Assignment 2 : Predicting California Housing Prices Using Multiple and Polynomial Regression¶

Majid Rostami	Ayush Juvekar	Aditya Karle	Shreyash Waghdhare
Michigan Technological University	*Michigan Technological University*	*Michigan Technological University*	*MTU*
majidr@mtu.edu	aajuveka@mtu.edu	arkarle@mtu.edu	spwaghdh@mtu.edu

Abstract – This report outlines our methodology and results for a project aimed at prediction of California housing price based on an existing dataset. We employed four methods: multiple linear regression model, polynomial regression, XGBoost Regressor and finally sequential algorithm from deep learning networks. In addition, in the preprocessing part, we examined the effects of basic features of dataset, as well adding multiple derived interactive features. Moreover, from PANDAS library, correlation matrix is used to determine independence of features and eliminate highly correlated ones. Furthermore, logarithmic transformations are used to smooth the feature space and mitigate the effects of outliers. Regarding predictive capability metrics, we use RMSE, R^2 , Mean Residual Percentage, and the coefficient of variation (COV) of residuals. Comparing results, XGB Regressor possesses the highest accuracy among other methods with R^2 of around 84%.

Index Terms – Housing Price Prediction, Multiple Linear Regression Model, Polynomial Regression, XGB Regressor, Deep Learning.

INTRODUCTION

Accurate housing price prediction is essential for buyers, sellers, and policymakers to make informed decisions in the real estate market. Predicting California housing prices involves utilizing various regression models to capture the relationships between housing features and prices. Commonly used models include multiple linear regression, polynomial regression, and advanced techniques such as XGBoost Regressor and deep learning algorithms. Each model has its strengths and weaknesses in terms of accuracy and computational efficiency.

One of the primary challenges in this task is managing the dataset itself. Housing data often contains missing values, outliers, and highly correlated features, which can skew the results. Effective preprocessing steps, such as handling missing data, applying logarithmic transformations, and eliminating multicollinearity, are crucial for improving model performance. Additionally, the choice of evaluation metrics, such as RMSE (Root Mean Squared Error) and R^2 (Coefficient of Determination), plays a significant role in assessing the predictive capability of the models.

In this assignment, we attempt to address those challenges in our pipeline and see their effects on results.

METHODOLOGY

The dataset includes key features such as "longitude," "latitude," "housing median age," "total rooms," "total bedrooms," "population," "households," "median income," and "ocean proximity," while the target value is "median house value".

To handle missing values, we first identified the number of missing values in each column, finding 207 missing values in the "total_bedrooms" column. We chose to fill these missing values with the mean of the respective column, as replacing them with zero would not be neutral and could alter the distribution probability function. After filling the missing values, we confirmed that there were no remaining missing values in any column. This approach ensures that the dataset is complete and ready for further analysis and modeling.

In the next step, we performed feature engineering to enhance the predictive power of our regression models. Initially, we encoded the categorical "ocean_proximity" feature using a label encoder. We then selected relevant features and the target variable, "median_house_value." To handle outliers, we provided an optional method using the Interquartile Range (IQR) method to filter out extreme values, ensuring a more robust dataset.

Next, we calculated the correlation matrix to identify and remove highly correlated features, so addressing multicollinearity. We also applied logarithmic transformations to smooth the feature space and mitigate the effects of outliers. Additionally, we created interactive features, such as interactions between rooms and income, and between latitude and longitude, to capture more complex relationships within the data. To have more profound insight into different parameters of the model, we

will conduct a sensitivity analysis by removing different features to assess their impact on the results.

To demonstrate the distribution of features, we used histograms and box plots. Histograms provide a visual representation of the frequency distribution of each feature, while box plots highlight the spread and potential outliers. Histograms show must features demonstrate a normal distribution.

In the Preprocessing part, finally, we applied two types of normalization to the features and observed their impacts on the results. First, we used the StandardScaler to scale the features based on their mean and standard deviation. Alternatively, we provided an option to use the MinMaxScaler to scale the features to a range between 0 and 1.

Multiple Linear Regression (MLR) models the relationship between a dependent variable and multiple independent variables, aiming to find the best-fitting hyperplane that minimizes prediction errors. Key assumptions include linearity, independence, homoscedasticity, normality, and no multicollinearity. Coefficients are estimated using the Least Squares Method (LSM), which minimizes the sum of squared residuals. Model evaluation metrics such as Mean Squared Error (MSE) and R-squared (R^2) are used to assess performance. MLR is advantageous for its simplicity, interpretability, and efficiency but has limitations like sensitivity to multicollinearity and outliers, and it may struggle with non-linear data. Applications range from real estate pricing to healthcare and finance.

Polynomial regression extends linear regression by modeling the relationship between a dependent variable and one or more independent variables using an n-th degree polynomial. This approach captures non-linear relationships, making it suitable for complex data patterns. The model transforms original features into polynomial terms, allowing it to fit curves rather than straight lines. Key differences from linear regression include a higher risk of overfitting and increased model complexity. Coefficients are estimated using the least squares method, and the model is evaluated using metrics like R-squared (R^2) and Root Mean Squared Error (RMSE). Polynomial regression is advantageous for its flexibility and ability to model non-linear trends but can be sensitive to outliers and harder to interpret with higher degrees. Applications range from stock price prediction to engineering and healthcare.

The XGBRegressor from the XGBoost library is a powerful and flexible tool for regression tasks, using gradient boosting techniques to predict continuous target variables. It's designed to handle various types of data and is known for its high accuracy and efficiency, making it ideal for large datasets and complex problems. The model supports a range of parameters that can be fine-tuned, such as learning rate, maximum depth of trees, and the number of boosting rounds. It also includes built-in mechanisms to handle missing values and prevent overfitting, like regularization and early stopping. This makes XGBRegressor a popular choice for tasks like predicting housing prices, stock prices, and other continuous outcomes.[1]

The Sequential model from TensorFlow's Keras API is a simple and effective way to build neural networks by stacking layers sequentially. In our example, we used the Sequential model to create a regression model with three layers: two hidden layers with 64 and 32 neurons, respectively, and an output layer with one neuron. We compiled the model with the Adam optimizer and mean squared error loss function, then trained it on our dataset.

In this analysis, we used several metrics to evaluate the performance of our models. The Root Mean Squared Error (RMSE) measures the average magnitude of the prediction errors, with lower values indicating better performance. The R-squared (R^2) metric assesses how well the model explains the variance in the target variable, with values closer to 1 indicating a better fit. We also calculated the mean residual percentage to understand the average percentage error in predictions and the coefficient of variation (CV) to assess the relative variability of the residuals. These metrics provide a comprehensive evaluation of the model's accuracy and reliability.

RESULTS

In this part, we start by displaying heatmaps to show the correlation matrix between features, followed by histograms and bar plots to illustrate the distribution of features.

We then present the results for the four different methods in separate tables, including metrics and the outcomes of some sensitivity analysis.

[1]https://xgboost.readthedocs.io/en/latest/python/python_api.html

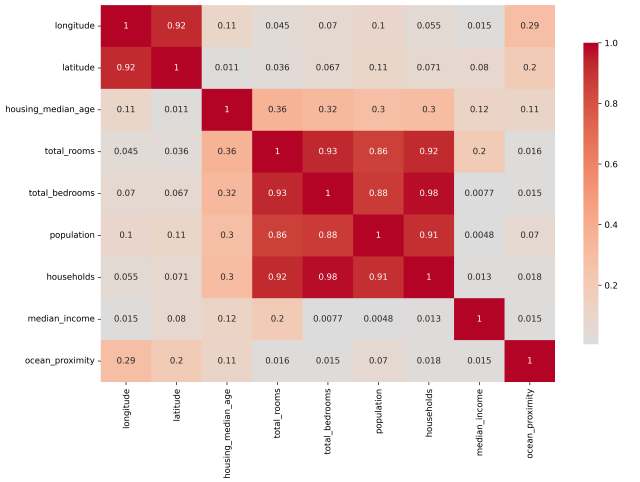

FIGURE 1 HEATMAP FOR BASIC FEATURES

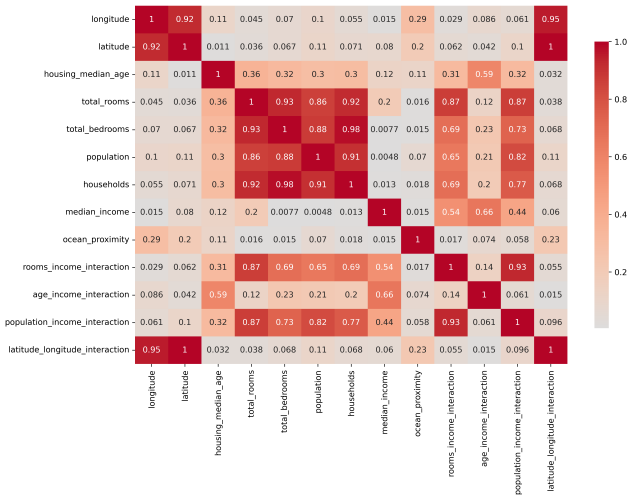

FIGURE 2 HEATMAP FOR BASIC AND INTERACTIVE FEATURES

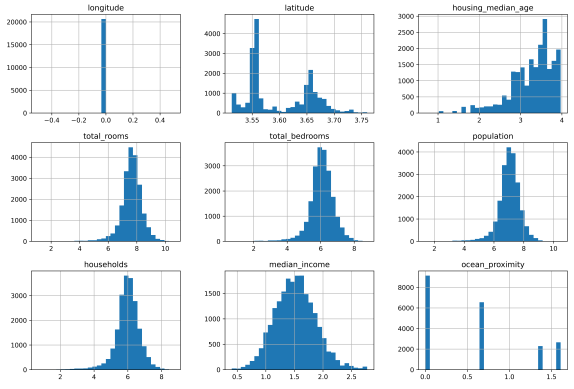

FIGURE 3 FEATURES HISTOGRAMS AFTER LOG TRANSFORM

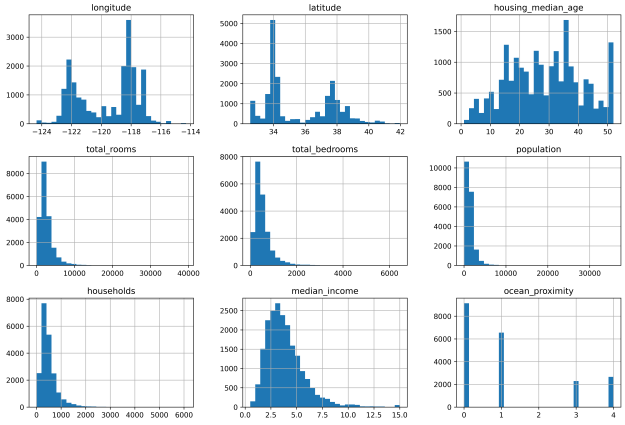

FIGURE 4 FEATURES HISTOGRAMS BEFORE LOG TRANSFORM

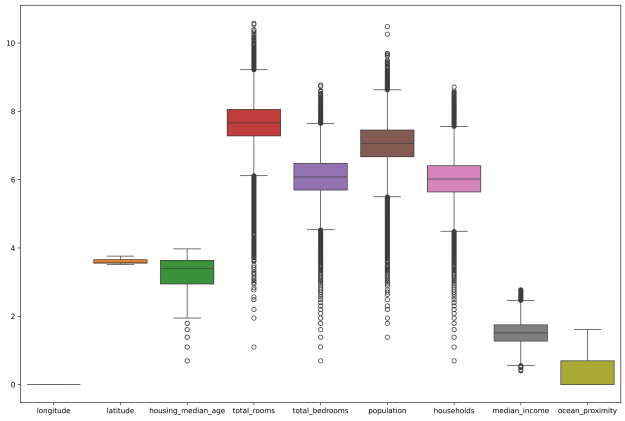

FIGURE 5 BAGPLOTS OF FEATURES AFTER LOG TRANSFORM

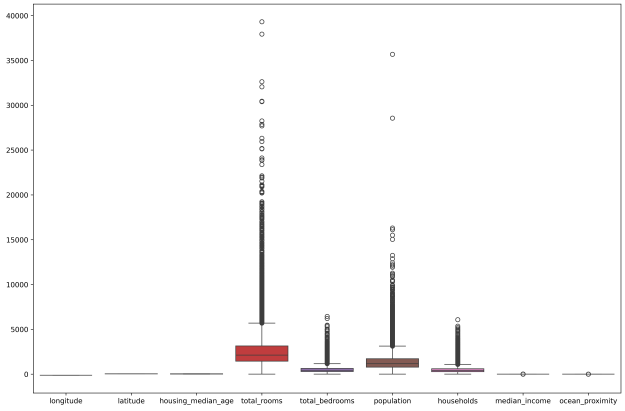

FIGURE 6 BAGPLOTS BEFORE LOG TRANSFORM

TABLE 1. RESULTS FOR MULTIPLE LINEAR REGRESSION MODEL BASIC FEATURES (1 FOR ACTIVE, 0 FOR INACTIVE).

RMSE Train	RMSE Test	RMSE Train	RMSE Test	Mean Residual	CV	Features Correlation (80% for removal)	Feature Log Transform	IQR Method
69361	71098	0.64	0.61	31.08%	0.938	0	0	0
79980	80424	0.52	0.51	37.46%	0.903	1	0	0

RMSE Train	RMSE Test	RMSE Train	RMSE Test	Mean Residual	CV	Features Correlation (80% for removal)	Feature Log Transform	IQR Method
69474	73492	0.63	0.58	32.96%	0.943	1	1	0
57831	59926	0.57	0.55	27.91%	0.953	1	1	1
53450	55464	0.63	0.62	25.88%	0.96	0	0	1

TABLE 2. RESULTS FOR MULTIPLE LINEAR REGRESSION MODEL WITH INTERACTIVE FEATURES (1 FOR ACTIVE, 0 FOR INACTIVE).

RMSE Train	RMSE Test	RMSE Train	RMSE Test	Mean Residual	CV	Features Correlation (80% for removal)	Feature Log Transform	IQR Method
69474	73492	0.63	0.58	32.96%	0.943	0	0	0
79903	80408	0.52	0.50	37.36%	0.906	1	0	0
79717	80371	0.52	0.51	37.42%	0.89	1	1	0
69474	73492	0.63	0.58	32.96%	0.943	1	1	1

TABLE 3. RESULTS FOR DIFFERENT DATA NORMALIZER (1 FOR ACTIVE, 0 FOR INACTIVE).

RMSE Train	RMSE Test	RMSE Train	RMSE Test	Mean Residual	CV	Interactive Features	Normalizing Method
69361	71098	0.64	0.61	31.08%	0.938	0	StandardScaler
69474	73492	0.63	0.58	32.96%	0.943	1	StandardScaler
69361	71098	0.64	0.61	31.08%	0.938	0	MinMaxScaler
51745	53729	0.66	0.64	25.15%	0.971	1	MinMaxScaler

TABLE 4. RESULTS FOR POLYNOMIAL REGRESSION (N=2) WITHOUT ADDITIONAL OPERATIONS WITH MIN-MAX-SCALAR (1 FOR ACTIVE, 0 FOR INACTIVE).

RMSE Train	RMSE Test	RMSE Train	RMSE Test	Mean Residual	CV	Features Correlation (80% for removal)	Interactive Features
62646	67376	0.71	0.65	26.84%	1.03	0	0
71773	72643	0.61	0.60	30.08%	0.96	1	0
46120	47767	0.73	0.72	21.42%	1.03	0	1
52646	54717	0.64	0.63	25.63%	0.98	1	1

TABLE 5. RESULTS FOR XGB REGRESSOR WITHOUT ADDITIONAL OPERATIONS WITH MIN-MAX-SCALAR (1 FOR ACTIVE, 0 FOR INACTIVE).

RMSE Train	RMSE Test	RMSE Train	RMSE Test	Mean Residual	CV	Features Correlation (80% for removal)	Interactive Features
17531	45803	0.98	0.84	17.38%	1.14	0	0
30992	59013	0.92	0.73	22.65%	1.06	1	0
15181	46450	0.97	0.73	20.87%	1.03	0	1

TABLE 6. RESULTS FOR DEEP LEARNING SEQUENTIAL WITHOUT ADDITIONAL OPERATIONS WITH MIN-MAX-SCALAR; WITHOUT HYPERPARAMETER OPTIMIZATION (1 FOR ACTIVE, 0 FOR INACTIVE).

RMSE Train	RMSE Test	RMSE Train	RMSE Test	Mean Residual	CV	Features Correlation (80% for removal)	Interactive Features
73535	74383	0.60	0.58	32.43%	0.94	0	0

CONCLUSION

In conclusion, our analysis shows that the XGBoost Regressor is the best model for predicting California housing prices. It achieved the highest accuracy with an R^2 value of 0.84 and the lowest RMSE values in both training (17,531) and testing (45,803) phases. This success is due to its ability to handle different data types, use regularization techniques, and manage missing values and outliers effectively.

On the other hand, the multiple linear regression and polynomial regression models performed less well, with lower R^2 values and higher RMSE scores. For example, the multiple linear regression model with basic features had an R^2 of 0.64 and an RMSE of 71,098 in the test phase. Additionally, removing highly correlated features did not significantly improve performance. The deep learning sequential model also lagged behind XGBoost, with an R^2 of 0.58 and an RMSE of 74,383 in the test phase.

Furthermore, the MinMaxScaler normalization method proved to be more effective than the StandardScaler, as evidenced by the lower RMSE values in the results tables.

These findings highlight the importance of using proper method for normalization and learning model.

CONTRIBUTIONS

We, the undersigned, confirm that this assignment is a group project. All steps and deliverables were completed by the members (specify the roles) listed below.

Majid Rostami: Coding, Report Writing
Ayush Jovekar: Coding, Report Writing
Adita Karle: Coding, Report Writing
Shreyash Waghdhare: Coding, Report Writing

Signatures:
Majid Rostami
Ayush Jovekar
Adita Karle
Shreyash Waghdhare