**California Housing Price Prediction**

Lakkum Sai Aditya  
 2021NOVVUGP0054

**1. Introduction**

1.1 **Background**

Develop a model that forecasts district housing prices in California based on a variety of variables. District-level details are included in the dataset. Developing a machine learning model to forecast for unknown districts and generalize insights is a challenge. intends to provide a useful resource for comprehending and navigating the ever-changing California real estate market.

1.2 **Problem**

The objective is to create a predictive model based on a dataset containing multiple metrics for distinct districts or block groups that will forecast median house values in California. The objective is to develop a machine learning model that, while taking other pertinent factors into account, can efficiently learn from the data provided and predict the median housing price for any given district.

1.3 **Interest**

Building a housing price prediction model for California presents an interesting challenge as it examines district-level information impacted by location and demographics. A machine learning tool that not only gains knowledge from data but also generalizes findings to make precise predictions is the aim. The goal of this project is to give decision-makers a clear grasp of the complex California housing market.

**2. Data Acquisition and Cleaning**

2.1 **Data Acquisition**

They collected information on the variables using all the block groups in California from the 1990 Census. In this sample a block group on average includes 1425.5 individuals living in a geographically compact area. Naturally, the geographical area included varies inversely with the population density. They computed distances among the centroids of each block group as measured in latitude and longitude. They excluded all the block groups reporting zero entries for the independent and dependent variables. The final data contained 20,640 observations on 10 characteristics.

2.2 **Data Cleaning**

There were null values on the attribute **total\_bedrooms**. To fill those NA values. We imputed that missing data with the attribute median value.

**Feature Engineering**

I have added 6 more extra attributes into the dataset which are:

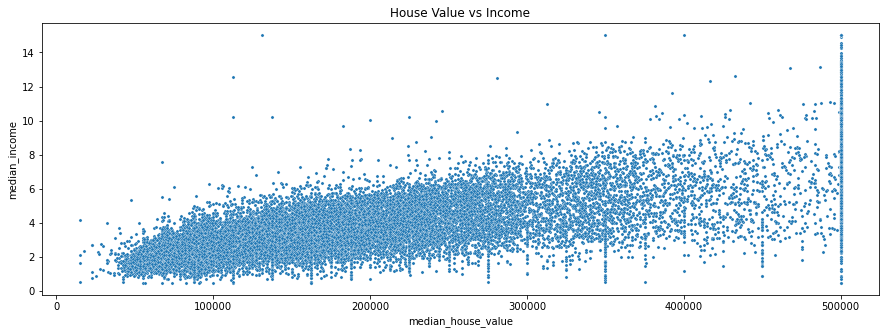
1. **rooms\_bedroom\_ratio**: this gives ratio between total\_rooms and total\_bedrooms.
2. **population\_per\_household**: this gives ration between population and households.
3. **avg\_rooms\_per\_household**: this gives ration between total\_rooms and households.
4. **income\_per\_capita**: this gives ration between median\_income and population.
5. **median\_age\_income\_ratio**: this gives ration between housing\_median\_age and median\_income.
6. **bedrooms\_per\_household**: this gives ration between total\_bedrooms and households

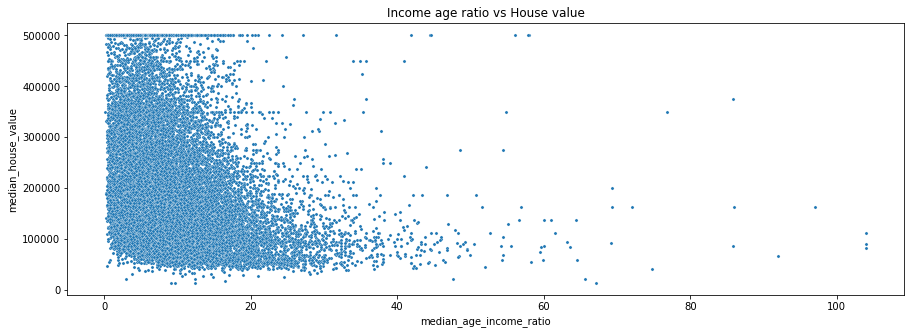
**3. Methodology**

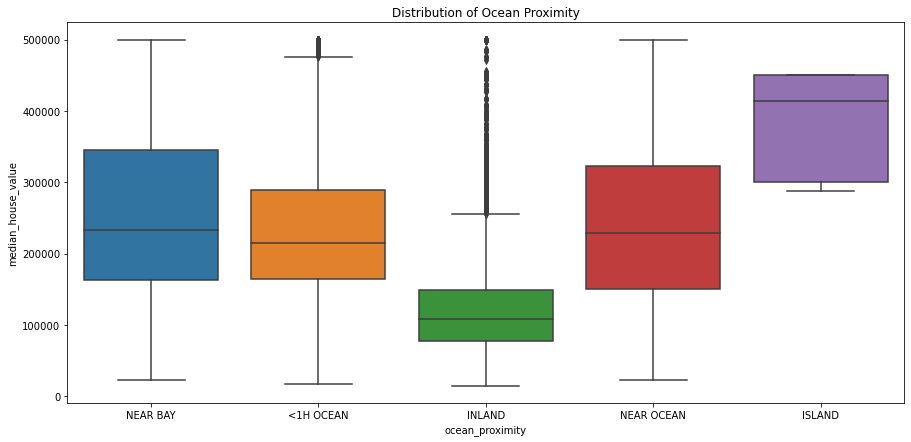
3.1 **Exploratory Data Analysis**

These are plots which the relation between

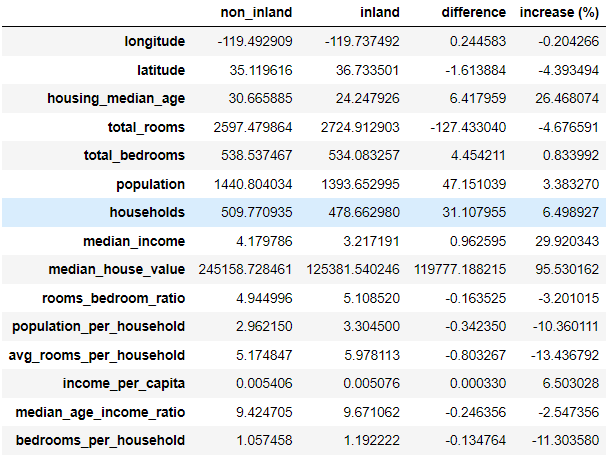
1. Plot: this plot returns us a scatter plot between House Value and Income.



1. Plot: this plot returns us a scatter plot between Income Age ratio and House Value.
2. Plot: this returns a box distribution of the catergorical attribute ocean\_proximity



The INLAND outliers take up almost 32% of the data. Comparing the INLAND outliers statistics against other houses that are near the ocean whose median house value is **greater than the maximum** (41775.0) of INLAND houses.

The below image gives information about the NON\_INLAND households with respect to INLAND households.

Compared to near-ocean houses in the same median house value range, on average, the outlier INLAND houses:

1. are relatively new (by roughly 6 years and 4 months)
2. are 96% cheaper than non-inland houses
3. people who live in them earn 6.5% less than non-inland house owners
4. are seemingly larger than non-inland houses indicated by their greater average bedroom size and average number of rooms per household (3.4% larger bedrooms and 13.4% more rooms respectively).

3.2 **Modelling**

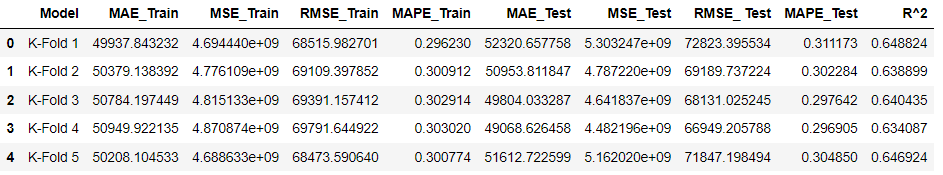
For Model building I have used 5 models of linear regression

1. K-Fold Cross Validation
2. Recursive Feature Elimination
3. Principal Component Analysis
4. Lasso Regression
5. Ridge Regression
6. Elastic Net Regression

**4. Results**

1. **K-Fold Cross Validation**

For this regression, I have used 5 folds. For preparation did ordinal encoding on the categorical variable in the dataset which is ocean\_proximity and transformed the dataset. The score of the model is shown below



1. **Recursive Feature Elimination**

This regression returns the attributes which are suitable for further predicting the model based on the in-built feature ranking in the library RFE. The dataset used for this should include only numerical attribute. We build a model for the on the features selected, the result is shown below



1. **Principal Component Analysis**

Principal Component Analysis (PCA) is used for dimensionality reduction of the datasets. The dataset should consist of all numerical attributes and should be standardized. We have selected 8 components for further model building for detailed code you can check the jupyter notebook. The result for this model shows as



1. **Lasso Regression**

Feature Selection: Lasso regression includes a regularization term that introduces sparsity by driving some regression coefficients to exactly zero. This encourages feature selection, making it useful when dealing with a large number of potential predictors.



1. **Ridge Regression**

Handling Multicollinearity: Ridge regression is effective in dealing with multicollinearity, a situation where predictor variables are highly correlated. In real estate datasets, it is common for features like number of bedrooms, and number of bathrooms to be correlated. Ridge regression helps stabilize the model and prevents overfitting, making it robust in scenarios with multicollinear features. This is the given result



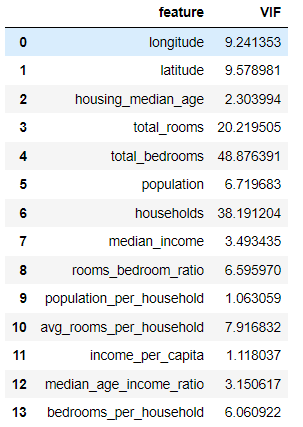
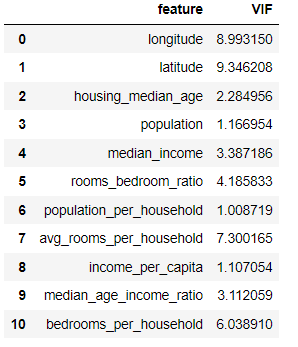
1. **Elastic Net Regression**

Elastic Net is a hybrid regression technique, combining Lasso and Ridge strengths. By introducing both L1 and L2 regularization terms, it excels at feature selection (like Lasso) and managing multicollinearity (like Ridge). In California house pricing, where diverse factors impact prices, Elastic Net provides a balanced approach, maintaining relevant features while handling correlations. This is the given result



**5. Discussion**

As we know that this dataset is in the domain of real estate and finance. There were a lot of attributes which were dependent on each other i.e., there is a very high multi-collinearity between some attributes. For us to make a better model we must remove those attributes which are highly dependent on others. To check of multi-collinearity, in this project we have used Variation Inflation Factor (VLF), a package from the library statsmodels.stats.outliers\_influence. This image below (left) is the variation score of all the attributes in the dataset. We can see that some of the attributes have very high score. Now, I removed all the attributes which have a score greater than 10.

****

**Before After**

We can see that after dropping the attributes for high collinearity we can see that the score of the other attributes also have been changed. All the attributes in the image (right) we use those for building our models.

**6. Conclusion**

In the below image we can see the score of all the models:

1. The K-Fold cross-validation models provide a solid baseline for predicting California house prices, demonstrating good generalization to unseen data.
2. Feature selection methods like RFE and RFE-VIF offer competitive performance, emphasizing the importance of selected features in predictive accuracy.
3. PCA, while reducing dimensionality, results in a noticeable drop in predictive performance, suggesting caution in adopting aggressive dimensionality reduction.
4. Multiple Linear Regression with Lasso, Ridge, and ElasticNet showcases their effectiveness in managing feature selection and multicollinearity, with ElasticNet offering a balanced compromise.

In summary, the choice of the model may depend on specific priorities such as interpretability, feature importance, or the trade-off between bias and variance. But I would be taking the **Ridge model** for further optimization and fine-tuning could enhance the overall performance of the models.