

MANAGEMENT SCIENCE FINAL REPORT

110306011 盧柏瑜
110306002 許馨文
110306047 吳堃豪
110306068 張奕奇
110306017 陳奕利



REPORT OUTLINE

01

BACKGROUND

Basic introduction
about our report

02

PREPROCESS DATASET

Preprocess the dataset
to make it easier to
analyze

03

MODELS BUILDING

Building
linear regression model
and
lasso regression model

04

MODELS EVALUATION

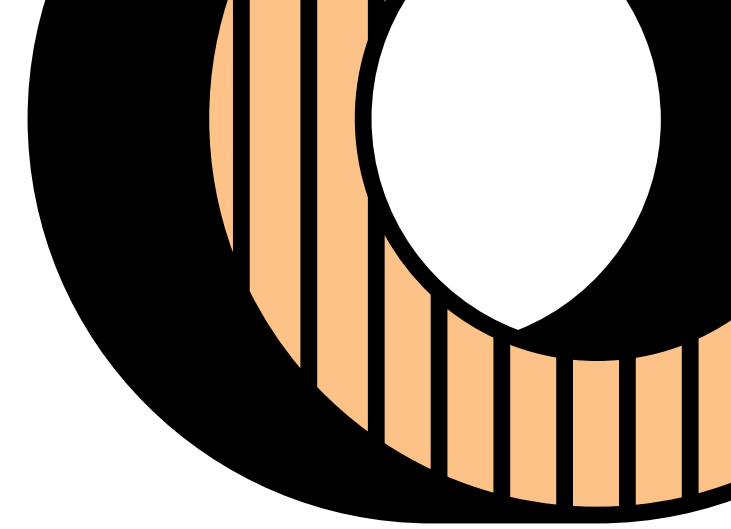
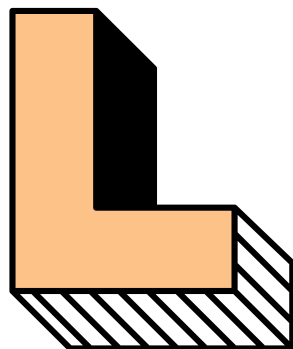
Evaluate all three
regression models with
MAE, MAPE and MSE

05

CONCLUSION

06

VIDEO LINK





BACKGROUND

CHOSEN SUBJECT:

We chose the dataset “California Housing Prices Data” from Kaggle. The data pertains to the houses found in a given California district and some summary about them based on the 1990 census data.

“

The dataset contains information about various housing attributes and their respective values. By analyzing this dataset, we can explore relationships between the median house value and factors such as income, age, size of the property, population, and proximity to the coast.





BACKGROUND

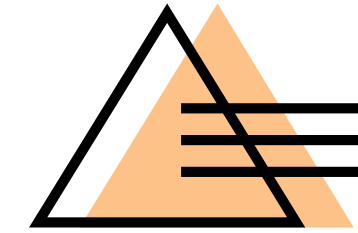
MOTIVATION:

1. The dataset on California housing prices provides valuable insights into the housing market in a specific region.
2. Understanding housing trends is crucial for individuals and organizations involved in the real estate industry.
3. Market analysis, based on the dataset, provides insights into the dynamics of the housing market in California.
4. The dataset also offers insights relevant to urban planning.





PREPROCESS DATASET- INITIAL DATASET



1.INITIAL DATASET:

Number of initial rows: 20640

Number of initial columns: 10

Top five rows of the dataset display right

```
df.head()
```

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

2.DESRIPTIVE STATISTICS OF A DATAFRAME:

Such as count, mean, standard deviation, minimum value, quartiles, and maximum value.

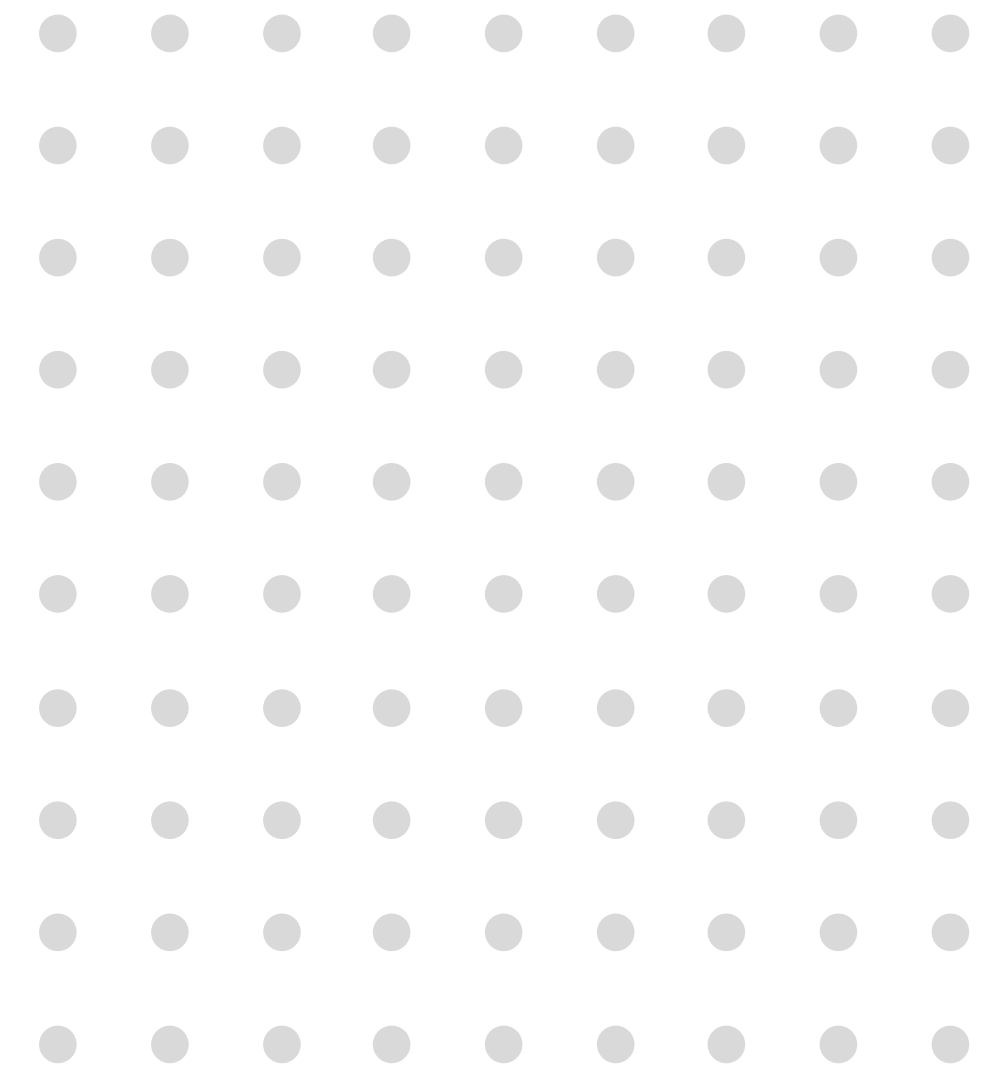
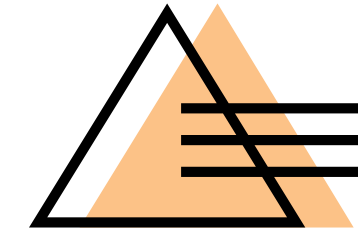
ocean_proximity does not have a numeric value, so it is not incorporated.

```
df.describe()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671	206855.816909
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822	115395.615874
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400	119600.000000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800	179700.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250	264725.000000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100	500001.000000



PREPROCESS DATASET- INITIAL DATASET



3.CORRELATIONS:

It calculates the correlation coefficient between every pair of numeric columns in the DataFrame.

The correlation coefficient is a statistical measure that quantifies the strength and direction of the linear relationship between two variables.

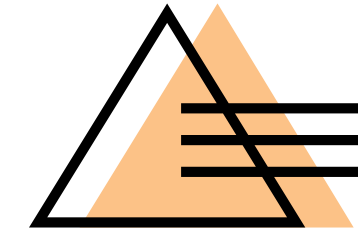
```
df.corr()
```

<ipython-input-659-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will be removed. Use df.corr(numeric_only=True) to silence this warning.

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
longitude	1.000000	-0.924664	-0.108197	0.044568	0.069608	0.099773	0.055310	-0.015176	-0.045967
latitude	-0.924664	1.000000	0.011173	-0.036100	-0.066983	-0.108785	-0.071035	-0.079809	-0.144160
housing_median_age	-0.108197	0.011173	1.000000	-0.361262	-0.320451	-0.296244	-0.302916	-0.119034	0.105623
total_rooms	0.044568	-0.036100	-0.361262	1.000000	0.930380	0.857126	0.918484	0.198050	0.134153
total_bedrooms	0.069608	-0.066983	-0.320451	0.930380	1.000000	0.877747	0.979728	-0.007723	0.049686
population	0.099773	-0.108785	-0.296244	0.857126	0.877747	1.000000	0.907222	0.004834	-0.024650
households	0.055310	-0.071035	-0.302916	0.918484	0.979728	0.907222	1.000000	0.013033	0.065843
median_income	-0.015176	-0.079809	-0.119034	0.198050	-0.007723	0.004834	0.013033	1.000000	0.688075
median_house_value	-0.045967	-0.144160	0.105623	0.134153	0.049686	-0.024650	0.065843	0.688075	1.000000



PREPROCESS DATASET- PROCESS DATASET METHOD1



1.FIND MISSING VALUES

We found out that the column "total_bedrooms" has 207 missing values in the DataFrame.

All other columns, including "longitude", "latitude", "housing_median_age", "total_rooms", "population", "households", "median_income", "median_house_value", and "ocean_proximity" have no missing values (count of missing values is 0).

2.REMOVE MISSING DATA

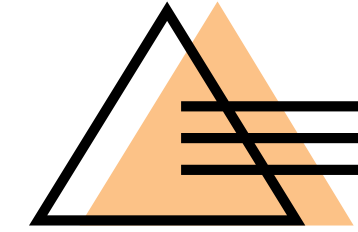
Any row containing at least one missing value will be dropped.

```
# total bedroom缺值
df.isnull().sum()

longitude      0
latitude       0
housing_median_age  0
total_rooms    0
total_bedrooms 207
population     0
households     0
median_income  0
median_house_value  0
ocean_proximity 0
dtype: int64
```




PREPROCESS DATASET- PROCESS DATASET METHOD1

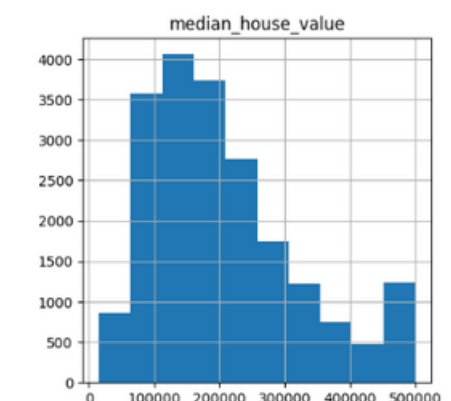
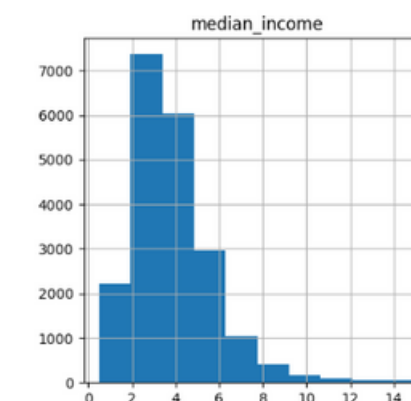
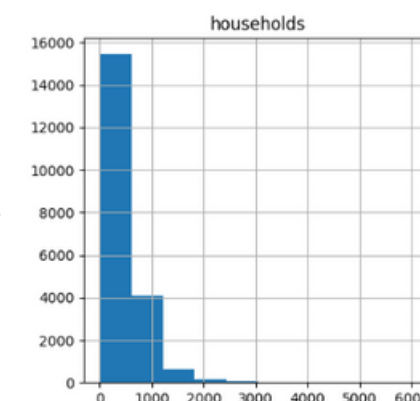
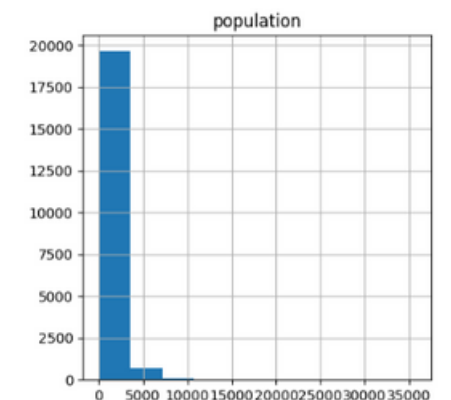
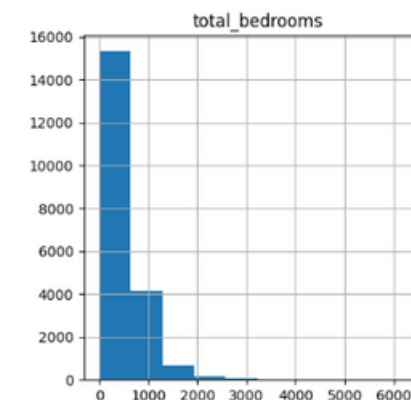
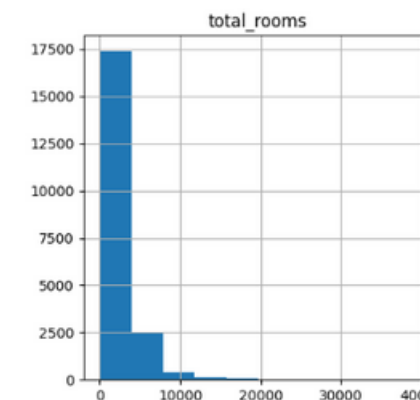
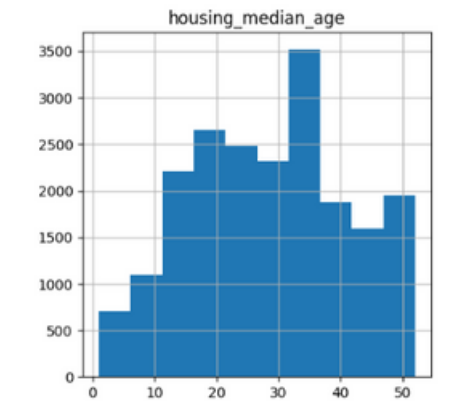
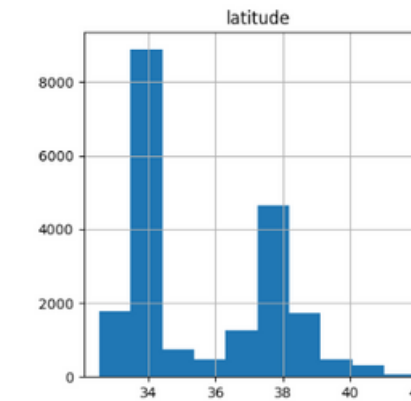
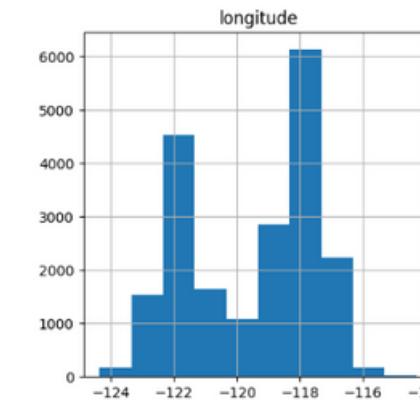


3.DRAW HISTOGRAM

Create a histogram for each numeric column in the DataFrame.

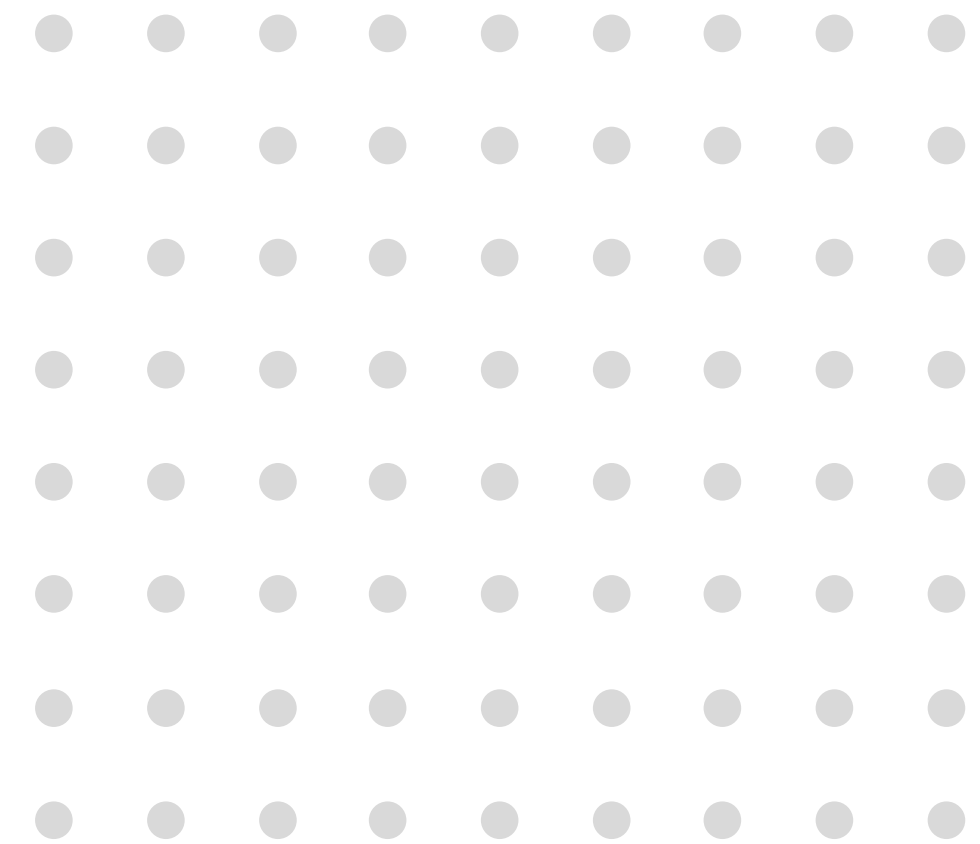
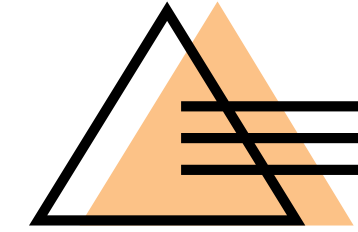
It visualizes the distribution of values in each column using a set of bins. `figsize=(16, 16)` is an optional parameter that specifies the size of the figure in inches.

It determines the width and height of the resulting histogram plot. By setting `figsize=(16, 16)`, the resulting plot will have a larger size, making it easier to view and analyze the histograms





PREPROCESS DATASET- PROCESS DATASET METHOD2



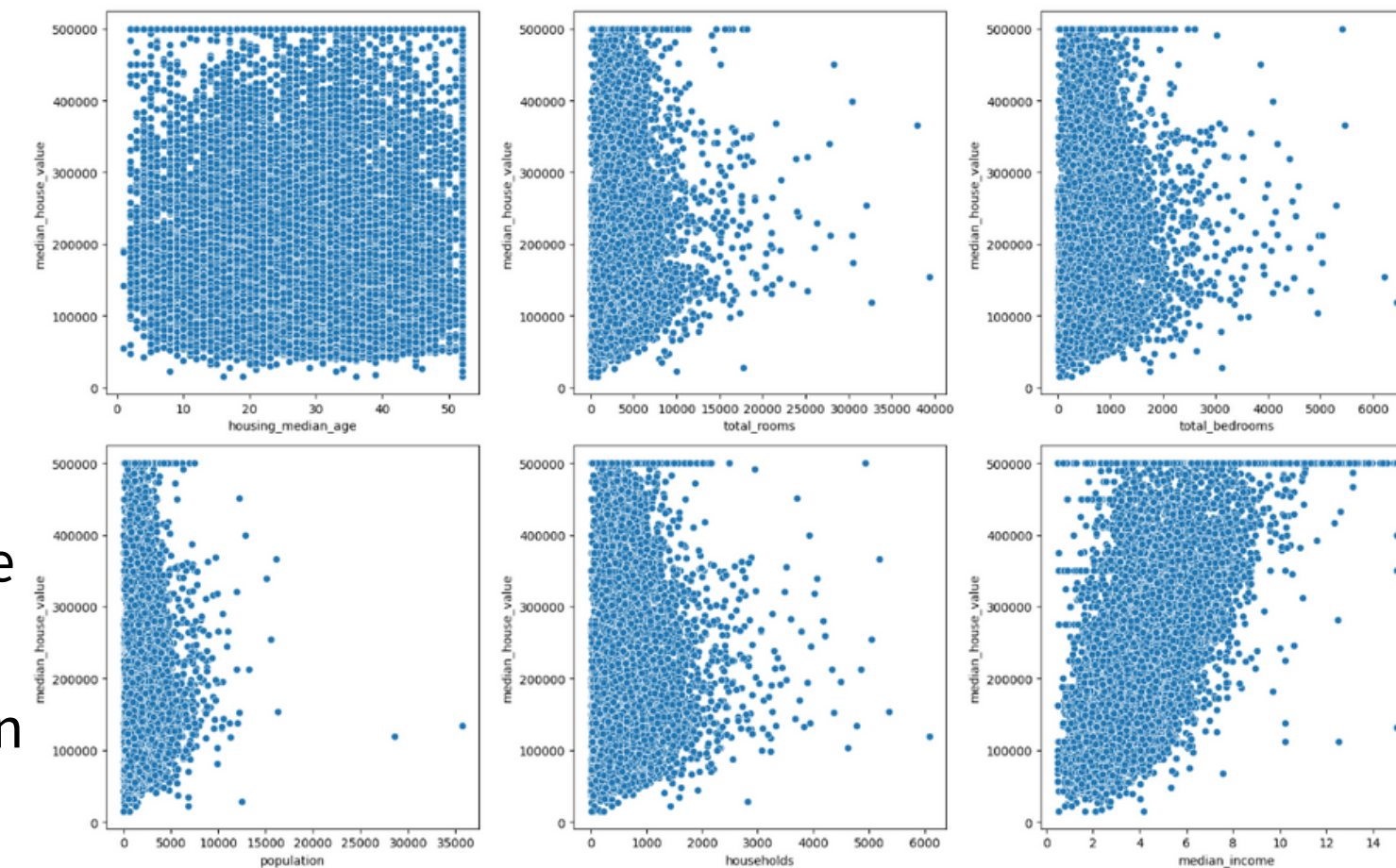
1.FIND OUTLIERS

Creating a scatter plot for each selected variable against the "median_house_value" column in the DataFrame. This allows us to visually identify potential outliers or extreme values.

2.REMOVE OUTLIERS

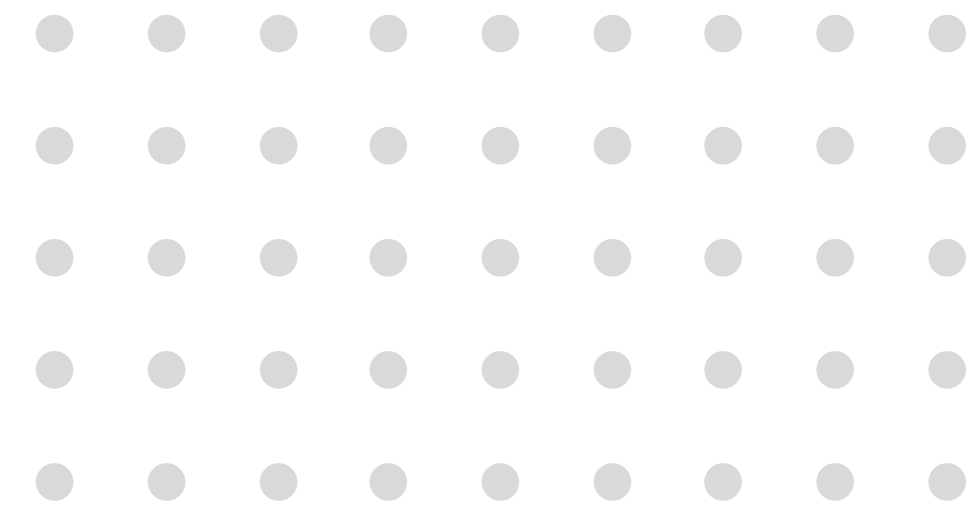
Remove rows where the values in the "total_rooms" column are less than or equal to 20000, the values in the "total_bedrooms" column are less than or equal to 3800, the values in the "population" column are less than or equal to 12000, and the values in the "households" column are less than or equal to 3000.

After removing, we have a remaining of 20379 columns.





PREPROCESS DATASET- PROCESS DATASET METHOD3&4



PROCESS DATASET METHOD3 – ADD NEW COLUMN "BEDROOM_PERCENTAGE"

This column represents the percentage of bedrooms in relation to the total number of rooms in each row. After executing the code, the DataFrame df will have a new column called "bedroom_percentage" that contains the calculated bedroom percentages for each row. This column can be used for further analysis, visualization, or modeling tasks.

Note that this calculation assumes that both the "total_bedrooms" and "total_rooms" columns contain valid numerical values and have been appropriately handled for missing or erroneous data.

PROCESS DATASET METHOD4 – ENCODING CATEGORICAL VARIABLES INTO NUMERIC LABELS

By using LabelEncoder, the code converts the categorical values in the "ocean_proximity" column into numeric labels. Each unique category in the column is assigned a different integer label. After executing the code, the "ocean_proximity" column in df will be replaced with the encoded numeric labels obtained from the LabelEncoder transformation.



LINEAR

REGRESSION MODEL



MODELS BUILDING -LINEAR REGRESSION MODEL



1.SETTING X AND Y:

X will contain the independent variables, excluding 'median_house_value' and 'total_bedrooms'. Y will contain the dependent variable, which is 'median_house_value'.

2.SPLITS THE DATA

Splits data into training and testing sets for both the independent variables (x) and the dependent variable (y) using the train_test_split function from scikit-learn. 30% of the data will be allocated to the testing set, and the remaining 70% will be used for training. Random_state=42 sets the random seed to ensure reproducibility.

3.CALCULATE THE COEFFICIENT OF DETERMINATION

Score: 0.6654808607029031. A higher R-squared score indicates that the linear regression model provides a better fit to the test data, with a higher proportion of the variance explained by the independent variables.





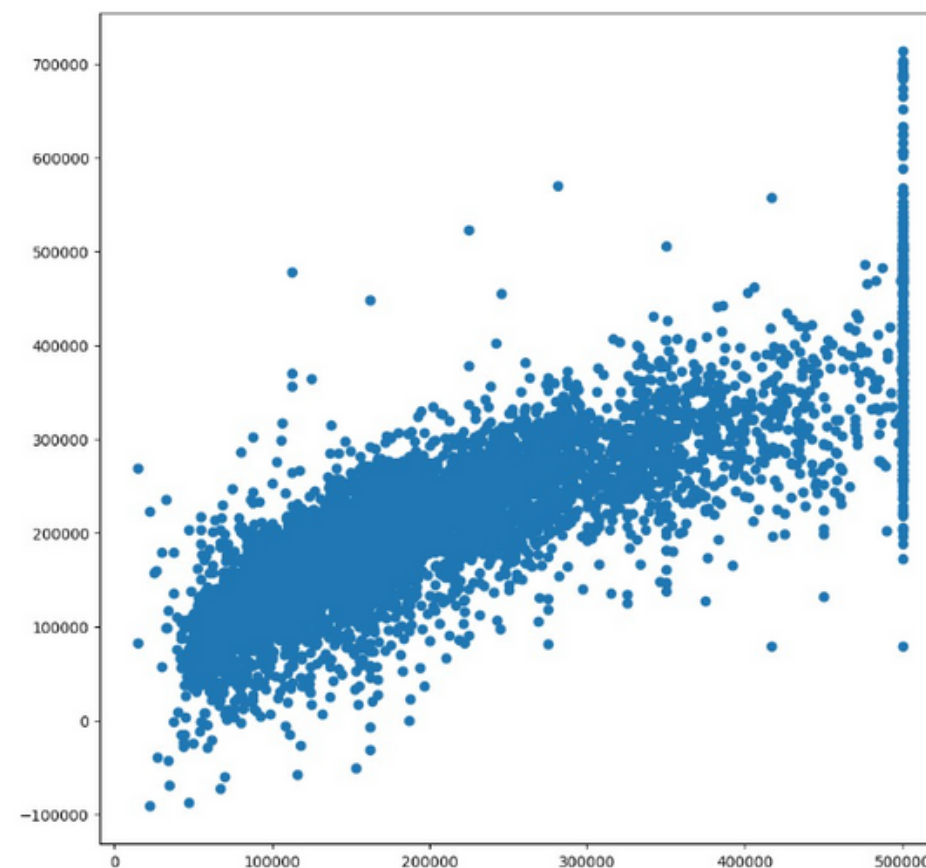
MODELS BUILDING

-L0 REGRESSION MODEL

1.FIND BEST INTERCEPT AND COEFFICIENT ESTIMATES

Applying L0_regression to the x_train and y_train datasets with a specified random seed of 10101. The function in the code L0_regression() will search for the best number of non-zero coefficients

2.DRAW THE PLOT



Creates a scatter plot to visualize the relationship between the actual house prices and the predicted house prices using the L0 regression model. The plot helps in evaluating the performance of the model by comparing the predicted values to the actual values.

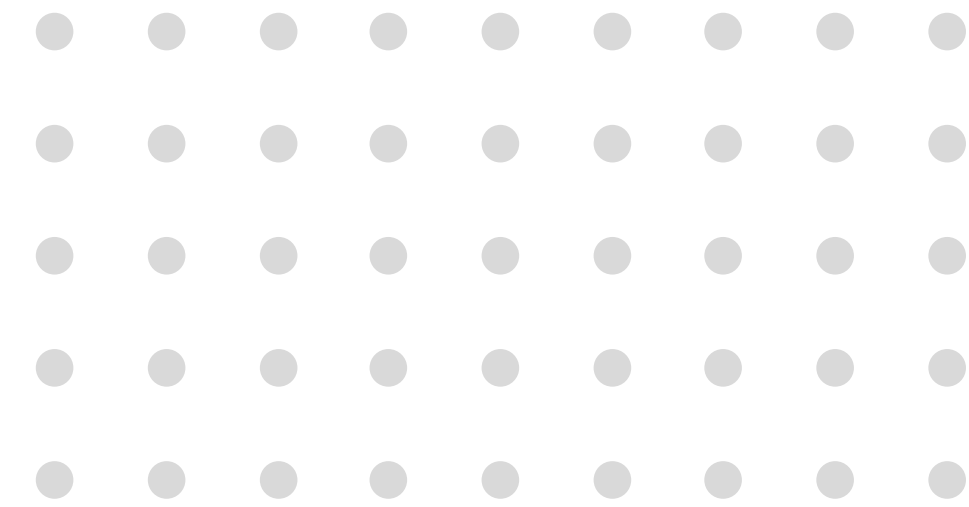


If the points in the scatter plot lie close to a diagonal line ($y = x$), it indicates that the model's predictions are in good agreement with the actual values. On the other hand, if the points are scattered far from the diagonal line, it suggests that the model's predictions deviate from the actual values.



MODELS BUILDING

-LASSO (L1) REGRESSION MODE



1.FIND ALPHA VALUE

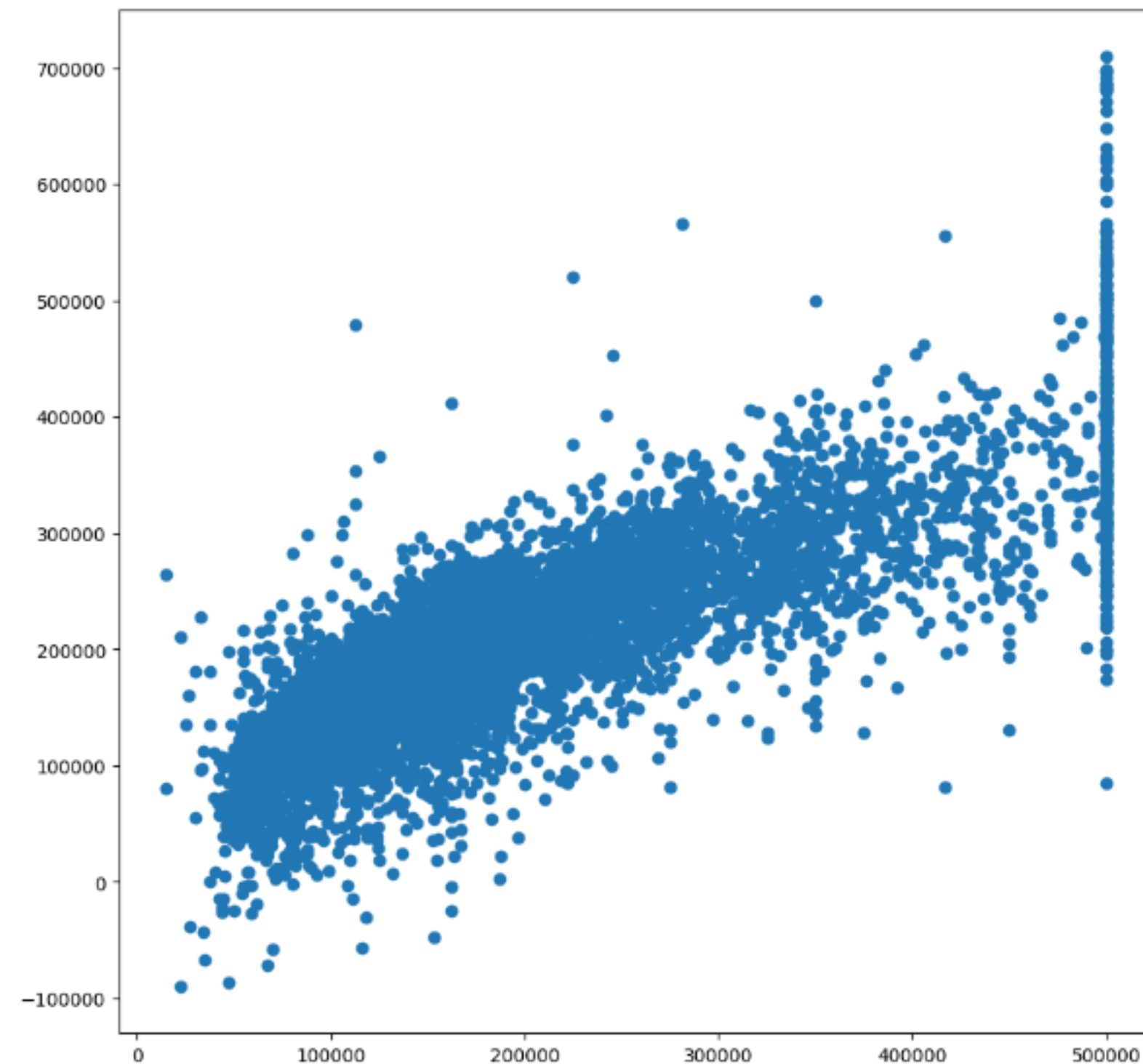
Obtain the optimal value of the regularization parameter (alpha) selected by the LassoCV model using cross-validation. We use 5-fold cross-validation. This value is crucial for controlling the amount of regularization applied in the subsequent Lasso regression model.

Random_state=0 sets the random seed to ensure reproducibility of the cross-validation process. Max_iter=10000 sets the maximum number of iterations for the Lasso algorithm to converge. This parameter determines the maximum number of iterations the algorithm will perform if it does not converge earlier.

2.CALCULATE THE COEFFICIENT OF DETERMINATION (R-SQUARED) :

After fitting the data, we calculate the R-squared score. We get => Score: 0.6656716894638972.





MODELS BUILDING

-LASSO (L1) REGRESSION MODE

DRAW THE PLOT

Creates a scatter plot to visualize the relationship between the actual house prices (y_{test}) and the predicted house prices (lasso_predict) using the Lasso regression model. The plot helps in evaluating the performance of the model by comparing the predicted values to the actual values.

If the points in the scatter plot lie close to a diagonal line ($y = x$), it indicates that the model's predictions are in good agreement with the actual values. On the other hand, if the points are scattered far from the diagonal line, it suggests that the model's predictions deviate from the actual values.



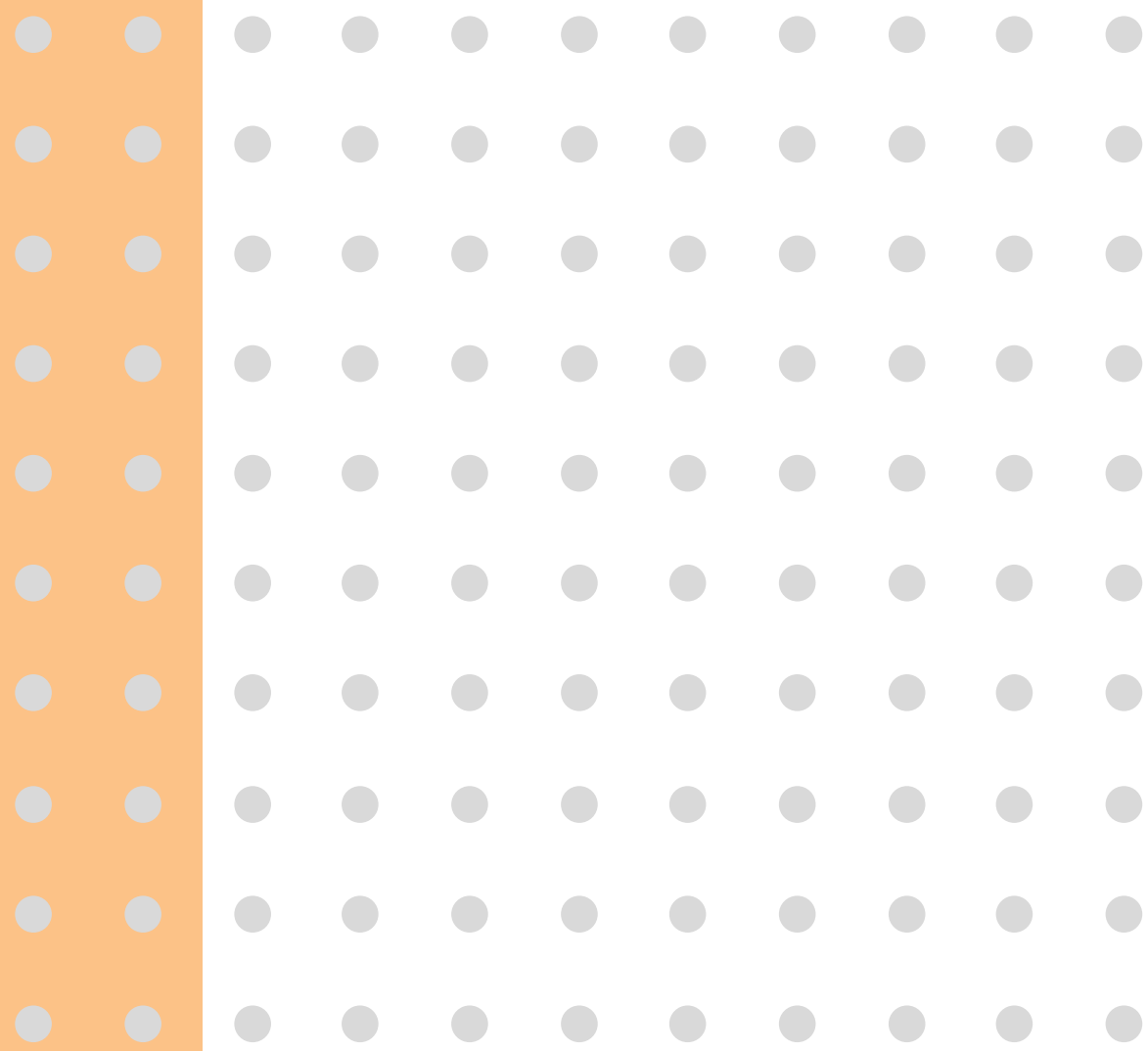
MODELS EVALUATION



	MAE	MAPE	MSE
Linear Regression	49435.207873	0.306684	4.440456e+09
Lasso Regression	49493.530515	0.306709	4.437923e+09
L0 Regression	49433.174008	0.306685	4.440404e+09

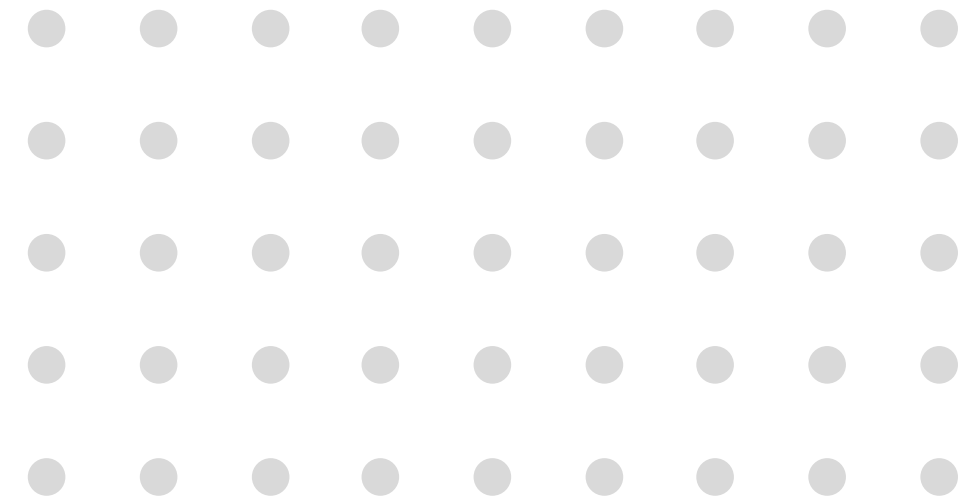
From the evaluation metrics, we can observe that all three regression models have similar performance in terms of MAE, MAPE, and MSE. The differences between the models in these metrics are minimal.

Therefore, based on these results, we can conclude that the three regression models, namely Linear Regression, Lasso Regression, and L0 Regression, perform similarly in predicting the median house values. It is recommended to consider other factors such as model complexity, interpretability, and computational efficiency when choosing the best model for the specific application.





CONCLUSION



1

All three models, namely Linear Regression, Lasso Regression, and L0 Regression, provide reasonably similar performance in predicting house prices.

5

The Lasso Regression and L0 Regression models, which incorporate regularization techniques, do not significantly outperform the Linear Regression model in this scenario.

2

The mean absolute error (MAE) for all models is around 49,000 USD, indicating an average absolute difference between the predicted and actual house prices.

6

The evaluation metrics suggest that the chosen predictors in the models may not capture the full complexity and variability of the California housing prices.

3

The mean absolute percentage error (MAPE) for all models is approximately 0.3067. This indicates that, on average, the predictions deviate from the actual values by around 30.67%.

7

Additional factors, such as geographic location, specific neighborhood characteristics, housing market trends, and other socioeconomic variables, may need to be considered to improve the accuracy of the models.

4

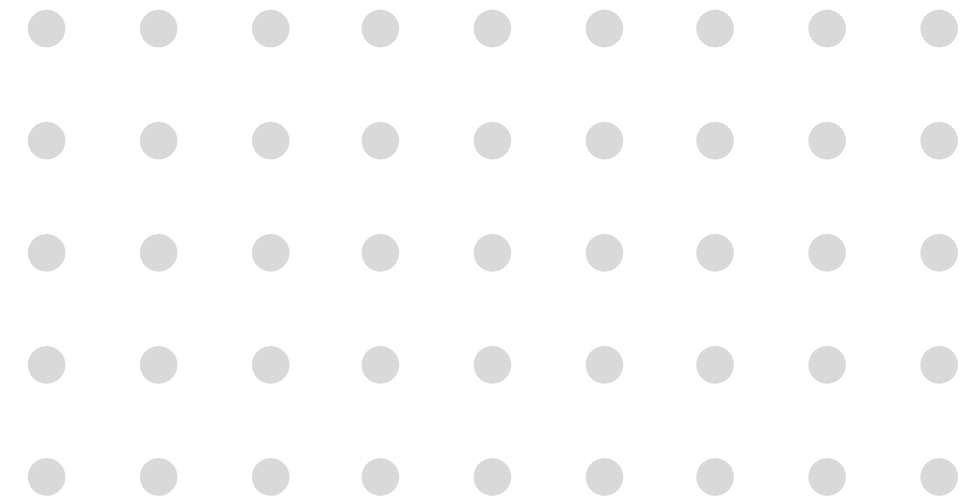
The mean squared error (MSE) for all models is on the order of 10^9 . This metric gives higher weight to larger errors.

8

It is recommended to further explore and refine the feature selection process, consider alternative regression techniques, and potentially incorporate domain knowledge or additional data sources to enhance the predictive performance for the California house price dataset.

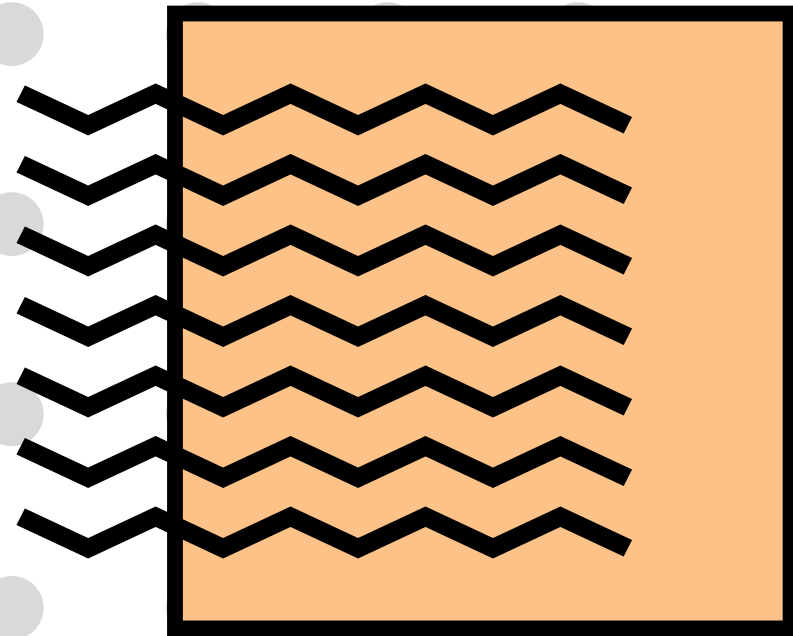


VIDEO LINK



<https://youtu.be/GZ2I1ZjNuNA>

-



**THANKS FOR
WATCHING**

