Computer Vision

Predicting housing prices using a ML algorithm – a KMeans approach

Student:
Vișinescu Ioan-Valentin

Table of Contents

1. INTRODUCTION

In today’s world, we encounter a very dynamic landscape in the domain of house prices, a fact that affects our everyday life. The intersection of housing markets, geographic locations, and local salaries constitutes a dynamic landscape that profoundly influences economic well-being, lifestyle choices, and community development within a region. A better location of your home can offer a person access to better schools, services, and most importantly more time, as you might get whatever you need close to your home. This will reflect in the home prices in a certain region, prices that will be closely related to the purchasing power of the people living in that neighborhood or city.

With this information, I’ve decided to analyze the prices of houses in California and correlate it with the household income. The intricate relationships between housing affordability, geographic location, and income levels form the backbone of this investigation. The California housing market, known for its diversity and complexity, provides a unique opportunity to delve into the dynamics that shape the real estate landscape in one of the most economically vibrant regions.

As we embark on this analytical journey, the use of advanced techniques such as K-Means clustering allows us to uncover hidden patterns within the vast dataset, offering insights into the clustering of house prices and the economic regions they represent. By combining the power of data analytics with geospatial visualization, we aim to provide a comprehensive understanding of how different clusters of house prices align with the economic well-being of various regions in California.

Our investigation goes beyond mere statistical analyses; it aims to decode the nuanced correlations between house prices, location advantages, and local income distribution. The ultimate goal is to empower individuals, investors, policymakers, and urban planners with actionable insights for making informed decisions in the dynamic Californian real estate landscape.

Next, we will navigate through the intricacies of California’s housing market, unraveling the stories told by data and drawing meaningful connections that can shape the future of our communities.

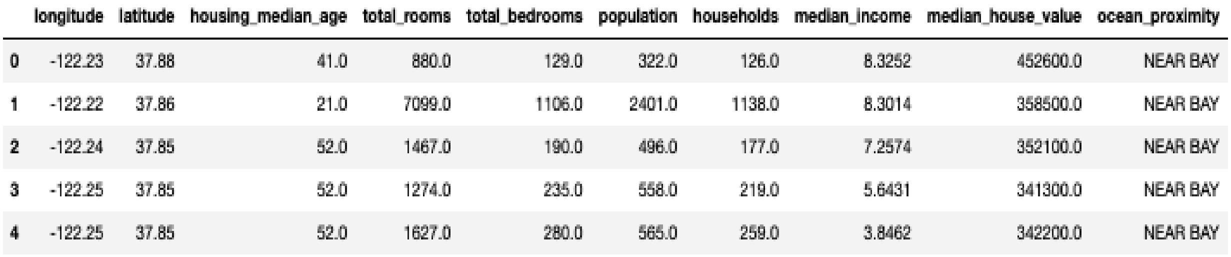
2. DATASET OVERVIEW

First and foremost, the foundation of our analytical journey rested on acquiring an adequate dataset. The search for a comprehensive and relevant dataset was a crucial initial step, as the accuracy and richness of our findings heavily depended on the quality and depth of the data at hand. This process involved meticulous exploration and selection to ensure that the dataset encapsulated the essential elements—longitude, latitude, house prices, and salaries—essential for unraveling the intricate dynamics of California's real estate landscape. The quest for the right dataset marked the commencement of our exploration into the complex interplay between housing prices, geographical locations, and income levels.

The dataset selected is called "California Housing Prices" as it included the longitude and latitude of the house, the *median_house_value* and the *median_inclome*. This dataset contains the information from the 1990 California census, so although it may not help with predicting current housing prices, it is a good start in machine learning. The dataset was obtained from Kaggel [1]. This dataset contains:

- longitude
- latitude
- housing_median_age
- total_rooms
- total_bedrooms
- population
- households
- median_income
- median_house_value
- ocean_proximity

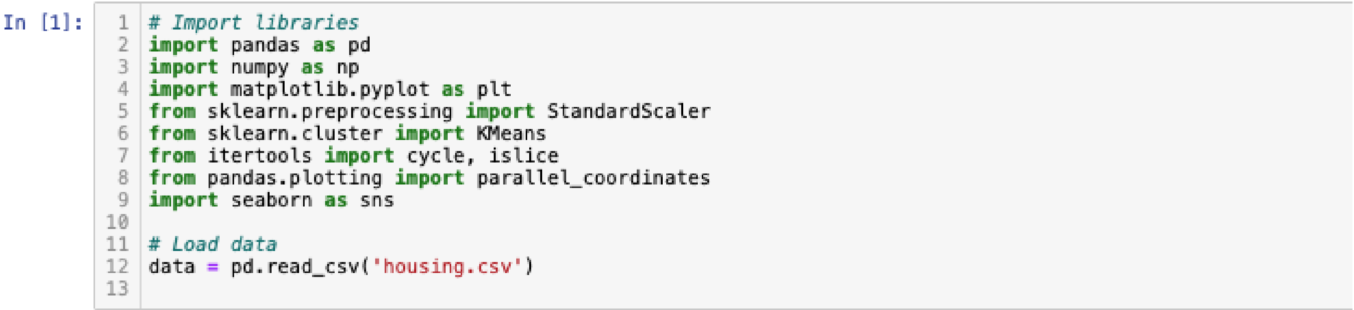
The goal of this project it is to see the correlation between the area, the household income and the home price. All this data is saved in a CSV file in order to be read by python.



	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY

3. METHODOLOGY

The first step in this project was the import of all necessary libraries for data processing, numerical operations, machine learning and plotting. These are: pandas, numpy, sklearn and matplotlib. At the end library cartopy was added in order to overlay the house locations with the map of California. Our data frame was simply defined as "data" and the information was read from "housing.csv"



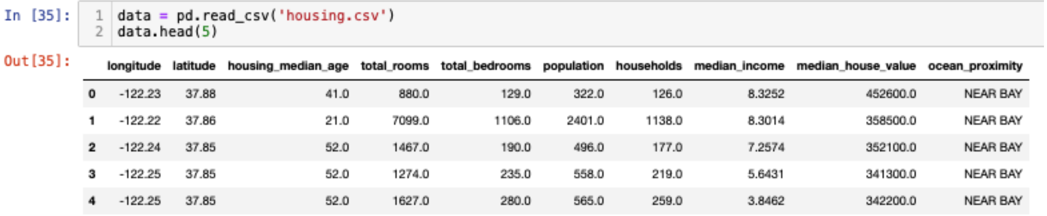
```
In [1]:
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from itertools import cycle, islice
from pandas.plotting import parallel_coordinates
import seaborn as sns

# Load data
data = pd.read_csv('housing.csv')

```

As it can be seen in the code above, each library has a specific role in the project:

- Pandas is used for reading and writing spreadsheets.
- Numpy for carrying out efficient computations.
- Matplotlib for visualization

In the initial stages of this project, it is imperative to perform some preprocessing on the data contained in the CSV file. [2] This involves the careful selection of features essential for the project, discarding those that are extraneous. This strategic feature selection ensures that the input data is streamlined, facilitating ease of readability and enhancing the overall effectiveness of our analysis. The features selected are the *longitude, latitude, median_house_value* and the *median_income.*

To see the data imported the head of the table was printed out in Jupyter notebooks.

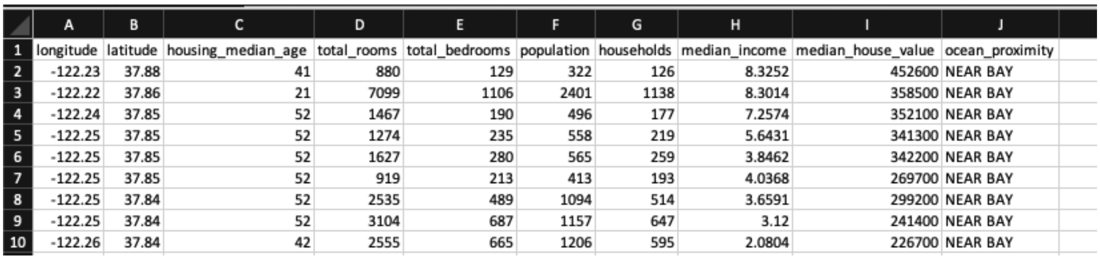


```
In [35]:
data = pd.read_csv('housing.csv')
data.head(5)
```

Out[35]:

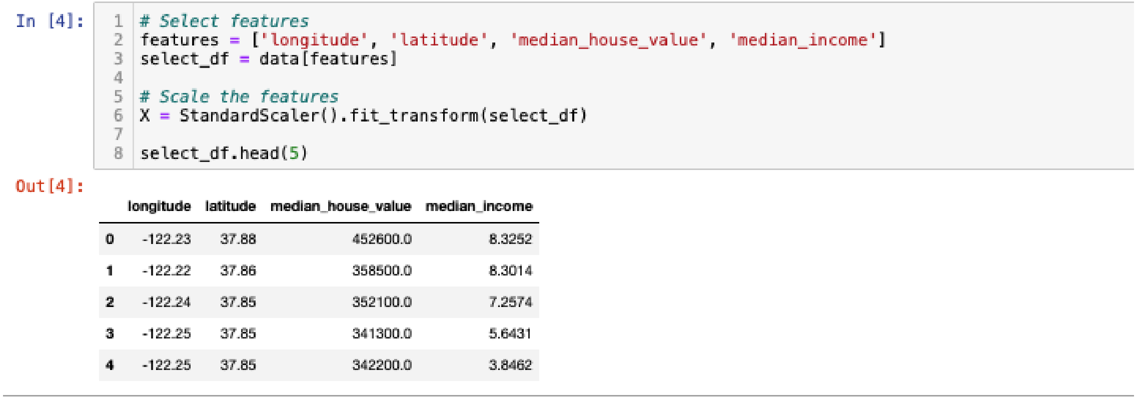
	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY

As it can be observed, the data is the same as the one present in CSV file.



	A	B	C	D	E	F	G	H	I	J
1	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
2	-122.23	37.88	41	880	129	322	126	8.3252	452600	NEAR BAY
3	-122.22	37.86	21	7099	1106	2401	1138	8.3014	358500	NEAR BAY
4	-122.24	37.85	52	1467	190	496	177	7.2574	352100	NEAR BAY
5	-122.25	37.85	52	1274	235	558	219	5.6431	341300	NEAR BAY
6	-122.25	37.85	52	1627	280	565	259	3.8462	342200	NEAR BAY
7	-122.25	37.85	52	919	213	413	193	4.0368	269700	NEAR BAY
8	-122.25	37.84	52	2535	489	1094	514	3.6591	299200	NEAR BAY
9	-122.25	37.84	52	3104	687	1157	647	3.12	241400	NEAR BAY
10	-122.26	37.84	42	2555	665	1206	595	2.0804	226700	NEAR BAY

And afterwards the relevant data columns were selected.



In [4]:

```
# Select features
features = ['longitude', 'latitude', 'median_house_value', 'median_income']
select_df = data[features]

# Scale the features
X = StandardScaler().fit_transform(select_df)

select_df.head(5)
```

Out[4]:

	longitude	latitude	median_house_value	median_income
0	-122.23	37.88	452600.0	8.3252
1	-122.22	37.86	358500.0	8.3014
2	-122.24	37.85	352100.0	7.2574
3	-122.25	37.85	341300.0	5.6431
4	-122.25	37.85	342200.0	3.8462

The method selected to perform a spatial data analysis is K-Means clustering. This algorithm is one of the simplest and most popular unsupervised machine learning algorithms. From an online article [3] we can get a good definition of k-means clustering and why is it that powerful: “The objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (*k*) of clusters in a dataset.”

A cluster refers to a collection of data points aggregated together because of certain similarities. In our case we will have two types of clusters, one based on the house price and one based on the income of the household. [3]

There is a need to define a target number k, which refers to the number of centroids. A centroid can be defined as the “imaginary or real location representing the center of the cluster”. Each data point is then allocated to a cluster through reducing the incluster sum of squares. More explicitly, the K-means algorithm “identifies *k* number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.” [3]

The K-means algorithm is applied through Scikit-learn library. This library offers a wide range of machine learning algorithms, integration with other Python libraries, tools for data preprocessing, model evaluation, and cross-validation. All the data points in a cluster should be similar to each other and also different clusters should be as different as possible.

In order to see if there is a connection between the home prices and the incomes, the number of cluster is needed. In order to find it the elbow method was used. The elbow method is a technique used to determine the optimal number of clusters in a clustering algorithm, particularly in K-Means clustering. It involves running the clustering algorithm for a range of cluster numbers (k) and plotting the explained variation or inertia as a function of k. The "elbow" of the curve is then inspected to identify the point at which adding more clusters does not significantly reduce the inertia. There are some steps needed to find the right point: [2]

1. **Run K-Means for Different Cluster Numbers (k)** - Perform K-Means clustering for a range of cluster numbers (for example, from 1 to 10 clusters).

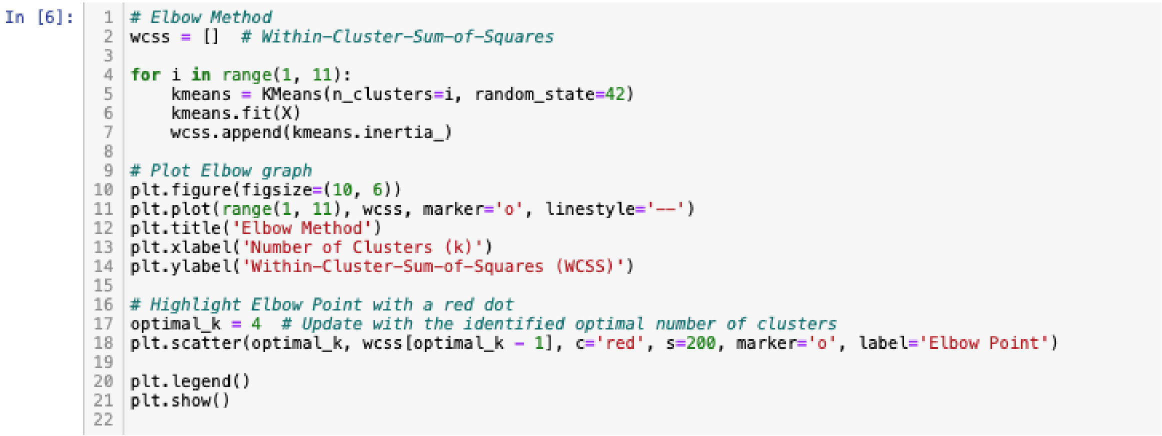
2. **Calculate Inertia (Within-Cluster Sum of Squares)** - For each value of k, calculate the inertia, which is the sum of squared distances between data points and their assigned cluster centers.

3. **Plot the Elbow Curve** - Plot the values of k against the corresponding inertias.The curve typically resembles an "elbow" shape.

4. **Identify the Elbow Point** - Look for the point on the curve where adding more clusters provides diminishing returns in terms of reducing inertia. The "elbow" is the point where the rate of decrease in inertia slows down.

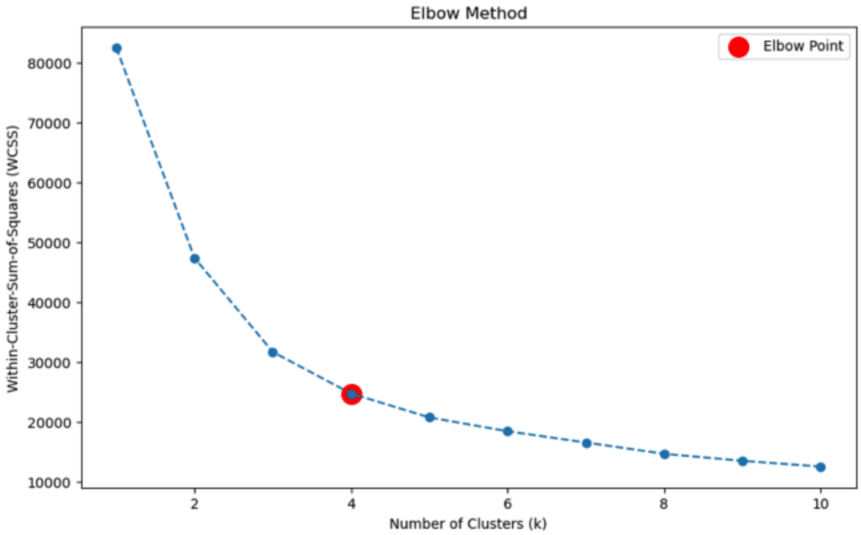
5. **Choose the Optimal Number of Clusters** - The number of clusters corresponding to the elbow point is considered the optimal choice for k.

The rationale behind the elbow method is that as the number of clusters increases, the inertia tends to decrease. However, beyond a certain point, the reduction in inertia becomes less significant (forming an "elbow" in the curve), indicating that additional clusters may not capture substantially more information about the data's structure.



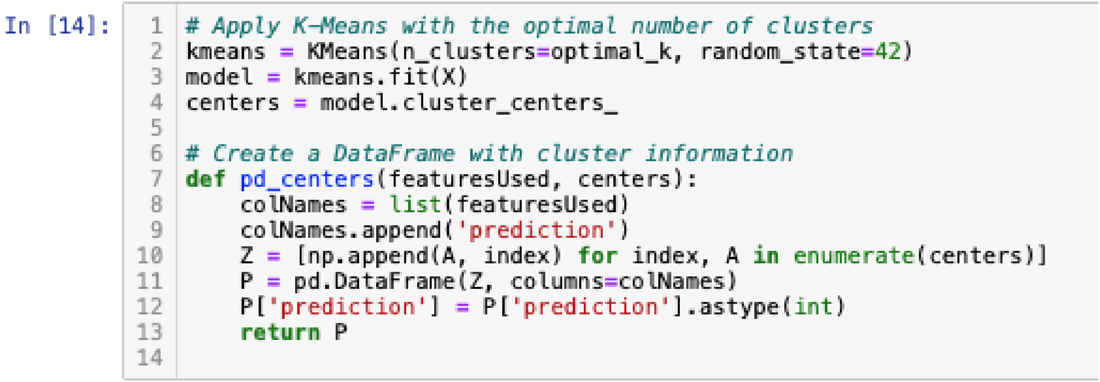
```
# Elbow Method
wcss = []  # Within-Cluster-Sum-of-Squares

for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

# Plot Elbow graph
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), wcss, marker='o', linestyle='--')
plt.title('Elbow Method')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Within-Cluster-Sum-of-Squares (WCSS)')

# Highlight Elbow Point with a red dot
optimal_k = 4  # Update with the identified optimal number of clusters
plt.scatter(optimal_k, wcss[optimal_k - 1], c='red', s=200, marker='o', label='Elbow Point')

plt.legend()
plt.show()

```

The result is in the form of a graph:



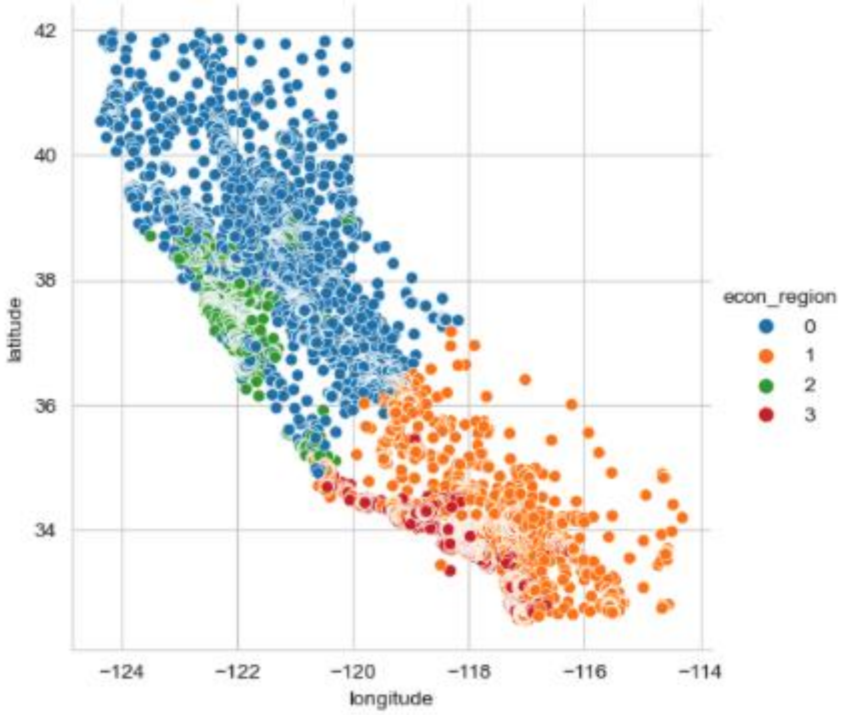
As it can be seen in the graph above the curve is more linear after the k = 4, so we can conclude that inertia is less significant after this point and the optimal number of clusters is 4.

After finding the right number of clusters, K-means is applied.



```
In [14]:
# Apply K-Means with the optimal number of clusters
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
model = kmeans.fit(X)
centers = model.cluster_centers_

# Create a DataFrame with cluster information
def pd_centers(featuresUsed, centers):
    colNames = list(featuresUsed)
    colNames.append('prediction')
    Z = [np.append(A, index) for index, A in enumerate(centers)]
    P = pd.DataFrame(Z, columns=colNames)
    P['prediction'] = P['prediction'].astype(int)
    return P

```

Then the statistics for each cluster is printed:

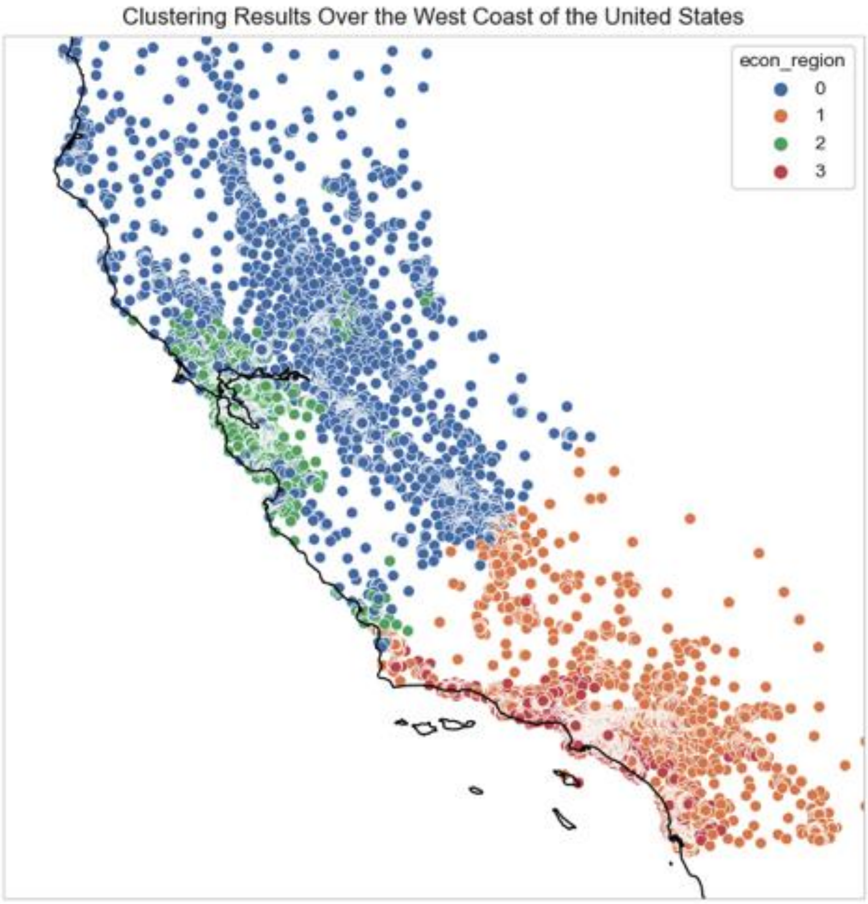


The median house value (in USD) for each econ_region is as it follows:

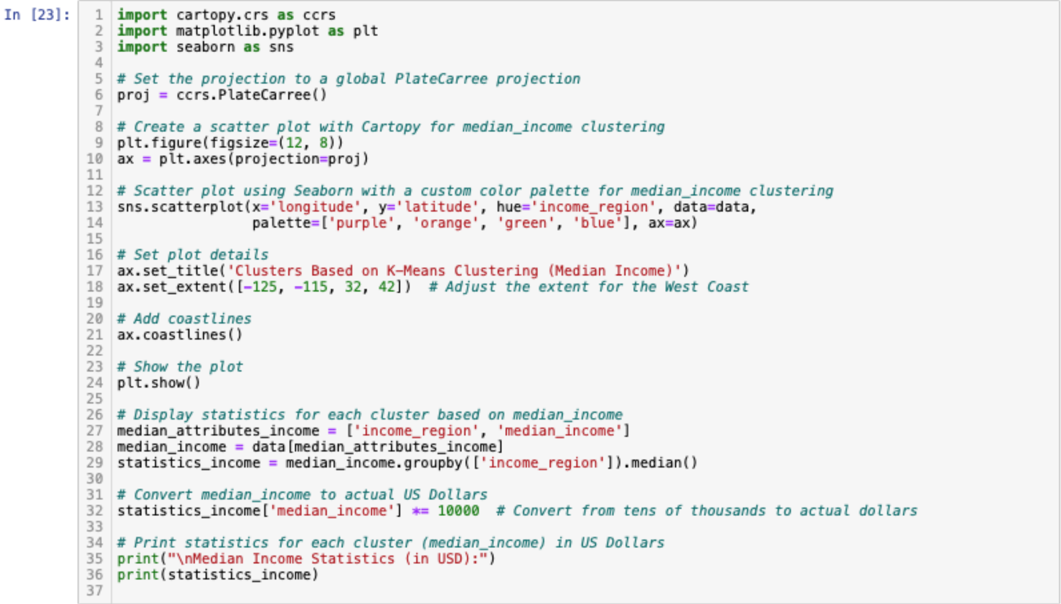
Econ_region	Value (in USD)
0	$122.000
1	$162.050
2	$319.100
3	$353.100

Upon examining the table, a notable trend emerges: residences situated inland exhibit an appraisal value that is more than 50% lower than their counterparts nearer to the ocean. This observation underscores the substantial impact of location on housing prices, prompting further consideration of the nuanced factors influencing real estate valuations.

Over the map presented above, the map of California was also printed to have a more user-friendly interface.

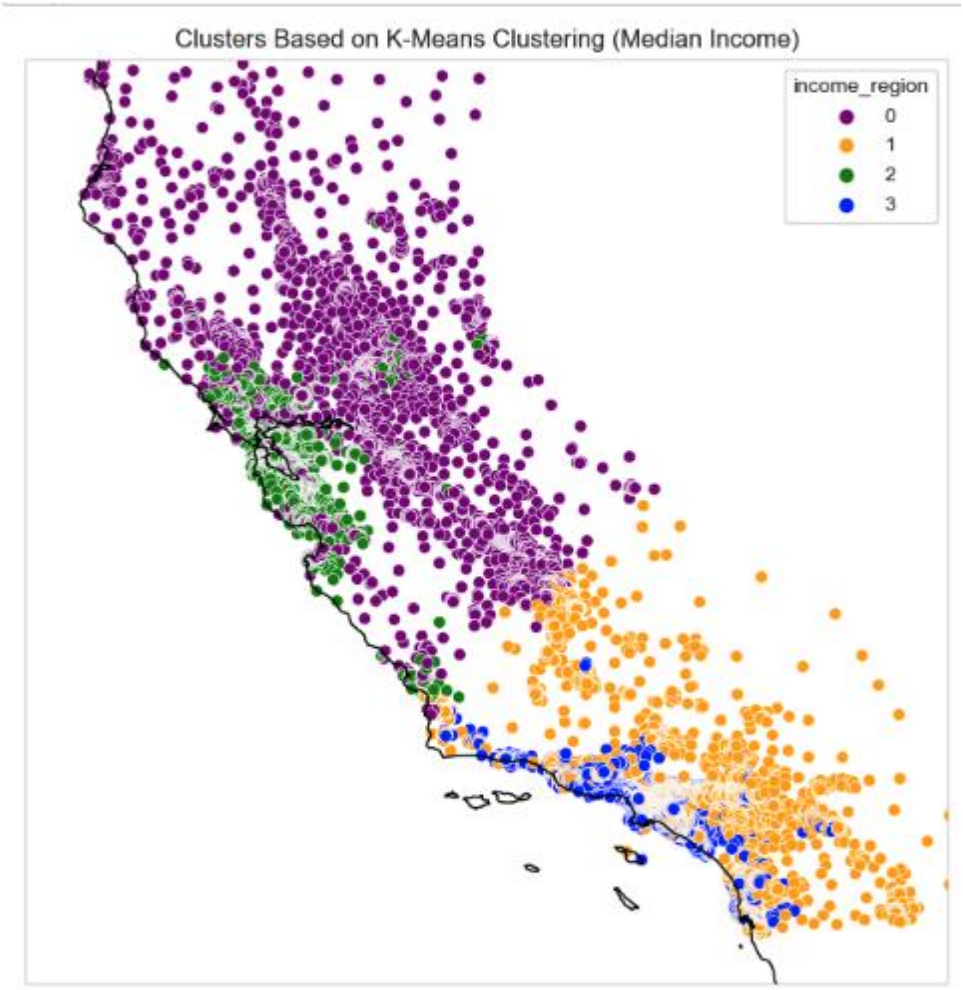


After that the median income was analyzed trough the K-means clustering algorithm also.



```python
import cartopy.crs as ccrs
import matplotlib.pyplot as plt
import seaborn as sns

# Set the projection to a global PlateCarree projection
proj = ccrs.PlateCarree()

# Create a scatter plot with Cartopy for median_income clustering
plt.figure(figsize=(12, 8))
ax = plt.axes(projection=proj)

# Scatter plot using Seaborn with a custom color palette for median_income clustering
sns.scatterplot(x='longitude', y='latitude', hue='income_region', data=data,
                palette=['purple', 'orange', 'green', 'blue'], ax=ax)

# Set plot details
ax.set_title('Clusters Based on K-Means Clustering (Median Income)')
ax.set_extent([-125, -115, 32, 42])  # Adjust the extent for the West Coast

# Add coastlines
ax.coastlines()

# Show the plot
plt.show()

# Display statistics for each cluster based on median_income
median_attributes_income = ['income_region', 'median_income']
median_income = data[median_attributes_income]
statistics_income = median_income.groupby(['income_region']).median()

# Convert median_income to actual US Dollars
statistics_income['median_income'] *= 10000  # Convert from tens of thousands to actual dollars

# Print statistics for each cluster (median_income) in US Dollars
print("\nMedian Income Statistics (in USD):")
print(statistics_income)

```

The output it is also splited in four clusters so the data can be compared. Then it is plotted over the map of California:



The median income (in USD) for each econ_region is as it follows:

Econ_region	Value (in USD)
0	$28.750
1	$31.387
2	$52.855
3	$58.872

If we compare these two tables, we can observe that the value of the house is closely related to the median income. This can result in more affordable housing options being clustered in regions with lower median incomes, while higher-income areas may exhibit higher house values.

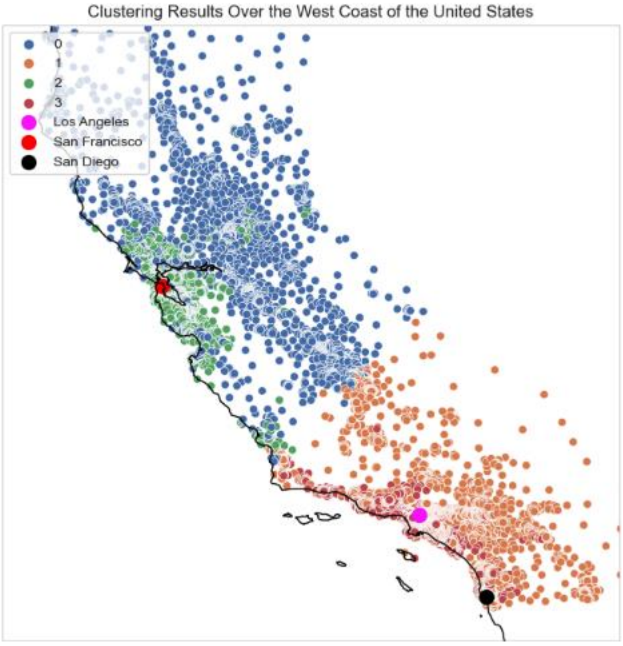
This interdependence underscores the socioeconomic factors influencing the real estate landscape. The affordability of housing appears to be intricately linked to the economic well-being of the community. Such insights are crucial for various stakeholders, including potential homebuyers, policymakers, and investors, as they navigate the complex terrain of the housing market.

For individuals, understanding this correlation can guide decisions related to home purchasing, helping them align their preferences with their budgetary constraints. Policymakers might leverage this information to formulate targeted strategies aimed at addressing housing affordability challenges in specific income brackets.

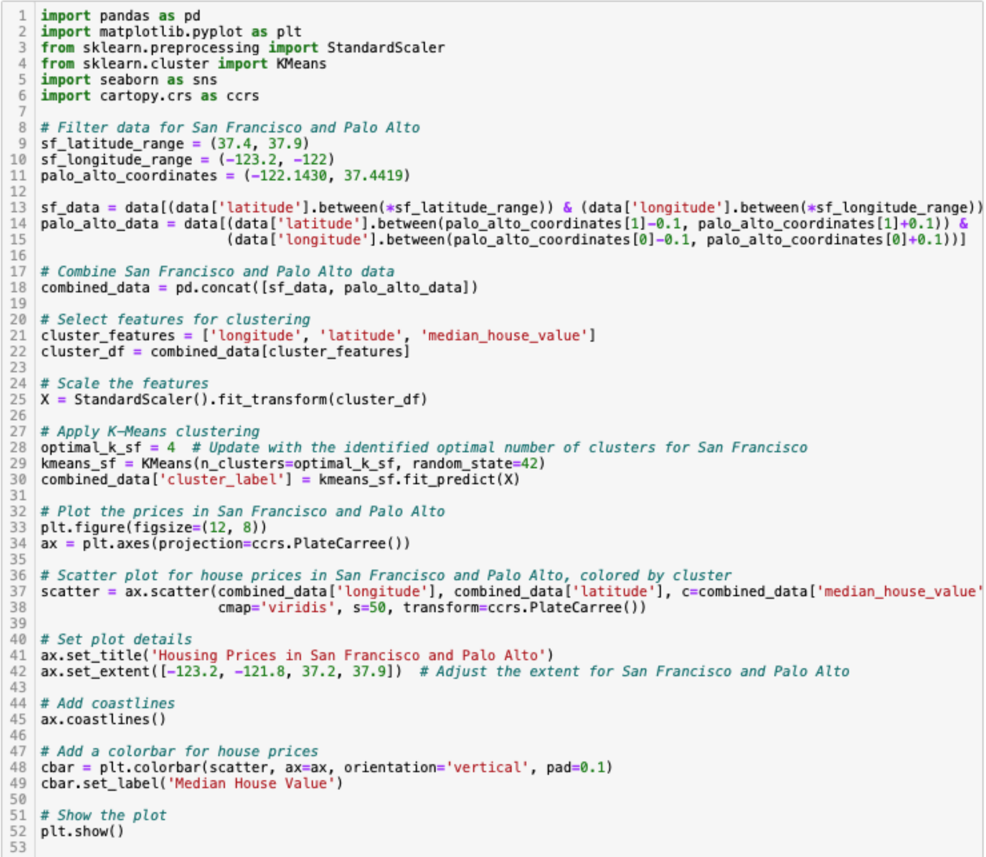
Investors, on the other hand, could use these insights to make informed decisions about real estate portfolios, recognizing the potential for varying returns based on the economic characteristics of different regions. By acknowledging and analyzing these relationships, a more comprehensive understanding of the housing market emerges, offering valuable perspectives for strategic decision-making and informed interventions in the broader economic landscape.

Moreover, this may result in poorer and under-developed areas because investors might be less attracted to invest in business in the lower income zones.

We can also see that the four types of prices are sometimes overlayed like in these areas around the big cities

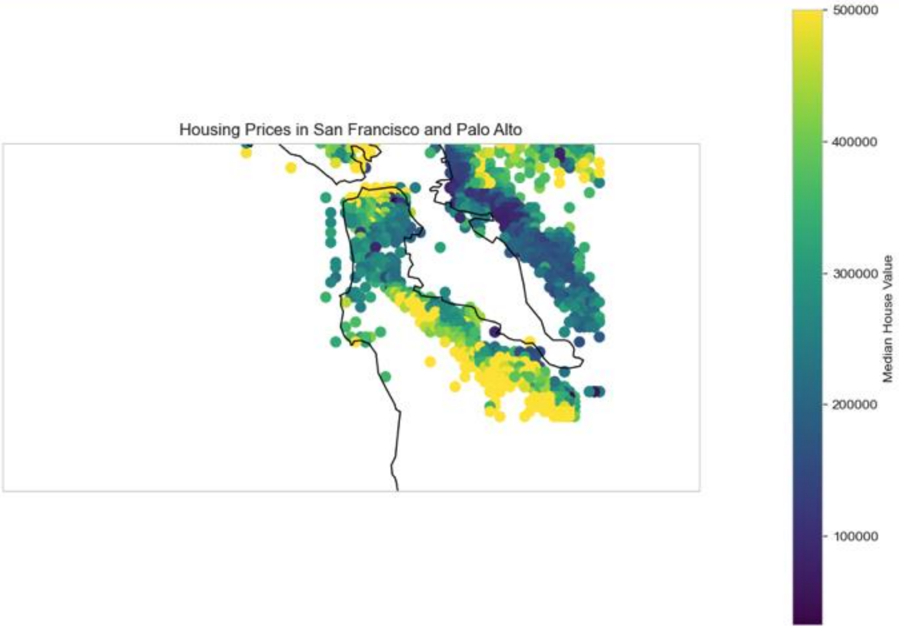


In order to check if this is a trend that goes on into the urban city areas, the researched then focused on San Francisco.

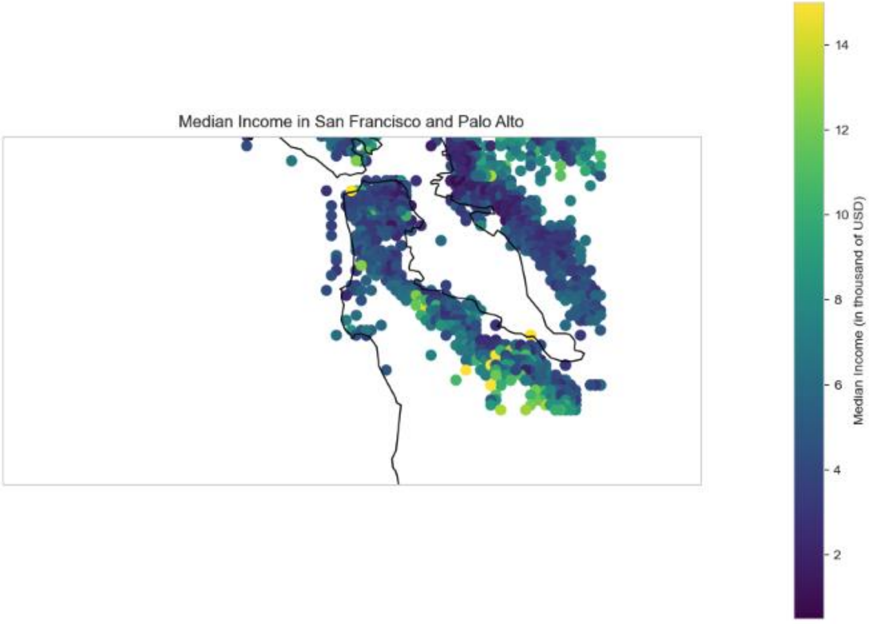


```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import seaborn as sns
import cartopy.crs as ccrs

# Filter data for San Francisco and Palo Alto
sf_latitude_range = (37.4, 37.9)
sf_longitude_range = (-123.2, -122)
palo_alto_coordinates = (-122.1430, 37.4419)

sf_data = data[(data['latitude'].between(*sf_latitude_range)) & (data['longitude'].between(*sf_longitude_range))
palo_alto_data = data[(data['latitude'].between(palo_alto_coordinates[1]-0.1, palo_alto_coordinates[1]+0.1)) &
                      (data['longitude'].between(palo_alto_coordinates[0]-0.1, palo_alto_coordinates[0]+0.1))]

# Combine San Francisco and Palo Alto data
combined_data = pd.concat([sf_data, palo_alto_data])

# Select features for clustering
cluster_features = ['longitude', 'latitude', 'median_house_value']
cluster_df = combined_data[cluster_features]

# Scale the features
X = StandardScaler().fit_transform(cluster_df)

# Apply K-Means clustering
optimal_k_sf = 4  # Update with the identified optimal number of clusters for San Francisco
kmeans_sf = KMeans(n_clusters=optimal_k_sf, random_state=42)
combined_data['cluster_label'] = kmeans_sf.fit_predict(X)

# Plot the prices in San Francisco and Palo Alto
plt.figure(figsize=(12, 8))
ax = plt.axes(projection=ccrs.PlateCarree())

# Scatter plot for house prices in San Francisco and Palo Alto, colored by cluster
scatter = ax.scatter(combined_data['longitude'], combined_data['latitude'], c=combined_data['median_house_value'
                     cmap='viridis', s=50, transform=ccrs.PlateCarree())

# Set plot details
ax.set_title('Housing Prices in San Francisco and Palo Alto')
ax.set_extent([-123.2, -121.8, 37.2, 37.9])  # Adjust the extent for San Francisco and Palo Alto

# Add coastlines
ax.coastlines()

# Add a colorbar for house prices
cbar = plt.colorbar(scatter, ax=ax, orientation='vertical', pad=0.1)
cbar.set_label('Median House Value')

# Show the plot
plt.show()
```

As it can be observed, the prices tend to be higher in the area around the bay and the ocean. Over the other side of San Francisco the prices tend to be lower, with some exceptions. Also, we can see that the houses in the region of Palo Alto, where multiple tech businesses such as Apple and Google are located and also where the Stanford University is located, are higher.



Due to the salaries in the tech area we can also observe that the median income for that area tends to be a bit higher as well but not for all the people as the higher income households are way fewer than the number of houses with a high values as it can be seen on this graph:


Median Income in San Francisco and Palo Alto
Median income (in thousand of USD)
14
12
10
8
6
4
2

4. CONCLUSION

The clustering analysis of median income in San Francisco and Palo Alto reveals a discernible relationship between income levels and housing values. In many areas, the median income appears to be closely related to the median house value, indicating a proportional economic landscape where housing affordability aligns with local incomes.

However, it is crucial to acknowledge the disparities observed in certain regions. Despite the correlation between income and house value, there are instances where housing prices surpass the local income levels. This discrepancy may lead to higher poverty rates as residents face challenges in affording housing within these areas.

These findings emphasize the need for a nuanced approach to housing policies, taking into account the specific economic dynamics of each region. Addressing the affordability gap requires targeted interventions that consider not only the median income but also the distribution of house values within different income brackets. Additionally, broader socioeconomic factors influencing housing disparities should be thoroughly examined to formulate comprehensive strategies aimed at fostering equitable and sustainable communities.

BIBLIOGRAPHY

[1] "kaggel," [Online]. Available: https://www.kaggle.com/datasets/camnugent/california-housing-prices/discussion. [Accessed 27 12 2023].

[2] P. Sharma, "analyticsvidhya," 3 11 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/. [Accessed 07 01 2024].

[3] "towardsdatascience," 13 09 2018. [Online]. Available: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1. [Accessed 10 01 2024].

[4] "wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/K-means_clustering. [Accessed 07 01 2024].