MM20B007 Tutorial 2

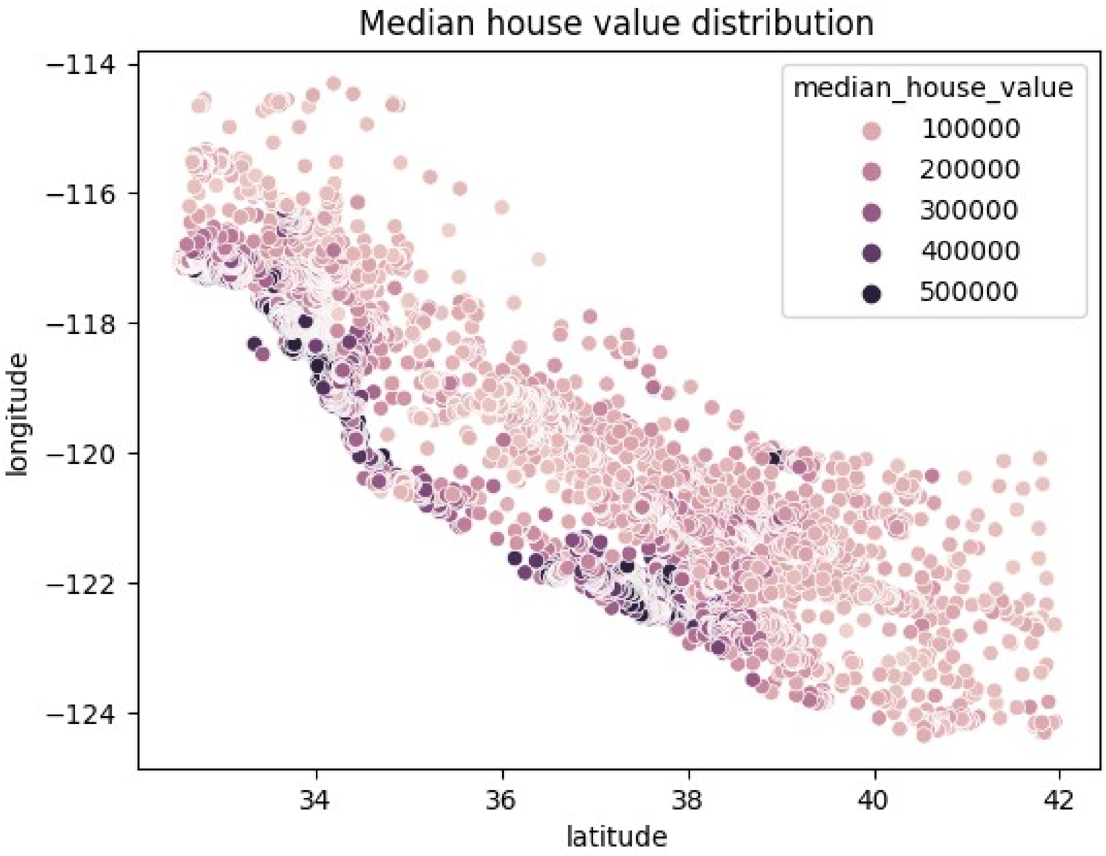
Getting necessary packages

```
import pandas as pd
from sklearn import preprocessing
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import silhouette_score
```

Reading and Visualizing the data

```
data = pd.read_csv('/content/drive/MyDrive/sem 7/ID5055/k-
Means_Tutorial/housing.csv', usecols = ['longitude', 'latitude',
'median_house_value'])
df = data.drop(columns = ['median_house_value'])
df_n = preprocessing.normalize(df)
plt.figure()
plt.title('Median house value distribution')
sns.scatterplot(data = data, x = 'latitude', y = 'longitude', hue=
'median_house_value')
```

```
<Axes: title={'center': 'Median house value distribution'},
xlabel='latitude', ylabel='longitude'>
```



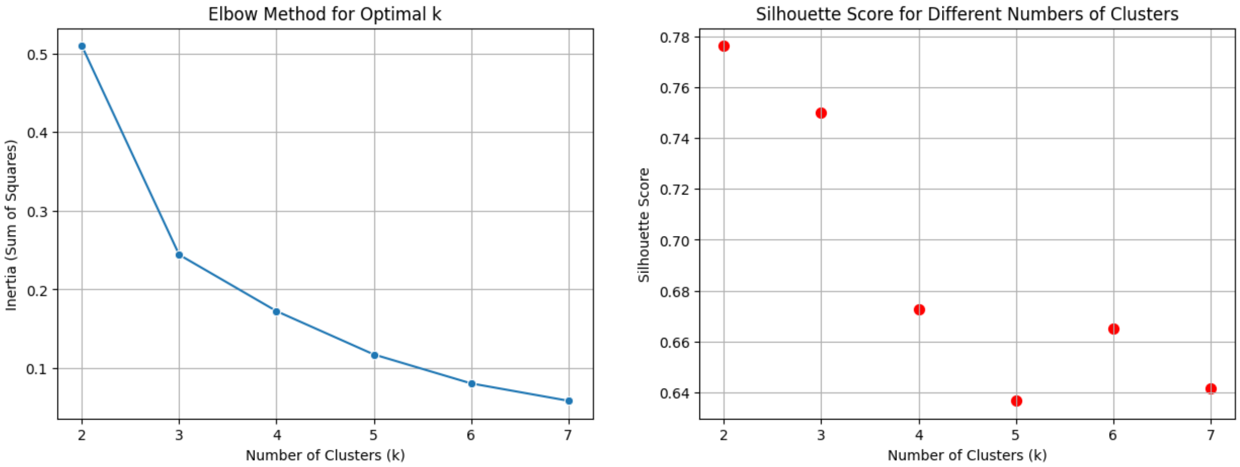
Trying different values of k within the interval [2, 7]

```
k_values = range(2, 8)
inertia_values_elbow_met = []
inertia_values_silhouette_met = []

for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=0, n_init = 'auto')
    kmeans.fit(df_n)
    inertia_values_elbow_met.append(kmeans.inertia_)
    score = silhouette_score(df_n, kmeans.labels_, metric = 
'euclidean')
    inertia_values_silhouette_met.append(score)
```

Visualizing the k values to get the best value

```
fig, axs = plt.subplots(1, 2, figsize=(15, 5), gridspec_kw={'hspace': 
1})

# Silhouette Score Plot
plt.scatter(list(range(2, 8)), inertia_values_silhouette_met, 
color='red', marker='o', s=50)
axs[1].set_xlabel('Number of Clusters (k)')
axs[1].set_ylabel('Silhouette Score')
```

```
axs[1].set_title('Silhouette Score for Different Numbers of Clusters')
axs[1].grid(True)

# Plot the elbow curve
sns.lineplot(x = k_values, y = inertia_values_elbow_met, marker='o',
ax=axs[0])
axs[0].set_xlabel('Number of Clusters (k)')
axs[0].set_ylabel('Inertia (Sum of Squares)')
axs[0].set_title('Elbow Method for Optimal k')
axs[0].set_xticks(k_values)
axs[0].grid(True)

plt.tight_layout()
plt.show()

<ipython-input-54-5fae4ad98fe7>:18: UserWarning: This figure includes
Axes that are not compatible with tight_layout, so results might be
incorrect.
  plt.tight_layout()
```



1. A higher ***Silhouette score*** indicates that the data point is well matched to its own cluster and poorly matched to neighboring clusters. If the silhouette score is closer to 1, it suggests that the clusters are well-defined and appropriately separated. The optimal number of clusters can be determined by finding the value of k that maximizes the silhouette score. From the right graph it is visible that the ***k = 2*** is the most optimal value.

2. In the ***Elbow method***, a range of cluster numbers (K), spanning from 2 to 7, is investigated. For each chosen K value, the WCSS (Within-Cluster Sum of Squares) or inertia is calculated. This metric measures the sum of squared distances between individual data points and the centroids of their corresponding clusters. Constructing a graph that illustrates the connection between K and WCSS results in a distinctive elbow-shaped pattern. By closely observing the graph, a point of significant curvature change akin to an elbow is identified. The K value

corresponding to this turning point serves as the optimal selection for K, signifying the most appropriate number of clusters. From the left graph it is clear that most optimal value of ***k by Elbow method is 3***.