# Linear Regression

# Overview

This project analyzes California housing prices from the 1990 census. The data was downloaded from [www.kaggle.com - California Housing Prices](https://www.kaggle.com/camnugent/california-housing-prices). The dataset was downloaded from kaggle and placed onto a public folder in DropBox. The data can be downloaded from [here](https://www.dropbox.com/s/nj5cj6mkmmkdrjv/housing.csv?dl=1).

The goal was to perform a linear regression analysis on this dataset to identify correlations that can be used to help predict house prices. Our approach for this assignment includes three main phases: data cleaning, identify correlations and trends in the data, and linear regression analysis. See below for details for each phase.

## Data Cleaning

The dataset was first examined to identify and clean up any questionable data.

### Fill in missing values in **total\_bedrooms** column

When examining the data set to see if there is any data missing, we identified there were 207 missing values in the total\_bedrooms column. We filled the missing values with the mean value of this column.

### Convert **ocean\_proximity** from object type to Boolean type

We were not able to train the linear regression model with the **ocean\_proximity** because it is an object type with descrete text values. The two options were:

1. Convert the **ocean\_proximity** values to a numeric field. Example code:

from sklearn.preprocessing import LabelEncoder

cali\_dataset['ocean\_proximity']=LabelEncoder().fit\_transform(cali\_dataset['ocean\_proximity'])

This produces the r-squared score equal 0.635 when model was trained with all features.

1. Create individual Boolean (1/0) columns for each value. Example code:

from sklearn.preprocessing import LabelBinarizer

lb = LabelBinarizer()

cali\_dataset = cali\_dataset.join(pd.DataFrame(lb.fit\_transform(cali\_dataset["ocean\_proximity"]),

                          columns=lb.classes\_,

                          index=cali\_dataset.index))

cali\_dataset = cali\_dataset.drop(columns=['ocean\_proximity'])

The r-squared score for this option is 0.645 when model was trained with all features.

Option b provides a higher r-squared score, we decided to go with option b.

### Filter unwanted outliners

We generated boxplots to look for outliners in specific features. Several columns were added to the dataset and the columns **bedrooms\_per\_household**, **rooms\_per\_household**, and **population\_per\_household** have been identified as having unusual maximum values. For instance, some records indicated an average number of rooms per household as being > 140 which appears to be erroneous.

Records with unwanted outliers were removed, specifically, records that met these criteria were removed:

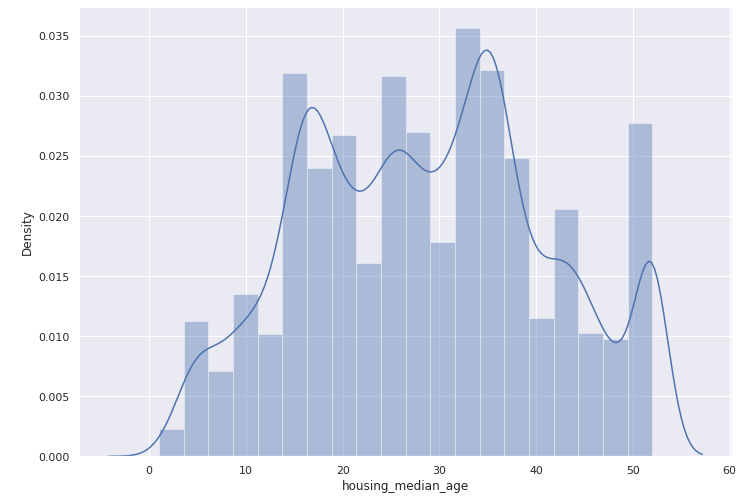
* bedrooms\_per\_household >= 15
* rooms\_per\_household >= 15
* population\_per\_household >= 20

The boxplots used to identify these outliers are shown below:

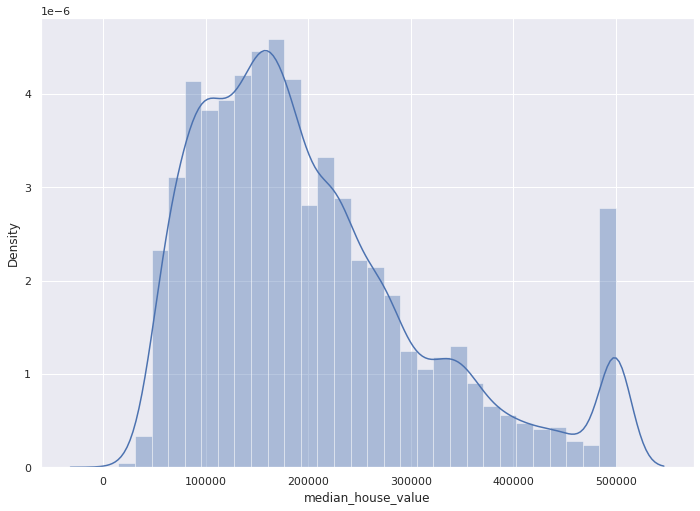
 



House ages appear clipped at 52 years.  There are significantly more houses in the 52-year bucket. Houses with an age of 52 were removed from the dataset.



Home prices appear to be clipped at $500,001. We originally looked at removing these outliers from the dataset but after running the analysis, leaving in the clipped home prices improved the accuracy of the model.



Homes with a proximity of 'island' may be outliers so look at the range of house values for homes on islands. There are only 5 records on 'islands' and their values are much higher than average. We decided to remove rows for homes on 'islands' because there may be other factors, not present in the data, that affect the prices of these homes. For instance, just being on an island may significantly increase the home price regardless of other features.

## Identify correlations and trends

After cleaning the dataset the next step is to look for specific trends and correlations to determine whether it is feasible to predict home prices using the given dataset and, if it’s possible, which features may provide the strongest correlation.

### Create new features

The data contained features that indicate the total number of rooms, number of bedrooms, and number of occupants in an area but that information may not be helpful in determining individual house prices. Rooms per household, bedrooms per household, and population per household might be more helpful in predicting individual prices.

The following features were added to the dataset:

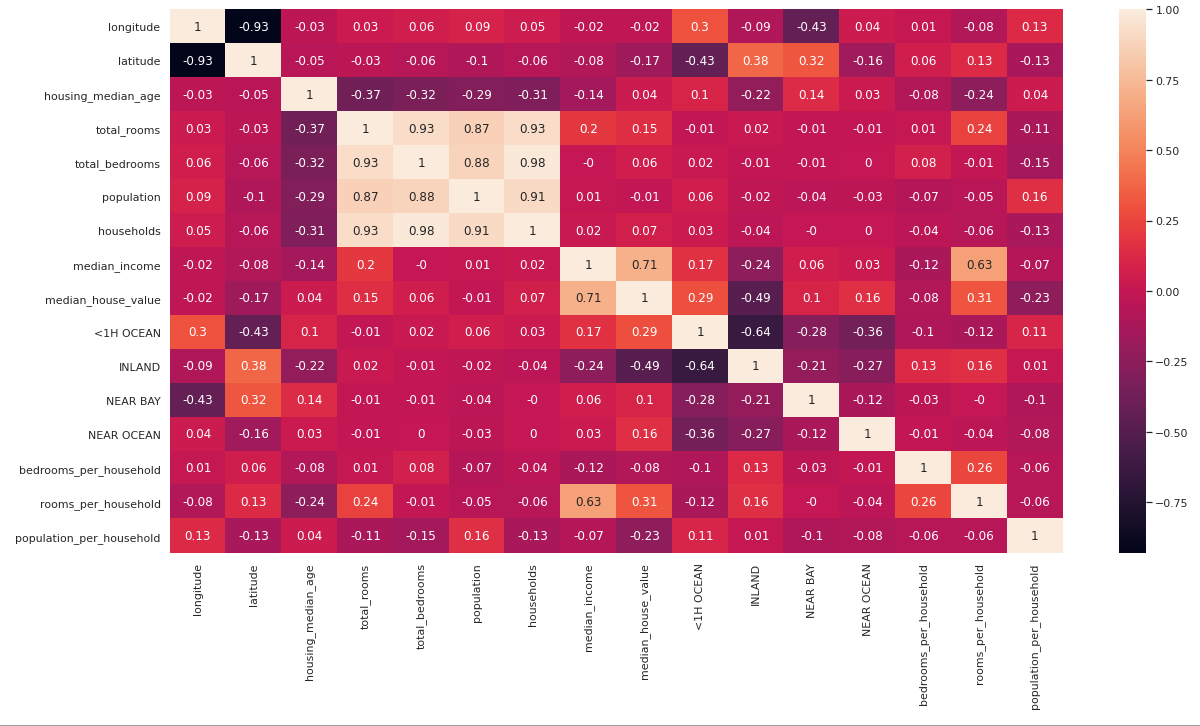
* **bedrooms\_per\_household** = total\_bedrooms / households
* **rooms\_per\_household** = total\_rooms / households
* **population\_per\_household** = population / households

### Correlate variables

After cleaning the data set, we generated a heat map for all the correlations between all features to determine which independent variables correlate most closely with median\_house\_value.

The strongest correlations identified are:

* **median\_income** has the strongest correlation to **median\_house\_value** (0.71)
* There is a negative correlation (0.49) between houses that are ‘inland’ vs those closest to the ocean.
* **rooms\_per\_household** has the next strongest correlation at 0.31
* The calculated values have a stronger correlation to **median\_house\_value** than their counterparts. For instance, **total\_rooms** has a 0.15 correlation while **rooms\_per\_household** has a 0.31.



## Linear Regression Analysis

Multiple linear regression models were built using different combinations of independent variables in order to see how the different groupings affected the r2 score.

In all cases, the input data set was separated into training and test data sets. 25% of the data was reserved for the test data set.

### First run - Model strongly correlated variables

Create linear regression model using only median\_income which has the highest correlation. The three features we used in this first run are longitude, latitude, and median\_income. The results from this run were:

**Training Data**

**RMSE** is 70894.42

**R2** score is 0.6003

**Absolute mean error** is 52287.02

**Coef** is [-46383.8245764 -46700.60423751 37321.69837484]

**Test Data**

**RMSE** is 70268.96

**R2** score is 0.6112

**Absolute mean error** is 51441.56

### Second run - Include moderately correlated variables

In this model, in addition to the three features in the first run, we also includes the INLAND, population\_per\_household features. The results from this run are.

These features were added to the model

* + INLAND
  + population\_per\_household
  + rooms\_per\_household

**The model performance for *training* set**

**RMSE** is 66237.28

**R2** score is 0.6511

**Absolute mean error** is 47882.62

**Coef** is [-27807.99414753 -28687.80079199 36717.85396794 -40232.39600969 -25791.61799398 -1672.35498863]

**The model performance for *testing* set**

**RMSE** is 65634.72

**R2** score is 0.6608

**Absolute mean error** is 47228.29

### Final run - Include all features

The final run included all features available in the input dataset. The results of this model were:

**The model performance for *training* set**

**RMSE** is 64974.50

**R2** score is 0.6643

**Absolute mean error** is 46876.05

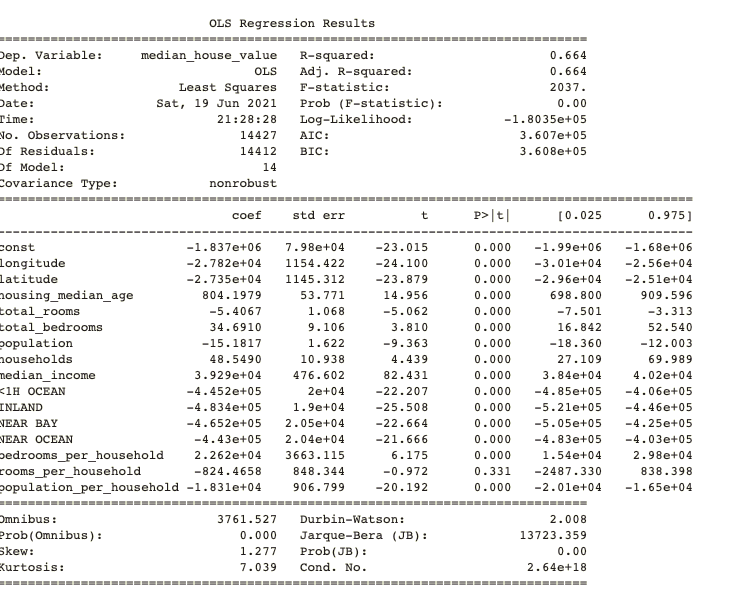
**Coef** is [-2.78211658e+04 -2.73493876e+04 8.04197872e+02 -5.40672110e+00 3.46909734e+01 -1.51817032e+01 4.85489820e+01 3.92867695e+04 1.39934874e+04 -2.42028116e+04 -5.99216336e+03 1.62014876e+04 2.26188250e+04 -8.24465760e+02 -1.83099849e+04]

**The model performance for *testing* set**

**RMSE** is 64675.45

**R2** score is 0.6707

**Absolute mean error** is 46752.18



# Conclusion

The goal was to use linear regression analysis on a dataset containing house prices in California. Data cleaning is an important step before performing analysis on a data set. It helps ensuring data are consistent and usable. Linear regression model seems simple to implement. The more features to train the linear regression model, the better the score becomes. By cleaning the data, calculating additional features, and adjusting the training/test split we were able to achieve an r2-score of **0.6643** on the training data and **0.6707** on the test data.

A higher score value would be preferrable, but a significantly higher score does not seem achievable using linear regression alone. Other models such as **RandomForest** might be able to produce better scores, but they are not purely linear regression based.

## Key points

* The data set had a strong relationship between **median\_income** and **median\_house\_value**.
* There was missing and potentially erroneous data that needed to be cleaned up before analyzing the dataset.
* Identifying other relationships within the data and creating additional features strengthened the final score. The dataset included **total\_rooms, total\_bedrooms, and total\_population.** Calculating **per\_household** versions, **rooms\_per\_household**, provided two benefits:
  + The new features had a stronger correlation to house prices and improved the r2 score.
  + The new features helped identify outliers and potentially erroneous data.
* Including all features in the dataset resulted in a higher score, even if some of the features did not have a strong relationship to house prices.
* Adjusting the parameters of the **sklearn.linear\_model.LinearRegression**() object (values such as **fit\_intercept=**) had little impact on the results.
* Scores between training and test data were similar in all cases.
* A ratio of 75% training/25% test data seemed to be optimal.
* Other **sklearn** linear models where used, including **Ridge**, **ElasticNet**, **Lars**, and **BayseianRidge**, but results from these models were not different from the standard **LinearRegression** model for this dataset.