California Housing Prices Statistical Analysis and Prediction

1. Introduction:

House prices in California depend on several socio-economic and geographical factors; hence, the prediction becomes multi-faceted and very complex. With the increasing demand for affordable housing, it is vital that policymakers, investors, and residents be abreast of the dynamics that drive property values. Using a dataset from the 1990 California census, in this project, we attempt to predict median house prices using machine learning based on influential variables: median income, location, and property characteristics.

2. Objective:

The objective of this project is to carry out a thorough statistical analysis and develop predictive models that can estimate median housing prices in California. Various methods of multiple regression analysis, such as linear regression, ridge, lasso, polynomial, and random forest regressions, will be considered in this study. It will try to find the most accurate model to predict house prices. Given

this, the project tries to find the key drivers of California housing prices using feature importance and evaluate the performance of the model using Machine Learning in real estate valuation.

3. Dataset:

The California Housing Prices dataset from Kaggle shows a snapshot taken from the 1990 census. This dataset has mainly been used for regression tasks in machine learning regarding median house price prediction. Let's take a glimpse into the dataset:

Dataset Features: The dataset includes 20,640 entries with the following 10 columns:

1. longitude: The geographical longitude of the district (float).
2. latitude: The geographical latitude of the district (float).
3. housing_median_age: The median age of the houses in the district (float).
4. total_rooms: The total number of rooms in the district (float).
5. total_bedrooms: The total number of bedrooms in the district (float, some missing values).
6. population: The population of the district (float).
7. households: The number of households in the district (float).
8. median_income: The median income for households in the district, in tens of thousands of dollars (float).
9. median_house_value: The median house value for households in the district, in US dollars (float, target variable).
10. ocean_proximity: The categorical variable indicating proximity to the ocean, with values like NEAR BAY, INLAND, NEAR OCEAN, and ISLAND.

Package Requirements:

In [ ]:
```
!pip install folium
```

Imports:

In [180…
```
import pandas as pd
import seaborn as sns
import folium
```

```
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.preprocessing import PolynomialFeatures
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import root_mean_squared_error, r2_score, mean_absolute_error, roc_curve, auc
from folium.plugins import HeatMap

# Setting up inline plotting
%matplotlib inline
```

Data Import:

In [180…

```
data = pd.read_csv('dataset/housing_original.csv')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object 
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

In [180…

```
data.head()
```

Out[180…

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

These statistics provide a foundational understanding of the dataset's structure, indicating the diverse housing and economic landscape across California.

1. **Longitude and Latitude**: The dataset covers a geographic area with longitude ranging from -124.35 to -114.31 and latitude from 32.54 to 41.95, corresponding to California's general geographic boundaries.
2. **Housing Median Age**: The median housing age is around 28.6 years, with a minimum of 1 and a maximum of 52 years, indicating that most homes fall within a mid-aged range.
3. **Total Rooms and Bedrooms**: Total rooms range from 2 to 39,320, and total bedrooms from 1 to 6,445, highlighting a large variation in housing sizes. Median total rooms and bedrooms are 2,127 and 435, respectively, suggesting that most houses are moderately sized.
4. **Population and Households**: Population per block group varies from as low as 3 to as high as 35,682, while households range from 1 to 6,082, showing significant demographic differences across regions.
5. **Median Income**: Median income ranges widely, from $0.5kto$15k (in tens of thousands), with an average of $3.87k, indicating economic diversity across neighborhoods.
6. **Median House Value (Target Variable)**: House values span from $14,999to$500,001, with a mean of $206,855. The maximum value is capped at 500,001, suggesting the data might have been truncated.

In [180…

```
data.describe()
```

Out[180…

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_inc
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.00
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.87
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.89
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.49
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.56
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.53
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.74
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.00

4. Data Cleaning:

1. **Handling Missing Values**: The missing values in the total_bedrooms column are addressed by imputing them with the median value of the column. This approach maintains the central tendency and mitigates the impact of outliers in the data.

In [180…

```
# Drop rows with missing values
data.dropna(inplace=True)

# Drop duplicate rows
data.drop_duplicates(inplace=True)
```

2. **Detecting and Handling Outliers**: To ensure the dataset's integrity, the Interquartile Range (IQR) method is applied to detect and handle potential outliers, focusing on the median_house_value column. This process helps in reducing the skewness caused by extreme values.

In [180…
```
# Define a function to remove outliers based on the IQR method
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    # Filter out rows with values outside the lower and upper bounds
    return df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]

# Apply outlier removal for 'median_house_value'
data = remove_outliers(data, 'median_house_value').copy()
```

3. **Encoding the Categorical Variable**: The categorical feature ocean_proximity is encoded using one-hot encoding. This method converts the categorical variable into a numeric format that can be used in further analysis and predictive modeling.

In [180…
```
# Perform one-hot encoding for 'ocean_proximity'
data = pd.get_dummies(data, columns=['ocean_proximity'], drop_first=True)
```

4. **Verification of Data Cleaning**: Verifying that the dataset no longer contains missing values and that the categorical variable has been appropriately encoded. This ensures data consistency before proceeding to the analysis phase.

In [181…
```
# Check for any remaining missing values and confirm data structure
data.info()
data.head()
data.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 19369 entries, 0 to 20639
Data columns (total 13 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   longitude                   19369 non-null  float64
 1   latitude                    19369 non-null  float64
 2   housing_median_age          19369 non-null  float64
 3   total_rooms                 19369 non-null  float64
 4   total_bedrooms              19369 non-null  float64
 5   population                  19369 non-null  float64
 6   households                  19369 non-null  float64
 7   median_income               19369 non-null  float64
 8   median_house_value          19369 non-null  float64
 9   ocean_proximity_INLAND      19369 non-null  bool
 10  ocean_proximity_ISLAND      19369 non-null  bool
 11  ocean_proximity_NEAR BAY    19369 non-null  bool
 12  ocean_proximity_NEAR OCEAN  19369 non-null  bool
dtypes: bool(4), float64(9)
memory usage: 1.6 MB
```

Out[181…

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_inc
count	19369.000000	19369.000000	19369.000000	19369.000000	19369.000000	19369.000000	19369.000000	19369.00
mean	-119.563902	35.655784	28.344158	2620.710930	539.893335	1442.285043	501.303991	3.66
std	2.005895	2.151468	12.503931	2187.046669	422.650225	1145.780125	383.339200	1.55
min	-124.350000	32.540000	1.000000	2.000000	2.000000	3.000000	2.000000	0.49
25%	-121.760000	33.930000	18.000000	1440.000000	297.000000	798.000000	282.000000	2.52
50%	-118.510000	34.270000	28.000000	2110.000000	437.000000	1181.000000	411.000000	3.44
75%	-117.990000	37.730000	37.000000	3119.000000	648.000000	1746.000000	606.000000	4.57
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.00

5. Exploratory Data Analysis:

Exploratory Data Analysis (EDA) provides insights into the California housing dataset by visualizing the distribution of house values and exploring relationships between features like median income, location, and ocean proximity. This helps identify key predictors for housing prices and prepares the data for modeling.

1. Distribution Analysis

The Distribution Analysis examines the spread and central tendencies of key features, such as median house value and median income, to understand their variability and highlight any skewness in the dataset. 1. Histogram of Median House Value 1. Distribution of Median Income

In [181…

```
plt.style.use('ggplot')

plt.figure(figsize=(12, 8))
# Plotting the histogram of 'median_house_value' with more bins for granularity
sns.histplot(data['median_house_value'], kde=True, color='#3498db', bins=60)

# Adding vertical lines to show mean and median values for better insight
plt.axvline(x=data['median_house_value'].mean(), color='#e74c3c', linestyle='--', linewidth=2, label='Mean Value')
plt.axvline(x=data['median_house_value'].median(), color='#2ecc71', linestyle='-', linewidth=2, label='Median Value'
plt.title('Distribution of Median House Value in California', fontsize=18)
plt.xlabel('Median House Value (in USD)', fontsize=14)
plt.ylabel('Frequency', fontsize=14)
plt.legend(fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
mean_value = data['median_house_value'].mean()
median_value = data['median_house_value'].median()
plt.text(mean_value + 5000, plt.ylim()[1] * 0.7, f'Mean: ${mean_value:.0f}', color='red', fontsize=12, rotation=90)
plt.text(median_value + 5000, plt.ylim()[1] * 0.5, f'Median: ${median_value:.0f}', color='green', fontsize=12, rotat
plt.show()
print("Figure 5.1.1: Distribution of Median House Value in California")
```

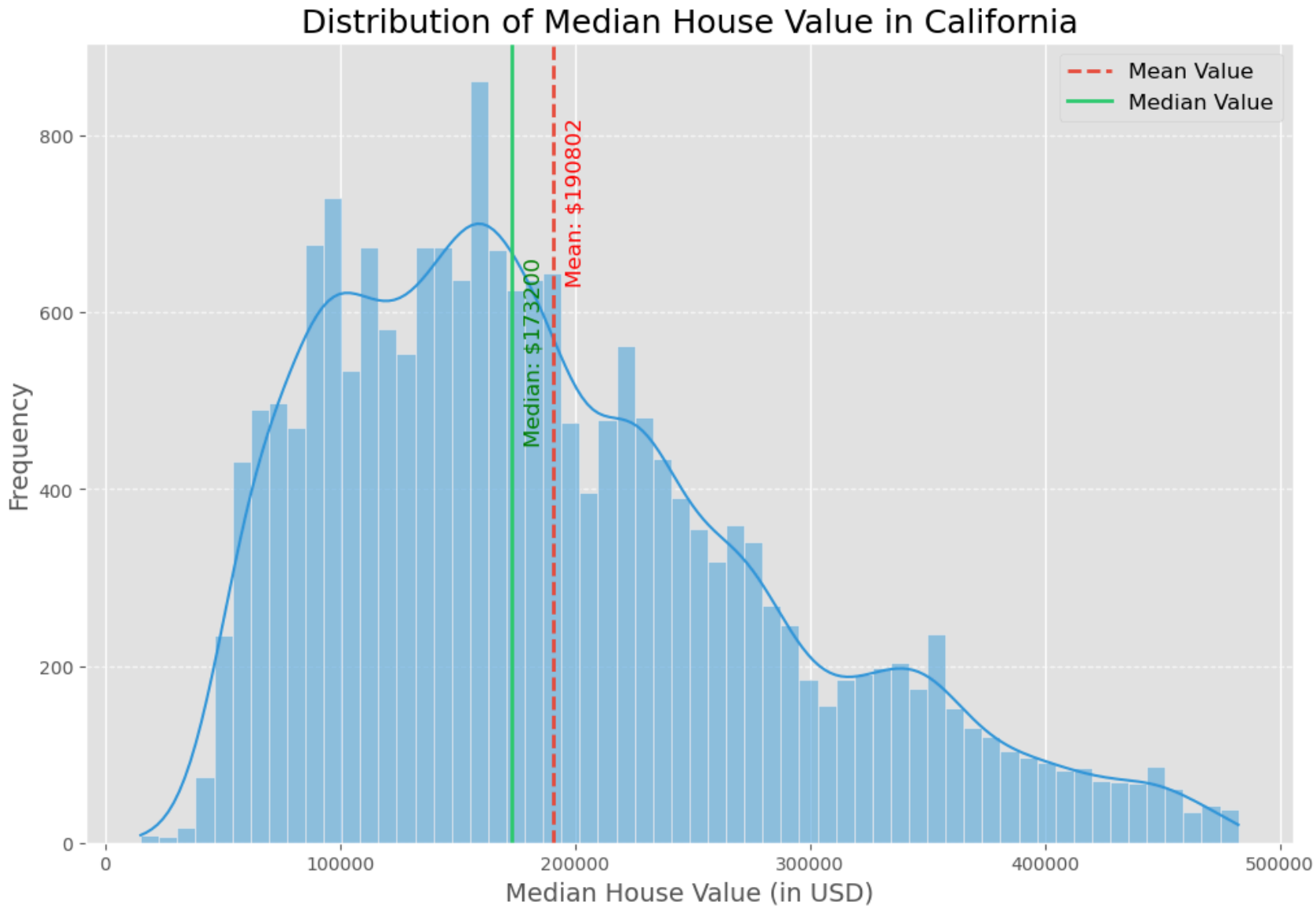


Figure 5.1.1: Distribution of Median House Value in California

In [181…

```
# Distribution of Median Income Visualization
plt.figure(figsize=(12, 8))
sns.histplot(data['median_income'], kde=True, color='#3498db', bins=50)

# Adding vertical lines for the mean and median income for better insight
```

```
plt.axvline(x=data['median_income'].mean(), color='#e74c3c', linestyle='--', linewidth=2, label='Mean Income')
plt.axvline(x=data['median_income'].median(), color='#2ecc71', linestyle='-', linewidth=2, label='Median Income')
plt.title('Distribution of Median Income in California Housing Data', fontsize=18)
plt.xlabel('Median Income (in tens of thousands)', fontsize=14)
plt.ylabel('Frequency', fontsize=14)
plt.legend(fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Annotating the mean and median values
mean_income = data['median_income'].mean()
median_income = data['median_income'].median()
plt.text(mean_income + 0.1, plt.ylim()[1] * 0.7, f'Mean: ${mean_income:.2f}', color='red', fontsize=12, rotation=90)
plt.text(median_income + 0.1, plt.ylim()[1] * 0.5, f'Median: ${median_income:.2f}', color='blue', fontsize=12, rotat
plt.show()
print("Figure 5.1.2: Distribution of Median Income in California Housing Data")
```

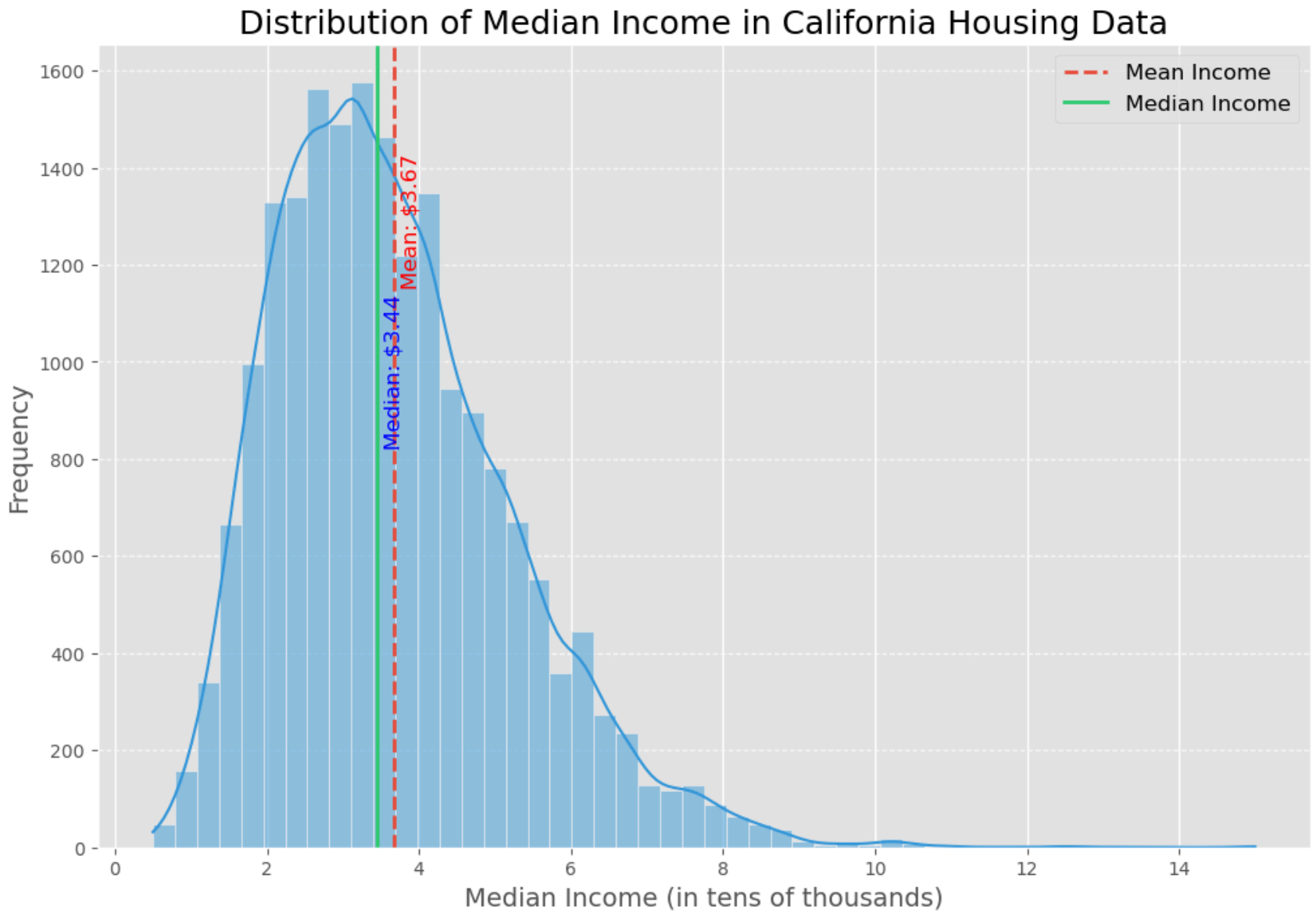


Figure 5.1.2: Distribution of Median Income in California Housing Data

2. **Relationship Analysis**

The Relationship Analysis explores how different features, such as median income and geographic location, correlate with median house value, providing insights into key factors that influence housing prices in the dataset. 1. Scatter Plot: Median Income vs. Median House Value 1. Scatter Plot: Latitude and Longitude vs. Median House Value

In [181…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='median_income', y='median_house_value', data=data, alpha=0.6,
                hue='median_house_value', size='median_income', palette='viridis', sizes=(20, 200))

# Adding a regression line for better insight into the trend
sns.regplot(x='median_income', y='median_house_value', data=data, scatter=False, color='red', line_kws={'linestyle':
plt.title('Relationship between Median Income and Median House Value', fontsize=18)
plt.xlabel('Median Income (in tens of thousands)', fontsize=14)
plt.ylabel('Median House Value (in USD)', fontsize=14)
plt.legend(title='House Value', loc='upper left', fontsize=12, title_fontsize=14)
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
print("Figure 5.2.1: Relationship between Median Income and Median House Value")
```

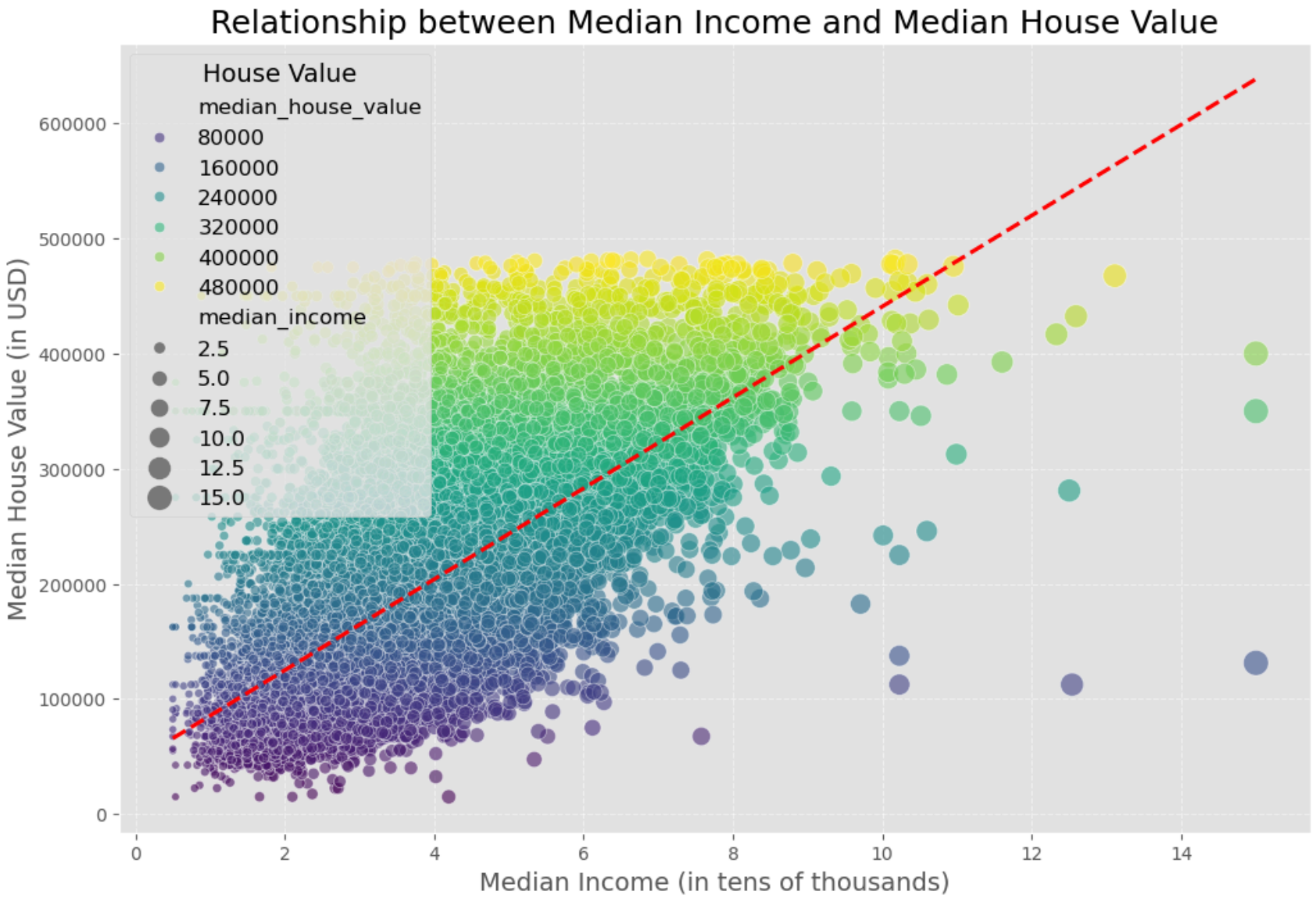


Figure 5.2.1: Relationship between Median Income and Median House Value

In [181…

```
# Create a figure with 1 row and 3 columns
fig, axs = plt.subplots(1, 3, figsize=(24, 8))

print("Figure 5.2.2: Relationship between Median Income and Median House Value by Ocean Proximity")
# Plot 1: Median Income by Location
```

```
scatter1 = axs[0].scatter(data['longitude'], data['latitude'], c=data['median_income'], cmap='plasma', alpha=0.6, edg
axs[0].set_xlabel('Longitude', fontsize=12)
axs[0].set_ylabel('Latitude', fontsize=12)
axs[0].set_title('Median Income by Location', fontsize=16)
fig.colorbar(scatter1, ax=axs[0], label='Income ($k)')

# Plot 2: Housing Median Age by Location
scatter2 = axs[1].scatter(data['longitude'], data['latitude'], c=data['housing_median_age'], cmap='cividis', alpha=0.
axs[1].set_xlabel('Longitude', fontsize=12)
axs[1].set_ylabel('Latitude', fontsize=12)
axs[1].set_title('Housing Median Age by Location', fontsize=16)
fig.colorbar(scatter2, ax=axs[1], label='Housing Age (Years)')

# Plot 3: Median House Value by Location
scatter3 = axs[2].scatter(data['longitude'], data['latitude'], c=data['median_house_value'], cmap='magma', alpha=0.6,
axs[2].set_xlabel('Longitude', fontsize=12)
axs[2].set_ylabel('Latitude', fontsize=12)
axs[2].set_title('Median House Value by Location', fontsize=16)
fig.colorbar(scatter3, ax=axs[2], label='House Value ($)')

plt.tight_layout(pad=3.0)
plt.subplots_adjust(top=0.92)
fig.suptitle('Geospatial Analysis of California Housing Data', fontsize=20, y=1.05)
plt.show()


# Creating a map centered around California
print("Figure 5.2.3: Heatmap of California Housing Data")
california_map = folium.Map(location=[34.986504,-118.716892], zoom_start = 6, min_zoom=4)
df_map = data[['latitude', 'longitude']]
heatmap_data = [[row['latitude'], row['longitude']] for index, row in df_map.iterrows()]
HeatMap(
    heatmap_data,
    radius=15,
    blur=10,
    max_zoom=1,
    min_opacity=0.3,
).add_to(california_map)
california_map
```

Figure 5.2.2: Relationship between Median Income and Median House Value by Ocean Proximity

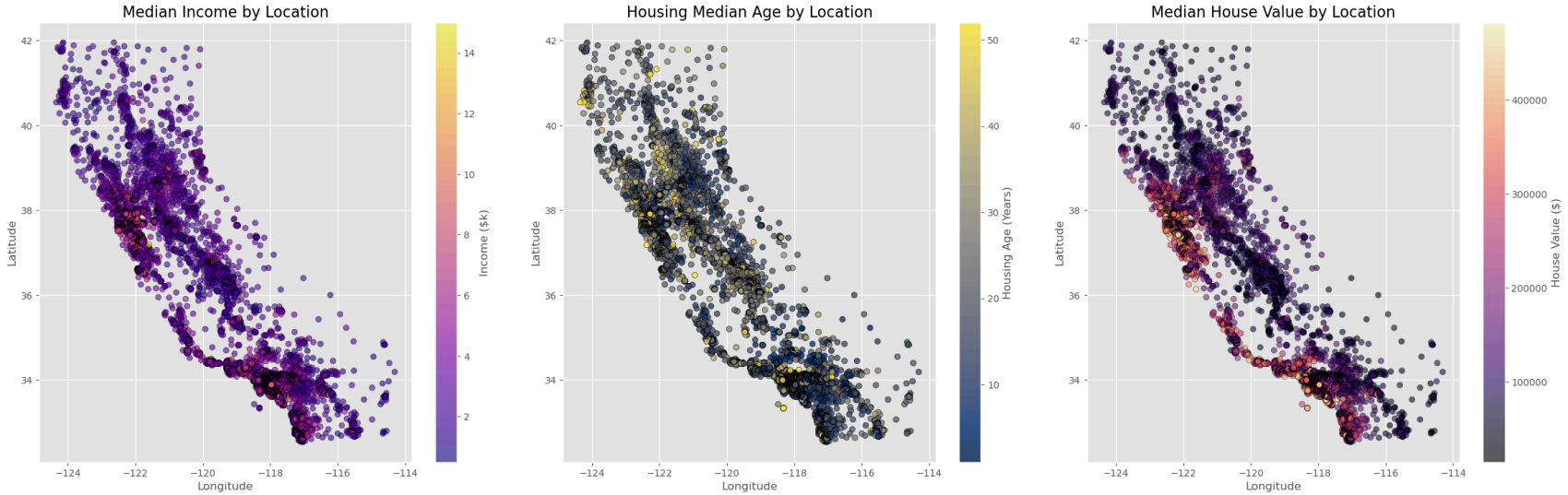
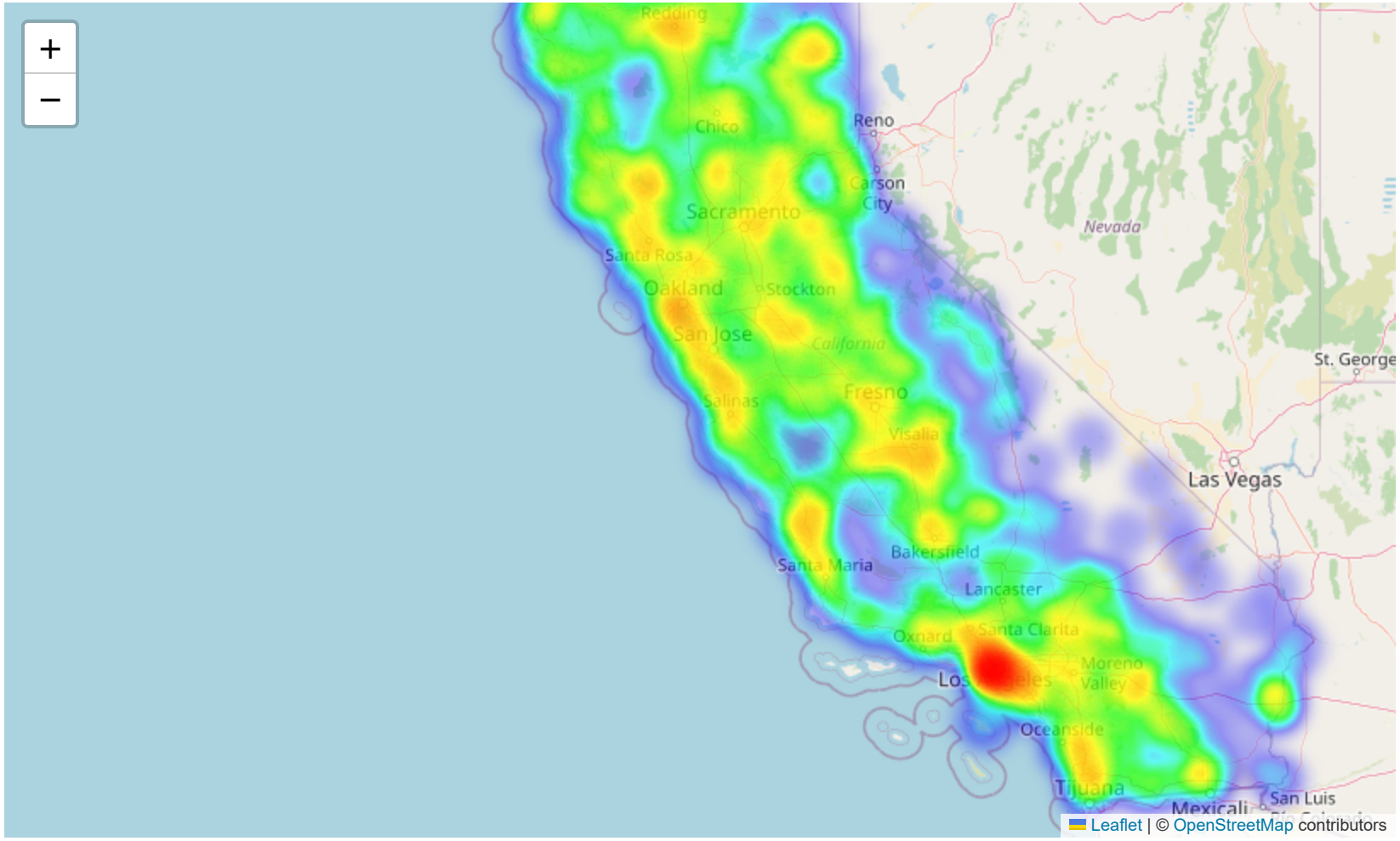


Figure 5.2.3: Heatmap of California Housing Data

Out[181…



3. **Correlation Analysis**

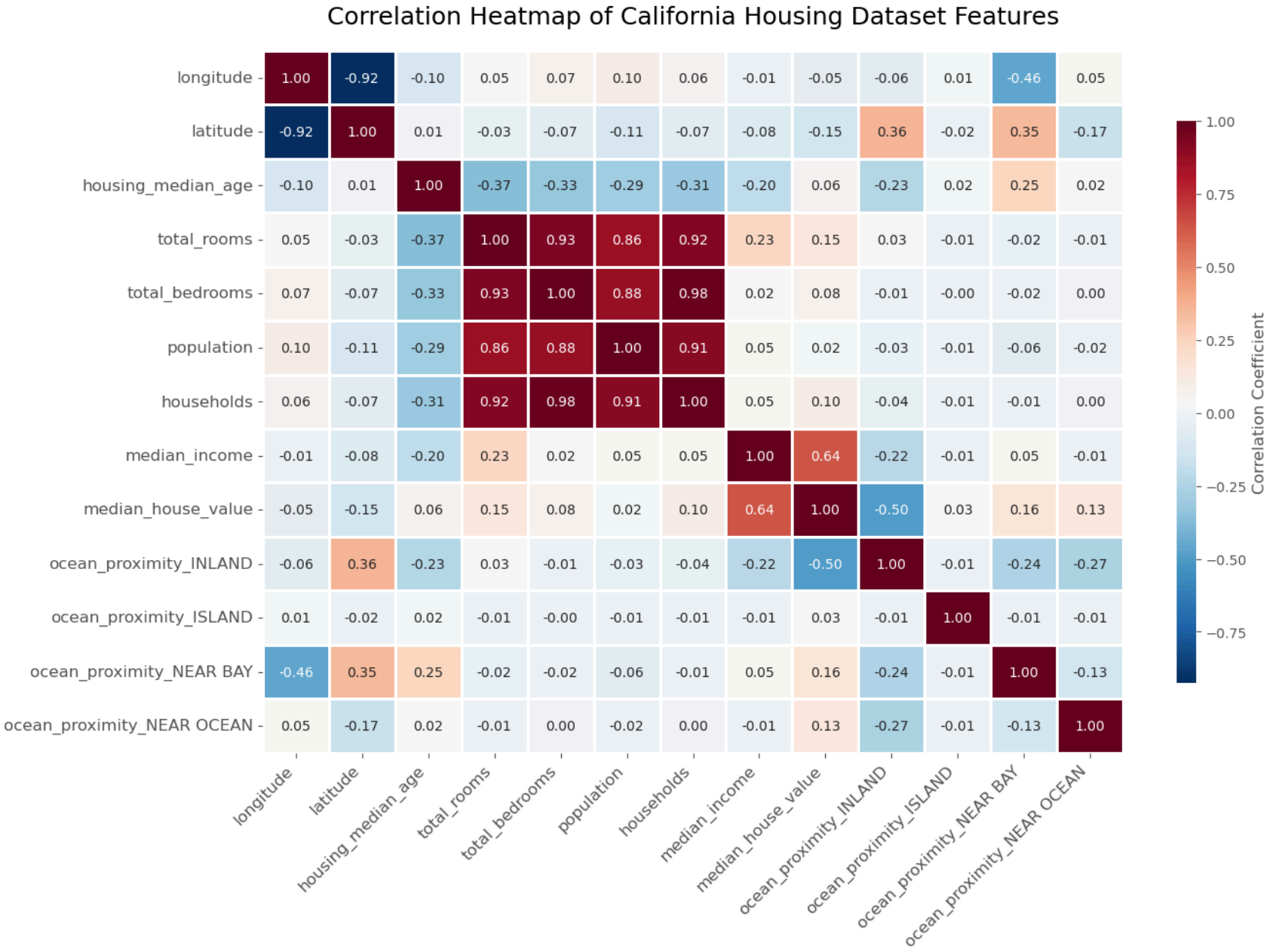
Correlation analysis identifies the strength of linear relationships between numerical features in the dataset, highlighting which variables are most closely associated with median house value and helping to detect potential multicollinearity.

In [181…

```
print("Figure 5.3.1: Correlation Heatmap of California Housing Dataset Features")
plt.figure(figsize=(14, 10))
sns.heatmap(
    data.corr(),
```

```
    annot=True,
    cmap='RdBu_r', # Reversed Red-Blue colormap to highlight both positive and negative correlations distinctly
    linewidths=1,
    linecolor='white',
    annot_kws={'size': 10},
    fmt=".2f",
    cbar_kws={'shrink': 0.8, 'aspect': 30, 'label': 'Correlation Coefficient'}
)
plt.title('Correlation Heatmap of California Housing Dataset Features', fontsize=18, pad=20)
plt.xticks(fontsize=12, rotation=45, ha='right')
plt.yticks(fontsize=12, rotation=0)
plt.tight_layout()
plt.show()
```

Figure 5.3.1: Correlation Heatmap of California Housing Dataset Features



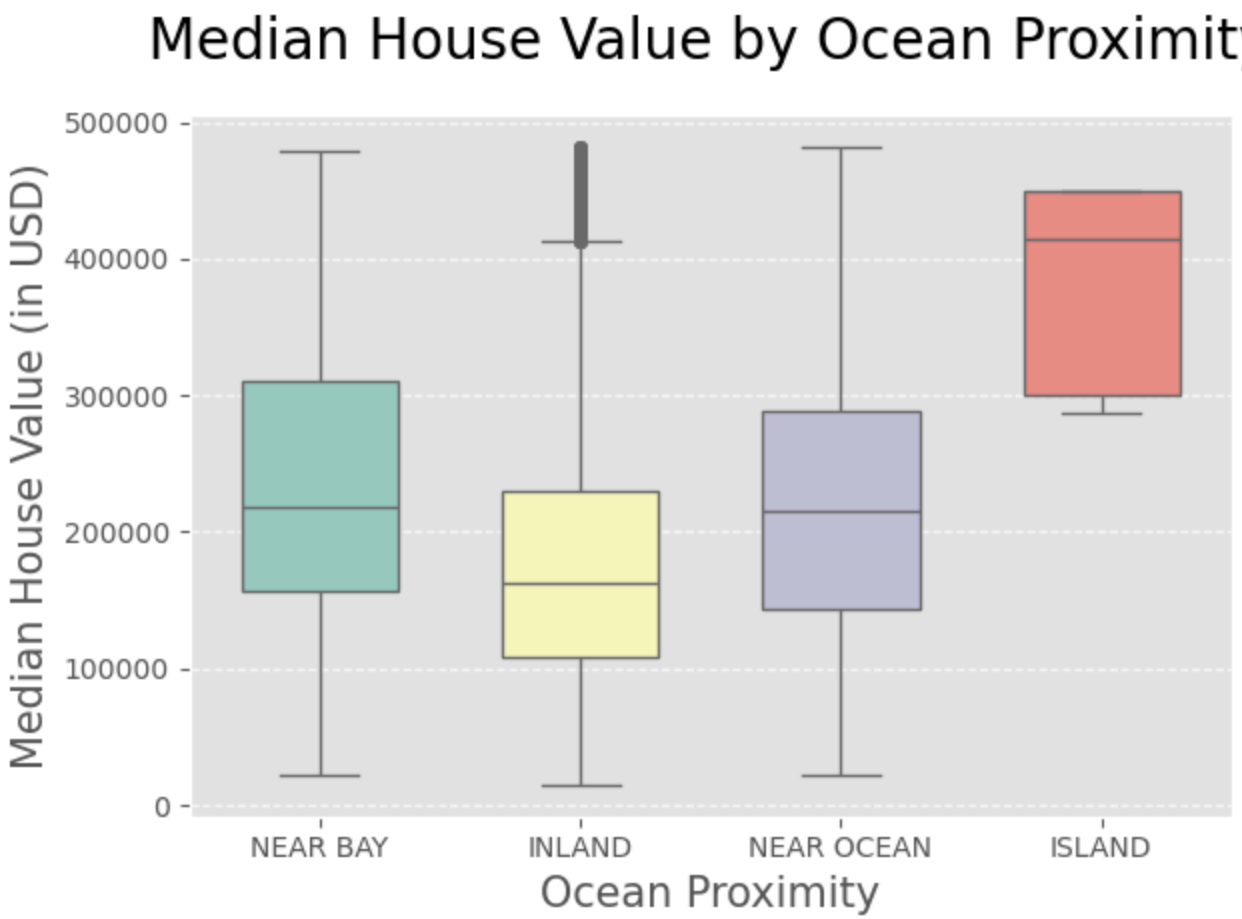
4. Box Plot Analysis

Box Plot Analysis provides a visual summary of how median house values vary across different categories, such as ocean proximity, helping to identify differences and patterns in housing prices across categorical groups in the dataset.

In [181…

```
data['ocean_proximity'] = data[['ocean_proximity_INLAND', 'ocean_proximity_ISLAND', 'ocean_proximity_NEAR BAY', 'ocea
print("Figure 5.4.1: Boxplot of Median House Value by Ocean Proximity")
sns.boxplot(x='ocean_proximity', y='median_house_value', data=data, hue='ocean_proximity', palette='Set3', width=0.6,
plt.title('Median House Value by Ocean Proximity', fontsize=20, pad=20)
plt.xlabel('Ocean Proximity', fontsize=15)
plt.ylabel('Median House Value (in USD)', fontsize=15)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

Figure 5.4.1: Boxplot of Median House Value by Ocean Proximity



5. **Pair Plot Analysis**

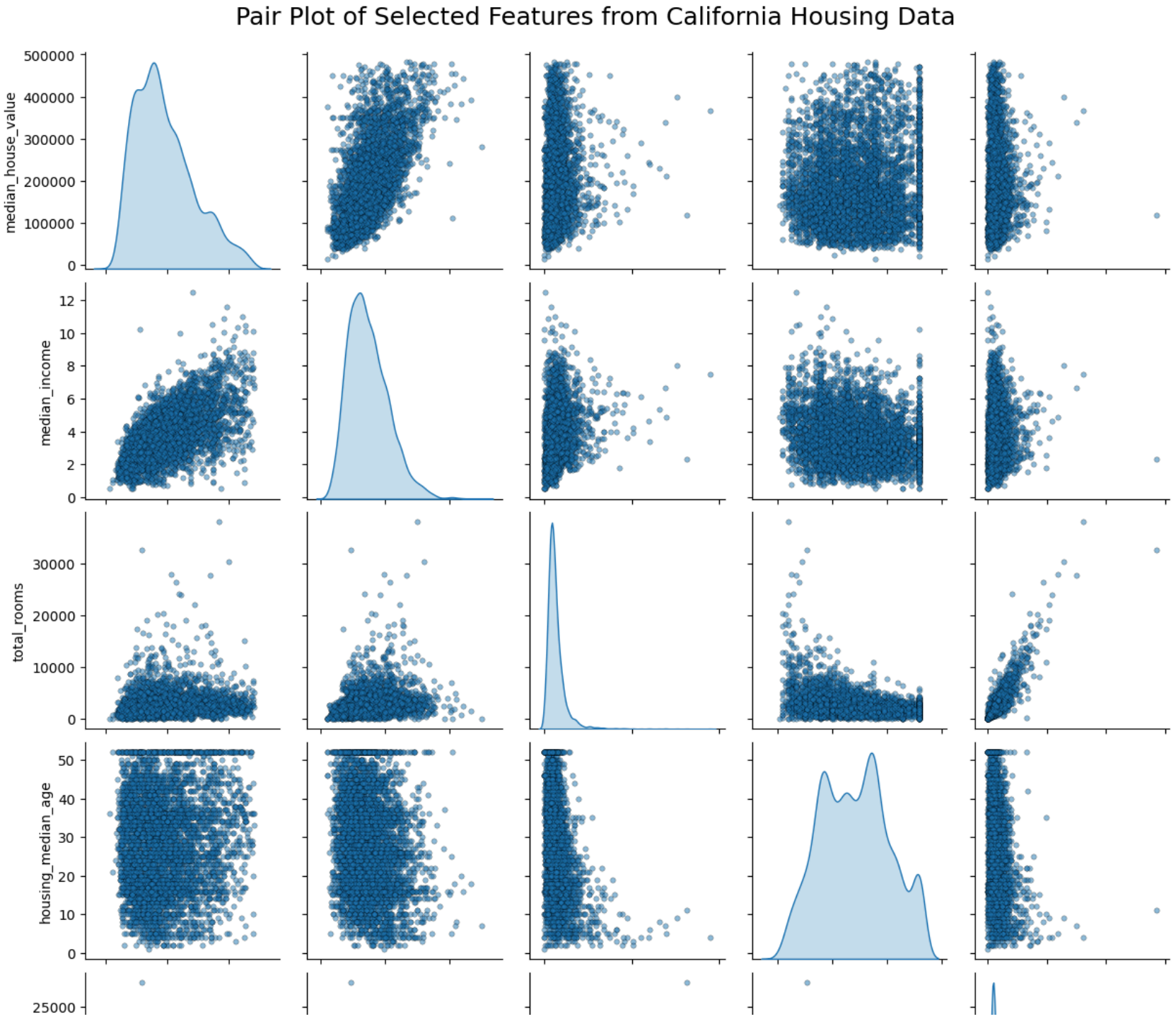
The Pair Plot provides a visual overview of pairwise relationships between selected features, helping to identify trends, correlations, and potential interactions in the dataset.

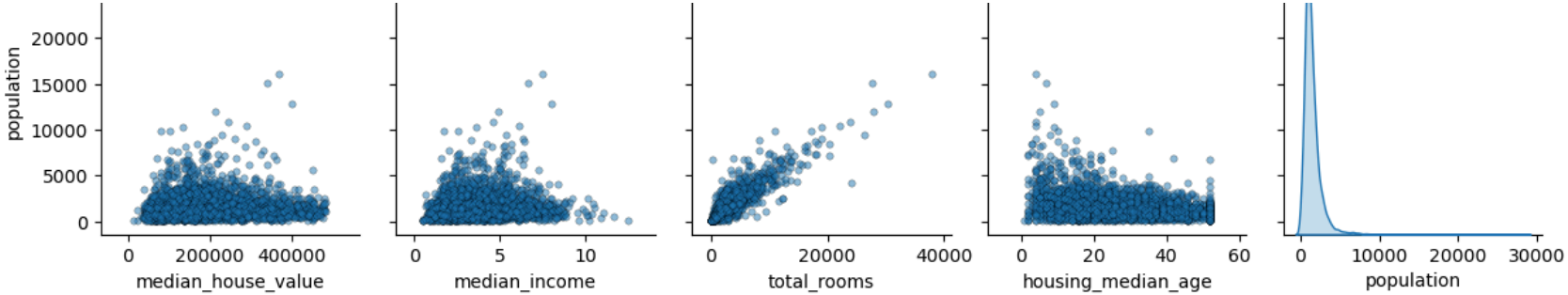
In [181…

```
plt.style.use('default')
selected_features = ['median_house_value', 'median_income', 'total_rooms', 'housing_median_age', 'population']
# Sampling the data for better readability and performance
sampled_data = data[selected_features].sample(5000, random_state=42)
print("Figure 5.5.1: Pair Plot of Selected Features from California Housing Data")
sns.pairplot(sampled_data, diag_kind='kde', kind='scatter',
```

```
                plot_kws={'alpha': 0.5, 's': 15, 'edgecolor': 'k'}, diag_kws={'fill': True})
plt.suptitle('Pair Plot of Selected Features from California Housing Data', y=1.02, fontsize=18)
plt.show()
```

Figure 5.5.1: Pair Plot of Selected Features from California Housing Data


Pair Plot of Selected Features from California Housing Data
median_house_value
median_income
total_rooms
housing_median_age
0
100000
200000
300000
400000
500000
0
2
4
6
8
10
12
0
10000
20000
30000
0
10
20
30
40
50
25000



6. Exploratory Data Analysis Summary

1. **Distribution Analysis**:
   - Median House Value: The distribution is right-skewed, with most houses priced below $300,000. The mean value is slightly higher than the median, showing the influence of higher-priced houses.
   - Median Income: The median income distribution is also right-skewed, with the majority of households earning between $2,000 and 5,000 (in tens of thousands), indicating some high-income households skewing the average.
2. **Relationship Analysis**:
   - Median Income vs. Median House Value: The scatter plot shows a clear positive relationship between income and house value. Higher income tends to lead to higher house values, as demonstrated by a distinct upward trend in the plot.
   - Geospatial Relationship: House values and incomes are significantly higher near the coast, particularly in areas like Los Angeles and the Bay Area, emphasizing the importance of location in determining house prices.
3. **Correlation Heatmap**:
   - The correlation heatmap shows that median income has a strong positive correlation with median house value (0.64). Other features such as total rooms, population, and households exhibit strong internal correlations but have limited predictive power for housing prices.
4. **Box Plot Analysis by Ocean Proximity**:
   - The box plot of median house value by ocean proximity shows that properties ISLAND, NEAR BAY, and NEAR OCEAN have much higher median values compared to INLAND properties. The coastal proximity of properties significantly increases their value.
5. **Pair Plot Analysis**:

- The pair plot reveals a strong positive correlation between median income and median house value, whereas relationships between other features like total rooms and median house value are less distinct. The distributions on the diagonal further confirm the skewness seen in earlier analyses.

6. **Heatmap**:
   - The heatmap provided an interactive visualization of house values across California. Hotspots of high median house values were observed along the coast, particularly near the Bay Area and Southern California. This visualization provided an intuitive understanding of how geography impacts housing prices, reinforcing the importance of location.

7. **Exploratory Data Analysis Conclusion**

- Income is the strongest predictor of housing price, with clear positive correlations visible in both scatter plots and the correlation heatmap.
- Geography has a significant impact on house prices, with properties in coastal areas being far more valuable compared to inland ones. The Folium heatmap and geographic scatter plots emphasize this geographic trend.
- Ocean Proximity is a strong determinant of property prices. Homes closer to the coast or near bays and oceans are consistently higher in value than those located inland.
- Distributions of Income and Housing Prices show that both are right-skewed, with the presence of high-income households and expensive properties increasing the average beyond the median.

6. Model Selection:

The goal of the Model Selection phase is to identify the best predictive model for estimating median house value based on features like median income, location, and other characteristics of California housing data. Model selection is a critical step in machine learning where compare the performance of different models to determine which one fits the data best while maintaining a good balance between bias and variance.

Multiple regression models will be tested to determine their predictive power:

1. **Linear Regression**: This model helps to establish a linear relationship between features like median income and median house value. It is simple, interpretable, and a good baseline for understanding the relationships in the data.

2. **Ridge and Lasso Regression**: These are forms of regularized linear regression that help control overfitting. Ridge Regression adds a penalty for the magnitude of coefficients to make the model more robust, while Lasso Regression performs feature selection by driving some feature coefficients to zero.

3. **Polynomial Regression**: This model captures the non-linear relationship between median income and median house value, allowing us to model complex data patterns that may not fit a straight line.

4. **Random Forest Regression**: This is an ensemble model that builds multiple decision trees and averages their predictions. Random Forest helps capture complex interactions between features and is useful for handling large datasets.

Model Selection Criteria:

Each model will be evaluated based on performance metrics such as:

- **R-Squared (R^2)**: This metric measures the proportion of variance in the target variable that is explained by the features. A higher R^2 value indicates a better fit.
- **Mean Squared Error (MSE)**: MSE measures the average squared difference between actual and predicted values, providing an idea of how accurate the model is.
- **Mean Absolute Error (MAE)**: This metric captures the average difference between actual and predicted values, offering a direct interpretation of prediction accuracy.

By comparing these metrics across different models, the aim is to find a balance between underfitting and overfitting, thereby selecting the model that best generalizes to unseen data.

Planned Models for Selection:

1. Univariate Linear Regression (Median House Value using Median Income)
2. Bivariate Linear Regression (Median House Value using Longitude and Latitude)
3. Multivariate Linear Regression (Median House Value using Median Income, Total Rooms, Population)
4. Multivariate Linear Regression (Median House Value using Median Income, Longitude, Latitude and Housing Median Age)

5. Ridge Regression (Median House Value using Median Income, Longitude, Latitude and Housing Median Age)
6. Lasso Regression (Median House Value using Median Income, Longitude, Latitude and Housing Median Age)
7. Polynomial Regression (Median House Value using Median Income, Longitude, Latitude and Housing Median Age)
8. Random Forest Regression (Median House Value using Median Income, Longitude, Latitude and Housing Median Age)

Steps for Model Selection:

1. **Data Preparation**: Splitting the data into training and testing sets for model evaluation.
2. **Model Implementation**: Implementing each regression model, starting with Linear Regression.
3. **Model Evaluation**: Evaluating models using metrics like R-Squared, MSE, and MAE.

7. Model Analysis:

In the Model Analysis phase, each of the trained models will be assessed based on their predictive performance, error metrics, and ability to generalize to unseen data. This involves evaluating the metrics of **R-Squared (R^2)**, **Mean Squared Error (MSE)**, and **Mean Absolute Error (MAE)**. Additionally, model behavior, residual analysis, and robustness checks will be considered to determine the best-performing model among those used.

Model Analysis Criteria:

- Performance Metrics:
  - **R-Squared (R^2)**: Represents the proportion of variance explained by the model. Higher R^2 indicates better predictive power.
  - **Mean Squared Error (MSE)**: Measures the average squared error between predictions and actual values, indicating how well the model is performing.
  - **Mean Absolute Error (MAE)**: Captures the average magnitude of prediction errors without considering their direction, providing an overall error value in the original units.
- Model Complexity:
  - Assessing whether adding complexity improves the model's performance significantly or if it leads to overfitting.

    - Ridge and Lasso Regression are regularized models that help in mitigating overfitting and feature importance, while Polynomial Regression and Random Forest Regression account for potential non-linear relationships.
- Residual Analysis:
    - Analyzing residuals for patterns that may indicate model biases or misspecifications.
    - Homogeneity of Variance: Residuals should have constant variance.
    - Normality: Residuals should be normally distributed.

Sampling:

Sampling in machine learning involves splitting data into a training set for model learning and a testing set for evaluation. This approach helps assess the model's performance on unseen data, providing an estimate of its accuracy and generalization ability.

1. **Train-Test Split**: The dataset is divided into training and testing sets, typically with an 80-20 ratio, to allow the model to learn from a subset (training data) and evaluate its performance on unseen data (test data). This approach helps assess the model's generalization ability.
2. **Random Sampling**: A random seed (like random_state=42) ensures that the sampling is reproducible, meaning the same train-test split can be achieved each time, which is useful for consistent results and comparisons across different model runs.

In [181…
```
# Define the sample size and random state for splitting the data
sample_test_size = 0.8
sample_random_state = 42

# Selecting the feature and target for the first model: Predicting Median House Value using Median Income, Median Age
X = data[['longitude', 'latitude', 'housing_median_age','median_income']]
y = data['median_house_value'] #target variable
```

Helper Functions:

In [181…
```
# Dictionary to store evaluation metrics
results = {
    'Model': [],
    'MSE': [],
    'R²': [],
    'MAE': []
}
```

```
# Helper function to evaluate and store results
def evaluate_model(X_train, X_test, y_train, y_test, model, model_no, model_name):
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    mse = root_mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    mae = mean_absolute_error(y_test, y_pred)
    if model_name in results['Model']:
        index = results['Model'].index(model_name)
        results['MSE'][index] = mse
        results['R²'][index] = r2
        results['MAE'][index] = mae
    else:
        results['Model'].append(model_name)
        results['MSE'].append(mse)
        results['R²'].append(r2)
        results['MAE'].append(mae)
    print(f"{str(model_no)}. {model_name} Linear Regression Model Evaluation:")
    print("Mean Squared Error (MSE):", mse)
    print("Mean Absolute Error (MAE):", mae)
    print("R-Squared (R²):", r2)
    return y_pred


# Helper function to plot residuals vs actual
def plot_residuals_vs_actual(y_test, y_pred, model_no, model_name, title_suffix):

    section = 'Figure 7.' + str(model_no)
    # Calculate residuals
    residuals = y_test - y_pred

    # Residual Distribution Plot
    print(f"\n{section}.1: Residual Distribution for {model_name} Linear Regression ({title_suffix})")
    plt.style.use('ggplot')
    plt.figure(figsize=(14, 6))
    sns.histplot(residuals, kde=True, color='#3498db', bins=30)

    # Adding vertical lines for mean and median
    plt.axvline(residuals.mean(), color='#e74c3c', linestyle='--', linewidth=2, label='Mean')
    plt.axvline(residuals.median(), color='#2ecc71', linestyle='-', linewidth=2, label='Median')

    plt.title(f'Residual Distribution for {model_name} Linear Regression ({title_suffix})', fontsize=16)
```

```
    plt.xlabel('Residuals')
    plt.ylabel('Frequency')
    plt.legend()
    plt.grid(axis='y', linestyle='--', alpha=0.6)
    plt.show()

    # Predicted vs Actual Plot
    plt.figure(figsize=(16, 6))
    sc = plt.scatter(y_test, y_pred, alpha=0.6, c=y_pred, cmap='coolwarm', s=40)

    # Line of perfect prediction
    print(f"{section}.2: Predicted vs. Actual Median House Value for {model_name} Linear Regression ({title_suffix})"
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', linewidth=2, label='Perfect Predictio
    plt.xlabel('Actual Median House Value', fontsize=12)
    plt.ylabel('Predicted Median House Value', fontsize=12)
    plt.title(f'Predicted vs. Actual Median House Value ({model_name} Linear Regression)', fontsize=16)
    plt.colorbar(sc, label='Predicted House Value')
    plt.grid(True)
    plt.legend()
    plt.show()


# Heatmap for Predicted Median House Values
def plot_geospatial_heatmap(X_test, y_test, y_pred, model_no, model_name, title_suffix, poly_transformer=None):
    # Prepare data for heatmap
    section = 'Figure 7.' + str(model_no)

    print(f"{section}.3: Heatmap of {title_suffix} for {model_name} Linear Regression")

    # Merge y_test with X_test to get latitude and longitude
    y_test_df = pd.DataFrame(y_test).reset_index(drop=True)

    # Convert y_test to DataFrame and merge with X_test
    if poly_transformer:
        X_test_df = pd.DataFrame(X_test, columns=poly_transformer.get_feature_names_out(X.columns)).reset_index(drop=
        merged_df = pd.concat([X_test_df, y_test_df], axis=1)
        heat_data = [[row['latitude'], row['longitude'], prediction] for row, prediction in zip(merged_df.to_dict('re
    else:
        y_test_df = pd.concat([y_test_df, X_test.reset_index(drop=True)], axis=1)
        heat_data = [[row['latitude'], row['longitude'], prediction] for row, prediction in zip(y_test_df.to_dict('re

    # Create base map
```

```
    map = folium.Map(location=[34.986504,-118.716892], zoom_start=6, min_zoom=4)

    # Add HeatMap layer
    HeatMap(heat_data, radius=8, blur=15, max_zoom=13).add_to(map)
    return map
```

1. Univariate Linear Regression (Median House Value using Median Income)

This model predicts the median house value based on median income, establishing a simple linear relationship to understand how income influences housing prices.

$$\text{Median House Value} = \beta_0 + \beta_1 \cdot \text{Median Income} + \epsilon$$

where:

- β_0 is the intercept,
- β_1 is the coefficient for Median Income,
- ϵ represents the residuals (errors).

In [182…

```
model_no=1
model_name= 'Univariate'
title_suffix = 'Median Income vs. Median House Value'

# Splitting the dataset into training and testing sets
X_Uni = data[['median_income']]
X_train, X_test, y_train, y_test = train_test_split(X_Uni, y, test_size=sample_test_size, random_state=sample_random_

# Creating and training the Linear Regression model
model = LinearRegression()

# Evaluate the model
y_pred = evaluate_model(X_train, X_test, y_train, y_test, model, model_no, model_name)

# Plat residuals vs. actual for the univariate model
plot_residuals_vs_actual(y_test, y_pred, model_no, model_name, title_suffix)
```

```
1. Univariate Linear Regression Model Evaluation:
Mean Squared Error (MSE): 72870.60208438961
Mean Absolute Error (MAE): 55706.32803939273
R-Squared (R²): 0.4153178441063634
```

Figure 7.1.1: Residual Distribution for Univariate Linear Regression (Median Income vs. Median House Value)

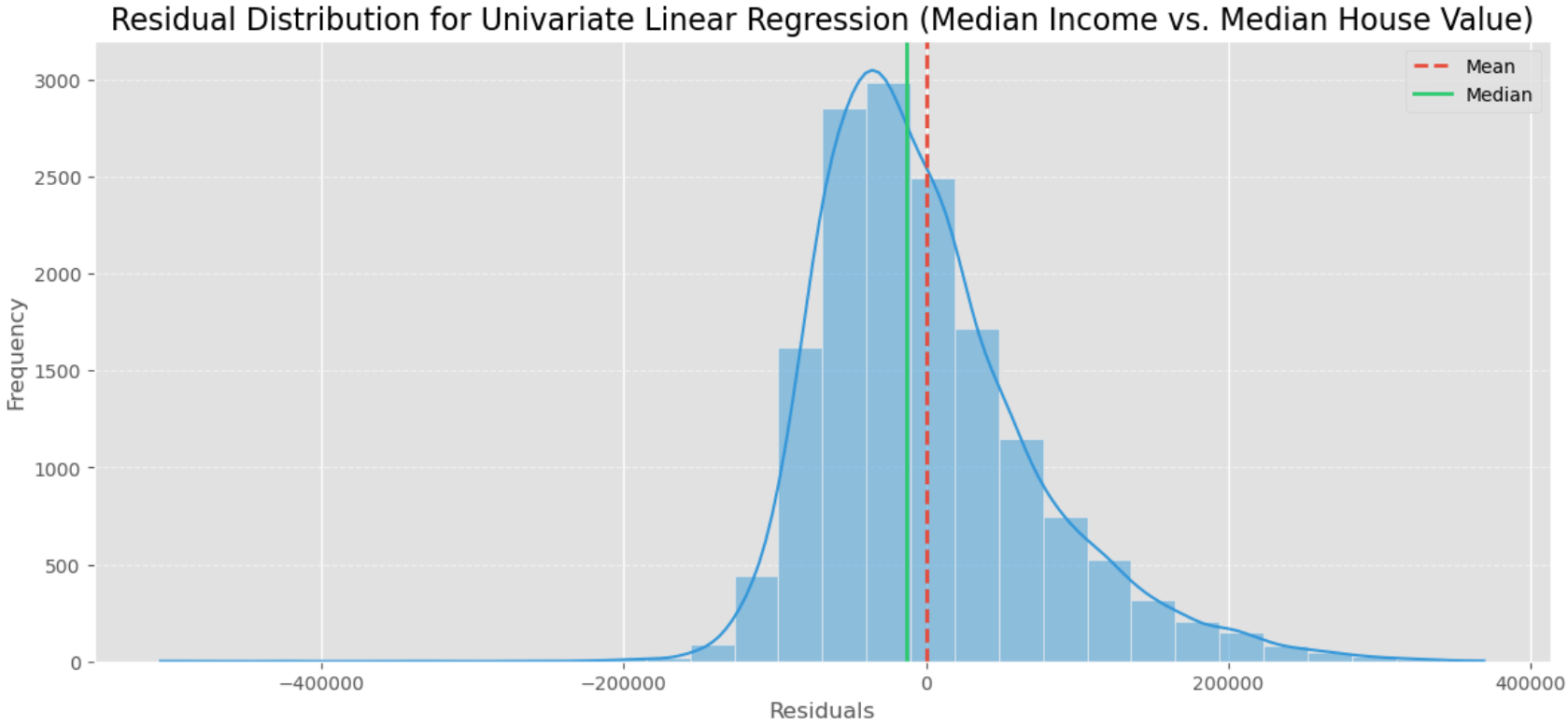
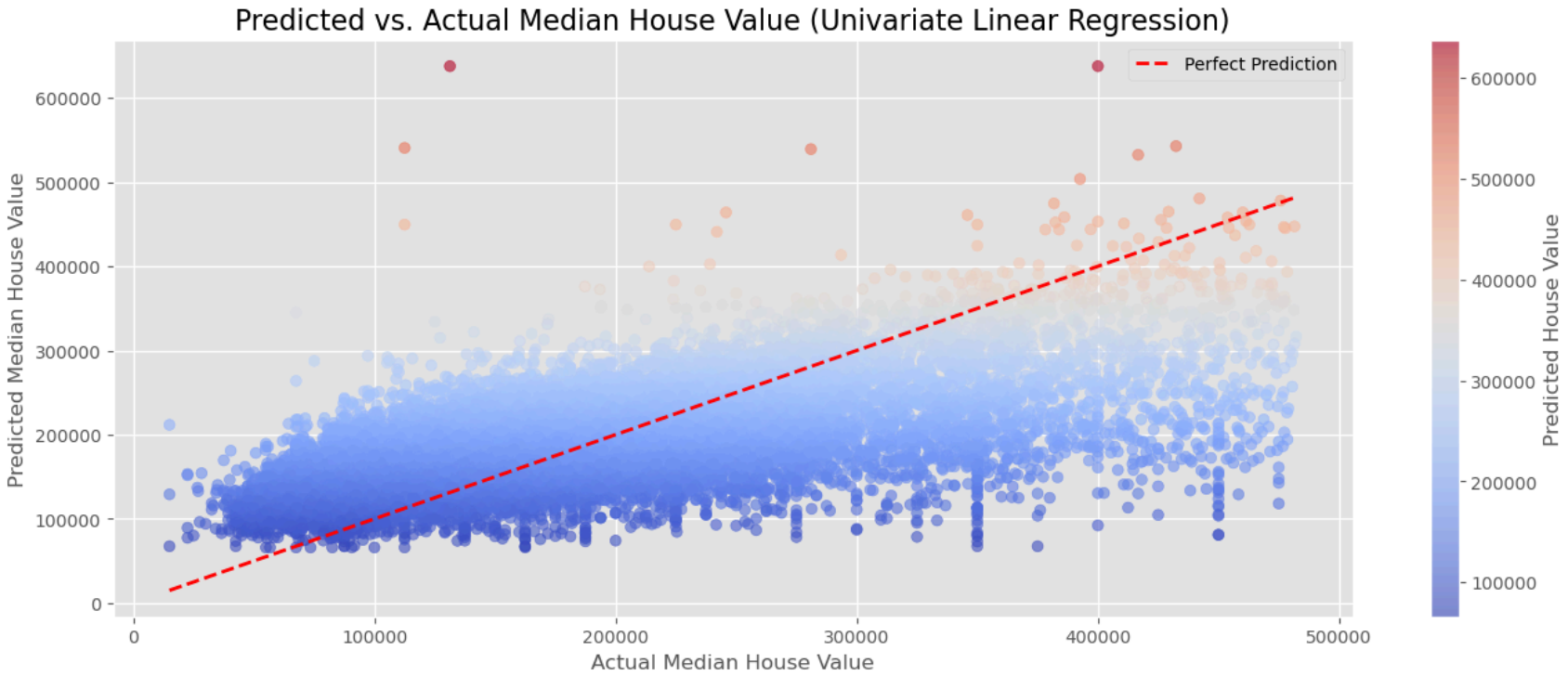


Figure 7.1.2: Predicted vs. Actual Median House Value for Univariate Linear Regression (Median Income vs. Median House Value)



Interpretation: The Univariate Linear Regression model, using median income to predict median house value, yields the following insights:

- MSE of approximately 72870 indicates moderate prediction errors, showing that the model struggles with accuracy in predicting housing prices based on only one feature, i.e., Median Income.
- MAE of \$55,706, the model's average prediction error shows considerable deviation from actual values. This suggests that a simple linear relationship between median income and house prices does not fully capture the variability.
- R^2 of 0.415 implies that about 41.5% of the variance in median house values is explained by median income. Although this explains a moderate amount of the variance, it indicates that other variables significantly influence house prices, which are not captured by this univariate model.

Chart Analysis:

- **Predicted vs. Actual Scatter Plot**:
  - The scatter plot shows a positive trend, indicating that as median income increases, the predicted house values also increase.

- However, a significant spread is visible, particularly in the lower and higher house value ranges, which deviates from the red dashed line representing perfect prediction. This suggests that the model struggles with extreme values and outliers.
- The color gradient (from blue to red) indicates the predicted values, where higher predicted values are generally more scattered, reflecting prediction limitations in high-priced areas.

- **Residual Distribution Plot**:
  - The residuals are centered around zero, indicating no strong bias in the predictions. However, the wide distribution with significant errors on both the negative and positive sides shows that the model has large residuals, particularly for homes at the lower and higher ends of the price spectrum.
  - The residuals' mean and median being close to zero confirms that there is no systematic bias, but the model's precision is affected by the missing influential factors.
  - The long tails in the distribution indicate the presence of outliers and errors, particularly in high-value house predictions.

2. **Bivariate Linear Regression (Median House Value using Longitude and Latitude)**

This model predicts median house value based on geographical coordinates, aiming to capture the impact of location on housing prices.

$$\text{Median House Value} = \beta_0 + \beta_1 \cdot \text{Longitude} + \beta_2 \cdot \text{Latitude} + \epsilon$$

where:

- β_0 is the intercept,
- β_1 and β_2 are the coefficients for Longitude and Latitude,
- ϵ represents the residuals (errors).

In [182…

```
model_no=2
model_name= 'Bivariate'
title_suffix = 'Longitude & Latitude vs. Median House Value'

# Selecting features (longitude, latitude) and target (median house value)
X_loc = data[['longitude', 'latitude']]

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_loc, y, test_size=sample_test_size, random_state=sample_random_
```

```
# Creating and training the Linear Regression model
model = LinearRegression()

# Evaluate the model
y_pred = evaluate_model(X_train, X_test, y_train, y_test, model, model_no, model_name)

# Plot residuals vs. actual for the bivariate model
plot_residuals_vs_actual(y_test, y_pred, model_no, model_name, title_suffix)

# Display the map
california_map = plot_geospatial_heatmap(X_test, y_test, y_pred, model_no, model_name, title_suffix)
california_map
```

```
2. Bivariate Linear Regression Model Evaluation:
Mean Squared Error (MSE): 82255.96212488494
Mean Absolute Error (MAE): 64215.74292257116
R-Squared (R²): 0.2550109369137644
```

Figure 7.2.1: Residual Distribution for Bivariate Linear Regression (Longitude & Latitude vs. Median House Value)

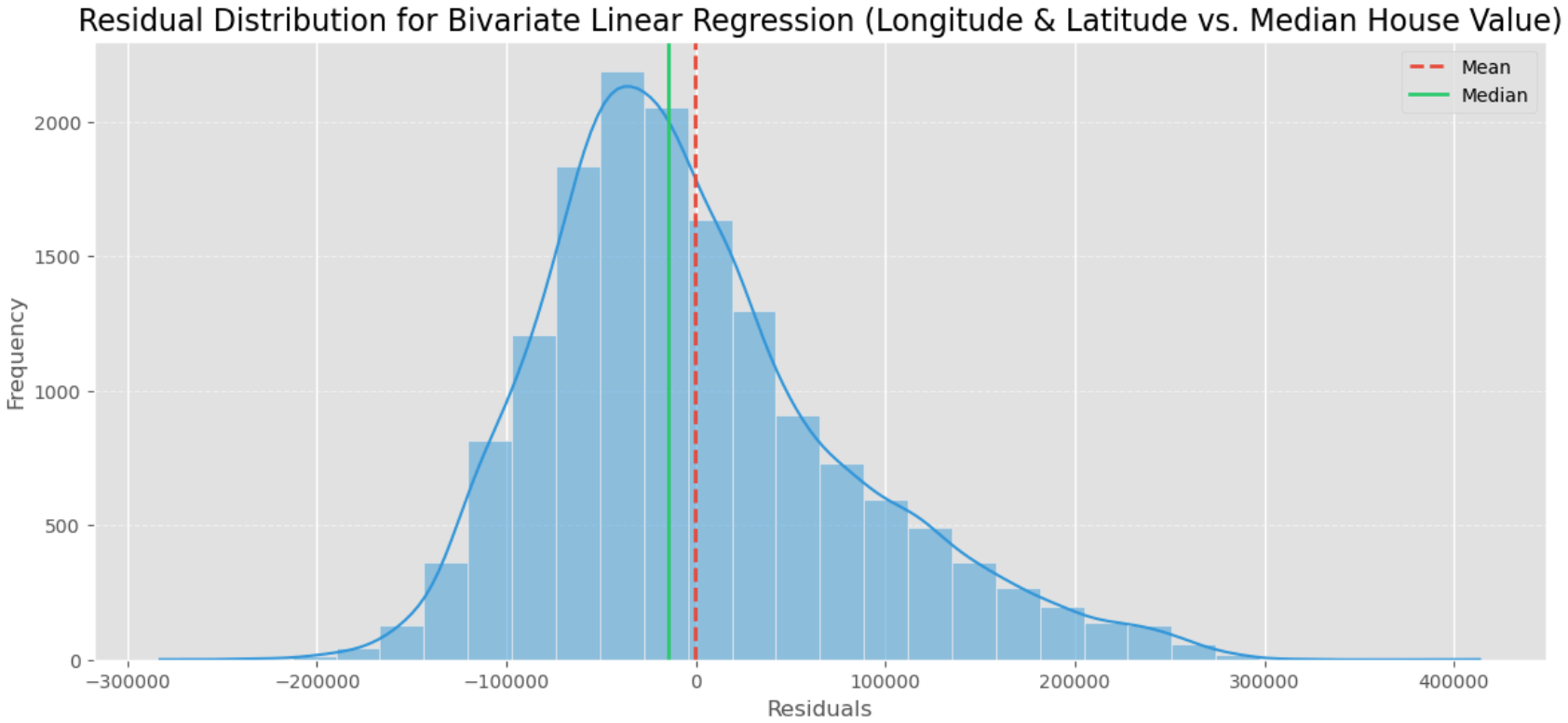


Figure 7.2.2: Predicted vs. Actual Median House Value for Bivariate Linear Regression (Longitude & Latitude vs. Median House Value)

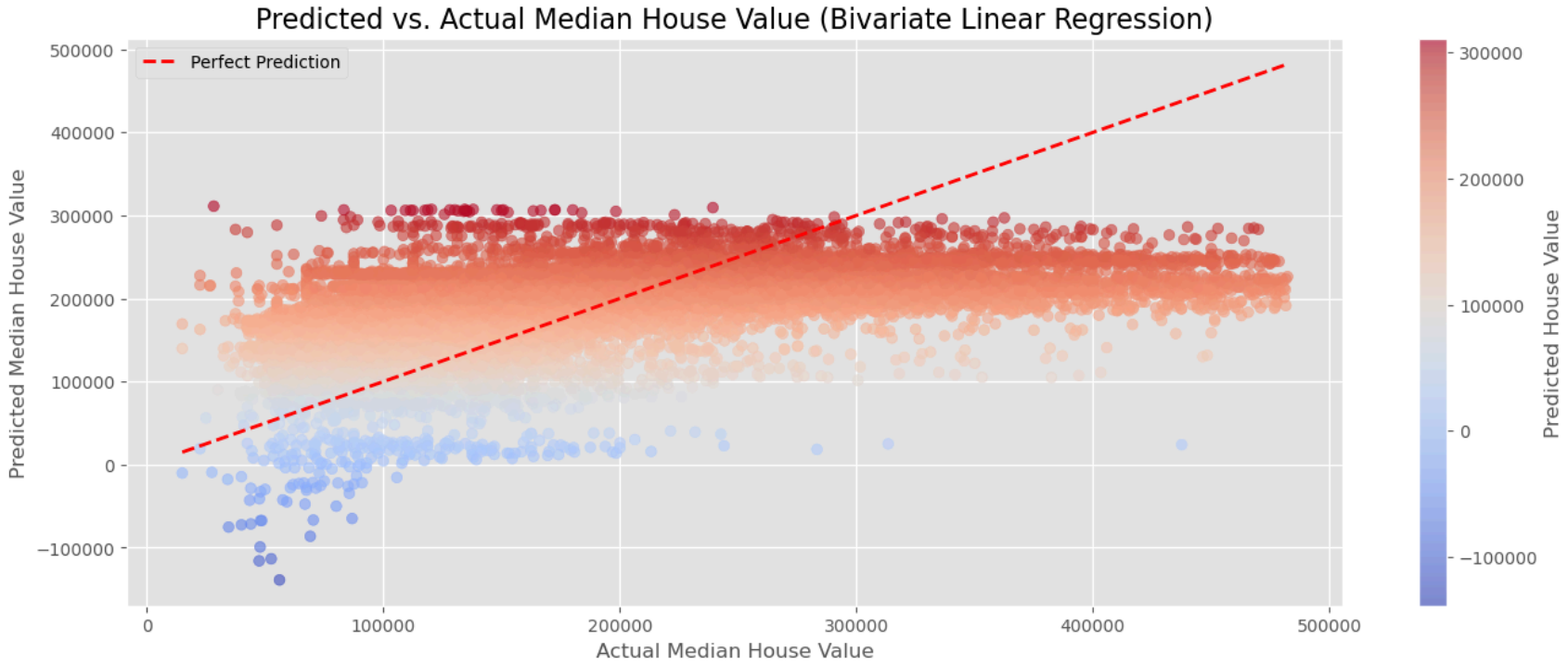
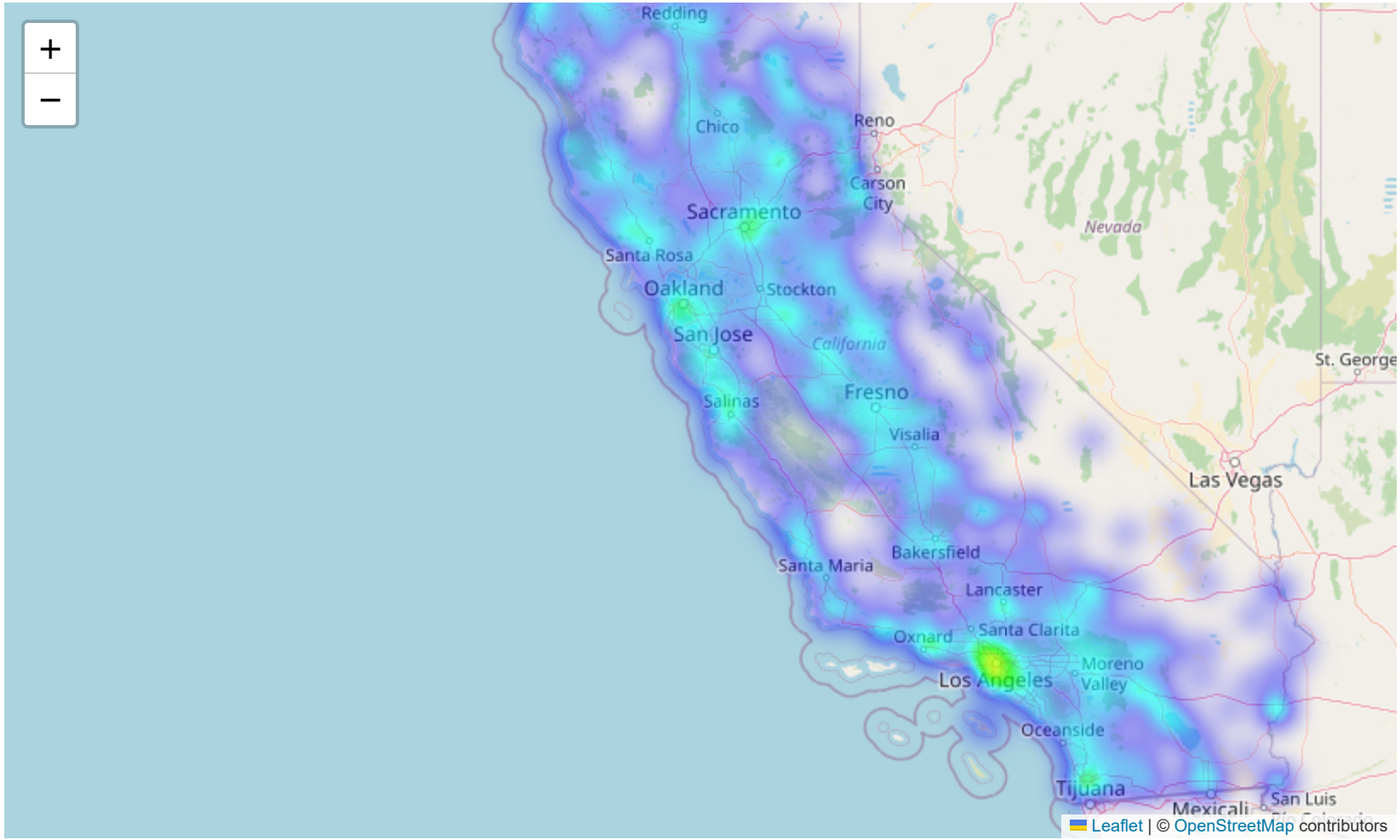


Figure 7.2.3: Heatmap of Longitude & Latitude vs. Median House Value for Bivariate Linear Regression

Out[182...



Interpretation: The Bivariate Linear Regression model using longitude and latitude to predict median house value yields the following insights:

- MSE of approximately 82255, which reflects significant prediction errors. This suggests that location alone (longitude and latitude) does not provide sufficient information for accurately predicting house prices, especially in high-value areas.
- MAE of $64,215 indicates the average error between predicted and actual house prices. This is a relatively high value, further suggesting the limited predictive power of the model based solely on geographical coordinates.
- R^2 of 0.255 explains only 25.5% of the variance in median house values, which indicates a weak model performance. It suggests that additional features are needed to better explain the variability in housing prices.

Chart Analysis:

- **Residual Distribution Plot**:
  - The residuals are centered around zero, showing no major bias in the model. However, the distribution is broad, with a high concentration of values around zero and long tails on both sides. This indicates substantial errors for some predictions, especially for outliers.
  - The slight skew to the right and long tails in the distribution suggest that the model struggles with both underestimating and overestimating house prices for some data points.
  - The mean and median residuals being close to zero confirm no systematic bias, but the long tail suggests the presence of errors, particularly for high-value homes.
- **Predicted vs. Actual Scatter Plot**:
  - The scatter plot reveals a positive trend but with significant dispersion from the line of perfect prediction, especially for high house values.
  - Points closer to the line represent more accurate predictions, while the widespread deviation, especially for higher house values, shows that the model underestimates the prices for more expensive houses.
  - The color gradient from blue (lower predicted values) to red (higher predicted values) demonstrates that the model struggles with over and underestimating prices, particularly at the extremes of the dataset.
- **Heatmap**:
  - The heatmap shows the distribution of predicted house values across California, with hotter spots indicating higher predicted prices. The heatmap shows dense activity in metropolitan areas like Los Angeles and San Francisco, reflecting high predicted values in these regions.
  - While the heatmap provides useful geographical insights, it cannot capture other factors such as house size, income levels, and neighborhood quality, which contribute to house price variability.

3. **Multivariate Linear Regression (Median House Value using Median Income, Total Rooms, Population)**

This model predicts the median house value based on median income, total rooms, and population, establishing a linear relationship to understand how these features collectively influence housing prices.

$$\text{Median House Value} = \beta_0 + \beta_1 \cdot \text{Median Income} + \beta_2 \cdot \text{Total Rooms} + \beta_3 \cdot \text{Population} + \epsilon$$

where:

- β_0 is the intercept,
- β_1, β_2, and β_3 are the coefficients for Median Income, Total Rooms, and Population,
- ϵ represents the residuals.

In [182…

```
model_no = 3
model_name= 'Multivariate 1'
title_suffix = 'Income, Total Rooms, Population vs. Median House Value'

# define our predictor variables
X_multi = data[['median_income', 'total_rooms', 'population']]

# split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_multi, y, test_size=sample_test_size, random_state=sample_rando

# train the multiple linear regression model
model = LinearRegression()

# Evaluate the model
y_pred = evaluate_model(X_train, X_test, y_train, y_test, model, model_no, model_name)

# Plot residuals vs. actual for the Multivariate model
plot_residuals_vs_actual(y_test, y_pred, model_no, model_name, title_suffix)
```

```
3. Multivariate 1 Linear Regression Model Evaluation:
Mean Squared Error (MSE): 72922.80396725831
Mean Absolute Error (MAE): 55949.512409422096
R-Squared (R²): 0.41447985345442007

Figure 7.3.1: Residual Distribution for Multivariate 1 Linear Regression (Income, Total Rooms, Population vs. Median
House Value)
```

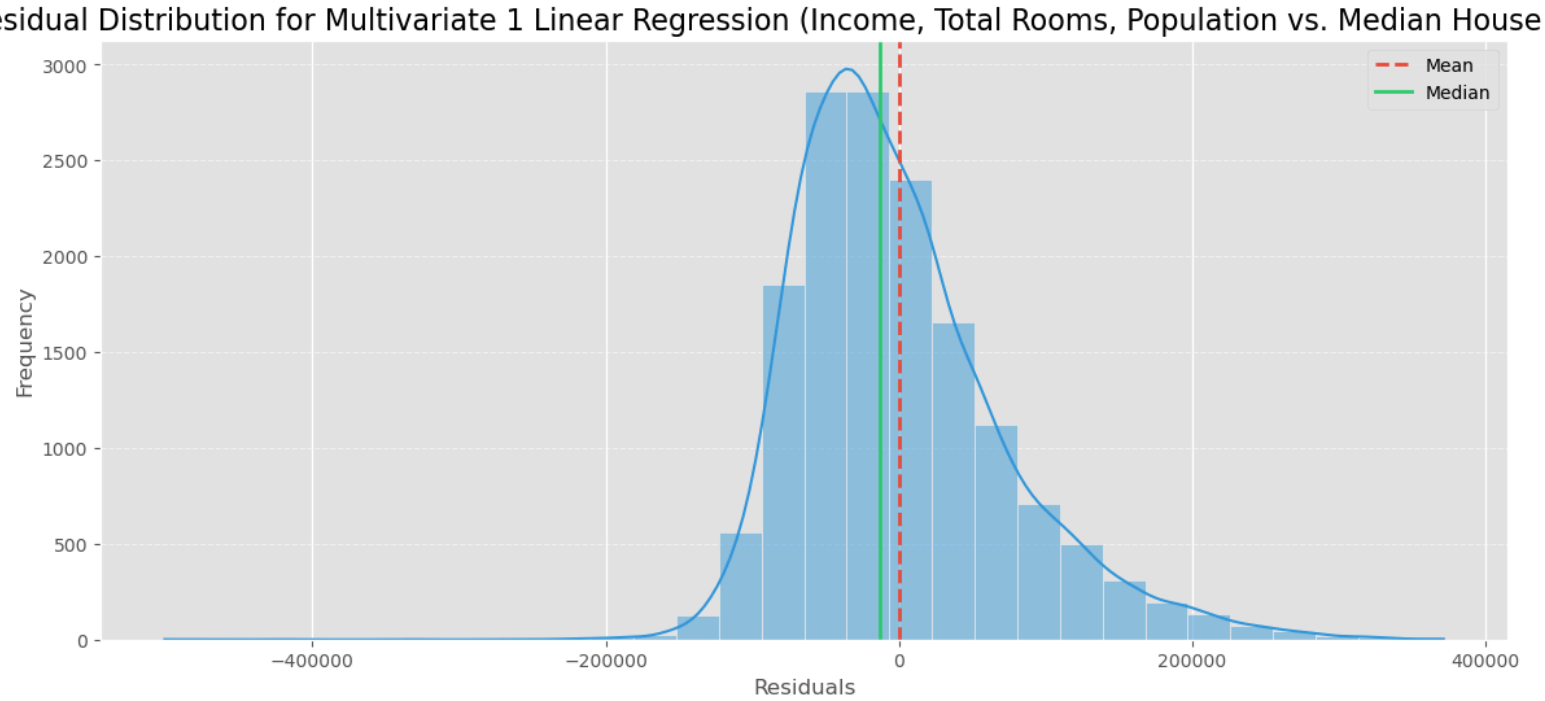
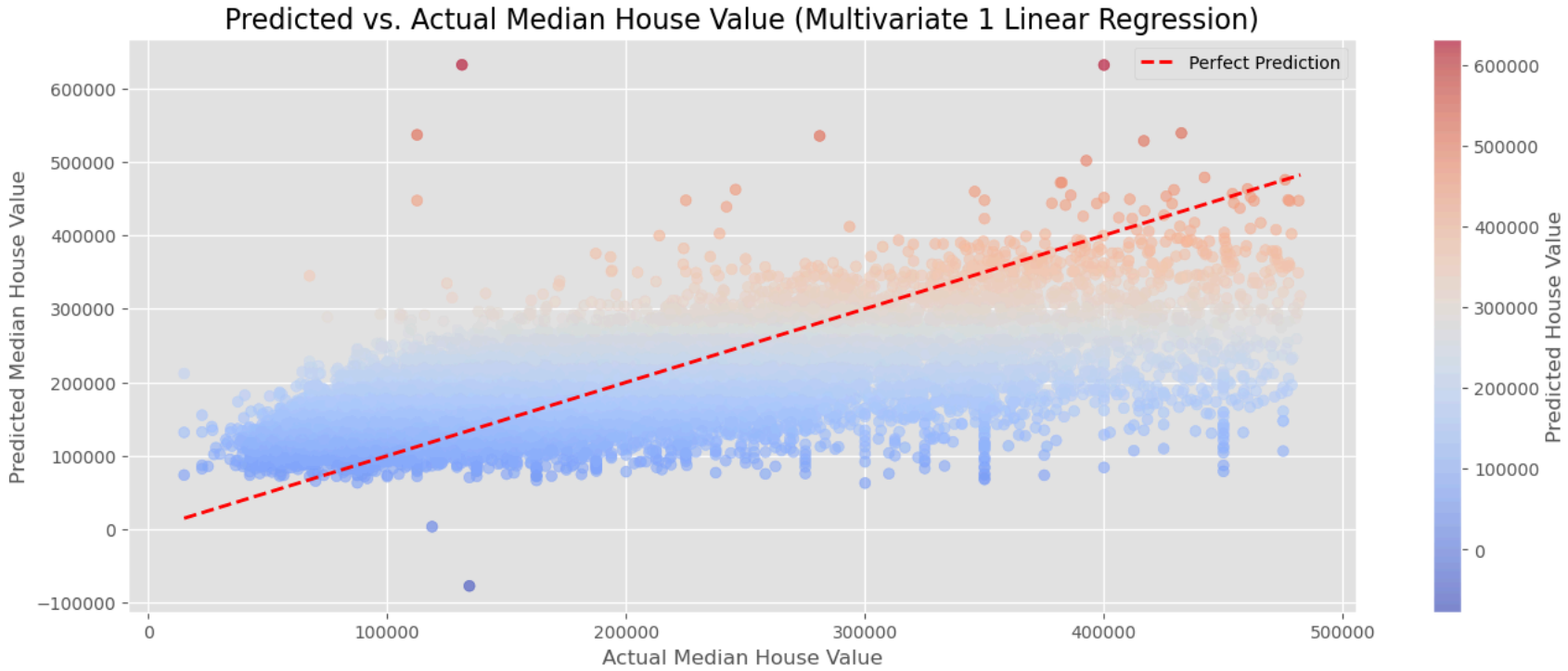


Figure 7.3.2: Predicted vs. Actual Median House Value for Multivariate 1 Linear Regression (Income, Total Rooms, Population vs. Median House Value)



Interpretation: This model, which uses median income, total rooms, and population to predict the median house value, shows moderate performance.

- MSE of approximately 72922 indicates notable prediction errors, particularly in the higher and lower house price ranges.
- R^2 of 0.414 suggests that about 41.4% of the variance in house prices can be explained by these features, which is decent but implies that additional factors likely influence house prices.
- MAE of approximately $55,950 indicates a relatively high average deviation between predicted and actual values.

Chart Analysis:

- **Residual Distribution Plot**:
  - The residuals are centered around zero, which indicates no strong bias in the predictions.
  - However, the residual distribution is fairly broad, indicating considerable prediction errors, especially for higher house values.
  - The mean and median lines are close, showing that there is no significant skew, but the long tails in both directions suggest that the model struggles to predict extreme outliers.

- **Predicted vs. Actual Scatter Plot**:
  - The scatter plot demonstrates a positive correlation, with predicted values generally increasing with actual values. However, notable deviations from the perfect prediction line exist, particularly at higher values.
  - Clustering at lower house prices indicates better model accuracy for mid-range prices, but errors grow significantly for more expensive properties.
  - The predicted values (shown by the gradient color) are generally lower than actual high-priced houses, suggesting the model struggles to capture extreme price variations.

4. Multivariate Linear Regression (Median House Value using Median Income, Longitude, Latitude and Housing Median Age)

This model attempts to predict the median house value based on four key features: median income, longitude, latitude, and housing median age. By incorporating both geographical and socioeconomic factors, this model aims to improve the prediction of housing prices.

$$\text{Median House Value} = \beta_0 + \beta_1 \cdot \text{Median Income} + \beta_2 \cdot \text{Longitude} + \beta_3 \cdot \text{Latitude} + \beta_4 \cdot \text{Housing Median Age} + \epsilon$$

where:

- β_0 is the intercept,
- β_1 is the coefficient for Median Income,
- β_2 is the coefficient for Longitude,
- β_3 is the coefficient for Latitude,
- β_4 is the coefficient for Housing Median Age,
- ϵ represents the residuals (errors).

In [182…

```
model_no = 4
model_name= 'Multivariate 2'
title_suffix = 'Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value'


# training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=sample_test_size, random_state=sample_random_stat

# create a Linear Regression model
```

```
model = LinearRegression()

# Evaluate the model
y_pred = evaluate_model(X_train, X_test, y_train, y_test, model, model_no, model_name)

# Plot residuals vs. actual for the Multivariate model
plot_residuals_vs_actual(y_test, y_pred, model_no, model_name, title_suffix)

# Display the map
california_map = plot_geospatial_heatmap(X_test, y_test, y_pred, model_no, model_name, title_suffix)
california_map
```

```
4. Multivariate 2 Linear Regression Model Evaluation:
Mean Squared Error (MSE): 63197.01070376449
Mean Absolute Error (MAE): 47686.56049986861
R-Squared (R²): 0.560247622440619
```

Figure 7.4.1: Residual Distribution for Multivariate 2 Linear Regression (Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value)

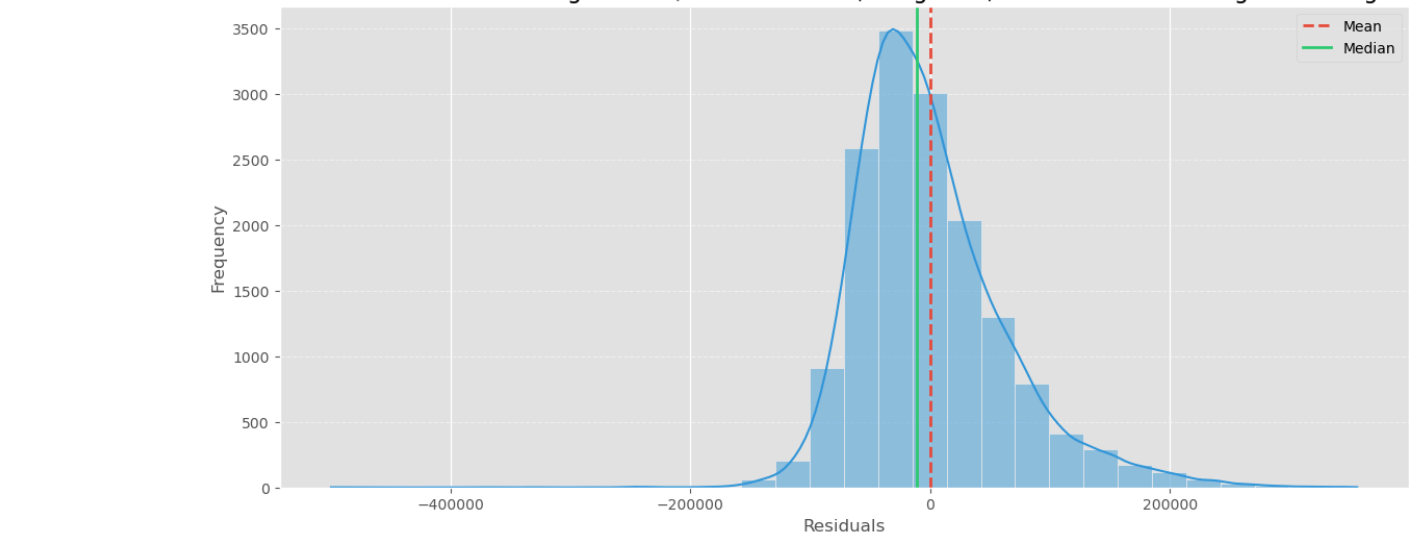


Figure 7.4.2: Predicted vs. Actual Median House Value for Multivariate 2 Linear Regression (Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value)

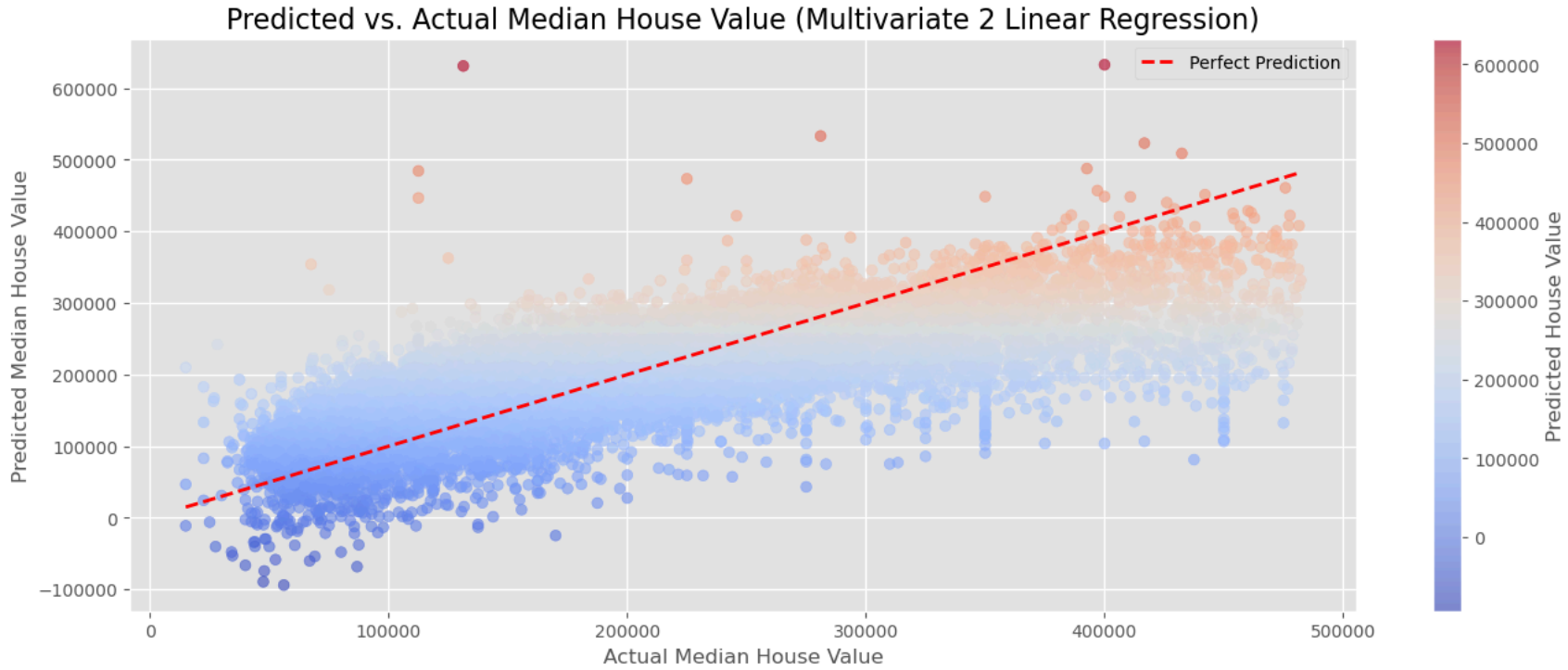
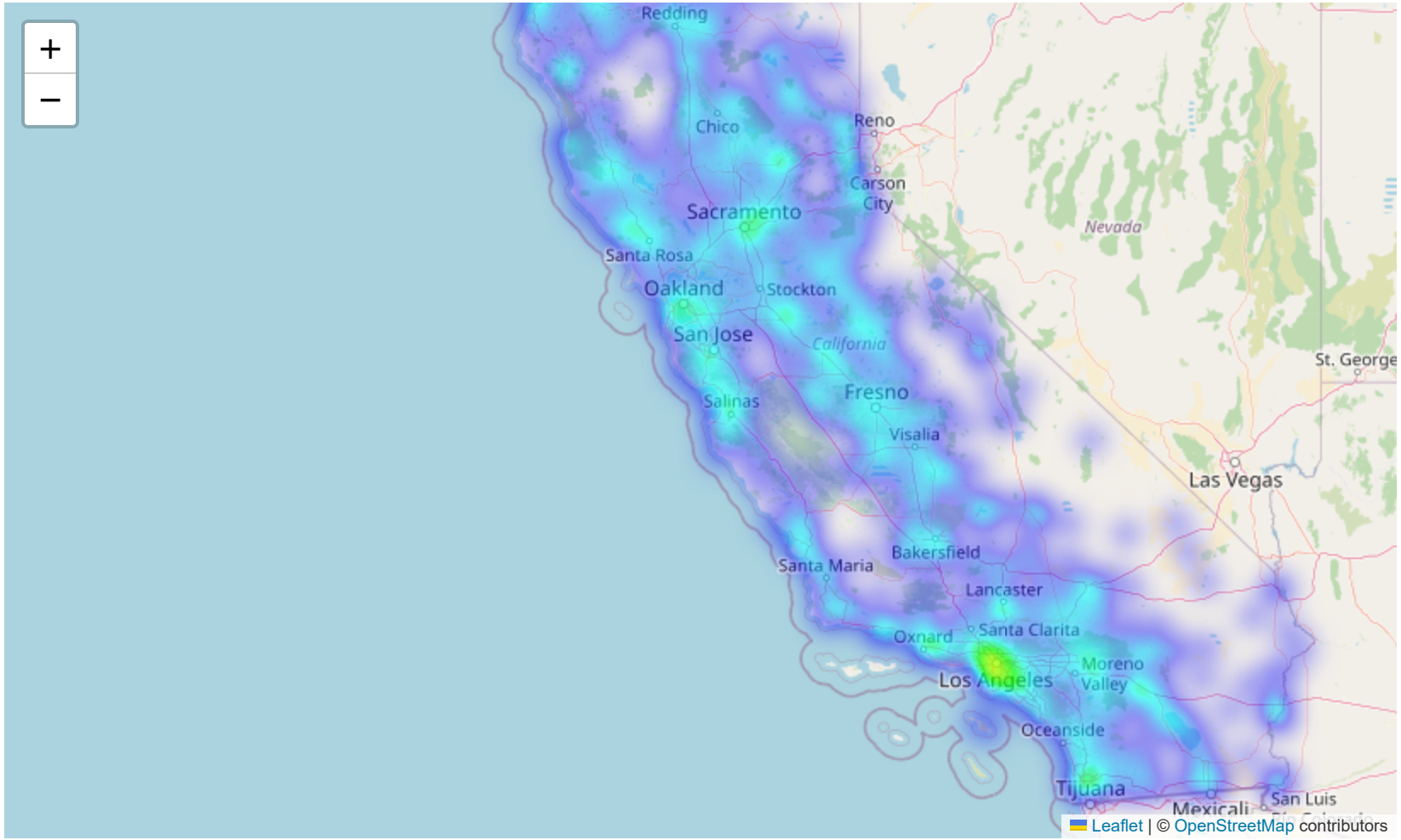


Figure 7.4.3: Heatmap of Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value for Multivariate 2 Linear Regression

Out[182...



Interpretation: The multivariate linear Regression model, using longitude, latitude, and housing median age to predict median house value, yields the following insights:

- MSE of approximately 63197. This value, while still large, represents a notable reduction in error compared to simpler models. The addition of more features (income, longitude, latitude, housing age) has improved the model's predictive performance.
- MAE of approximately $47,686 highlights that, on average, the model's predictions deviate from actual house prices by this amount. While there are still significant errors for higher-priced homes, the model performs better than previous models in reducing the average prediction error.

- R^2 of 0.560 shows that around 56% of the variance in median house values can be explained by this model. The inclusion of these features contributes significantly to the model's ability to predict house prices.

Chart Analysis:

- **Residual Distribution Plot**:
  - The residuals are more tightly centered around zero compared to prior models, indicating improved accuracy across the board.
  - The slight skew to the right and the long tails still suggest some issues with outliers, but the errors are more concentrated near zero, implying better prediction quality.
  - The mean and median residuals remain close to zero, affirming that the model is generally unbiased.
- **Predicted vs. Actual Scatter Plot**:
  - The plot reveals a more robust trend of predicted values aligning with the actual house prices.
  - Points closer to the line of perfect prediction indicate increased model accuracy across a wider range of house prices.
  - The color gradient (from blue to red) demonstrates that the model captures housing price variability more effectively, particularly in urban areas and high-income regions.
- **Heatmap**:
  - The heatmap shows clearer "hot spots" of predicted high house values across California, focusing on urban centers like Los Angeles, San Francisco, and San Diego.
  - By including housing age and geographical coordinates, the heatmap now reflects a more realistic distribution of predicted house values across the state.
- **Predictor Refinement**:
  - As we see from (Figure 7.4.3) adding additional predictors helps to make model more robust. We can see that our MSE and MAE are reduced by 33%. We also noticed the R^2 value has doubled.
  - Refining the dataset further, and adding more predictors will most likely allow model to be more accurate. However, we want to avoid overfitting, model complexity, and want to ensure feature importance.

5. **Ridge Regression (Median House Value using Median Income, Longitude, Latitude and Housing Median Age)**

Ridge Regression is a regularization technique used to prevent overfitting by adding a penalty term to the cost function. This regression model uses median income, geographical coordinates (longitude, latitude), and housing median age to predict the median house value.

$$\text{Median House Value} = \beta_0 + \beta_1 \cdot \text{Median Income} + \beta_2 \cdot \text{Longitude} + \beta_3 \cdot \text{Latitude} + \beta_4 \cdot \text{Housing Median Age} + \lambda \sum_{i=1}^{n} \beta_i^2 + \epsilon$$

Where:

- β_0 is the intercept,
- $\beta_1, \beta_2, \beta_3, \beta_4$ are the coefficients for median income, longitude, latitude, and housing median age,
- λ is the regularization parameter that shrinks the coefficients,
- ϵ represents the residuals (errors).

In [182…

```
model_no = 5
model_name = 'Ridge'
title_suffix = 'Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value'

# split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=sample_test_size, random_state=sample_random_stat

# train the Ridge regression model
model = Ridge(alpha=1.0)

# Evaluate the model
y_pred = evaluate_model(X_train, X_test, y_train, y_test, model, model_no, model_name)

# Plot residuals vs. actual for the Multivariate model
plot_residuals_vs_actual(y_test, y_pred, model_no, model_name, title_suffix)

# Display the map
california_map = plot_geospatial_heatmap(X_test, y_test, y_pred, model_no, model_name, title_suffix)
california_map
```

```
5. Ridge Linear Regression Model Evaluation:
Mean Squared Error (MSE): 63196.63009691232
Mean Absolute Error (MAE): 47686.2707734986
R-Squared (R²): 0.560252919281792
```

Figure 7.5.1: Residual Distribution for Ridge Linear Regression (Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value)

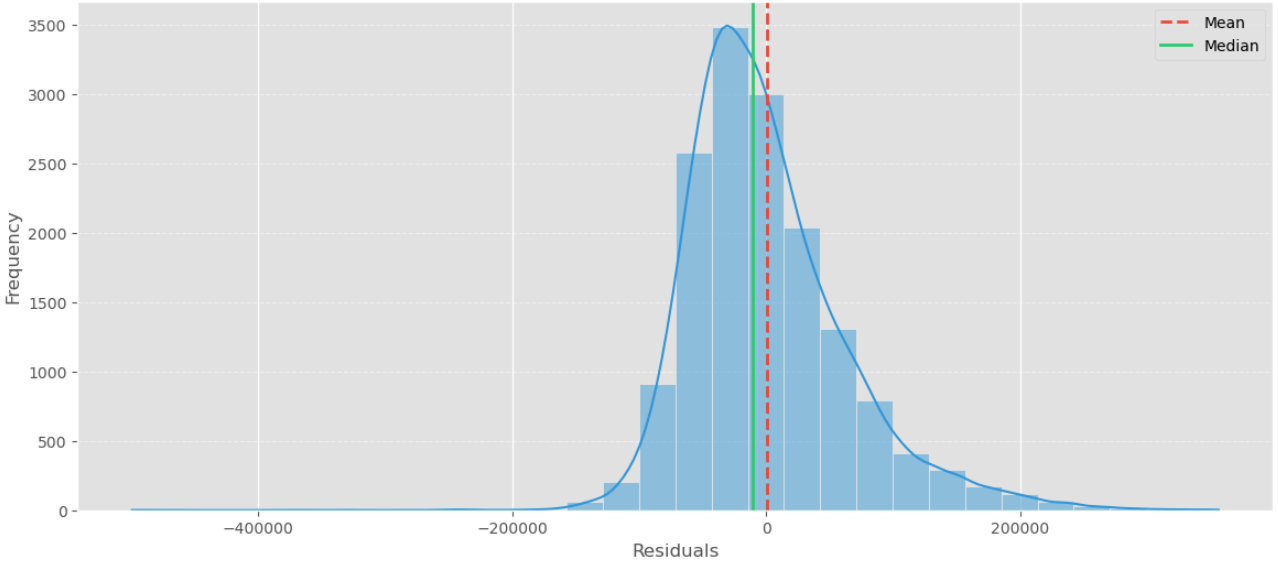


Figure 7.5.2: Predicted vs. Actual Median House Value for Ridge Linear Regression (Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value)

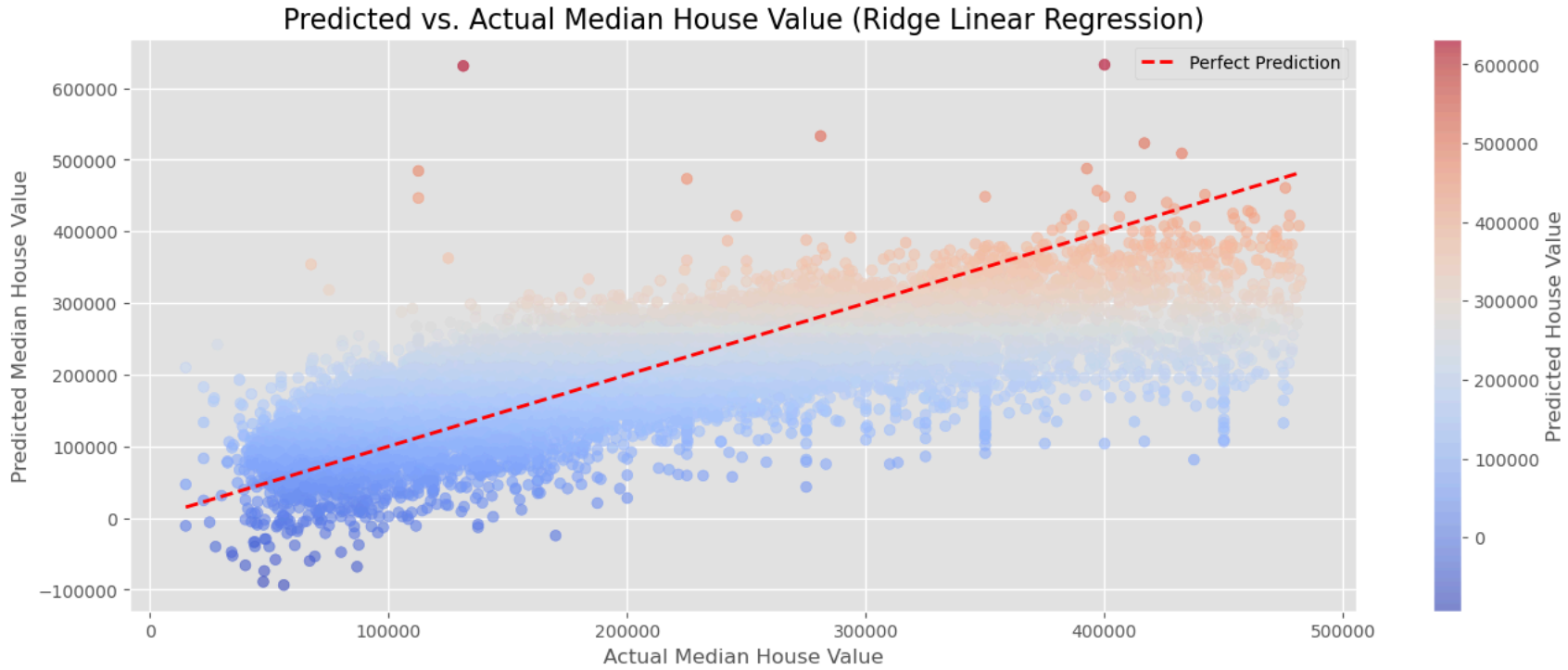
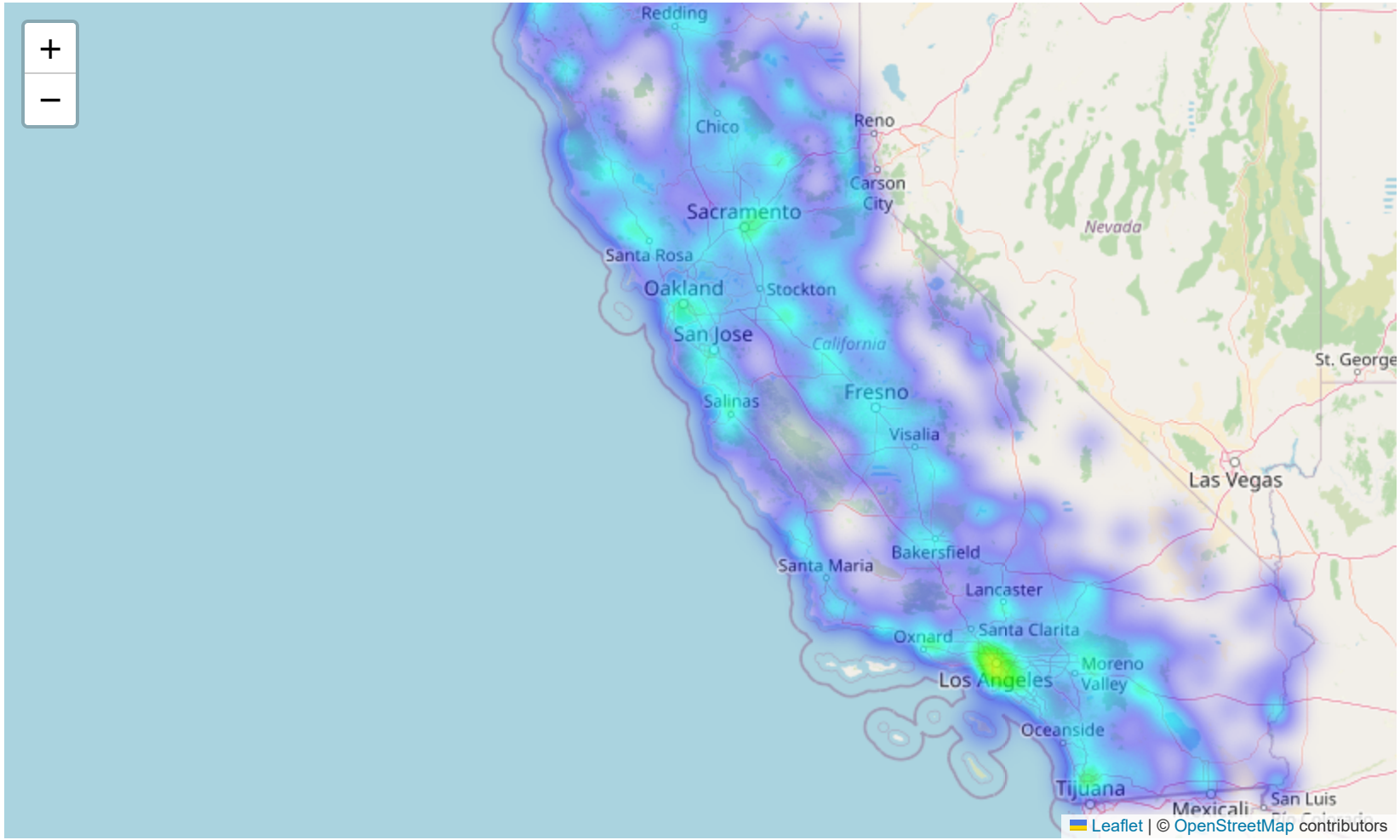


Figure 7.5.3: Heatmap of Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value for Ridge Linear Regression

Out[182...



Interpretation: The Ridge Regression model, using median income to predict median house value, provides the following insights:

- MSE of 63196 is an improvement over simpler models, showing reduced prediction errors, though still relatively high in terms of housing price prediction.
- MAE of 47,686 indicates that predictions, on average, deviate by about $47,686 from actual house prices, a reasonable improvement compared to simpler linear regression models.
- R^2 of 0.560 suggests that about 56% of the variance in median house prices is explained by this model, which shows that the inclusion of multiple features improves predictive performance.

Chart Analysis:

- **Residual Distribution Plot**:
  - The residuals are symmetrically distributed around zero, suggesting minimal bias in predictions.
  - The slight skew and the tails of the distribution indicate that the model has difficulty with very high or very low house prices, leading to increased prediction errors in those cases.
  - The mean and median residuals being close to zero confirm that the model does not have systematic bias, but still struggles with extreme outliers.
- **Predicted vs. Actual Scatter Plot**:
  - The scatter plot shows a clear positive trend, indicating that the model generally performs well.
  - The points are more tightly clustered around the line of perfect prediction, though there are still some outliers where the predictions deviate significantly from actual values.
  - The color gradient shows that higher predicted values tend to have greater deviations from the actual prices, which is common in housing price models due to the complex factors influencing high-value properties.
- **Heatmap**:
  - The heatmap shows strong geographical trends, with higher predicted prices concentrated around urban centers like San Francisco and Los Angeles, indicating the model's sensitivity to location as a key predictor.
  - This spatial representation highlights the importance of combining both geographical and economic factors, such as median income, to improve the model's accuracy.

6. Lasso Regression (Median House Value using Median Income, Longitude, Latitude and Housing Median Age)

Lasso Regression (Least Absolute Shrinkage and Selection Operator) is a regularization method that penalizes the absolute value of coefficients, allowing it to reduce some coefficients to zero and effectively select relevant features. In this model, we aim to predict median house value using median income, geographical location (longitude, latitude), and housing median age.

$$\text{Median House Value} = \beta_0 + \beta_1 \cdot \text{Median Income} + \beta_2 \cdot \text{Longitude} + \beta_3 \cdot \text{Latitude} + \beta_4 \cdot \text{Housing Median Age} + \lambda \sum_{i=1}^{n} |\beta_i| +$$

Where:

- β_0 is the intercept,
- $\beta_1, \beta_2, \beta_3, \beta_4$ are the coefficients for median income, longitude, latitude, and housing median age,

- λ is the regularization parameter that penalizes coefficients to avoid overfitting,
- ϵ represents the residuals (errors).

In [182…

```
model_no = 6
model_name = 'Lasso'
title_suffix = 'Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value'

# training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=sample_test_size, random_state=sample_random_stat

# Train the Lasso regression model
lasso_model = Lasso(alpha=1.0)

# Evaluate the model
y_pred = evaluate_model(X_train, X_test, y_train, y_test, model, model_no, model_name)

# Plot residuals vs. actual for the Lasso model
plot_residuals_vs_actual(y_test, y_pred, model_no, model_name, title_suffix)

# Display the map
california_map = plot_geospatial_heatmap(X_test, y_test, y_pred, model_no, model_name, title_suffix)
california_map
```

```
6. Lasso Linear Regression Model Evaluation:
Mean Squared Error (MSE): 63196.63009691232
Mean Absolute Error (MAE): 47686.2707734986
R-Squared (R²): 0.560252919281792

Figure 7.6.1: Residual Distribution for Lasso Linear Regression (Median Income, Longitude, Latitude and Housing Media
n Age vs. Median House Value)
```

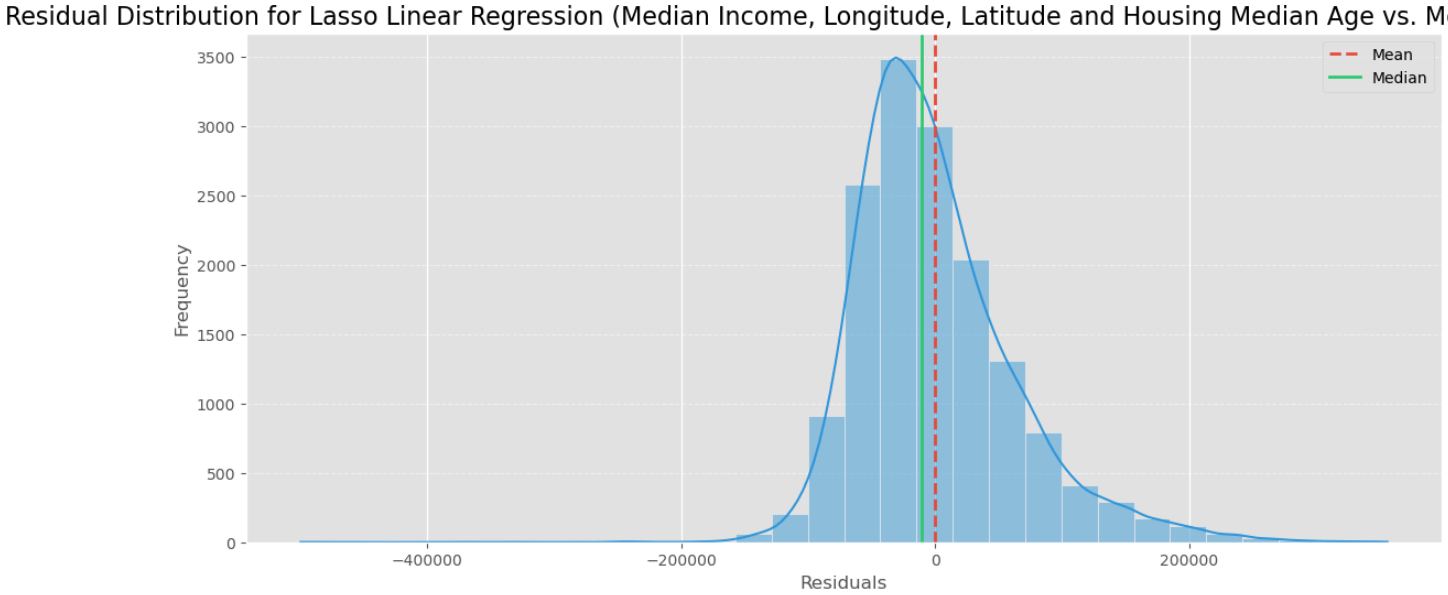


Figure 7.6.2: Predicted vs. Actual Median House Value for Lasso Linear Regression (Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value)

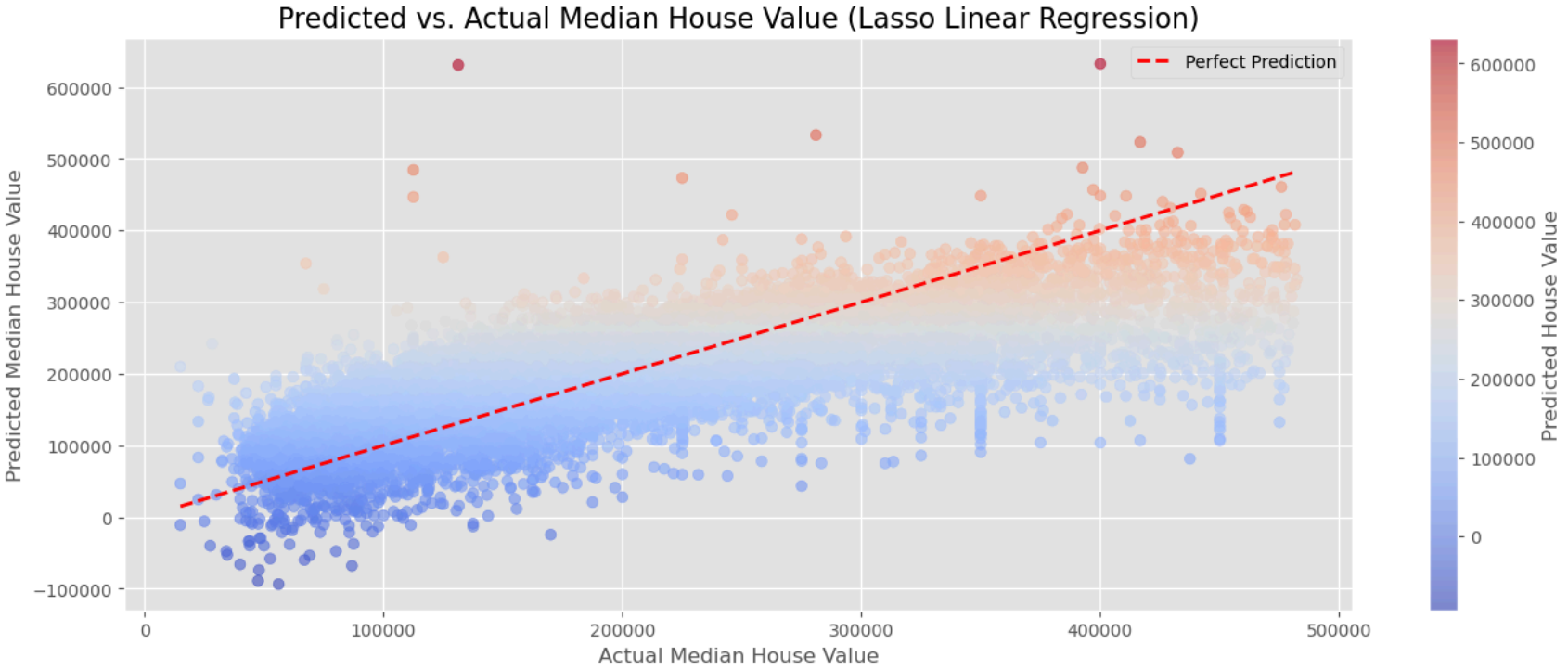
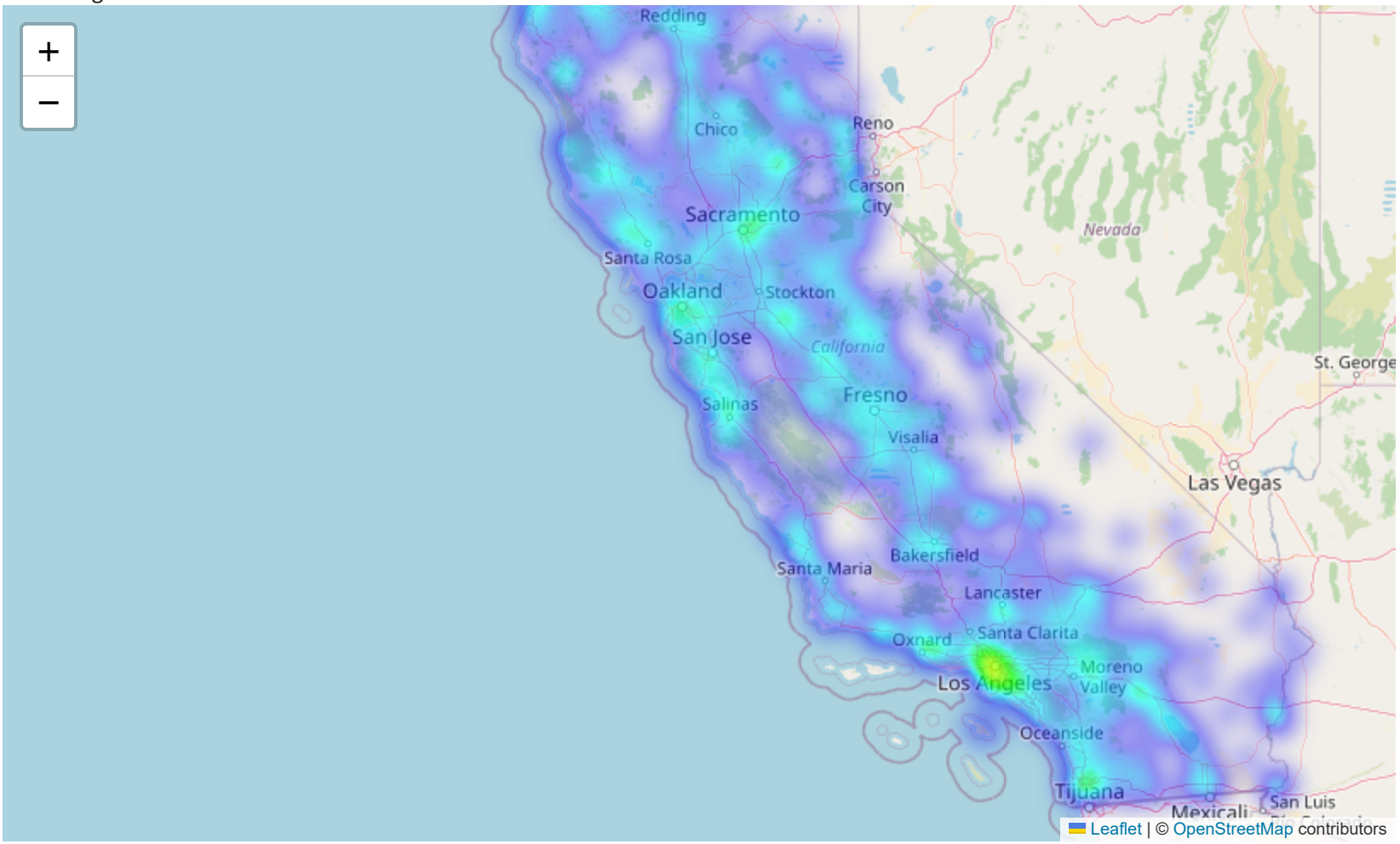


Figure 7.6.3: Heatmap of Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value for Lasso Linear Regression

Out[182…



Interpretation: The Lasso Regression model, using median income to predict median house value, provides the following insights:

- MSE of 63196 is similar to Ridge regression, showing reasonable prediction performance, though errors remain high, particularly for more expensive homes.
- The MAE of 47,686 shows that predictions deviate, on average, by about $47,686 from the true house prices. While this is an improvement over simpler models, it still suggests that other factors not included in the model (e.g., home size, quality) may play a significant role.

- R^2 value of 0.560 indicates that about 56% of the variance in house prices is explained by the model. This regularized model performs similarly to Ridge Regression, suggesting both models benefit from including geographic and demographic features.

Chart Analysis:

- **Residual Distribution Plot**:
  - The residuals show a normal-like distribution centered around zero, indicating that the model is relatively unbiased.
  - However, the distribution is wider at the tails, showing that the model struggles with extreme house values, both low and high.
  - The close alignment of mean and median residuals suggests that the model does not have any systematic bias.
- **Predicted vs. Actual Scatter Plot**:
  - The scatter plot shows a clear positive trend, with predicted values closely following the actual values, though some deviations are still present, especially at higher price ranges.
  - The color gradient shows that higher predicted values are spread further from the perfect prediction line, reflecting higher errors for more expensive homes.
- **Heatmap**:
  - The heatmap visualization highlights higher predicted house prices in urban areas such as San Francisco, Los Angeles, and San Diego, confirming that the model captures the effect of location well.
  - However, geographical features alone do not fully explain house price variations, as demonstrated by the prediction errors in more expensive regions.

7. **Polynomial Regression (Median House Value using Median Income, Longitude, Latitude and Housing Median Age)**

Polynomial regression allows the model to account for non-linear relationships between the predictors and the target variable. In this model, we aim to predict the median house value using the features median income, longitude, latitude, and housing median age. By using a polynomial transformation, we enable the model to capture more complex patterns that a simple linear regression may miss.

$$\text{Median House Value} = \beta_0 + \beta_1 \cdot \text{Median Income} + \beta_2 \cdot \text{Longitude} + \beta_3 \cdot \text{Latitude} + \beta_4 \cdot \text{Housing Median Age} + \epsilon$$

where:

- β_0 is the intercept,
- $\beta_1, \beta_2, \beta_3, \beta_4$ are the coefficients for median income, longitude, latitude, and housing median age,
- ϵ represents the residuals (errors),
- The model includes polynomial terms to capture non-linear relationships.

In [182…

```
model_no = 7
model_name = 'Polynomial'
title_suffix = 'Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value'

# Transforming the feature set into polynomial features
poly_transformer = PolynomialFeatures(degree=2)  # Degree can be adjusted
X_poly_transformed = poly_transformer.fit_transform(X)

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_poly_transformed, y, test_size=sample_test_size, random_state=s

# Creating and training the Polynomial Regression model
model = LinearRegression()

# Evaluate the model
y_pred = evaluate_model(X_train, X_test, y_train, y_test, model, model_no, model_name)

# Plot residuals vs. actual for the Lasso model
plot_residuals_vs_actual(y_test, y_pred, model_no, model_name, title_suffix)

# Display the map
california_map = plot_geospatial_heatmap(X_test, y_test, y_pred, model_no, model_name, title_suffix, poly_transformer
california_map
```

```
7. Polynomial Linear Regression Model Evaluation:
Mean Squared Error (MSE): 60901.114569328405
Mean Absolute Error (MAE): 44511.59213809611
R-Squared (R²): 0.5916189271856771
```

Figure 7.7.1: Residual Distribution for Polynomial Linear Regression (Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value)

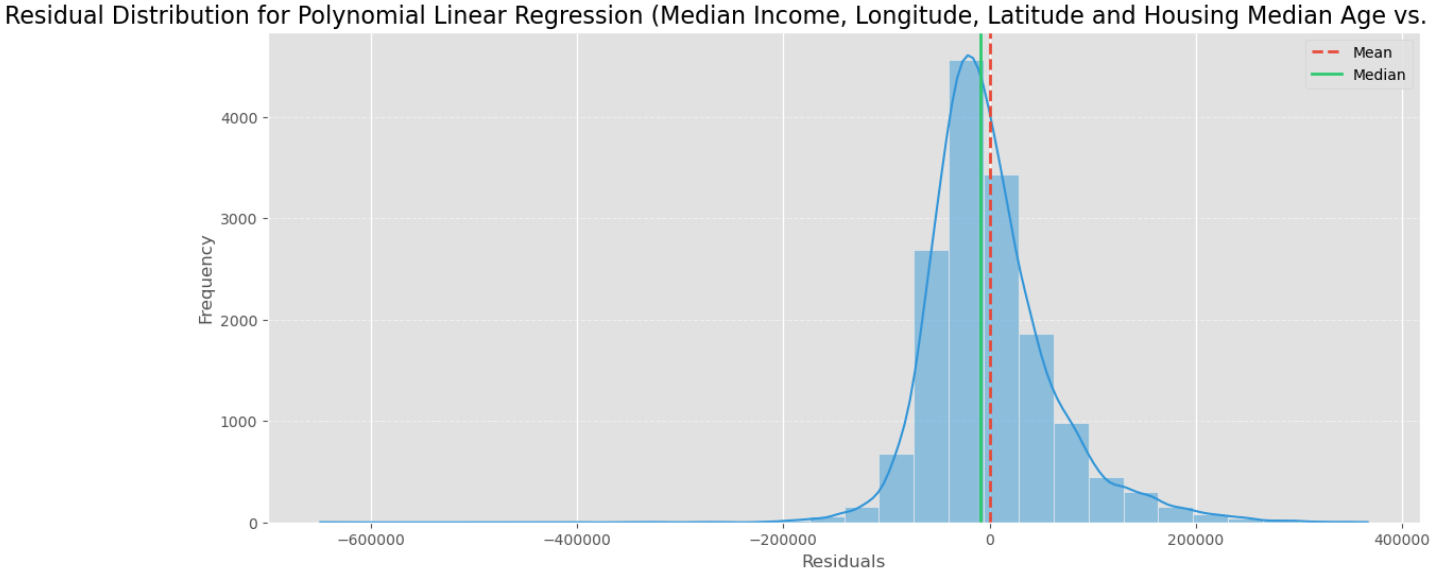


Figure 7.7.2: Predicted vs. Actual Median House Value for Polynomial Linear Regression (Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value)

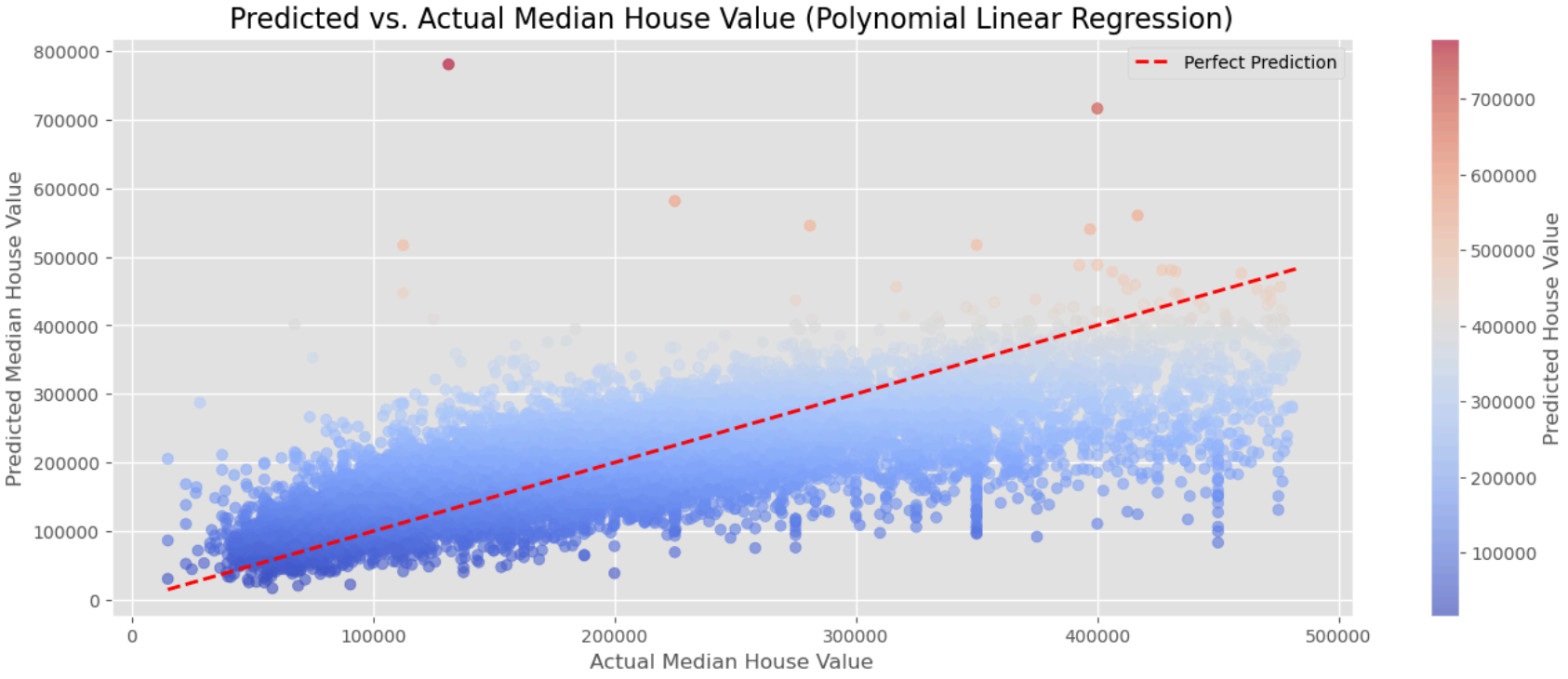
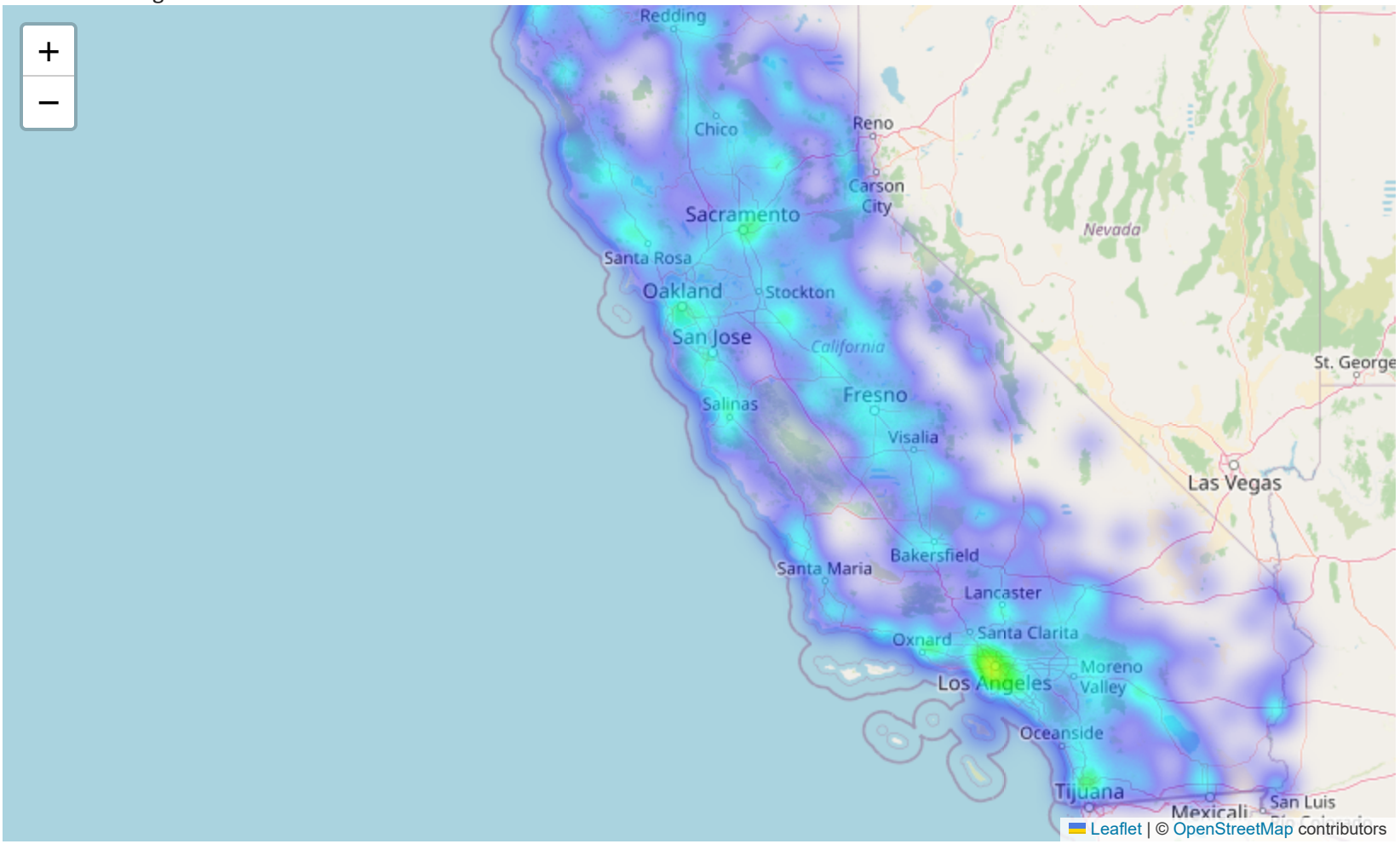


Figure 7.7.3: Heatmap of Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value for Polynomial Linear Regression

Out[182…



Interpretation: The Polynomial Regression model, using median income, longitude, latitude, and housing median age to predict median house value, yields the following insights:

- MSE of 60901 indicates a substantial improvement in prediction accuracy compared to simpler models. The lower error highlights the model's ability to capture non-linear relationships in the data.
- MAE of \$44,511 reflects a notable reduction in average prediction error compared to univariate and bivariate models, indicating improved accuracy.

- R^2 of 0.592 shows that approximately 59.2% of the variance in median house values is explained by this combination of features, representing a moderate improvement over linear models.

Chart Analysis:

- **Residual Distribution Plot**:
  - The residuals are mostly centered around zero, suggesting no major bias in the model's predictions.
  - The distribution is relatively symmetrical with slight skewness. The higher number of residuals near zero shows that the model has better predictive accuracy than previous models.
  - Both the mean and median residuals being near zero indicate that the errors are evenly distributed.
- **Predicted vs. Actual Scatter Plot**:
  - A more pronounced positive trend is visible, with fewer points deviating from the line of perfect prediction compared to other models. This suggests improved prediction performance.
  - Higher predicted house prices (in the red region) generally align more closely with actual values, reducing the overestimation observed in previous models.
  - The color gradient represents predicted values, showing that the model performs better for mid-range and higher house prices, but some errors still persist at the lower end.
- **Heatmap**:
  - The heatmap clearly shows areas with higher predicted house values concentrated in major metropolitan regions such as Los Angeles, San Francisco, and San Diego. This visualization emphasizes the geographic distribution of house prices.
  - However, while useful for geographical insights, the heatmap alone may not capture the nuanced relationship between predictors like income and house age, which are critical to this model's accuracy.

8. **Random Forest Regression (Median House Value using Median Income, Longitude, Latitude and Housing Median Age)**

This model aims to predict the median house value based on a combination of geographical (longitude, latitude), economic (median income), and housing (housing median age) features using a Random Forest regression model.

$$\hat{y} = \frac{1}{N_{\text{trees}}} \sum_{i=1}^{N_{\text{trees}}} T_i(\text{Median Income}, \text{Longitude}, \text{Latitude}, \text{Housing Median Age})$$

Where:

- $\hat{y}$ is the predicted median house value.
- N_{trees} is the number of decision trees in the random forest.
- $T_i(\text{Median Income}, \text{Longitude}, \text{Latitude}, \text{Housing Median Age})$ is the prediction from the i-th decision tree for the given features: Median Income, Longitude, Latitude, and Housing Median Age.

In [182…

```
model_no = 8
model_name = 'Random Forest'
title_suffix = 'Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value'

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=sample_test_size, random_state=sample_random_stat

# Creating and training the Random Forest Regressor model
model = RandomForestRegressor(n_estimators=100, random_state=sample_random_state)

# Evaluate the model
y_pred = evaluate_model(X_train, X_test, y_train, y_test, model, model_no, model_name)

# Plot residuals vs. actual for the Lasso model
plot_residuals_vs_actual(y_test, y_pred, model_no, model_name, title_suffix)

# Display the map
california_map = plot_geospatial_heatmap(X_test, y_test, y_pred, model_no, model_name, title_suffix)
california_map
```

```
8. Random Forest Linear Regression Model Evaluation:
Mean Squared Error (MSE): 47994.39438105085
Mean Absolute Error (MAE): 33100.54105962829
R-Squared (R²): 0.7463726081256825
```

Figure 7.8.1: Residual Distribution for Random Forest Linear Regression (Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value)

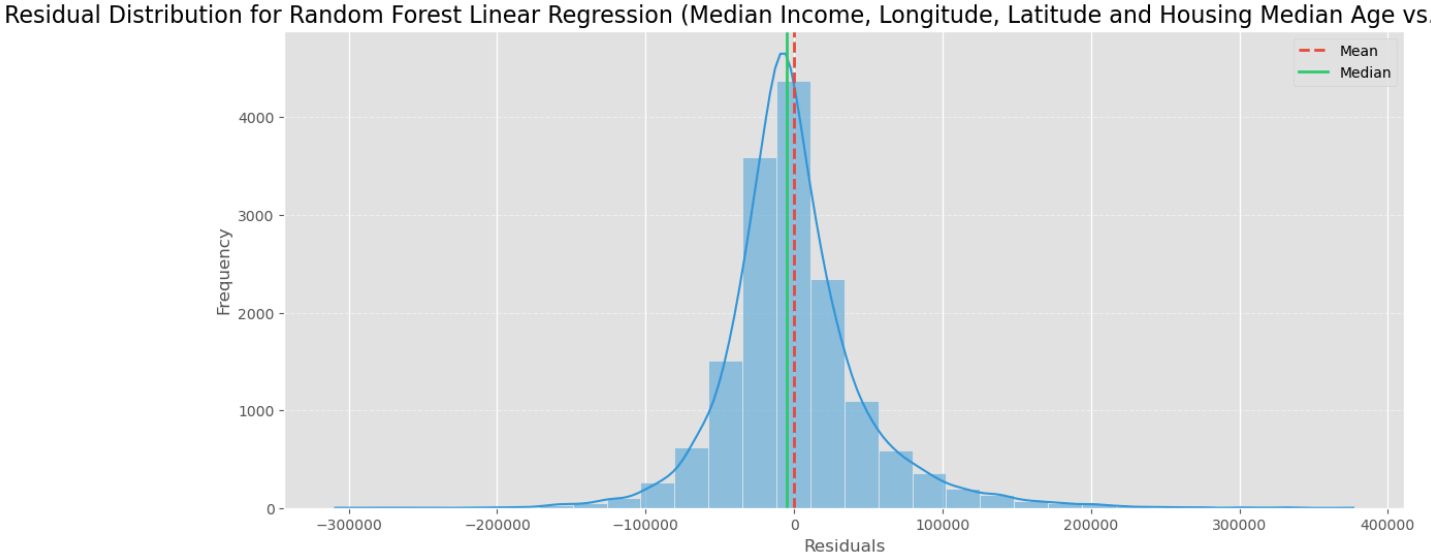


Figure 7.8.2: Predicted vs. Actual Median House Value for Random Forest Linear Regression (Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value)

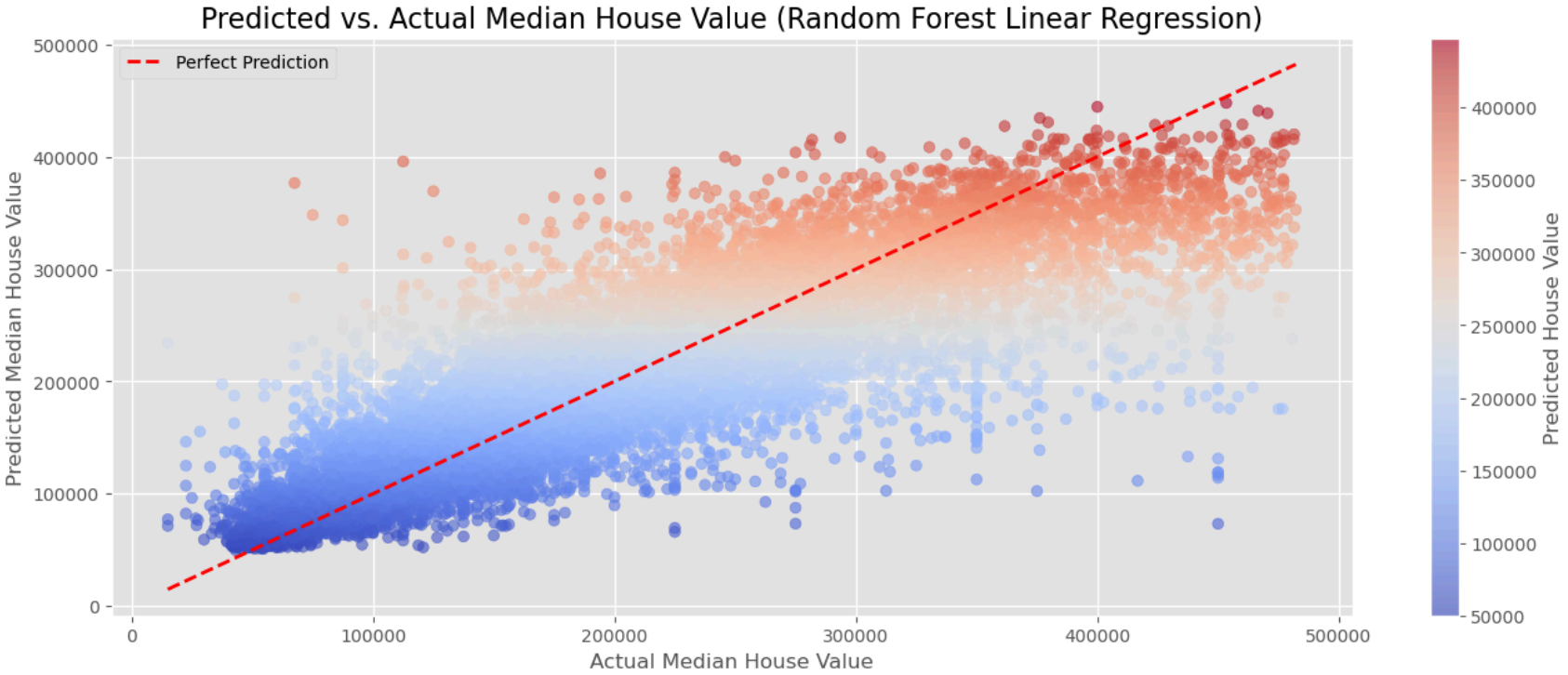
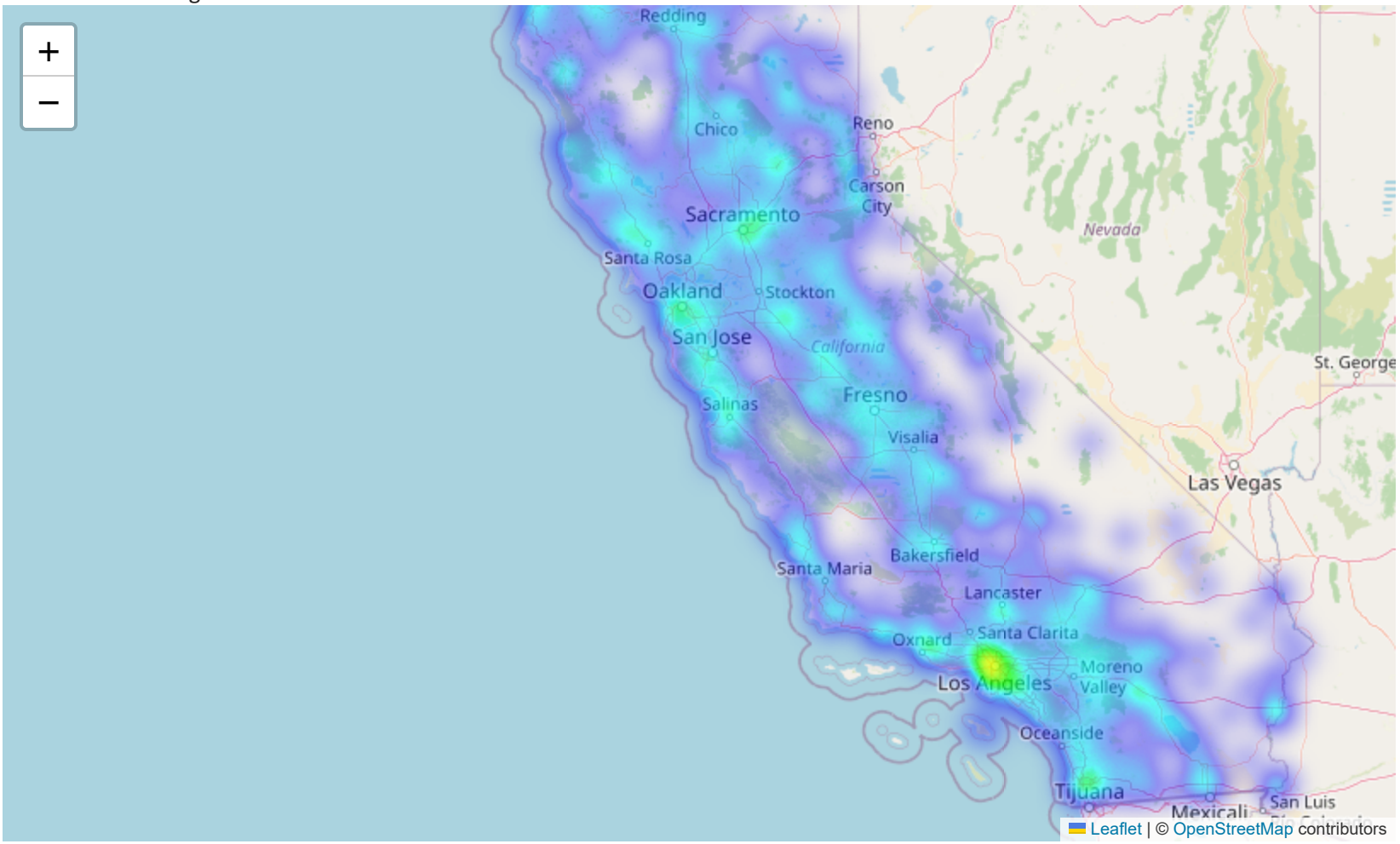


Figure 7.8.3: Heatmap of Median Income, Longitude, Latitude and Housing Median Age vs. Median House Value for Random Forest Linear Regression

Out[182…



Interpretation: The Random Forest Regression model, using Median Income, Longitude, Latitude, and Housing Median Age to predict Median House Value, offers several key insights:

- MSE of approximately 47994 indicates significantly lower prediction errors compared to other linear regression models.
- MAE of \$33,100 reflects relatively smaller average prediction errors, indicating improved accuracy in predicting housing prices compared to simpler models.
- R^2 of 0.746 shows that the predictors explain approximately 74.6% of the variance in house prices, demonstrating a strong relationship between these variables and median house value.

Chart Analysis:

- **Residual Distribution Plot**:
  - The residuals are tightly centered around zero, indicating a low bias in the predictions. This suggests that the Random Forest model performs better at capturing the underlying patterns of the data compared to other models.
  - The distribution of residuals is narrower and more symmetric than in previous models, with a shorter tail on the right side, showing that this model deals better with outliers.
  - The mean and median residuals are almost aligned, further indicating minimal bias in the model's predictions.
- **Predicted vs. Actual Scatter Plot**:
  - The scatter plot shows a much stronger positive trend compared to other models, with most of the points tightly clustered around the line of perfect prediction. This indicates better prediction accuracy overall.
  - The color gradient from blue to red reflects predicted values, with more points clustered around higher actual values. The lower dispersion of points around the perfect prediction line emphasizes the Random Forest model's superior ability to predict house prices.
  - There is a visible reduction in under- and overestimation for mid-range housing values, suggesting that the model better captures variability across different price ranges compared to previous models.
- **Heatmap**:
  - The heatmap indicates higher predicted median house values concentrated around metropolitan areas like Los Angeles, San Francisco, and Sacramento, which aligns well with actual data. The Random Forest model produces a more nuanced and accurate geographic distribution of predicted house prices, especially in densely populated regions.

8. Comparative Analysis:

It provides a brief evaluation of multiple regression models, assessing their effectiveness in predicting California housing prices. By analyzing metrics such as MSE, R-squared, and MAE, this section highlights each model's performance, strengths, and limitations, ultimately guiding the selection of the most suitable model for robust predictions.

In [182…

```
# Convert results dictionary to DataFrame
results_df = pd.DataFrame(results)

# Displaying the results table for reference
```

```
print("\nFigure 8.1: Evaluation Metrics for models")
print(results_df)

print("\nFigure 8.2: Model Comparison for California Housing Data")

# Combined plot for Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-Squared (R²)
plt.figure(figsize=(12, 8))
fig, ax1 = plt.subplots(figsize=(12, 8))

# Define color palette for a modern look
mse_color = '#66c2a5'   # Soft green for MSE
mae_color = '#fc8d62'   # Soft coral for MAE
r2_color = '#8da0cb'    # Soft blue for R²

# Plot for Mean Squared Error (MSE)
ax1.plot(results_df['Model'], results_df['MSE'], marker='o', linestyle='-', color=mse_color, linewidth=2, markersize=
ax1.set_xlabel("Model", fontsize=14)
ax1.set_ylabel("MSE", fontsize=14, color=mse_color)
ax1.tick_params(axis='y', labelcolor=mse_color)
for i, value in enumerate(results_df['MSE']):
    ax1.text(i, value, f"{value:.2e}", ha='center', va='bottom', fontsize=10, color=mse_color)
ax1.grid(True, linestyle='--', alpha=0.5)

# Plot for Mean Absolute Error (MAE)
ax2 = ax1.twinx()
ax2.plot(results_df['Model'], results_df['MAE'], marker='o', linestyle='-', color=mae_color, linewidth=2, markersize=
ax2.set_ylabel("MAE", fontsize=14, color=mae_color)
ax2.tick_params(axis='y', labelcolor=mae_color)
for i, value in enumerate(results_df['MAE']):
    ax2.text(i, value, f"{value:.2f}", ha='center', va='bottom', fontsize=10, color=mae_color)

# Plot for R-Squared (R²)
ax3 = ax1.twinx()
ax3.spines['right'].set_position(('outward', 60))
ax3.plot(results_df['Model'], results_df['R²'], marker='o', linestyle='-', color=r2_color, linewidth=2, markersize=8,
ax3.set_ylabel("R²", fontsize=14, color=r2_color)
ax3.tick_params(axis='y', labelcolor=r2_color)
for i, value in enumerate(results_df['R²']):
    ax3.text(i, value, f"{value:.2f}", ha='center', va='bottom', fontsize=10, color=r2_color)

# Adding legends
fig.legend(loc='upper left', bbox_to_anchor=(0.1, 1), bbox_transform=ax1.transAxes, frameon=False)
```

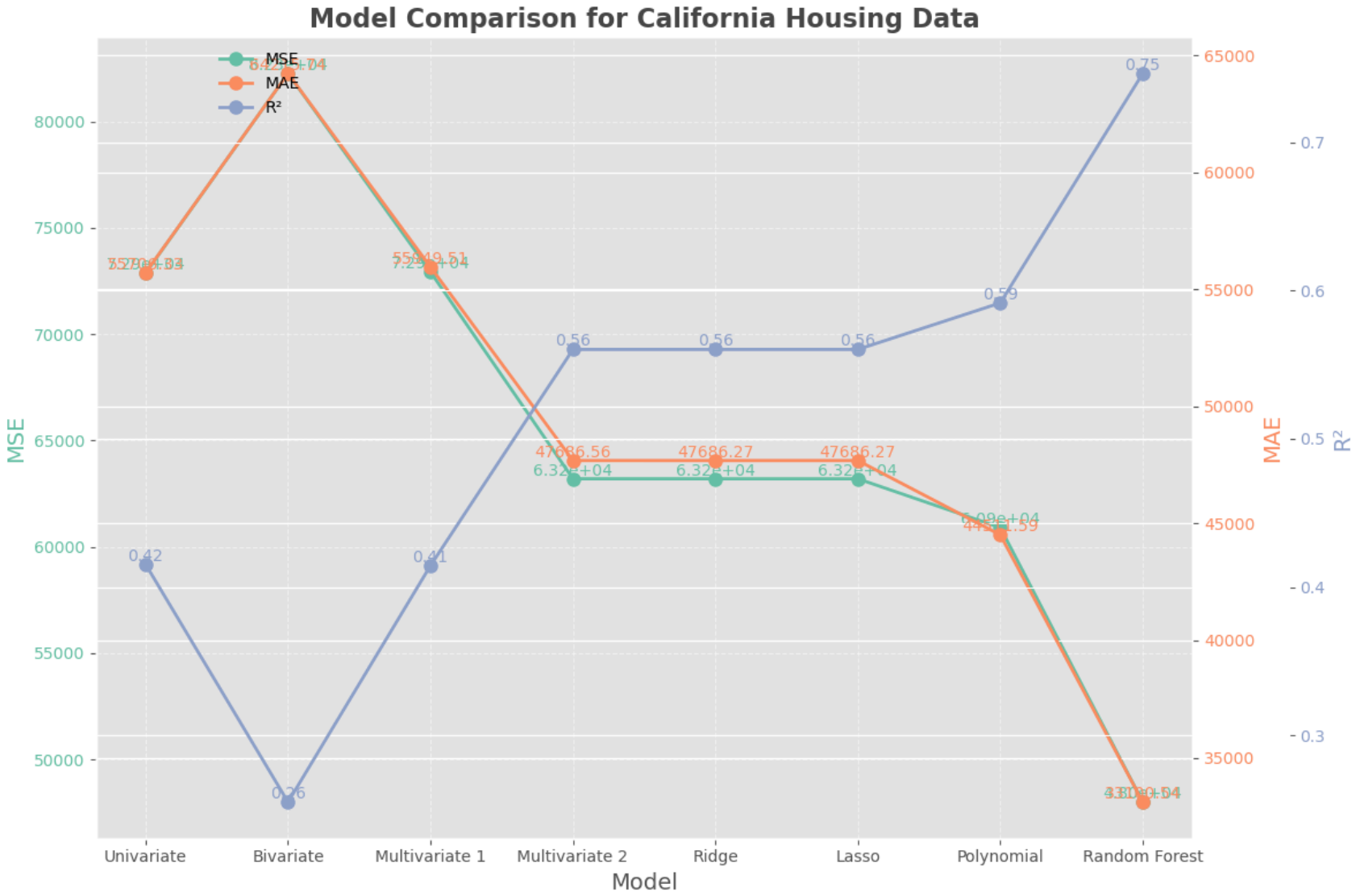
```
# Title enhancement
plt.title("Model Comparison for California Housing Data", fontsize=16, weight='bold', color='#4a4a4a')

plt.tight_layout()
plt.show()
```

Figure 8.1: Evaluation Metrics for models

	Model	MSE	R^2	MAE
0	Univariate	72870.602084	0.415318	55706.328039
1	Bivariate	82255.962125	0.255011	64215.742923
2	Multivariate 1	72922.803967	0.414480	55949.512409
3	Multivariate 2	63197.010704	0.560248	47686.560500
4	Ridge	63196.630097	0.560253	47686.270773
5	Lasso	63196.630097	0.560253	47686.270773
6	Polynomial	60901.114569	0.591619	44511.592138
7	Random Forest	47994.394381	0.746373	33100.541060

Figure 8.2: Model Comparison for California Housing Data

<Figure size 1200x800 with 0 Axes>



Analysis Summary:

1. **Univariate**: The model is limited, with a relatively high error because it uses only one feature, median income.
2. **Bivariate**: Shows the poorest performance, suggesting that longitude and latitude alone are insufficient to predict housing prices.
3. **Multivariate 1**: Adding total rooms and population to the model does not improve performance significantly.

4. **Multivariate 2**: Adding housing median age along with geographical factors and income improves performance considerably.
5. **Ridge/Lasso**: Both regularization techniques stabilize the model, providing similar results to Multivariate 2.
6. **Polynomial Regression**: This model identifies higher-order relationships, achieving better predictions than the linear models.
7. **Random Forest**: The standout model, performing best with significantly lower MSE and MAE and a higher R^2 value, capturing complex non-linear patterns in the data.

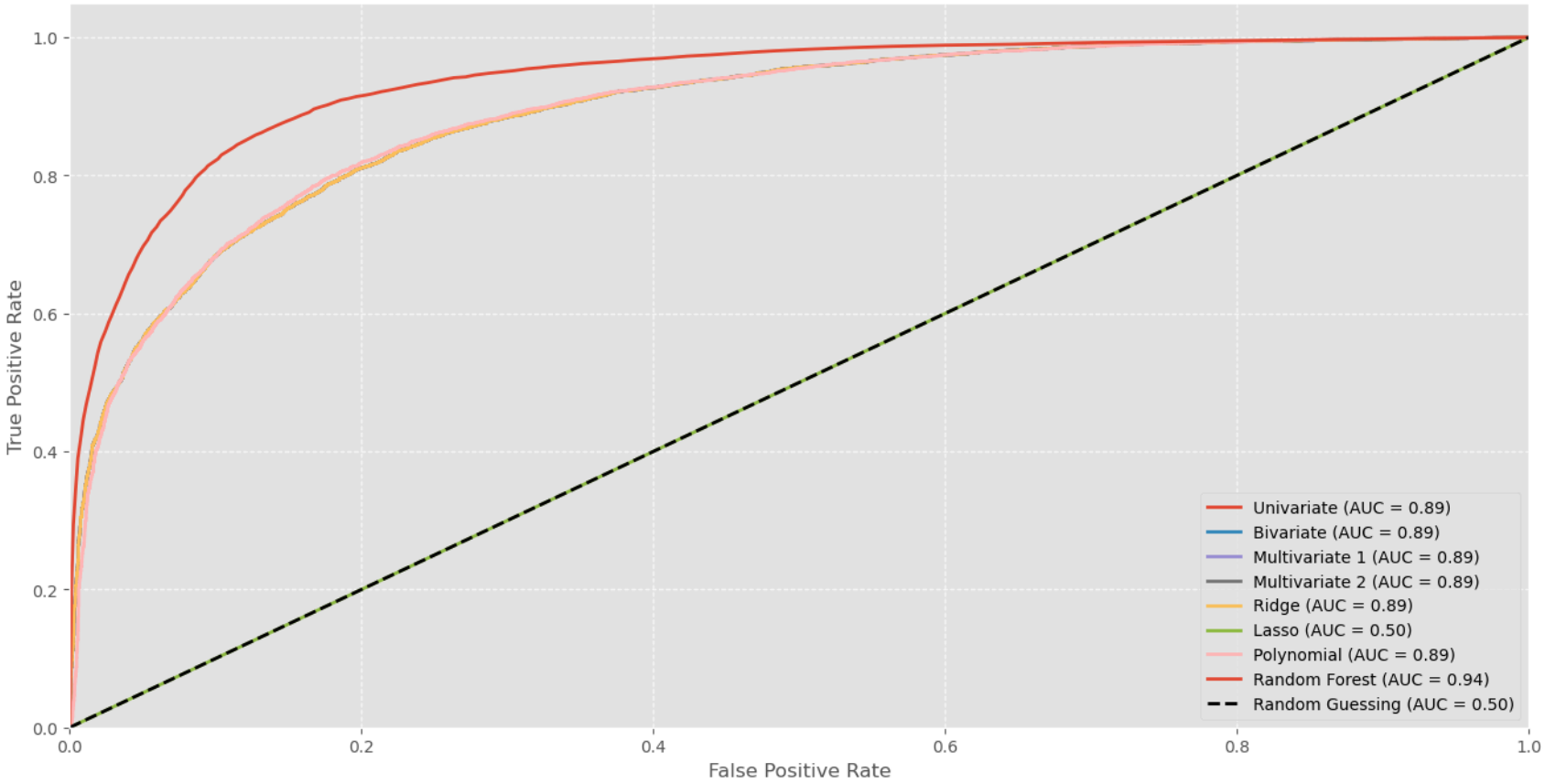
ROC (Receiver Operating Characteristic) curves offer a graphical method to assess the performance of different models by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The Area Under the Curve (AUC) value associated with each ROC curve quantifies the model's ability to distinguish between classes, where a higher AUC signifies better classification accuracy.

In [182…

```
# Convert the regression problem to binary classification
# Threshold can be the median of median_house_value
median_value = data['median_house_value'].median()
y_binary = (data['median_house_value'] >= median_value).astype(int)  # 1 for high-value houses, 0 for low-value

# Prepare the features
X = data[['median_income', 'longitude', 'latitude', 'housing_median_age']]
X_train, X_test, y_train, y_test = train_test_split(X, y_binary, test_size=sample_test_size, random_state=sample_ran

# Define models
models = {
    "Univariate": LinearRegression(),
    "Bivariate": LinearRegression(),
    "Multivariate 1": LinearRegression(),
    "Multivariate 2": LinearRegression(),
    "Ridge": Ridge(alpha=1.0),
    "Lasso": Lasso(alpha=1.0),
    "Polynomial": PolynomialFeatures(degree=2),
    "Random Forest": RandomForestRegressor(n_estimators=100, random_state=42)
}

plt.figure(figsize=(16, 8))

# Loop through each model, fit, predict, and calculate ROC curve
for model_name, model in models.items():
    if model_name == "Polynomial":
        # Polynomial features transformation
        poly_transformer = PolynomialFeatures(degree=2)
```

```
        X_train_poly = poly_transformer.fit_transform(X_train)
        X_test_poly = poly_transformer.transform(X_test)
        poly_model = LinearRegression()
        poly_model.fit(X_train_poly, y_train)
        y_prob = poly_model.predict(X_test_poly)
    else:
        # Train other models
        model.fit(X_train, y_train)
        y_prob = model.predict(X_test)

    # Calculate ROC curve and AUC for the model
    fpr, tpr, _ = roc_curve(y_test, y_prob)
    roc_auc = auc(fpr, tpr)

    # Plot the ROC curve
    plt.plot(fpr, tpr, lw=2, label=f'{model_name} (AUC = {roc_auc:.2f})')

# Plot the diagonal line for random guessing
plt.plot([0, 1], [0, 1], 'k--', lw=2, label='Random Guessing (AUC = 0.50)')

# Adding plot details
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel("False Positive Rate", fontsize=12)
plt.ylabel("True Positive Rate", fontsize=12)
plt.title("Figure 8.2: Model Comparison for California Housing Data (ROC Curve)", fontsize=14)
plt.legend(loc="lower right", fontsize=10)
plt.grid(True, linestyle='--', alpha=0.7)
plt.show()
```

Figure 8.2: Model Comparison for California Housing Data (ROC Curve)



The ROC curve compares the true positive rate to the false positive rate for different models, measuring their ability to distinguish between classes. Higher AUC values indicate better performance, with Random Forest (AUC = 0.94) performing best. Other models, including Univariate, Bivariate, and Polynomial Regression, have lower but still strong AUC values around 0.89, showing that they also perform well but are less effective than Random Forest. The diagonal line (AUC = 0.50) represents random guessing.

9. Hypothesis Validation and testing:

In [183…

```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=sample_test_size, random_state=sample_random_stat
```

```
# Function to perform hypothesis testing for OLS models
def hypothesis_test_ols(model):
    print(model.summary())  # This shows p-values, t-statistics, and more

# 1. Linear Regression Hypothesis Testing
X_train_const = sm.add_constant(X_train)
X_test_const = sm.add_constant(X_test)

linear_model = sm.OLS(y_train, X_train_const).fit()
hypothesis_test_ols(linear_model)  # Perform hypothesis testing for Linear Regression

# 2. Ridge Regression Hypothesis Testing (p-values not available directly in Ridge regression)
model = Ridge(alpha=1.0)
model.fit(X_train, y_train)

# Printing coefficients for Ridge, no p-values are available, so just the coefficients.
print("Ridge Regression Coefficients:", model.coef_)

# 3. Lasso Regression Hypothesis Testing (p-values not available directly in Lasso regression)
lasso_model = Lasso(alpha=1.0)
lasso_model.fit(X_train, y_train)

# Printing coefficients for Lasso
print("Lasso Regression Coefficients:", lasso_model.coef_)

# 4. Polynomial Regression Hypothesis Testing
poly_transformer = PolynomialFeatures(degree=2)
X_poly_train = poly_transformer.fit_transform(X_train)
X_poly_test = poly_transformer.transform(X_test)

poly_model = sm.OLS(y_train, X_poly_train).fit()
hypothesis_test_ols(poly_model)  # Perform hypothesis testing for Polynomial Regression

# 5. Random Forest Regression (Not suitable for hypothesis testing; feature importance can be used)
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Print feature importance
print("Random Forest Feature Importance:", rf_model.feature_importances_)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     median_house_value   R-squared:                       0.558
Model:                            OLS   Adj. R-squared:                  0.558
Method:                 Least Squares   F-statistic:                     1221.
Date:                Sat, 19 Oct 2024   Prob (F-statistic):               0.00
Time:                        17:28:45   Log-Likelihood:                -48338.
No. Observations:                3873   AIC:                         9.669e+04
Df Residuals:                    3868   BIC:                         9.672e+04
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                         coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const              -3.623e+06   1.28e+05    -28.210      0.000   -3.87e+06   -3.37e+06
median_income       3.543e+04    696.151     50.888      0.000    3.41e+04    3.68e+04
longitude          -4.303e+04   1450.068    -29.677      0.000   -4.59e+04   -4.02e+04
latitude            -4.15e+04   1339.799    -30.972      0.000   -4.41e+04   -3.89e+04
housing_median_age   639.9831     87.514      7.313      0.000     468.405     811.561
==============================================================================
Omnibus:                      610.092   Durbin-Watson:                   1.993
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1163.007
Skew:                           0.976   Prob(JB):                    2.86e-253
Kurtosis:                       4.843   Cond. No.                     1.61e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.61e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
Ridge Regression Coefficients: [ 35427.30136513 -42995.98199156 -41460.5514696     640.57925873]
Lasso Regression Coefficients: [ 35425.86090114 -43030.81672695 -41492.60712496    640.02879328]
                            OLS Regression Results
==============================================================================
Dep. Variable:     median_house_value   R-squared:                       0.594
Model:                            OLS   Adj. R-squared:                  0.592
Method:                 Least Squares   F-statistic:                     402.8
Date:                Sat, 19 Oct 2024   Prob (F-statistic):               0.00
Time:                        17:28:45   Log-Likelihood:                -48175.
No. Observations:                3873   AIC:                         9.638e+04
Df Residuals:                    3858   BIC:                         9.647e+04
Df Model:                          14
```

```
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       5.573e+07   1.04e+07      5.384      0.000    3.54e+07     7.6e+07
x1          -5.54e+05   9.25e+04     -5.991      0.000   -7.35e+05   -3.73e+05
x2          1.397e+06   2.25e+05      6.202      0.000    9.55e+05    1.84e+06
x3          1.533e+06   1.87e+05      8.216      0.000    1.17e+06     1.9e+06
x4         -6.103e+04   1.12e+04     -5.433      0.000   -8.31e+04    -3.9e+04
x5            123.5766    272.227      0.454      0.650    -410.147     657.300
x6         -6759.3123   1057.488     -6.392      0.000   -8832.601   -4686.024
x7         -6405.5299   1002.704     -6.388      0.000   -8371.411   -4439.649
x8            300.9656     58.779      5.120      0.000     185.725     416.206
x9            8744.9434   1248.565      7.004      0.000    6297.033    1.12e+04
x10         1.911e+04   2140.625      8.926      0.000    1.49e+04    2.33e+04
x11          -711.1416    128.805     -5.521      0.000    -963.675    -458.608
x12         1.034e+04   1021.434     10.120      0.000    8334.790    1.23e+04
x13          -724.9904    122.169     -5.934      0.000    -964.512    -485.468
x14            23.6474      6.332      3.735      0.000      11.234      36.061
==============================================================================
Omnibus:                      594.922   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1301.433
Skew:                           0.902   Prob(JB):                    2.50e-283
Kurtosis:                       5.194   Cond. No.                     1.63e+08
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.63e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
Random Forest Feature Importance: [0.49234558 0.22531584 0.20984466 0.07249392]
```

Interpretation:

1. **Model Fit**: The OLS regression (R^2 = 0.558) explains 55.8% of the variance in house prices. The polynomial model improves this slightly to 59.4%, indicating a better fit.

2. **Key Predictors**: Median income has a strong positive impact across all models, with house prices increasing by about 35,000 USD for each unit increase in income. Longitude and latitude have significant negative effects, reflecting lower prices inland.

3. **Housing Age**: Housing median age has a small positive effect on house prices, though it plays a lesser role compared to income and location.

4. **Multicollinearity**: High condition numbers suggest possible multicollinearity, meaning the predictors (especially location factors) are highly correlated, affecting the stability of the models.

5. **Random Forest Feature Importance**: The Random Forest model ranks median income as the most important feature (49.2%), followed by location factors (longitude and latitude), showing the dominance of income and location in predicting house prices.

10. Conclusion:

The project focused on an in-depth statistical analysis and predictive models using a range of regression techniques-linear, ridge, lasso, polynomial, and random forest regressions-on California house pricing to arrive at an estimation of median housing prices. In due course, during this analysis, several insights and some key findings on the predictive capability of different models that have emerged included the role of median_income as a key feature in determining house prices.

Model Performance:

- **Model Complexity**: As the complexity of the models increased—from univariate to multivariate and non-linear methods—there was a noticeable improvement in prediction accuracy. The inclusion of additional features (e.g., income, housing age, location) significantly reduced errors and improved the R^2 score.
- **Best Performing Models**: Random Forest and Polynomial Regression stood out with the highest accuracy, demonstrated by their low MSE and high R^2 values. These models better captured the complex interactions between the predictors, making them ideal for housing price predictions.
- **Feature Importance**: Median income consistently proved to be the most important feature in all models, suggesting its strong influence on housing prices. Adding geographical features such as longitude and latitude improved the models, especially when combined with housing characteristics like median age.
- **Trade-offs**: While simpler models, such as Univariate and Bivariate Linear Regression, are easier to interpret, they underperformed compared to more advanced models. More sophisticated techniques, like Random Forest, provided much better predictions at the cost of interpretability.

Findings:

- **Income as a Strong Predictor**: Median income consistently emerged as a key factor in housing price prediction, reinforcing the notion that income strongly influences housing affordability.
- **Geographical and Demographic Features**: While longitude and latitude alone were weak predictors, combining them with median income, housing age, and other features led to significant improvements. Location matters but is more effective when combined with other factors.
- **Increasing Model Complexity Improves Performance**: As models became more complex—moving from univariate to multivariate and non-linear approaches—performance improved. Polynomial Regression and Random Forest Regression, which handle non-linearities, yielded significantly better results than simple linear models.
- **Random Forest as the Best Performing Model**: Random Forest Regression outperformed all other models, providing the most accurate predictions with the lowest MSE and highest R^2 (0.746). Its ability to capture complex relationships and interactions between variables made it the most suitable model for predicting housing prices.

Recommendations:

1. **Use Non-Linear Models**: Non-linear models, such as Random Forest, should be preferred for predicting housing prices due to their superior ability to capture complex relationships.
2. **Feature Expansion**: Including a diverse set of predictors, such as geographical, economic, and housing characteristics, improves prediction accuracy. Incorporating additional socio-economic factors could further refine model performance.
3. **Further Model Exploration**: To enhance prediction accuracy further, advanced techniques like boosting algorithms or additional machine learning approaches could be explored.
4. **Addressing Multicollinearity**: Review high variance inflation factors (VIF) among predictor variables. Removing or combining highly correlated features could reduce multicollinearity, improving model stability.
5. **Improving Data Quality**: If possible, enrich the dataset with more recent or region-specific data. Ensure that all missing or incorrect values are addressed in the cleaning process to avoid introducing bias.

In conclusion, complex models like Random Forest Regression significantly outperform simpler ones, providing better insights into the variability of California housing prices.

11. References:

- Cam Nugent. (2023). California Housing Prices [Data set]. Kaggle. https://www.kaggle.com/datasets/camnugent/california-housing-prices/data
- Torgo, L. (1997). California Housing Prices. Retrieved from https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html