CSP571-Data Preparation and Analysis

Project Report

Topic: Predictive Analysis and Clustering of Housing Data

Prepared by:

Nachiketh Nallamaddi - A20549679

Anmol Rao Karukonda - A20554502

Somu Medaka - A20548401

1. Introduction

This project is to realize how different columns of this California housing dataset predict houses' prices. We shall apply dimensionality reduction techniques and explore unsupervised clustering methods with Gradient Boosting, XGBoost, and cross-validation. Lastly, we will use visualization techniques for further comprehension of the results obtained. We will do feature engineering, hyperparameter tuning, and ensemble methods to improve the accuracy of the model. The project would be helpful in many ways, such as real estate analysis, urban planning, and a lot more.

2. Dataset Exploration

Dataset Summary

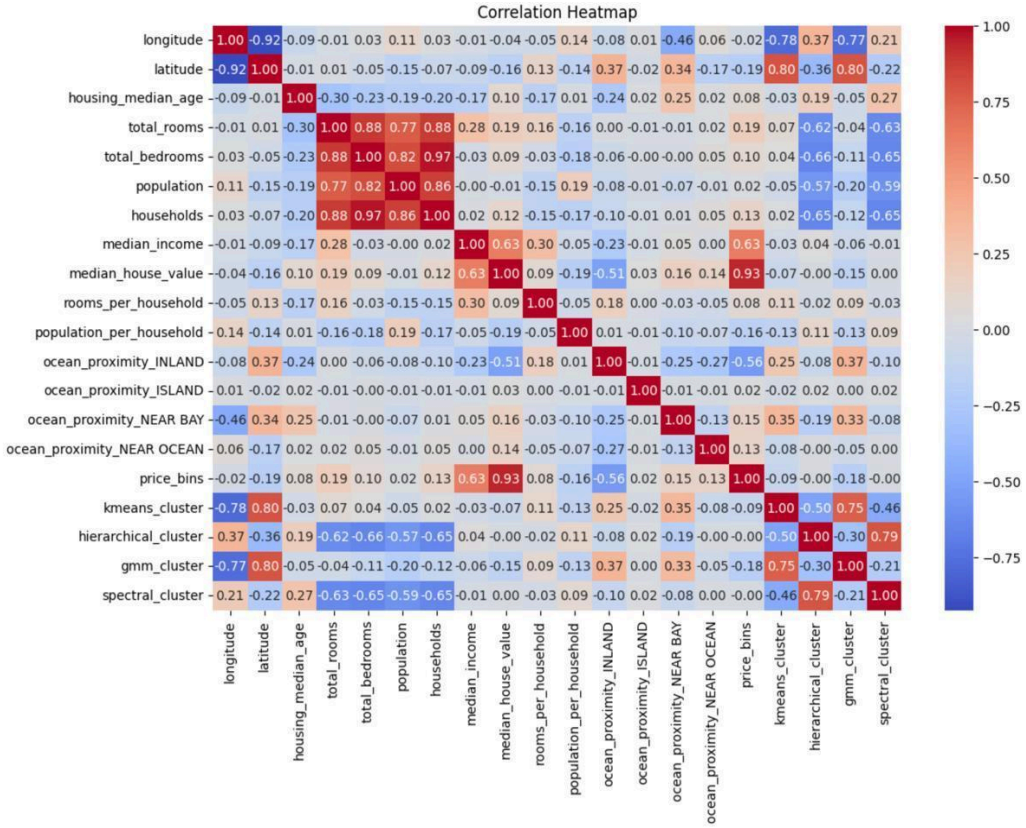
The dataset used for this analysis is "California Housing Prices," which is a derived dataset based on the 1990 census. This dataset has 20,640 rows and ten columns. The major task is to predict prices using several housing and demographic features. Features represent Total Rooms, Households, Total Bedrooms, Population, and Median Income.

```
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
```

Correlation Analysis

There were in-depth correlation analyses, relating the features to see their relations to the target variable. Each of these relations was calculated and visualized using a heatmap. The Median Income feature is highly correlated with the target variable. This means that areas with high median incomes amount to higher values of this particular target; hence, it stands out as

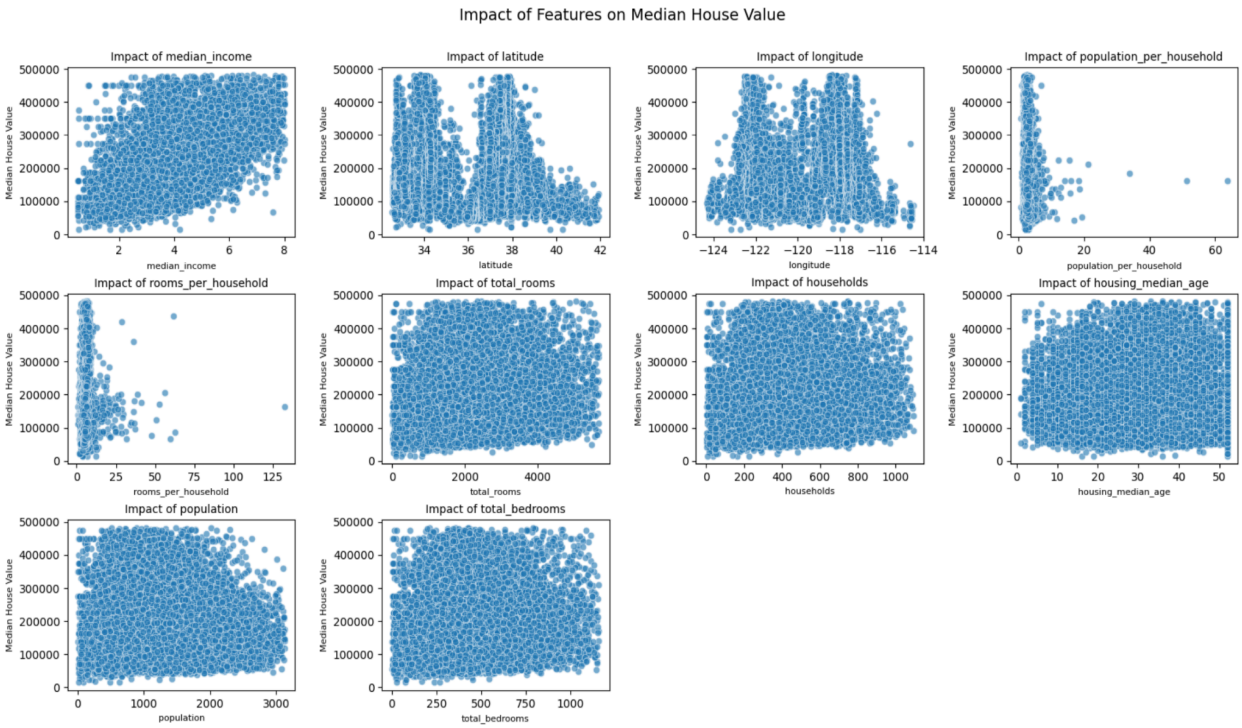
an important variable in predictive modeling. Total Rooms and Households are moderately positively correlated with the target variable, and some influence of them can be observed.



On the other hand, the features Population and Total Bedrooms show a weak or negligible correlation with the target variable; they do not directly influence the target, which may need further investigations for better transformation or their interaction with other features for predictive power. These findings present important recommendations concerning the prioritization of features for modeling and identification of those features where preprocessing or engineering may yield improvements.

Impact on Target

Scatter plots, pair plots, and distribution analyses are some of the techniques applied. This is applied to know the relations across features and the target variable. The most influencing predictor is Median Income because, with increased income level, there is increased target value. Thus, with just this relationship alone, it should be said to be needed for predictive modeling. This makes it a prioritized feature for analysis and modeling.



While, on the other hand, Total Rooms and Households did show positive features, being found at their decrease of influence at higher values; probably that's what represented saturation levels, more than which increases in the room count or households would barely increase the value of the target variable with an equivalent degree of variation. On the contrary, Population depicted a pretty minor effect at a really poor correlation level; probably this is very negligible help for an independent predictor.

Independence Assumptions

In order to check the assumption of independence, a Variance Inflation Factor analysis was conducted to check for multicollinearity among the features in the dataset.

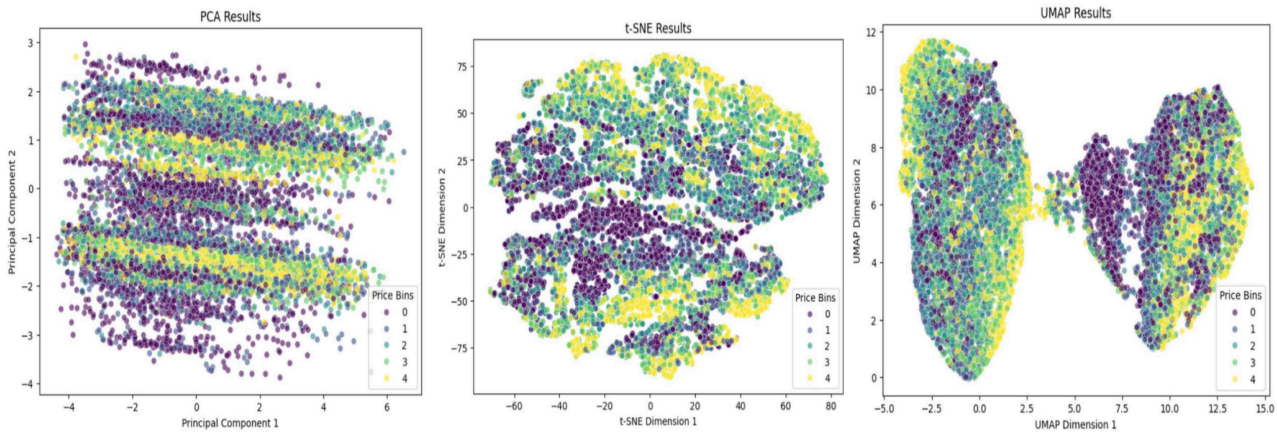
```
Variance Inflation Factor (VIF) Analysis:
              Feature        VIF
0           longitude   9.004144
1            latitude   9.486034
2  housing_median_age   1.227811
3         total_rooms   9.476551
4      total_bedrooms  19.287775
5          population   4.191276
6          households  21.167115
7       median_income   2.144942
```

VIF quantifies how much feature variance is inflated due to its linear correlation with other predictors. From the analysis, it is quite clear that Total Bedrooms and Households have a VIF of

19.29 and 21.17, respectively, hence highly multicollinear and redundant. The features of Longitude, Latitude, and Total Rooms also showed very high VIF values at about 9, indicating redundancy.

In contrast, Housing Median Age and Median Income show low VIFs of 1.23 and 2.14, respectively, indicating low multicollinearity and a high degree of independence.

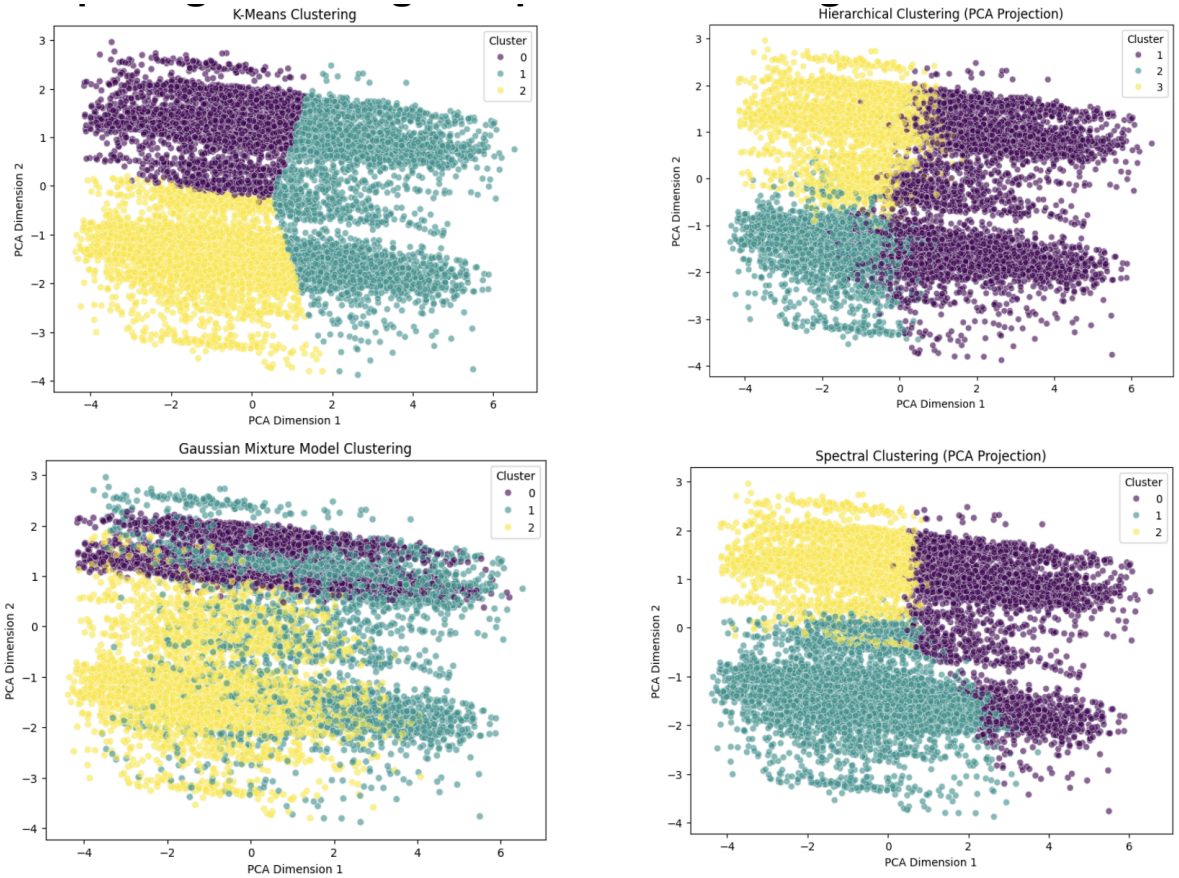
3. Data Visualization



Reduction techniques such as PCA, t-SNE, and UMAP were used to explore significant patterns and clusters in this dataset. These methods reduce the feature to two dimensions such that most of the intrinsic structures in the data are preserved. Principal component analysis also identified the two most contributory components to the dataset variance. It pointed out the feature of the space critical dimensions under which the features of variability are related and how it has been simplified in data representation. While the clustering in PCA was rigid, it has been used because, though linear, it makes a strong baseline as far as data distribution is concerned.

t-SNE and UMAP further captured the nonlinear relationship of the structure in this dataset. t-SNE focuses much more on localized patterns. The clusters obtained had fine-grained separations, which turned to be particularly effective at capturing small subgroups. However, UMAP captures a glimpse of the global structure, hence producing smoother, more coherent clusters, effectively leveraging broad trends and relationships; that is, clusters that were also generally well separated.

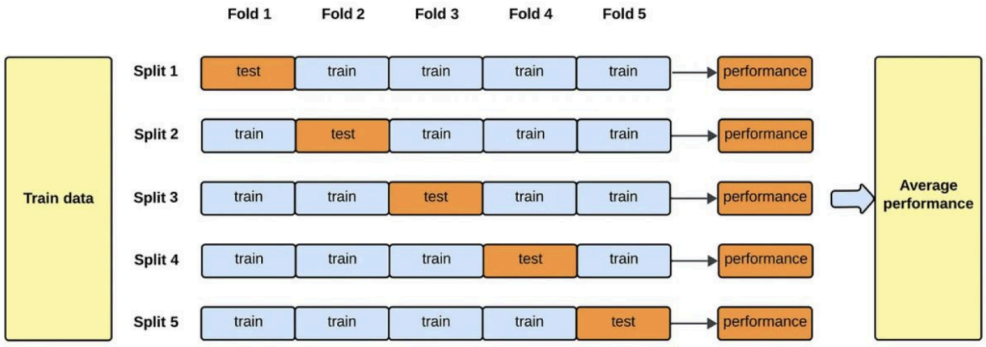
4. Data Exploration Using Unsupervised Learning



Techniques involved in the unsupervised analysis of this data consisted of K-Means clustering, Hierarchical Clustering, Gaussian Mixture Modeling, and Spectral Clustering – visualizing all projected within a PCA to circumvent complexities in higher-dimensional space. K-Means separated three well-defined classes by optimizing intra-cluster variance, minimizing inter-similarity in-class assignment, and proving highly suitable for linearly separable features. Hierarchical Clustering produced similar structures to K-Means but offered the flexibility of a tree-based approach, bringing in the hierarchical relationships intrinsic in data. GMM provided a view of probability, where clusters had soft boundaries, allowing for overlapping points, thereby capturing uncertainty in the allocation of data. Applying graph-based techniques, the spectral clustering identified complex nonlinear relations in the data by constructing subtle boundaries, thus identifying further nonlinear patterns similar to K-Means results.

5. Model Development

Cross-Validation Strategy

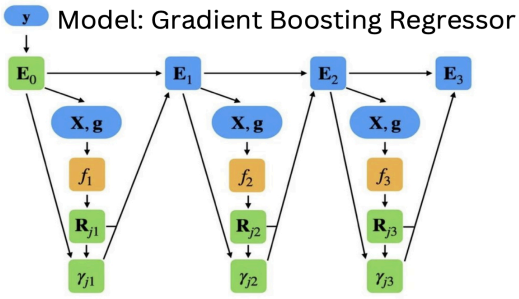


For stronger model evaluations and the avoidance of overfitting, k-fold cross-validation with k = 5 was performed. The process works by splitting the data into five equal-sized folds. It trains on four folds on every iteration and uses one-fold for validation. Once five rounds are complete, performance metrics are averaged across the different folds. This technique reduces the influence of data variability, hence offering a better estimate of model performance and generalization capability.

The data were split into further subsets, apart from cross-validation, for training, validation, and testing purposes. The training set comprised 70% and was used to fit the models. Then, 15% were used for hyperparameter tuning, which included early stopping, preventing overfitting of the model. The remaining 15% were used as the test set for the final evaluation to ensure fairness in evaluating the model on unseen data.

Simple Model

First to note, a Gradient Boosting Regressor was trained on its default hyperparameters to set a baseline of performance.



This model was selected since it is appropriate for a regression task and is pretty efficient in handling nonlinear complex feature-to-target variable relationships. The model was trained on the 70% training subset, and predictions were evaluated on the 15% validation subset. Performance metrics were 677.06 for the Validation MSE and 0.88 for the Validation R^2. The model performed quite well, with a very high R^2 value, indicating its strength in identifying critical patterns in the data.

6. Hyperparameter Tuning

The hyperparameter tuning was done by a Grid Search to optimize the performance of the Gradient Boosting Regressor. Important parameters of learning rate, number of estimators, and max depth were varied over a predefined range. Besides, a 5-fold cross-validation strategy was implemented to evaluate each combination of the parameters. Then, the best-performing hyperparameters were obtained using a learning rate of 0.1 with some estimators of 200 and a max depth of 5.

After obtaining these optimized parameters, manifold increase in model performance was obtained. The Validation MSE went down to 603.43, while the Validation R^2 came up to 0.92, meaning that the model explained 92% of the variance in the target variable.

7. Performance Improvement Strategies

Feature Engineering

Interaction terms and transformations were designed to enhance the predictability of the model like Rooms per Household, Bedrooms per Room, Population per Household, and Log Median Income. The interaction features that were engineered here were very strong in capturing important relationships in this dataset and, thus, greatly improved the generalization ability of this model. This gave a Validation MSE of 602.11 and an R^2 of 0.93 which illustrated the power of these custom features.

Polynomial Features

Added second-order polynomial features to the data, which helped in capturing the nonlinear interactions of variables; this extended the feature space and enhanced predictive power. However, it increased the risk of overfitting. The Validation MSE went up to 648.15, while the R^2 score stayed strong at 0.93, but again showed the requirement of balancing model complexity with generalization capability.

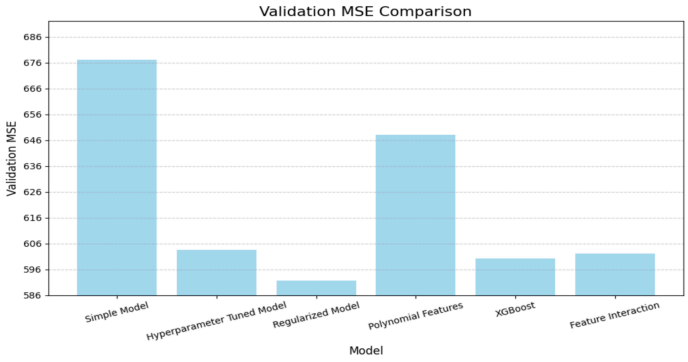
Regularization

Overfitting was addressed by regularization techniques like subsampling and L1/L2 penalties. These are strategies aimed at making the model more robust against its own complexities. In fact, a tuned and regularized version of the Gradient Boosting model performs best among all, while offering a very competitive result with a Validation MSE of 591.53 and an R^2 of 0.93, at minimized error and maintained predictive accuracy.

XGBoost

XGBoost was also applied since it is an efficient ensemble algorithm, making use of hyperparameter tuning and intrinsic regularization. This provided a performance with the Validation MSE equal to 600.12, and R^2 was equal to 0.93. XGBoost performed similarly to the regularized Gradient Boosting variant, conferring extra benefits on computational efficacy.

Comparison



Among these models, the Regularized Gradient Boosting Model had the best performance for the lowest Validation MSE, which means this model generalizes very well. XGBoost and Feature Interaction are performing closely to each other, while the Polynomial Features model offers an improved fit at the cost of a higher risk of overfitting due to its high degree of complexity.

8. Conclusion

In this project we have successfully applied data exploration, feature engineering, and advanced modeling to improve the predictive accuracy . Strong predictors such as Median Income and Rooms per Household cropped up as important features. Dimensionality reduction and clustering have provided critical preprocessing insights, while regularization and hyperparameter tuning proved highly instrumental for the performance improvements. The

front was led by the Regularized Gradient Boosting Model, which gave a Validation MSE of 591.53.

Github link: https://github.com/Nachiketh1717/CS571_Project/blob/main/DPA_Project.ipynb