carport_sqrt’,'livable_sqft','house_number', 'unit_number', 'half_bathrooms','garage_type','garage_sqft','has_pool', 'has_fireplace', 'has_central_heating', 'has_central_cooling'

Handling null values:

```
dataset2.isnull().sum()/dataset2.shape[0] * 100
```

```
year          0.000000
bedrooms      0.000000
bathrooms     0.000000
area          0.000000
zipcode       0.000000
price         0.000000
address       0.000000
dtype: float64
```

Removed all null values:

```
dataset2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42703 entries, 0 to 42702
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   year       42703 non-null  int64
 1   bedrooms   42703 non-null  int64
 2   bathrooms  42703 non-null  int64
 3   area       42703 non-null  int64
 4   zipcode    42703 non-null  int64
 5   price      42703 non-null  float64
 6   address    42703 non-null  object
dtypes: float64(1), int64(5), object(1)
memory usage: 2.3+ MB
```

Plotting and Correlation of Dataset:2

DATA AMALGAMATION

Dataset 1: Given by the professor

Dataset 2: Sourced from Zillow

Dataset 3: Scraped Dataset

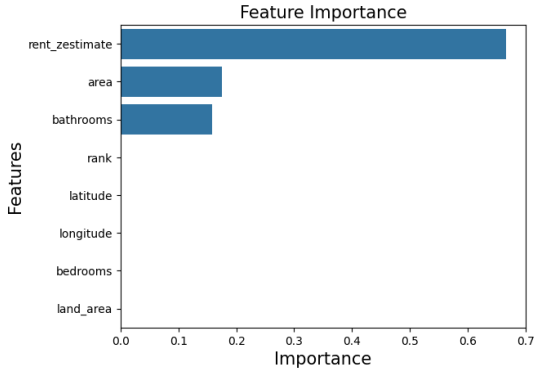
First Amalgamation:

The first amalgamation combines attributes from both Dataset 1 and Dataset 2, enhancing the dataset's richness and offering a more comprehensive view for analysis. This amalgamated dataset includes the following columns:

Attributes: Rank, Property ID, Address, Latitude, Longitude, Price, Bathrooms, Bedrooms, Area, Land Area, Zestimate (Zillow's estimated market value), Rent Zestimate (Zillow's estimated rent price), Days on Zillow, Is Zillow Owned, Image, Listing Type, Status Text, Broker Name, Input, Property URL, Street, City, State, Zip code

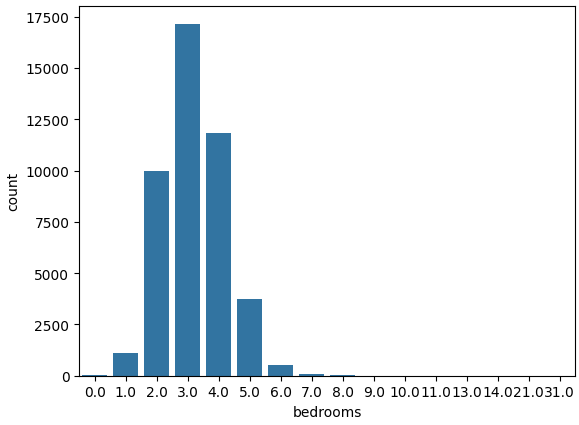
FEATURE IMPORTANCE

In this analysis, we assessed the importance of various features in predicting real estate investment outcomes using a Decision Tree Classifier. The model, set with specific criteria, was trained on a dataset incorporating property-related features. After training, feature importances were extracted to identify which attributes significantly influence investment decisions. A visualization of these importances was created with a bar plot, ranking each feature by its impact. This approach provides valuable insights into key property aspects to consider, guiding more informed investment strategies.

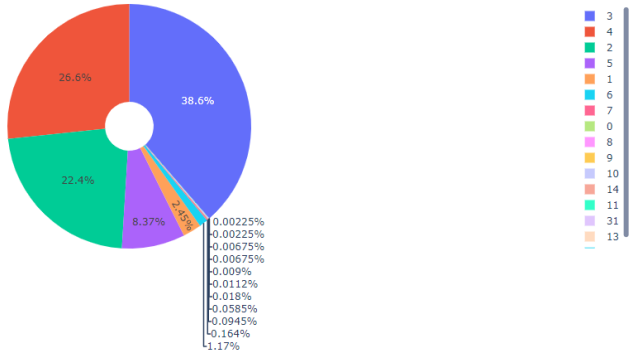


Understanding Market Trends: The Popularity of 3-Bedroom Houses

The analysis of the property data reveals a distinct trend in the real estate market: 3-bedroom houses are the most common preference among buyers. The data shows a higher count of 3-bedroom properties, indicating their popularity and potential as an investment focus. Additionally, the relationship between the number of bedrooms and property prices suggests that most buyer interest peaks for homes that combine affordability with sufficient living space, typically around the 400k to 500k price range. The average price of interest for prospective buyers hovers around 600k, providing a clear target for investors looking to cater to market demand.



Bedrooms Distribution:



Muller Loop for Classification

Classification can be used to predict whether a property is likely to be expensive or affordable. For example, the model can be trained to classify properties as either high-end or low-end based on their features. This information can help homeowners and property investors make informed decisions about whether to invest in a particular property or not. In a comparative study of machine learning models for classifying real estate properties as high-end or low-end based on their features, the Random Forest classifier outperformed others with an accuracy of 92.8%. This suggests its strength in handling complex, multi-featured datasets common in the real estate market. Other models like Decision Trees, AdaBoost, and Neural Networks also showed promising results with accuracies around 88% to 89.6%, indicating their potential usefulness in property valuation. Linear SVM and Nearest Neighbors yielded respectable performances, while RBF SVM lagged behind, possibly due to the non-linear nature of the data.

```
metrics_df = muller_classification(X_train, X_test, y_train, y_test)
```

```
Nearest Neighbors
Classifier = Nearest Neighbors, Score (test, accuracy) = 84.80,
Linear SVM
Classifier = Linear SVM, Score (test, accuracy) = 86.80,
RBF SVM
Classifier = RBF SVM, Score (test, accuracy) = 47.20,
Decision Tree
Classifier = Decision Tree, Score (test, accuracy) = 88.00,
Random Forest
Classifier = Random Forest, Score (test, accuracy) = 92.80,
Neural Net
Classifier = Neural Net, Score (test, accuracy) = 89.20,
AdaBoost
Classifier = AdaBoost, Score (test, accuracy) = 89.60,
Naive Bayes
Classifier = Naive Bayes, Score (test, accuracy) = 86.00,
Best --> Classifier = Random Forest, Score (test, accuracy) = 92.80
```

Overall, these models offer valuable insights for stakeholders making investment decisions, with Random Forest being the most reliable for this particular task.

Muller Loop for Regression

Regression can be used to predict the exact price of a property based on its features. For example, the model can be trained to predict the price of a house based on its square footage, location, and other features. This information can assist real estate agents and property investors in accurately pricing properties based on their features.

In the domain of real estate, regression models are employed to predict property prices with precision. Each model is trained using features like square footage and location to estimate a house's market value, aiding agents and investors in pricing decisions. Among the tested regressors, the RandomForest Regressor surfaced as the most effective, scoring the highest in accuracy at 75.55%. This model outperformed others including Linear Regression, MLP Regressor, and Gradient Boosting Regressor, as well as KNeighbors and SGD Regressors. Notably, the KernelRidge Regressor performed poorly with a negative accuracy score, indicating it was not suitable for this dataset. The RandomForest Regressor's strong performance suggests it is the most reliable for predicting house prices within the evaluated set.

```
metrics_df = muller_regression(X_train, X_test, y_train, y_test)
```

```
Regressor = Linear Regression, Score (test, accuracy) = 45.41,
Regressor = MLP Regressor, Score (test, accuracy) = 46.68,
Regressor = RandomForest Regressor, Score (test, accuracy) = 75.55,
Regressor = Gradient Boosting Regressor, Score (test, accuracy) = 71.98,
Regressor = KNeighbors Regressor, Score (test, accuracy) = 48.87,
Regressor = SGD Regressor, Score (test, accuracy) = 44.96,
Regressor = KernelRidge Regressor, Score (test, accuracy) = -46.59,
Best --> Regressor = RandomForest Regressor, Score (test, accuracy) = 75.55
```

CLUSTERING

Clustering can be useful in identifying areas with similar housing characteristics and prices. For example, the model can group together houses that have similar features, such as total area, number of bedrooms and bathrooms, and other amenities. This information can assist real estate companies and property investors in identifying areas with similar property characteristics and prices.

Fractal Distance Calculation:

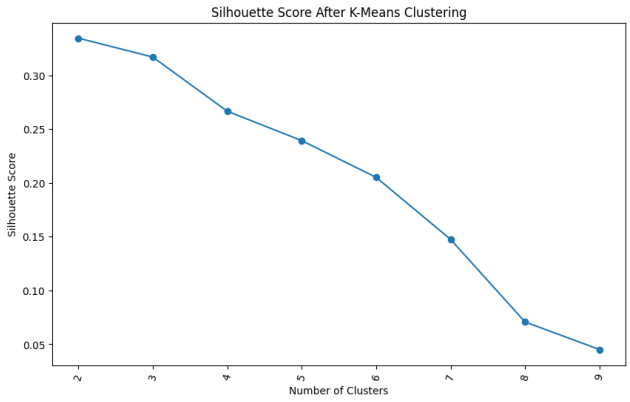
Fractal-clustering can be used to identify patterns in the dataset that may not be apparent through other techniques. For an instance, the model can be used to identify clusters of houses with similar pricing and features that may be located in different neighborhoods. This information can assist real estate companies and property investors in identifying areas with high potential for property appreciation.

```
# Print the result
print(f"The fractal distance between {p1} and {p2} is: {distance}")
```

```
The fractal distance between (0, 0) and (1, 1) is: 5.656854249492381
```

K-Means with Fractal distance

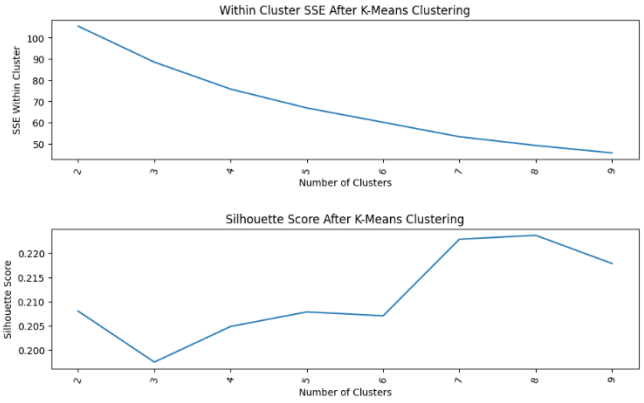
Implementing K-Means clustering with a custom distance metric, such as a fractal distance, involves modifying the traditional Euclidean distance calculation part of the algorithm. K-Means is inherently designed around the concept of minimizing variance (sum of squared Euclidean distances) within clusters, which makes altering its distance metric non-trivial, especially with algorithms optimized for speed and scalability that assume Euclidean geometry.




```
clustering performance
----------------------------------
silhouette score: 0.3169480178509633
```

K-Means with Euclidean distance

K-Means clustering with Euclidean distance is the standard implementation of the K-Means algorithm, widely used for partitioning a dataset into a set of k groups (or clusters) based on feature similarity. Here's a breakdown of how K-Means works with Euclidean distance, along with a simple implementation:




```
clustering performance
----------------------------------
Silhouette score: 0.2
SSE within cluster: 88.48930484344203
   Feature1  Feature2  Feature3  Feature4  Feature5  cluster
0  0.275298  0.732402  0.398371  0.488895  0.563980        0
1  0.257231  0.380934  0.899999  0.856762  0.184425        2
2  0.964931  0.425080  0.181365  0.306678  0.023743        1
3  0.918802  0.105483  0.834155  0.023496  0.812675        2
4  0.685496  0.215198  0.827909  0.731957  0.955359        2
```

Latent Variables

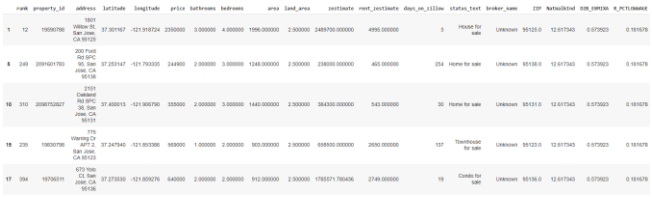
We scraped a beautiful dataset consisting of predominant latent variables which can be the form factors while choosing the right house. These latent variables predominantly act as a characteristic feature for houses which indirectly may impact on the popularity, pricing and considerations of houses of that location. For the location, we scraped walkability, mix of employment types in a block group (such as retail, office, or industrial), percentage of workers that are low, medium, or high wage (by home and work locations).

Merge Dataset:1 and Latent variables

Attributes: rank, property_id, address, latitude, longitude, price, bathrooms, bedrooms, area, land_area, zestimate, rent_zestimate, days_on_zillow, status_text, broker_name, ZIP, NatWalkInd, D2B_E8MIXA, R_PCTLOWWAGE

Amalgamation + Latent Variables

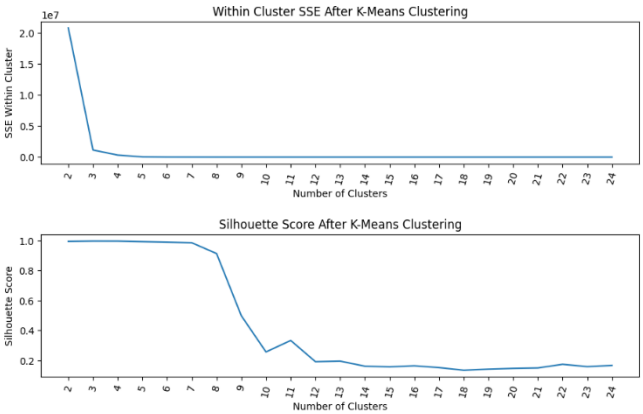
The dataset was augmented with three new features (NatWalkInd, D2B_E8MIXA, R_PCTLOWWAGE) derived from external sources based on the city extracted from the address field. Despite successfully adding these features, there are 1318 missing values for each new feature, suggesting gaps in the external data or mismatches in the city names used for mapping.

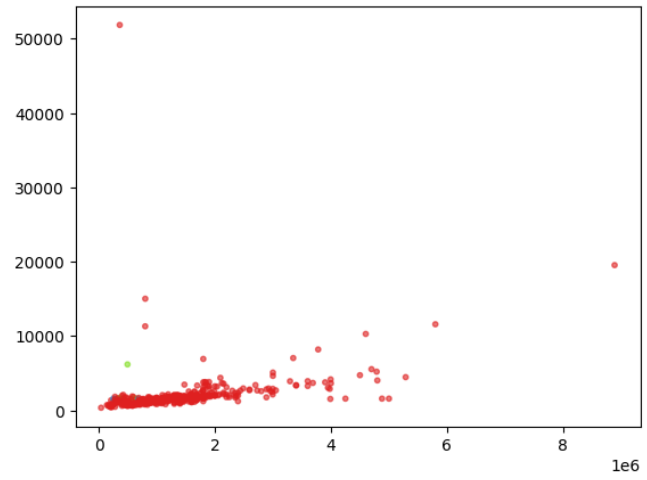


Implementing Machine Learning Algorithms

Fractal Clustering with Euclidean Distance:

First Iteration:





We see that after the first iterration there is 1 prominent cluster and hence we need to perform a second iteration on this cluster.

Second Iteration:

```
cluster_perf_df['cluster'].value_counts()
```

```
cluster  cluster
0        0          428
1        1            4
2        2            2
Name: cluster, dtype: int64
```

Golden Cluster: During the Clustering 2, we found that there are 2 clusters 0, 1 with each value mean value of price and rent zestimates. Cluster 0 being the golden cluster has 428 units.

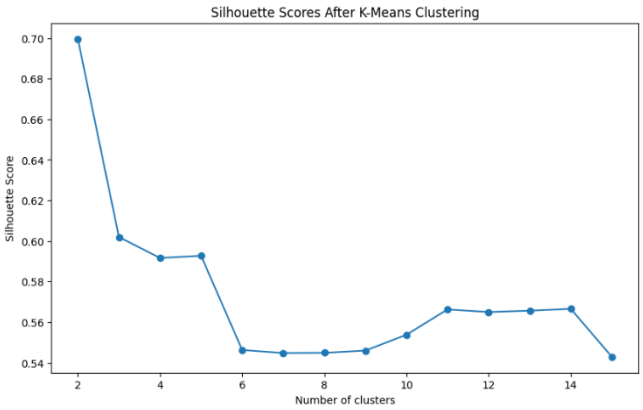
Fractal Clustering with Fractal Distance:

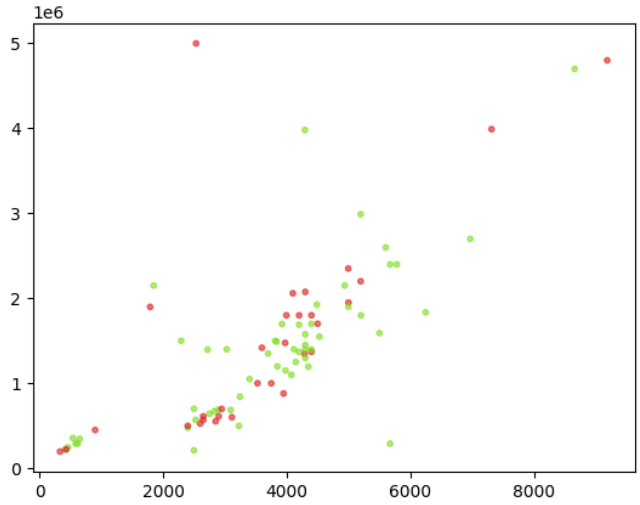
The code assesses clustering performance on a property dataset across different numbers of clusters using KMeans, leveraging silhouette scores to identify the optimal clustering scenario. High silhouette scores suggest better cluster cohesion and separation, guiding the choice of k for the most meaningful property groupings based on their features.

Silhouette score: 1.0: A silhouette score of 1.0 is perfect and indicates that every point is exactly on the border of the nearest cluster. This usually happens in very distinct or well-separated data, but it's unusual with real-world data and might indicate that the data is not clustered meaningfully, or there might be an issue with the data or the way clustering was applied.

SSE within cluster: 1146584.1824115708: This value represents the sum of the squared distances between each member of a cluster and the cluster's centroid. A lower SSE value suggests that the model has fewer errors and the clusters are tighter.

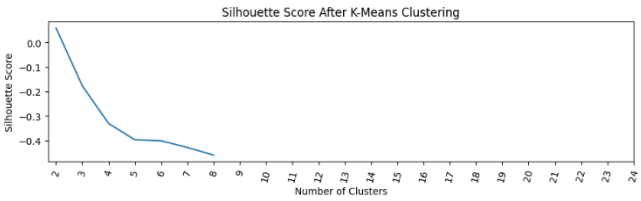
First Iteration:





We see that after the first iteration there is 1 prominent cluster and hence, we need to perform a second iteration on this cluster.

Second Iteration




```
clustering performance
----------------------------------
silhouette score: 0.05842694116700984
```

```
cluster_perf_df = second_trial.groupby('cluster')
cluster_perf_df['cluster'].value_counts()
```

```
cluster  cluster
0        0          20
1        1          34
Name: cluster, dtype: int64
```

Golden Cluster: During the Clustering 2, we found that there are 2 clusters 0, 1 with each value mean value of price and rent zestimates. Cluster 1 being the golden cluster has 34 units.

Classification using Muller Loop:

Adding HOA (Dataset 3) to Previous dataset for classification

HOA: Home Owner Association

For further in-depth investigation of house pricing and looking for some correlating factors, we introduce HOA, mortgage to our dataset. These are basically to calculate the appreciation trends of a property preleased property. On investigating we came to conclusion that the least interested houses are the houses with high price and mortgage fees. This reflects that irrespective of spatial features of houses and latent variables, unreasonable pricing caused the buyers to neglect such housing options.

Mortgage Fee Calculation:

Mortgage Fee = 0.0062 * Price (average based on multiple mortgage calculators online)

Amalgamating HOA from Dataset 3 and Dataset 1

```
X.head()
```

	rank	price	bathrooms	bedrooms	area	land_area	rent_zestimate	days_on_zillow	is_zillow_owned	hoa	mortgage_hoa
0	380	799000	1.000000	2.000000	810.000000	2.500000	3354.000000	10	False	330.000000	5283.800000
1	12	2350000	3.000000	4.000000	1996.000000	2.500000	4995.000000	3	False	370.000000	14940.000000
2	97	999900	3.000000	2.000000	1750.000000	2.500000	3875.000000	3	False	330.000000	6529.380000
3	88	795000	2.000000	2.000000	1400.000000	2.500000	3999.000000	2	False	330.000000	5259.000000
4	778	450000	1.000000	1.000000	716.000000	2.500000	2394.000000	61	False	330.000000	3120.000000

Next steps: View recommended plots

Results:

```
clf_metrics_amal_latent = muller_classification(X_train, X_test, y_train, y_test)
```

```
Nearest Neighbors
Classifier = Nearest Neighbors, Score (test, accuracy) = 95.42,
Linear SVM
Classifier = Linear SVM, Score (test, accuracy) = 99.24,
RBF SVM
Classifier = RBF SVM, Score (test, accuracy) = 95.42,
Decision Tree
Classifier = Decision Tree, Score (test, accuracy) = 97.71,
Random Forest
Classifier = Random Forest, Score (test, accuracy) = 96.95,
Neural Net
Classifier = Neural Net, Score (test, accuracy) = 93.13,
AdaBoost
Classifier = AdaBoost, Score (test, accuracy) = 98.47,
Naive Bayes
Classifier = Naive Bayes, Score (test, accuracy) = 96.95,
Best --> Classifier = Linear SVM, Score (test, accuracy) = 99.24
```

Regression Using Muller Loop:

Test Train Dataset 1

Dataset 1

```
X = data.drop([ 'address','price', 'mortgage_hoa'], axis=1)
y = data['price']
```

```
X = X.drop([ 'broker_name'],axis=1)
```

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
```

```
reg_metrics_data1 = muller_regression(x_train, x_test, y_train, y_test)
```

```
Regressor = Linear Regression, Score (test, accuracy) = 44.94,
Regressor = MLP Regressor, Score (test, accuracy) = 43.51,
Regressor = RandomForest Regressor, Score (test, accuracy) = 58.94,
Regressor = Gradient Boosting Regressor, Score (test, accuracy) = 65.37,
Regressor = KNeighbors Regressor, Score (test, accuracy) = 63.56,
Regressor = SGD Regressor, Score (test, accuracy) = -122258929576425138683904.00,
Regressor = KernelRidge Regressor, Score (test, accuracy) = 46.48,
Best --> Regressor = Gradient Boosting Regressor, Score (test, accuracy) = 65.37
```

Regression on Amalgamated dataset

```
X = df.drop(['address','property_id', 'zestimate', 'target', 'price', 'mortgage_hoa'], axis=1)
y = df['price']
```

```
X=X.drop(['status_text','broker_name'],axis=1)
```

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
```

```
reg_metrics_amal = muller_regression(x_train, x_test, y_train, y_test)
```

```
Regressor = Linear Regression, Score (test, accuracy) = 65.03,
Regressor = MLP Regressor, Score (test, accuracy) = 8.44,
Regressor = RandomForest Regressor, Score (test, accuracy) = 72.98,
Regressor = Gradient Boosting Regressor, Score (test, accuracy) = 76.90,
Regressor = KNeighbors Regressor, Score (test, accuracy) = 69.12,
Regressor = SGD Regressor, Score (test, accuracy) = -6866728760983783278284636160.00,
Regressor = KernelRidge Regressor, Score (test, accuracy) = 65.37,
Best --> Regressor = Gradient Boosting Regressor, Score (test, accuracy) = 76.90
```

Dataset 1 + Latent Variables:

```
X = data1_latent.drop(['address','property_id', 'zestimate', 'target', 'price', 'mortgage_hoa'], axis=1)
y = data1_latent['price']
```

```
X=X.drop(['status_text','broker_name'],axis=1)
```

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
```

```
reg_metrics_data1_latent = muller_regression(x_train, x_test, y_train, y_test)
```

```
Regressor = Linear Regression, Score (test, accuracy) = 4.47,
Regressor = MLP Regressor, Score (test, accuracy) = 6.29,
Regressor = RandomForest Regressor, Score (test, accuracy) = 75.84,
Regressor = Gradient Boosting Regressor, Score (test, accuracy) = 83.02,
Regressor = KNeighbors Regressor, Score (test, accuracy) = 54.46,
Regressor = SGD Regressor, Score (test, accuracy) = -9192947707826449586916950016.00,
Regressor = KernelRidge Regressor, Score (test, accuracy) = 6.53,
Best --> Regressor = Gradient Boosting Regressor, Score (test, accuracy) = 83.02
```

Amalgamation + Latent Variables:

```
X = df_latent.drop(['address','property_id', 'zestimate', 'target', 'price', 'mortgage_hoa'], axis=1)
y = df_latent['price']
```

```
X=X.drop(['status_text','broker_name'],axis=1)
```

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
```

```
reg_metrics_amal_latent = muller_regression(x_train, x_test, y_train, y_test)
```

```
Regressor = Linear Regression, Score (test, accuracy) = 69.33,
Regressor = MLP Regressor, Score (test, accuracy) = 9.06,
Regressor = RandomForest Regressor, Score (test, accuracy) = 80.68,
Regressor = Gradient Boosting Regressor, Score (test, accuracy) = 91.79,
Regressor = KNeighbors Regressor, Score (test, accuracy) = 59.96,
Regressor = SGD Regressor, Score (test, accuracy) = -2818934402155191331717120.00,
Regressor = KernelRidge Regressor, Score (test, accuracy) = 69.22,
Best --> Regressor = Gradient Boosting Regressor, Score (test, accuracy) = 91.79
```

OVERALL METRICS

Classification: Dataset 1:

```
clf_metrics_data1
```

	Classifier	MSE	MAE	RSquared	Test Accuracy	Recall	Precision
0	Naive Bayes	0.100000	0.200000	0.040000	95.077720	0.950777	0.939562
1	AdaBoost	0.020000	0.040000	0.800000	98.963731	0.989637	0.989750
2	Neural Net	0.040000	0.080000	0.600000	97.927461	0.979275	0.978527
3	Random Forest	0.030000	0.050000	0.750000	98.704663	0.987047	0.987222
4	Decision Tree	0.020000	0.030000	0.850000	99.222798	0.992228	0.992291
5	RBF SVM	0.110000	0.220000	-0.060000	94.559585	0.945596	0.894152
6	Linear SVM	0.010000	0.010000	0.950000	99.740933	0.997409	0.997527
7	Nearest Neighbors	0.110000	0.230000	-0.110000	94.300518	0.943005	0.894018

Dataset 1 + Latent Variables

clf_metrics_data1_latent

	Classifier	MSE	MAE	RSquared	Test Accuracy	Recall	Precision
0	Naive Bayes	0.050000	0.090000	0.480000	97.709924	0.977099	0.977636
1	AdaBoost	0.030000	0.060000	0.650000	98.473282	0.984733	0.984973
2	Neural Net	0.090000	0.180000	-0.050000	95.419847	0.954198	0.954198
3	Random Forest	0.030000	0.060000	0.650000	98.473282	0.984733	0.984973
4	Decision Tree	0.050000	0.090000	0.480000	97.709924	0.977099	0.975694
5	RBF SVM	0.090000	0.180000	-0.050000	95.419847	0.954198	0.910495
6	Linear SVM	0.020000	0.030000	0.830000	99.236641	0.992366	0.992427
7	Nearest Neighbors	0.090000	0.180000	-0.050000	95.419847	0.954198	0.910495

Results after Amalgamation:

clf_metrics_amal

	Classifier	MSE	MAE	RSquared	Test Accuracy	Recall	Precision
0	Naive Bayes	0.060000	0.120000	0.300000	96.946565	0.969466	0.966009
1	AdaBoost	0.020000	0.030000	0.830000	99.236641	0.992366	0.992427
2	Neural Net	0.080000	0.150000	0.130000	96.183206	0.961832	0.963300
3	Random Forest	0.030000	0.060000	0.650000	98.473282	0.984733	0.984973
4	Decision Tree	0.030000	0.060000	0.650000	98.473282	0.984733	0.984733
5	RBF SVM	0.090000	0.180000	-0.050000	95.419847	0.954198	0.910495
6	Linear SVM	0.020000	0.030000	0.830000	99.236641	0.992366	0.992427
7	Nearest Neighbors	0.090000	0.180000	-0.050000	95.419847	0.954198	0.910495

Results after second Amalgamation (includes Latent variables)

clf_metrics_amal_latent

	Classifier	MSE	MAE	RSquared	Test Accuracy	Recall	Precision
0	Naive Bayes	0.060000	0.120000	0.300000	96.946565	0.969466	0.969466
1	AdaBoost	0.030000	0.060000	0.650000	98.473282	0.984733	0.984973
2	Neural Net	0.140000	0.270000	-0.570000	93.129771	0.931298	0.925494
3	Random Forest	0.060000	0.120000	0.300000	96.946565	0.969466	0.970412
4	Decision Tree	0.050000	0.090000	0.480000	97.709924	0.977099	0.979219
5	RBF SVM	0.090000	0.180000	-0.050000	95.419847	0.954198	0.910495
6	Linear SVM	0.020000	0.030000	0.830000	99.236641	0.992366	0.992427
7	Nearest Neighbors	0.090000	0.180000	-0.050000	95.419847	0.954198	0.910495

Regression: Dataset 1:

reg_metrics_data1

	Regressor	MSE	MAE	RSquared	Test Accuracy
0	KernelRidge Regressor	402386.860000	1167810551602.830078	0.460000	46.477920
1	SGD Regressor	411254883193824440.000000	271328105045368041772963623418368.000000	-1222589295764251344896.000000	-1222589295764251386839040.000000
2	KNeighbors Regressor	405137.840000	808715395468.589966	0.640000	63.559736
3	Gradient Boosting Regressor	339201.640000	768483910744.979980	0.650000	65.372544
4	RandomForest Regressor	397405.020000	911246551959.619995	0.590000	58.939739
5	MLP Regressor	478074.610000	1253701502255.729980	0.440000	43.508910
6	Linear Regression	390917.150000	1221917020389.689941	0.450000	44.941101

Dataset 1 + Latent Variables:

reg_metrics_data1_latent

	Regressor	MSE	MAE	RSquared	Test Accuracy
0	KernelRidge Regressor	368807.530000	723865575082.270020	0.070000	6.526991
1	SGD Regressor	8435913225359648768.000000	7119116284924070812626480363234408448.000000	-919294770782644917460008960.000000	-919294770782644958691690016.000000
2	KNeighbors Regressor	318975.210000	352654845271.429993	0.540000	54.461539
3	Gradient Boosting Regressor	228415.600000	131527082847.050003	0.830000	83.015852
4	RandomForest Regressor	282772.880000	187063909641.640015	0.760000	75.844359
5	MLP Regressor	618921.400000	725702327602.449951	0.060000	6.289750
6	Linear Regression	351357.670000	739807053988.079956	0.040000	4.468401

Results after Amalgamation:

reg_metrics_amal

	Regressor	MSE	MAE	RSquared	Test Accuracy
0	KernelRidge Regressor	369928.480000	316356625574.770020	0.650000	65.369838
1	SGD Regressor	792002356231816908 8.000000	527295693919192038630642327890459361 28.000000	-6866728760983783450083328 0.000000	-6866728760983783278284636160.000000
2	KNeighbors Regressor	298006.330000	282063895641.359985	0.690000	69.123712
3	Gradient Boosting Regressor	265611.490000	211066919283.420013	0.770000	76.895437
4	RandomForest Regressor	300517.090000	246860675463.640015	0.730000	72.977253
5	MLP Regressor	649915.580000	836391663330.739990	0.080000	8.443901
6	Linear Regression	372544.560000	319498306421.650024	0.650000	65.025932

Results after second Amalgamation (includes Latent variables)

reg_metrics_amal_latent

	Regressor	MSE	MAE	RSquared	Test Accuracy
0	KernelRidge Regressor	348193.950000	301833586601.809998	0.690000	69.224852
1	SGD Regressor	151125916075310048.000000	27647278664740886185965554586123264.000000	-2818934402155191415603 2.000000	-2818934402155191331717120.000000
2	KNeighbors Regressor	353335.270000	392725417956.809998	0.600000	59.957463
3	Gradient Boosting Regressor	203049.040000	80566533894.410004	0.920000	91.785384
4	RandomForest Regressor	284714.240000	189473012260.239990	0.810000	80.681209
5	MLP Regressor	657178.950000	891869296021.880005	0.090000	9.064430
6	Linear Regression	344596.830000	300815235554.989990	0.690000	69.328684

CONCLUSION AND FUTURE SCOPE

In summarizing our exploration of the vast real estate data landscape, we've uncovered not just patterns and predictions but also the stories and dreams that animate the housing market. This research aimed to decode the numbers but ended up revealing the human experiences behind them. While our findings are rooted in analytics, they resonate with the aspirations of home-seekers, the calculations of investors, and a society's quest for stability. Each data point reflects a piece of a larger narrative of life and community.

As we continue, our analysis will delve deeper, considering environmental impacts, legislative changes, and technological advances, employing the next wave of analytical tools to uncover insights that currently lie beyond our reach. Our journey in real estate analytics has been enlightening, exposing the intricate connections between market mechanics and personal stories. With a universe of data still to uncover, we invite everyone to contribute to this ongoing narrative. The pursuit of understanding continues, and we eagerly anticipate the discoveries that await in the next phase of our adventure.

REFERENCES

[1] https://www.zillow.com/research/data/

[2]https://cde.ucr.cjis.gov/LATEST/webapp/#/pages/explorer/crime/crime-trend.

[3]https://gis.data.ca.gov/datasets/9a0f00ce466842f0bb9b5e1c95724a26_0/explore