Maximize Cash Flow Through Rental Properties in San Diego

Anisha Agarwal , Arshia Sali, Jeena Thampi, Pooja Prasannan

Abstract

In today's world people want to be independent. Investment is one such thing which can make a person independent and not relying on the money of others even in financial hardship. It means that you can buy an asset or put the capital money in the bank to get future interest from it. Invest today in order to increase the value in future. There can be different mediums of investment like, business, bonds, real estate property, or stocks. We are choosing Real Estate property because one can do a force appreciation, can get Tax benefits, can earn a regular cash flow, pass real estate down to their heirs, Equity can be used to increase the portfolio of real estate. A good investment in real estate is when we see the remains of Rent Zestimate is greater than Zero when subtracted with the overall Monthly Mortgage which includes HOA, Property Taxes and upkeeps.

In this project we are analyzing and predicting the good investment using different attributes from different datasets. Datasets used are Zillow scraped and publicly available Crime dataset. The attributes used are 30-year fixed rate to calculate Mortgage Value, property tax rate to calculate Property Tax price, HOA for accurate calculation of Good investment, Crime rate data and finally the hidden Average Income data.

Our objective is to find a good investment which is affordable, profitable and safe to invest, thus we have done multiple iterations of models to get an accurate prediction. As a result, we can see an increase in accuracy score as and when the attributes get added. The Classifier model increases from 80% to 92% and the Regressor model increases from 56% to 69%

Keywords: Rental Investment, Cash Flow, Data Mining, Fractal Clustering, Classifier, Regressor

1.Introduction

Nowadays, people wish to have an early retirement after working hard for the initial quarter of their life. Regular Savings accounts will not give them enough profit to make their life hassle free for a long term. As people grow old, chances of getting sick and prone to illness increases. Investing their savings in stocks could be risky. As the stock prices fluctuates and doesn't guarantee them a profitable return. There comes the value of investments in 'Real Estate'. Real Estate Investments are always guaranteed to give profit in a long term perspective without the risk factor.

Investing in real estate can be of several kinds. Out of which investing in rental properties are preferred for a positive and steady cash flow. This showcases a unique case of short term and long term investment. When a person invests in a rental property with a loan, he has to consider whether the mortgage payment, Homeowners Association (HOA) and property taxes combined together is less than the monthly rent he receives, only then there is a positive cash flow. This counts towards considering this as a short term investment. The value of the property appreciates with time. Selling this property after years would definitely give a high profit as the value of the property would have appreciated by then. This counts towards considering this investment as a long-term one. Thus, investing in rental properties is always a great choice without doubt.

In this study, we are focusing on properties listed in San Diego, California to shortlist the properties which would give the maximum returns to an investor. Data collected from various website listings and further information regarding the School District Ratings and Crime Data around the area are also used to enhance the property details. Machine Learning models are used to give a shortlisted

result set of properties which can be suggested to the investor.

2. Related Works

A lot of innovative evaluation methods in the real estate field have been published over the years. One such study proposes a method called "historical market price", which uses the mathematical, statistical and database-founded algorithms for valuation. This method involved the use of historical data and a lot of documentation such as ground plans, structure plans, legal documents such as insurance contracts, rental contracts, land registry documents, purchase contracts, work contracts, market price valuations, experts statements valuations,experts statements, and maintenance-related documents such as maintenance and replacement plans for individual construction units, invoices for construction work, auditing plans, expert technical inspections, servicing and maintenance activities.

In another such study it shows how using publicly available data streams and machine learning algorithms one can develop practical data driven services with no input from domain experts as a form of prior knowledge. Based on web crawling of publicly available real estate advertisements continuously and using building data from open street maps, they developed a system where they estimated the rental and sales price index.

In our project, we are focusing on the properties listed in San Diego which investors can invest on to make more profit.

3. Data

We have used multiple datasets to achieve our objective of good investment.

1. **Dataset-1**: Publicly available US housing Dataset https://raw.githubusercontent.com/AnishaA-git/SanDiego-Housing/master/Dataset_Realestate.csv
2. **Dataset-2**: Zillow scraped dataset https://raw.githubusercontent.com/AnishaA-git/SanDiego-Housing/master/df_z_prop_detail_output.csv
3. **Dataset-3**: Publicly available San Diego Crime Dataset https://github.com/AnishaA-git/SanDiego-Housing/blob/master/crimes_merge_all.csv

From the US housing dataset we are filtering out the San Diego data, amalgamating it with the Zillow scraped data and using different attributes from it like, HOA, 30 year fixed rate, property tax rate. Further amalgamating the Crime dataset we get Crime rate data which enhances the overall objective of good investment.

3.1 Data Filtering, Description and Preprocessing:

For US Housing:

- Filtering: fetching San Diego data

```
df_sd = df[df['address'].str.contains('San Diego')]
df_sd
```

	rank	property_id	address	latitude	longitude
	728	72109285	350 11th Ave UNIT 224, San Diego, CA 92101	32.708987	-117.154838

Figure 1. US Housing Dataset filtering. .

- Description

Type of Data - It contains all Numerical (continuous and discrete), Categorical and Text

```
Data Types:
rank                 int64
property_id          int64
address             object
latitude           float64
longitude          float64
price              float64
currency            object
bathrooms          float64
bedrooms           float64
area                object
land_area           object
zestimate          float64
rent_zestimate     float64
days_on_zillow     float64
sold_date          float64
is_zillow_owned       bool
image               object
listing_type        object
broker_name         object
input               object
property_url        object
listing_url         object
dtype: object
Rows and Columns:
(710, 22)
```

Figure 2. US Housing data type .

Column Names

```
Column Names:
Index(['rank', 'property_id', 'address', 'latitude', 'longitude', 'pr
       'currency', 'bathrooms', 'bedrooms', 'area', 'land_area', 'zes
       'rent_zestimate', 'days_on_zillow', 'sold_date', 'is_zillow_ow
       'image', 'listing_type', 'broker_name', 'input', 'property_url
       'listing_url'],
      dtype='object')
```

Figure 3. US Housing Dataset column names.

Null Values: There are a total 710 null fields which need to be processed.

```
1  # Checking to see if there are any null values in our dataset.
2  df_sd.isnull().sum(axis = 0).sort_values(ascending = True)
```

```
rank                0
input               0
listing_type        0
image               0
is_zillow_owned     0
property_url        0
listing_url         0
price               0
address             0
property_id         0
currency            0
longitude           1
latitude            1
days_on_zillow      2
bedrooms           25
bathrooms          27
area               32
rent_zestimate     81
zestimate         112
broker_name       505
land_area         690
sold_date         710
dtype: int64
```

Figure 4. US Housing Dataset null values.

PreProcessing

Checking for duplicates: Zero duplicates found

```
1  # Checking to see if there are any duplicated data in dataset
2  df_sd[df_sd.duplicated() == True]
```

```
rank  property_id  address  latitude  longitude  price  currency  ba
```

Figure 5. US Housing Dataset Preprocessing checking for duplicates.

Removing the Columns which are not used: ‘land_area’ and ‘sold_date’ removed

```
df_sd_clean = df_sd.drop(columns=['land_area', 'sc
```

```
df_sd_clean.shape
```

```
(710, 20)
```

Figure 6. US Housing Dataset Preprocessing removing columns.

3.2 Zillow scraped:

Filtering: Fetching the Data which is required to amalgamate with the First DataSet

i. HOA, Property Taxes Rate, 30 year Fixed Rate Calculate the accurate Monthly Mortgage

ii. Page View and Rating average for feature addition

```
scrape_df = df_z_prop_detail_output[['zpid','hoaFee','propertyTaxRate','listed_by.rating_average','pageViewCount','mortgageRates
scrape_df
```

Figure 7. Zillow scraped Dataset filtering.

Merging the above two datasets:

```
final_df = pd.merge(df_sd_clean,scrape_df, left_on='property_id', right_
final_df
```

Figure 8. Zillow scraped Dataset amalgamated.

Description

Type of Data - It contains all Numerical (continuous and discrete), Categorical and Text

```
Data Types:
rank                                    int64
property_id                             int64
address                                object
latitude                              float64
longitude                             float64
price                                 float64
currency                               object
bathrooms                             float64
bedrooms                              float64
area                                   object
zestimate                             float64
rent_zestimate                        float64
days_on_zillow                        float64
is_zillow_owned                          bool
image                                  object
listing_type                           object
input                                  object
property_url                           object
listing_url                            object
hoaFee                                float64
propertyTaxRate                       float64
listed_by.rating_average              float64
pageViewCount                           int64
mortgageRates.thirtyYearFixedRate     float64
monthly_mortgage                      float64
property_tax_price                    float64
investment                              int64
dtype: object
Rows and Columns:
(688, 27)
```

Figure 9. Zillow scraped Data type.

Column Names

```
Column Names:
Index(['rank', 'property_id', 'address', 'latitude', 'longitude', 'price',
       'currency', 'bathrooms', 'bedrooms', 'area', 'zestimate',
       'rent_zestimate', 'days_on_zillow', 'is_zillow_owned', 'image',
       'listing_type', 'input', 'property_url', 'listing_url', 'hoaFee',
       'propertyTaxRate', 'listed_by.rating_average', 'pageViewCount',
       'mortgageRates.thirtyYearFixedRate', 'monthly_mortgage',
       'property_tax_price', 'investment'],
      dtype='object')
```

Figure 10. Zillow scraped Dataset column names.

Null Value: There are a total 383 null values to be processed.

```
rank                                  0
propertyTaxRate                       0
listing_url                           0
property_url                          0
input                                 0
listing_type                          0
image                                 0
is_zillow_owned                       0
pageViewCount                         0
mortgageRates.thirtyYearFixedRate     0
bedrooms                              0
bathrooms                             0
currency                              0
price                                 0
longitude                             0
latitude                              0
address                               0
property_id                           0
days_on_zillow                        2
area                                 10
listed_by.rating_average             70
rent_zestimate                       70
zestimate                            92
hoaFee                              383
dtype: int64
```

Figure 11. Zillow scraped Dataset null values.

PreProcessing

Removing the rows: Null rows of 'bedrooms', 'bathrooms','latitude','longitude' as these are not useful, people would not be interested in buying a property as they do not have these important details.

```
# cleaning dataset
final_df_clean = final_df.dropna(subset=['bedrooms', 'bathrooms','latitude',
final_df_clean = final_df_clean.drop(columns=['zpid','broker_name'])

final_df_clean.shape
```

```
(688, 24)
```

Figure 12. Zillow scraped Dataset Preprocessing removing columns and rows.

Min, Max and Mean of the data:

```
final_df_clean.describe()
```

	rank	property_id	latitude	longitude	price	bathrooms	bedrooms	z
count	688.000000	6.880000e+02	688.000000	688.000000	6.880000e+02	688.000000	688.000000	5.96
mean	390.377907	2.333586e+08	32.789439	-117.129768	9.951255e+05	2.412791	3.068314	9.85
std	235.348779	5.986845e+08	0.106807	0.063982	7.337591e+05	1.230831	1.369128	7.05
min	1.000000	1.673506e+07	32.561281	-117.258185	8.500000e+04	1.000000	0.000000	1.21
25%	179.750000	1.693312e+07	32.720860	-117.170595	5.927500e+05	2.000000	2.000000	5.88
50%	383.500000	1.706564e+07	32.766891	-117.127857	7.990000e+05	2.000000	3.000000	8.01
75%	593.500000	6.395879e+07	32.826418	-117.079727	1.167400e+06	3.000000	4.000000	1.10
max	800.000000	2.133748e+09	33.052774	-116.999709	7.495000e+06	16.000000	12.000000	6.99

Figure 13. Zillow scraped Dataset Preprocessing Min,Max,Mean values1.

pageViewCount	mortgageRates.thirtyYearFixedRate
688.000000	688.000000
1843.579942	2.904580
1942.764924	0.045793
49.000000	2.884000
739.500000	2.884000
1428.500000	2.904000
2392.750000	2.904000
32160.000000	3.341000

Figure 14. Zillow scraped Dataset Preprocessing Min,Max,Mean values2.

Filling the Null values: filling these with the mean values as described below

```
final_df_clean = final_df_clean.fillna({'hoaFee':0})
final_df_clean = final_df_clean.fillna({'rent_zestimate':3807.3})
final_df_clean = final_df_clean.fillna({'zestimate':985250.3})
final_df_clean = final_df_clean.fillna({'days_on_zillow':0})
final_df_clean = final_df_clean.fillna({'listed_by.rating_average':3.2})
final_df_clean = final_df_clean.fillna({'area':"1500 sqft"})

final_df_clean.isnull().sum(axis = 0).sort_values(ascending = True)
```

Figure 15. Zillow scraped Dataset Preprocessing filling null values.

Final data of 1 and 2 after preprocessing:

```
rank                                  0
listed_by.rating_average              0
propertyTaxRate                       0
hoaFee                                0
listing_url                           0
property_url                          0
input                                 0
listing_type                          0
image                                 0
is_zillow_owned                       0
days_on_zillow                        0
rent_zestimate                        0
zestimate                             0
area                                  0
bedrooms                              0
bathrooms                             0
currency                              0
price                                 0
longitude                             0
latitude                              0
address                               0
property_id                           0
pageViewCount                         0
mortgageRates.thirtyYearFixedRate     0
dtype: int64
```

Figure 16. Zillow scraped Dataset Preprocessing after filling null values.

3.3 *Crime Data*:

Filtering: Fetching crime and Household median income data on the basis of Zip Code from address attribute.

```
street_list = final_df_clean['address'].values.tolist()
street_actual_list = []
for i in street_list:
    zipcode = i.split(',')[2].split(' ')[2]
    street_actual_list.append(zipcode)

# print(street_actual_list)
final_df_clean['zipCode'] = street_actual_list
final_df_clean['zipCode'] = final_df_clean['zipCode'].astype(st
final_df_clean

```

Figure 17. Crime Data filtering.

Description:

```
crime_df_final = df_crime_sd[['Total Crime Risk','Zip Code','Median house
crime_df_final.dtypes
```

```
Total Crime Risk          int64
Zip Code                  int64
Median household income   int64
dtype: object
```

Figure 18. Crime Dataset description.

Final DataSet: Merging the data, and the final data is prepared.

```
final_df_invest_crime = pd.merge(final_df_clean,crime_df_final, left_on='z
final_df_invest_crime

```

Figure 19. Second Dataset amalgamated.

Min, Max and Mean of the Final Dataset:

	rank	property_id	latitude	longitude	price	bathrooms	bedrooms	zestimate	rent_zestimate	d
count	521.000000	5.210000e+02	521.000000	521.000000	5.210000e+02	521.000000	521.000000	5.210000e+02	521.000000	
mean	380.120921	2.630822e+08	32.793946	-117.131590	1.016232e+06	2.404990	3.055662	9.982625e+05	3861.126104	
std	237.083706	6.327539e+08	0.116900	0.064619	7.916582e+05	1.331242	1.460628	7.019452e+05	2137.167784	
min	1.000000	1.673506e+07	32.561281	-117.258185	8.500000e+04	1.000000	0.000000	1.214000e+05	1397.000000	
25%	170.000000	1.692778e+07	32.720362	-117.170817	5.890000e+05	2.000000	2.000000	6.225730e+05	2799.000000	
50%	362.000000	1.708222e+07	32.760022	-117.133034	7.999000e+05	2.000000	3.000000	8.780000e+05	3499.000000	
75%	592.000000	6.740417e+07	32.847589	-117.080060	1.188000e+06	3.000000	4.000000	1.020500e+06	3934.000000	
max	800.000000	2.133748e+09	33.052774	-116.999709	7.495000e+06	16.000000	12.000000	6.990200e+06	23140.000000	

Figure 20. Final data Min,Max,Mean values1.

days_on_zillow	is_zillow_owned	hoaFee	propertyTaxRate	listed_by.rating_average	pageViewCount	mortgageRates.thirtyYearFixedRate	monthly_mortgage
521.000000	521.000000	521.000000	5.210000e+02	521.000000	521.000000	521.000000	521.000000
9.821497	0.038388	188.790787	7.600000e-01	3.329750	1810.261036	2.906115	4213.466561
9.144187	0.192315	348.345878	5.445321e-15	2.188374	2006.679931	0.051246	3160.792131
-17.000000	0.000000	0.000000	7.600000e-01	0.000000	49.000000	2.884000	353.180445
2.000000	0.000000	0.000000	7.600000e-01	0.000000	683.000000	2.884000	2497.878789
7.000000	0.000000	0.000000	7.600000e-01	4.900000	1399.000000	2.904000	3290.069746
16.000000	0.000000	310.000000	7.600000e-01	5.000000	2351.000000	2.904000	4876.713985
32.000000	1.000000	3260.000000	7.600000e-01	5.000000	32160.000000	3.341000	30177.640425

Figure 21. Final data Min,Max,Mean values2.

property_tax_price	investment	zipCode	Total Crime Risk	Median household income
521.000000	521.000000	521.000000	521.000000	521.000000
643.613718	0.537428	92116.907869	92.714012	72259.454894
501.383507	0.499076	14.173620	33.981737	23020.779230
53.833333	0.000000	92101.000000	27.000000	37985.000000
373.033333	0.000000	92105.000000	60.000000	57946.000000
506.603333	1.000000	92115.000000	102.000000	68260.000000
752.400000	1.000000	92127.000000	122.000000	86252.000000
4746.833333	1.000000	92154.000000	143.000000	132069.000000

Figure 22. Final data Min,Max,Mean values3.

4. Methods

For getting a good investment prediction, below steps could be followed:

- To select the important features required for this objective. We have used:
- PCA
- Gini Score

- Correlation Heat Map
- Shapley Values

Fractal Clustering for finding the cluster of good investment.

- We have used, Elbow method and Silhouette Score to find out the number of clusters.
- Three iterations of the clustering is performed to get the golden cluster which is nothing but the good investment cluster points.

Implementation of different ML models for both Classifiers and Regressors.

- Training and testing the dataset one that is US Housing
- Plotting its Confusion Metrics for the same
- Training and testing the dataset 2 and 3 that is Zillow scraped data and crime data.
- Plotting the Confusion Metrics for the same

Add a Latent Variable to enhance the results.

- Train and test the data with the latent variable again.
- Plot the Confusion matrix for the same.

5. Experiments and Results

Feature Selection:

***PCA*:** Initial attributes of the US housing Dataset gave an accuracy of 88% after adding important features like: HOA, Property Tax rate, 30 year rate, Property Income, Monthly Mortgage, Crime data. The accuracy increased to 90%.

Shapley Values: For the first dataset important features are Zestimate, rent zestimate, price.

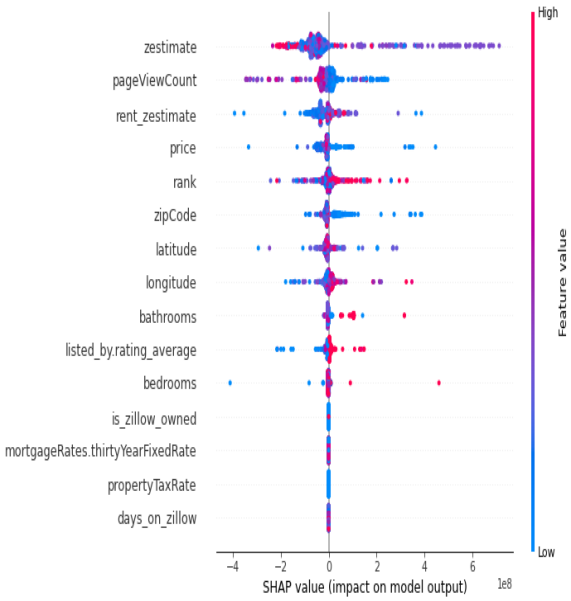

Figure 23. Shapley Values for dataset 1.

For 2nd and 3rd dataset: Important features are Zestimate, rent zestimate, monthly mortgage, HOA, Crime Risk

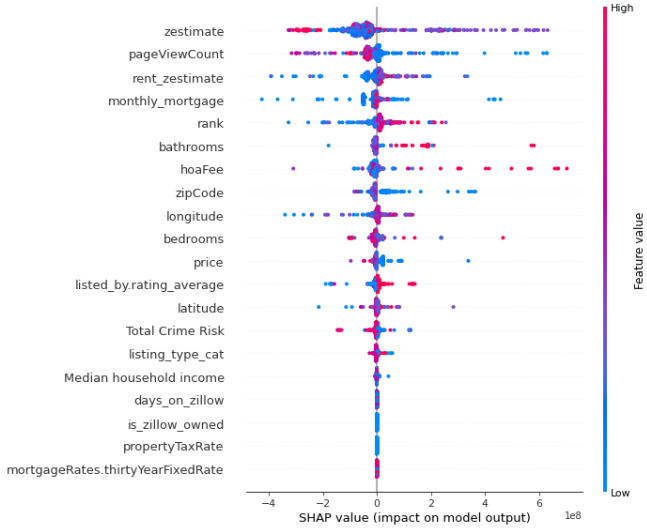

Figure 24. Shapley Values for dataset 2 and 3.

Inference: Zestimate, rent zestimate, monthly mortgage, HOA, Crime Risk are one of the important features which we want in our project. *Correlation Heat Map:* For the first dataset the important features are zestimate, rent zestimate, price, is zillow owned.

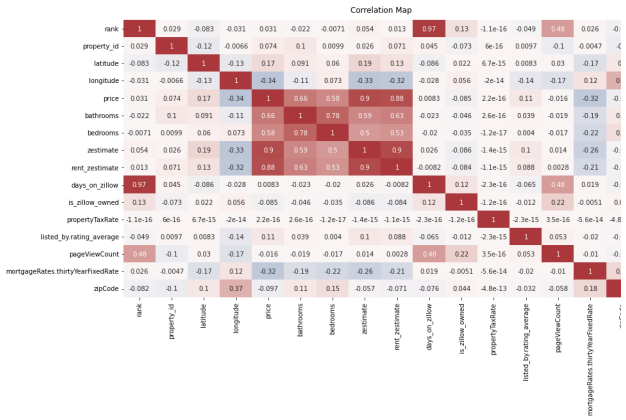

Figure 25. Correlation Heat map for dataset 1.

For the second dataset: the important features are zestimate, rent zestimate, price, is zillow owned, Monthly Mortgage, property tax price, HOA,Total Crime risk.

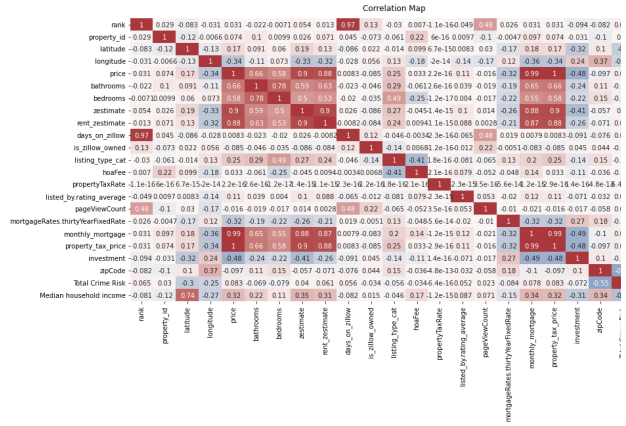

Figure 26. Correlation Heat map for dataset 2 and 3.

Inference: zestimate, rent zestimate, price, Monthly Mortgage, property tax price, HOA, Total Crime risk are the important features selected.

Gini Score: For first dataset. For 2nd dataset and for 3rd dataset

	score
property_id	0.998081
zestimate	0.992903
price	0.992000
zipCode	0.986579
rent_zestimate	0.982726
rank	0.978878
latitude	0.975247
days_on_zillow	0.970977
bedrooms	0.967440
bathrooms	0.966764
is_zillow_owned	0.959693
longitude	0.955854

	score
property_id	0.998081
zestimate	0.993013
price	0.991892
zipCode	0.986579
rent_zestimate	0.982318
pageViewCount	0.980197
hoaFee	0.975252
rank	0.974757
latitude	0.969682
days_on_zillow	0.965346
listed_by.rating_average	0.962492
bedrooms	0.959377
bathrooms	0.956455
mortgageRates.thirtyYearFixedRate	0.956084
listing_type_cat	0.951951
propertyTaxRate	0.945170
is_zillow_owned	0.944722
longitude	0.932821

	score
property_id	0.998081
zestimate	0.992903
price	0.992000
zipCode	0.986159
Median household income	0.983450
monthly_mortgage	0.977182
rent_zestimate	0.976428
pageViewCount	0.973673
property_tax_price	0.967484
hoaFee	0.965973
rank	0.965138
Total Crime Risk	0.958809
latitude	0.954460
days_on_zillow	0.950126
listed_by.rating_average	0.947455
bedrooms	0.944282
mortgageRates.thirtyYearFixedRate	0.940728
bathrooms	0.940104
listing_type_cat	0.936389
investment	0.930696
is_zillow_owned	0.928983
propertyTaxRate	0.927752
longitude	0.913628

Figure 29. Gini score for dataset 1, 2 and 3 respectively.

Inference: ***zestimate, rent zestimate, price, Monthly Mortgage, property tax price, HOA, Total Crime risk, Median household income*** are the important features selected finally.

Fractal Clustering: Our main objective is to shortlist the listings which would give maximum returns to the investor. For this we are performing clustering at multiple levels to find our golden cluster.

Elbow Method: Clusters selected are 5, as we can see the elbow bent to it.

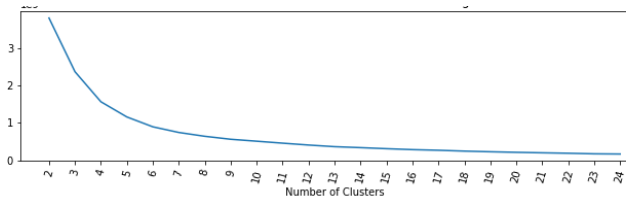

Figure 30. Elbow method for Fractal Clustering first iteration.

SSE Score: Clusters selected are 5, as we can see the score decreases at that point

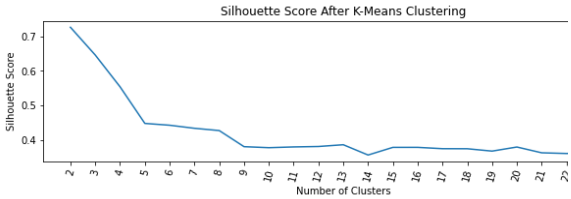

Figure 31. SSE score for Fractal Clustering second iteration.

Cluster Performance for 5 clusters:

```
clustering performance
------------------------------------
silhouette score: 0.45
sse with in cluster: 1156048392.0
Number of points in clusters:
 0    285
2     282
1      91
4      28
3       2
Name: cluster, dtype: int64
```

Figure 32. Performance for Fractal Clustering first iteration.

Plotting of first iteration of the cluster:

Figure 33. Plotting of Fractal Clustering first iteration.

Selecting Cluster 2 which is green in color. Plotting Elbow Method, SSE Score and Clustering Performance and plotting of clustering.
Elbow Method: Selecting Number of clusters 5 again.

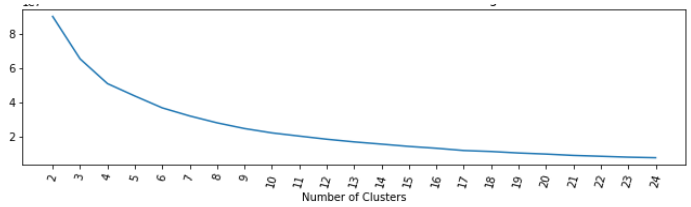

Figure 34. Elbow Method for Fractal Clustering second iteration.

SSE Score: Clusters selected are 5, as we can see the score decreases at that point

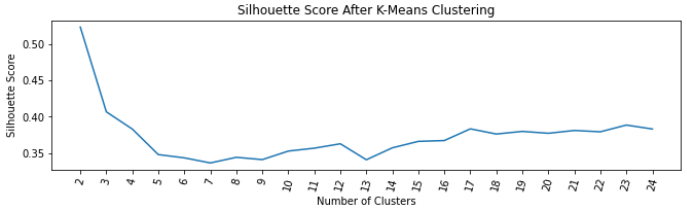

Figure 35. SSE Score for Fractal Clustering second iteration.

Cluster Performance for 5 clusters:

```
clustering performance
------------------------------------
silhouette score: 0.35
sse with in cluster: 43690526.0
Number of points in clusters:
 0    93
2     77
1     40
4     39
3     33
Name: cluster, dtype: int64
```

Figure 36. Performance of Fractal Clustering second iteration.

Plotting of second iteration of the cluster:

Figure 37. Plotting of Fractal Clustering second iteration.

Selecting Cluster number 0 which is blue in color. Plotting Elbow Method, SSE Score and Clustering Performance and plotting of clustering.
Elbow Method: Selecting Number of clusters 6.

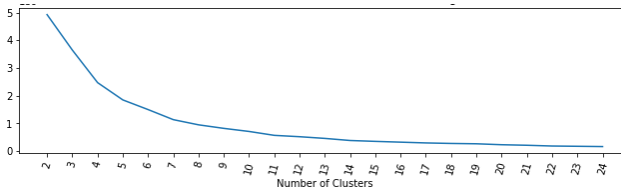

Figure 38. Elbow Method for Fractal Clustering third iteration.

SSE Score: Clusters selected are 6, as we can see the score decreases at that point

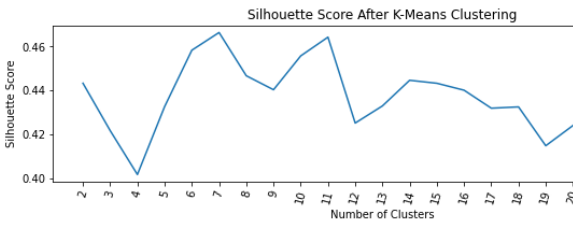

Figure 39. SSE Score for Fractal Clustering third iteration.

Cluster Performance for 6 clusters:

```
clustering performance
------------------------------
silhouette score: 0.46
sse with in cluster: 1500913.0
Number of points in clusters:
 0    19
4    18
1    18
3    16
2    15
5     7
Name: cluster, dtype: int64
```

Figure 40. Performance for Fractal Clustering third iteration.

Plotting of third iteration of the cluster:

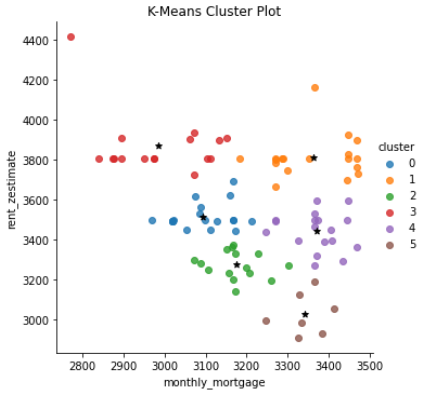

Figure 41. Plotting of Fractal Clustering third iteration.

Inference: Golden Cluster = 3, where rent_zestimate > monthly_mortgage, as we know rent_zestimate - monthly_mortgage(HOA, Property Taxes, upKeep) > 0 then its a good buy.

Implementation of Classifiers and Regressors with Metrics evaluation:

Classifiers for Dataset 1:

```
============================================================
Classifier:  Gradient Boost Classifier
              precision    recall  f1-score   support

           0       0.79      0.81      0.80        52
           1       0.87      0.86      0.87        79

    accuracy                           0.84       131
   macro avg       0.83      0.83      0.83       131
weighted avg       0.84      0.84      0.84       131

============================================================
```

Figure 42. Classifier for Dataset 1

Classifier for added Latent Variable:

```
=========================================================
Classifier:  Gradient Boost Classifier
              precision    recall  f1-score   support

           0       0.90      0.83      0.86        52
           1       0.89      0.94      0.91        79

    accuracy                           0.89       131
   macro avg       0.89      0.88      0.89       131
weighted avg       0.89      0.89      0.89       131

=========================================================
```

Figure 43. Classifiers for Latent Variable

Classifier for Dataset 2 and 3:

```
=========================================================
Classifier:  Gradient Boost Classifier
              precision    recall  f1-score   support

           0       0.90      0.85      0.87        52
           1       0.90      0.94      0.92        79

    accuracy                           0.90       131
   macro avg       0.90      0.89      0.90       131
weighted avg       0.90      0.90      0.90       131

=========================================================
```

Figure 44. Classifiers for Dataset 2 and 3

Confusion Matrix for Dataset 1:

```
Confusion matrix:
 [[37 15]
 [12 67]]
```

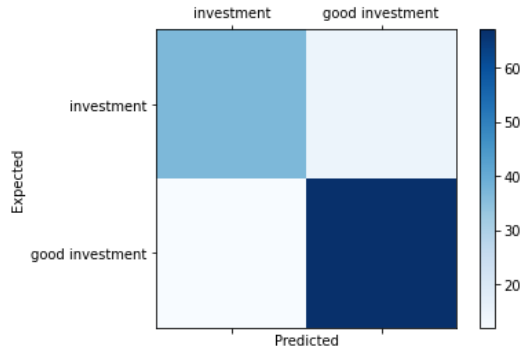

Figure 45. Confusion Matrix for Dataset 1

Confusion Matrix for Dataset 2 and 3:

```
Confusion matrix:
 [[42 10]
 [11 68]]
```

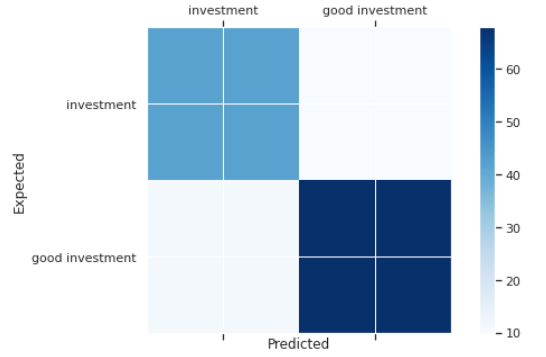

Figure 46. Confusion Matrix for Dataset 2 and 3

Confusion Matrix for Latent Variable:

```
Confusion matrix:
 [[41 11]
 [11 68]]
```

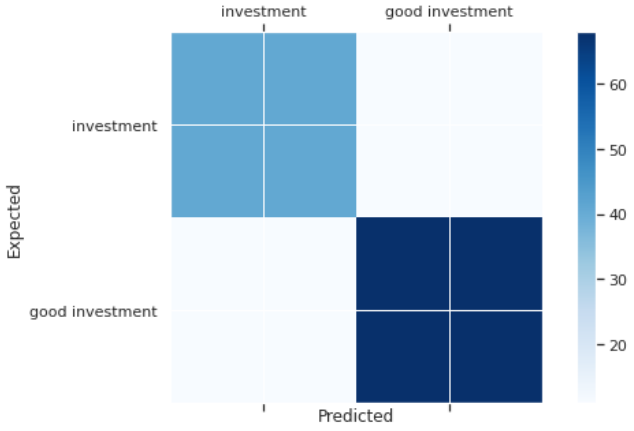

Figure 47. Confusion Matrix for Latent Variable

Regressors for Dataset 1:

```
Regressor:  KNN Regressor
Mean Absolute Error: 0.09116358325219098
================================================================================
Regressor:  Gradient Boost Regressor
Mean Absolute Error: 0.5639799816242481
================================================================================
Regressor:  Random Forest Regressor
Mean Absolute Error: 0.3546640416722404
================================================================================
Regressor:  MLP Regressor
Mean Absolute Error: -482786699.5416125
================================================================================
Regressor:  Linear Regressor
Mean Absolute Error: 0.32631677032487705
================================================================================
--------------------------------------------------------------------------------
Best --> Regressor = Gradient Boost Regressor, Accuracy Score = 0.56
```

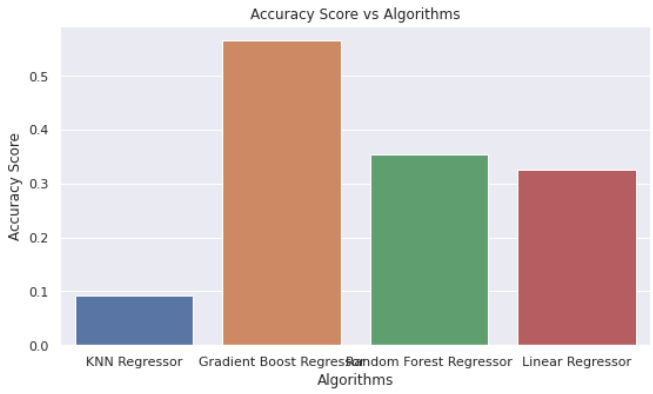

Figure 48. Regressors for Dataset 1

Regressors for Dataset 2 and 3:

```
Regressor:  KNN Regressor
Mean Absolute Error: 0.09116358325219098
================================================================================
Regressor:  Gradient Boost Regressor
Mean Absolute Error: 0.6760027323638835
================================================================================
Regressor:  Random Forest Regressor
Mean Absolute Error: 0.4000678655787151
================================================================================
Regressor:  MLP Regressor
Mean Absolute Error: -773665663.9689975
================================================================================
Regressor:  Linear Regressor
Mean Absolute Error: 0.3577438738216119
================================================================================
--------------------------------------------------------------------------------
Best --> Regressor = Gradient Boost Regressor, Accuracy Score = 0.68
```

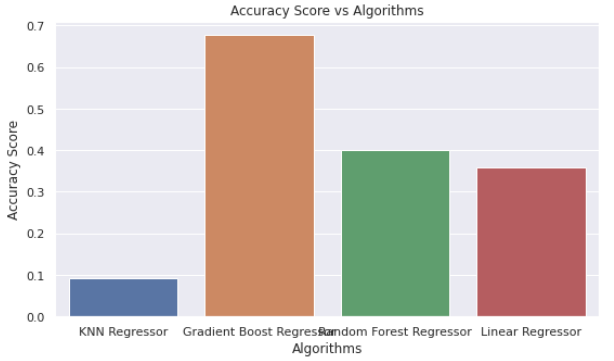

Figure 49. Regressors for Dataset 2 and 3

Regressor for Latent Variable:

```
Regressor:  KNN Regressor
Mean Absolute Error: 0.15494157740993197
================================================================================
Regressor:  Gradient Boost Regressor
Mean Absolute Error: 0.6846861585766824
================================================================================
Regressor:  Random Forest Regressor
Mean Absolute Error: 0.39888773643459075
================================================================================
Regressor:  MLP Regressor
Mean Absolute Error: -308012757916210.8
================================================================================
Regressor:  Linear Regressor
Mean Absolute Error: 0.36253656194448813
================================================================================
--------------------------------------------------------------------------------
Best --> Regressor = Gradient Boost Regressor, Accuracy Score = 0
```

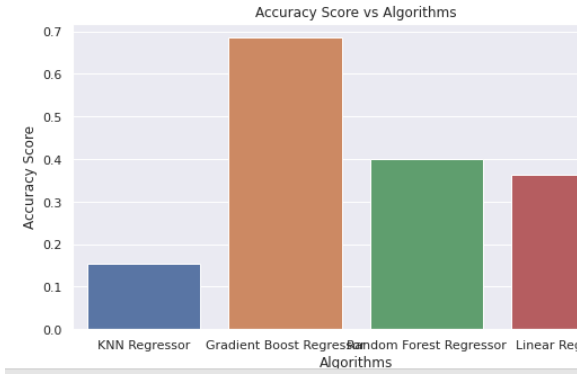

Figure 50. Regressors for Latent Variable

Inference:
From the above reports and matrix, it is clear that the accuracy has been increased from first dataset to 2nd and 3rd dataset to adding latent variables, i.e. from
Regressor: 0.56 -> 0.67 -> 0.69
Classifier: 0.80 -> 0.89 -> 0.90

6. Conclusion

Using fractal clustering, we can get more fine grained results which is more accurate than increasing the overall number of clusters initially. It helps in understanding which cluster to focus upon and do more in depth analysis on it. Also we found out feature importance can help a lot in using the important features and also reducing noise, for this we have leveraged gini scores, correlation matrix, PCA and shapley values. In this project, through fractal clustering we were able to find out the most profitable investment properties in San Diego through the Zillow dataset. These properties have the potential to generate positive cash flow through the property value minus expenses calculation. We can also conclude that enriching the data using amalgamation has resulted in more accuracy in classifiers and regressors. To find out which classifiers and regressors works best, we have created a loop to run all the classifiers and regressors and analyzed their metrics using accuracy, precision, recall and Mean Absolute Error. For future work, this project can be extended to other areas and also the data can be enriched with factors like past hazards, disaster prone zone, schools rating, neighborhood, amenities etc, which would give further insight into these properties.

7. References

[1] Eduard Hromada, *Real estate valuation using data mining software*, Creative Construction Conference 2016, CCC 2016, 25-28 June 2016

[2] Vahid Moosavi, *Urban Data Streams and Machine Learning: A case of swiss real estate market*

[3] Zahratu Shabrina, *Elsa Arcuate, Michael Batty, Airbnb's disruption of the housing structure in London*

[4] Victoria Rayskin, *Multivariate time series approximation by multiple trajectories of a dynamical system. Applications to internet traffic and COVID-19 data.*