ML Midterm Checkpoint - Group 96

Introduction/Background

Predicting house prices is an important indicator for real estate stakeholders, including buyers, sellers, and investors. With numerous factors influencing prices, such as location, property size, and condition, predicting accurate sale prices can be complex. House price prediction is a critical problem that goes beyond traditional metrics like the House Price Index (HPI), which measures broad market trends but lacks the granularity needed for individual home price prediction [1]. Recent studies have shown that machine learning models, by incorporating factors such as location, house age, and property features, can significantly enhance prediction accuracy [1][2]. Predicting house prices is crucial for real estate stakeholders and remains complex due to interdependent factors like location, property size, and amenities, as highlighted by studies using publicly available datasets in cities like Bengaluru [3].

Our project aims to explore these advanced methods, using the House Prices - Advanced Regression Techniques dataset from Kaggle, which provides a rich set of features—79 features—to predict housing costs based on property-specific attributes.

Problem Definement

The objective of this project is to develop 3 machine learning models that can predict the **SalePrice** of a house based on various features provided in the dataset. Accurate house price prediction is important for financial decision-making and investment planning. The problem lies in identifying the best model and preprocessing techniques to capture the underlying patterns in the data. This project aims to explore multiple machine learning algorithms to create an accurate predictive model.

Methods

Data Preprocessing

The primary methods of data preprocessing that we use include one-hot encoding, feature engineering with polynomial features, recursive feature elimination (RFE), and standardization to filter through the data. One-hot encoding helps in representing categories independently, allowing variables like price and square feet to be represented without implying relationships between them. Feature engineering with polynomial features involves creating interaction terms between existing features, enabling the model to capture more complex relationships within the data. Recursive feature elimination then helps to select only the most relevant features, ensuring that unnecessary or redundant interactions do not cause overfitting while retaining essential information. This was especially important to use since an estimated 46000 features were generated, so eliminating a majority of them is necessary. Finally, standardizing the resulting features to a uniform scale, with a mean of 0 and a standard deviation of 1, enhances the model's stability by preventing large features from skewing the optimization process.

Out of the 79 different features provided in the dataset, one-hot encoding was used to change the categorical features (string features) into binary features. For example, if the original features was color, then one-hot encoding would create `red`, `blue`, and `green` in place of "color" and assign boolean values to them, assuming that red, blue, and green are all string inputs in the original feature.

As such, one-hot encoding created 303 features, and through polynomial feature engineering, we created combinations of features, where `Feature1` is multiplied by `Feature2` to result in something like `Feature_1_x_Feature_2`, eventually giving us "303 choose 2", or `~46000` features. Due to us having only `~1400` data samples, having `46000` different features is unacceptable and would need to be reduced to a more reasonable value lower than `1400`.

In order to find the optimal value for RFE, we implemented grid search since it searches through a set of parameters to find a value that gives us the best performance. However, due to long runtimes of each iteration being exceedingly long, our grid search was smaller with bigger steps, with values being tested of `500`, `600`, `700`, `725`, `750`, `775`, `800`, `825`, `850`, `875`, `900`, `1000`, `1100`, and `1200`.

Algorithms/Models

Linear Regression

Linear regression is a method we use since the relationship between house prices and the features that the house are often linear. This provides a model that displays the different coefficients and factors that affect the price, and how much impact they have. Furthermore, linear regression is also computationally efficient and can handle large datasets, so the vast amount of housing data that is available can be easily sifted through.

Neural Network

Neural networks are utilized for predicting house prices due to their ability to capture complex non-linear relationships between features, including interactions between quantitative and qualitative factors, like square feet and neighborhood safety respectively. This algorithm benefits from both standardized features and reduced dimensions via recursive feature elimination, ensuring efficient handling of the thousands of features generated during preprocessing. The flexibility of neural networks allows them to weigh diverse inputs and automatically learn feature interactions without manual engineering. Additionally, the model architecture, potentially with multiple layers and activation functions, enables the neural network to generalize across the dataset and predict prices accurately, even with high-dimensional and varied data types.

Regression Tree

Regression trees used to model complex, non-linear relationships between features and target variables by iteratively splitting the data based on optimal feature thresholds. This approach enables the model to capture interactions between variables, such as the effect of neighborhood quality combined with house size, without requiring prior assumptions about the data's structure. Specifically,

DecisionTreeRegressor is particularly effective for mixed data types, as it handles both numerical and categorical variables without requiring any more preprocessing. The model's max_depth is set to control overfitting by limiting the number of splits, ensuring the model generalizes well to new data. Metrics like MSE and R^2 are used to validate performance, while the calculated feature importances offer insights into which variables affect price predictions.

Results and Discussion

Linear Regression

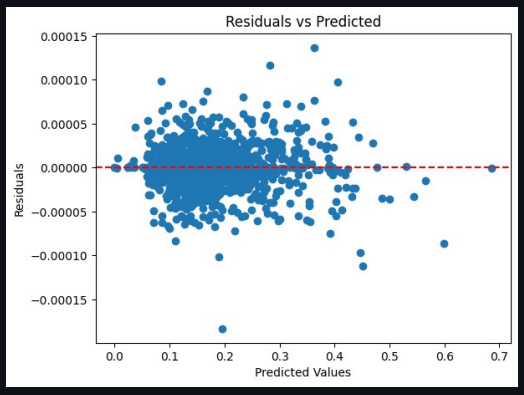


Figure 1 - Linear Regression Result

From the previous list of RFE values listed that we used, we found that lower values like `500` were underfitting, `1200` was overfitting with a magnitude of `~1,000,000,000`, while still giving a RMSE value of ~ `0.017`. Due to it being house prices, an RFE value of `775` gives a value `~400` features, resulting in RMSE being around `0.5%`, well within the bounds of acceptable error.

In this plot visualization of the resulting data, the residuals are centered around the red dotted zero line, indicating that the model's predictions are unbiased and do not contain over/under prediction. The y-axis scale being values extremely close to 0 also indicate that the randomness of the majority of the points do not create a pattern and the residuals being consistent across all predicted values.

However, there are some outlier points that may indicate that could negatively affect the model, but because of randomness in house prices and the quantifiable data like square feet, this could serve to be major exception and are not commonly found in real life beyond possibly 1 or 2 instances. As such, some next steps and ways to address the outliers could be to simply remove them from the data for their exceedingly rare occurrence in the real world, or to implement model regularization to penalize large coefficients and making the model less sensitive to their impact.

Neural Network

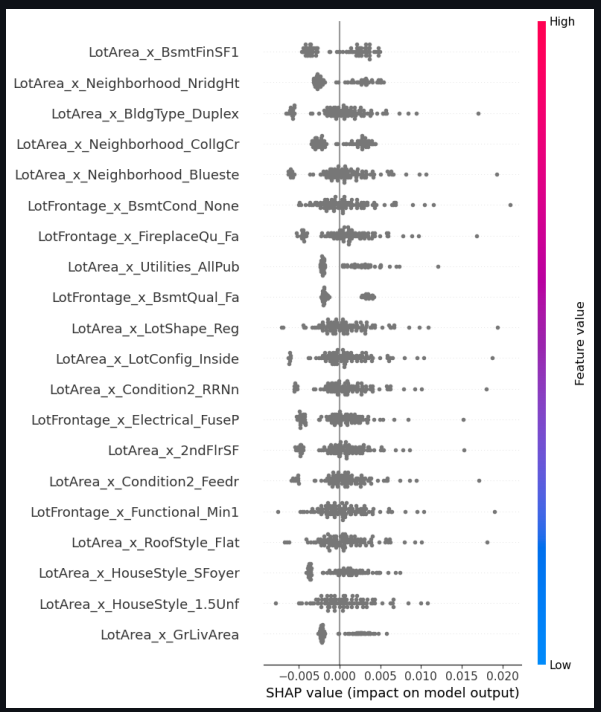


Figure 2 - Neural Network SHAP Results

The results from the residual plots and the *Figure 2* SHAP (Shapley Additive Explanations) analysis suggest that the neural network implemented predicted well in house prices and was able to capture meaningful relationships based on the features and target values. Because the residual plots centered around `0`, it shows that the model was unbiased in making its predictions. The SHAP shows that the model's focused on high impact key features, like `LotArea_x_BsmtFinSF1` and `LotArea_x_Neighborhood_NridgHt` to predictions. There were many expectations, where if the house had a

large square footage or was in higher-quality, then it would influence the house price much more.

However, there were some issues with the model, where sometimes the residuals would show a pattern of a higher predicted value in the test set, suggesting that the model struggles with keeping a constant variance across observations. Furthermore, it seems to have a difficulty in complex features. In the graph where categories were near the bottom of the gradient, such as `LotArea_x_HouseStyle_1.5Unf` and `LotArea_x_GrLivArea` where there is little impact. This could mean a multitude of things, such as redundancy or heavy noise in the data.

Regression Tree

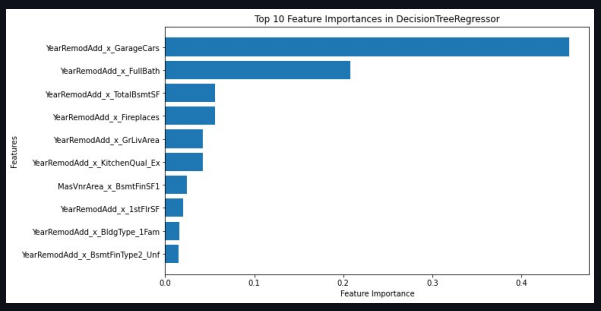


Figure 3 - Regression Tree Top 10 Resulting Features

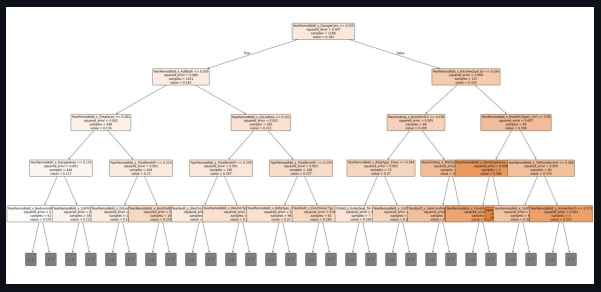


Figure 4 - Decision Tree Depth Possibility

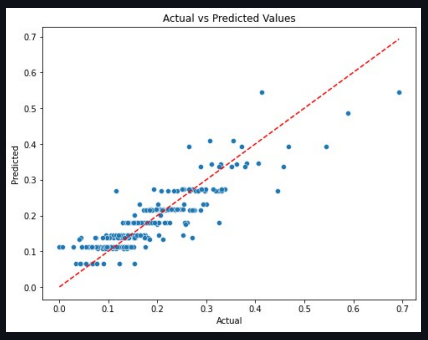


Figure 5 - Regression Tree Prediction Line

From the regression tree model, the model demonstrates that it was able to recognize a relationship between house features and house prices. In *Figure 5* with the prediction line, it shows a strong correlation along the diagonal, showing that the model performed well in predicting the house prices based on the data points. In *Figure 4* of the decision tree, the splits indicate how the model hierarchically identifies the factors that impact house prices sequentially starting at the top with the most important feature like `YearRemodAdd_x_GarageCars` and `YearRemodAdd_x_FullBath`, then working downwards to less impactful ones. This ties in well with *Figure 3*, where it confirms that certain features like the number of garage cars and number of full bathrooms are important predictors.

However, A potential downside of the regression tree is its likelihood to overfit, especially if the tree would have too large of a depth. Simply having a set value for `max_depth` would not be enough, since it introduces the possibility of also underfitting; it is finicky to dynamically determine a middle ground for all scenarios. Furthermore, regression trees are also sensitive to small changes in the data, which can cause variability in predictions, especially in the nodes that have the most importance.

Model Comparison

Linear regression, neural networks, and regression trees have strengths and weaknesses that make them suited for different scenarios in predicting house prices.

Out of the three, linear regression is the simplest, with just relying on the assumption of a relationship between the independent variables and the dependent variable, which in this case are the house features and house prices respectively. It calculates coefficients to minimize the sum of squared errors, making it easy to interpret and also being computationally efficient. However, linear regression struggles with non-linear relationships and interactions. It is sensitive to outliers, which can skew predictions when dealing with high-dimensional, noisy, or complex datasets.

Neural networks are more flexible and capable of modeling complex non-linear relationships between features and house prices. They consist of layers of connected neurons that automatically learn feature interactions through backpropagation, making them effective for datasets like house pricing and the

multitude of house features, such as interactions between house characteristics like neighborhood quality, lot size, and amenities. However, this model requires lots of computation time with large preprocessing, longer training times, and the need to tune hyperparameters. Furthermore, they are less interpretable compared to linear regression and regression trees, making it challenging to explain the impact of individual features.

Regression trees are able to naturally handle both quantitative and qualitative features without requiring much preprocessing and provide insights into feature importance. However, they are prone to overfitting, particularly if the tree depth is too large, capturing noise in the data. They are less efficient with high-dimensional data compared to linear regression and may require techniques like random-forest to improve stability and predictive performance.

Next Steps

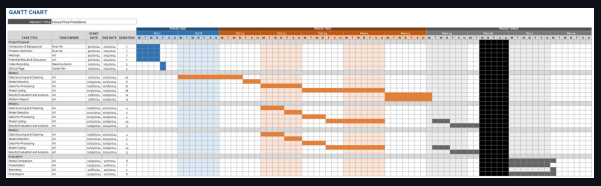
From our research, we can improve our current models, each specifically:

- *Linear Regression* - use higher order terms for certain features, especially if the data is non-linear.
- *Regression Tree* - instead of a decision tree or single tree, using a random forest or regression forest can give us more robust predictions as well as prevent overfitting.
- *Neural Network* - tune hyperparameters (as well for all models).

There are other models to utilize as well, such as SVM (Support Vector Machine), and possibly combine models together.

Furthermore, because our models and research focus on predicting a human quantifiable topic in real life, with house pricing and favorable features, creating an interface that can visually display information to non-technical users would be very helpful for them in deciding on a house.

Contributions



Name	Final Contributions
Ryan He	Data and summary validation, continued work on linear regression model, Video (Preprocessing, Linear regression), built slide deck
Maximus Genio	Regression tree, Video (introduction, table of contents, next steps, regression tree)
Daniel Pan	Streamlit page, data review and summary, put video together, discussed model comparisons
Joseph Yoo	Data and summary validation, built out neural network model, Video (Neural Network)

References

[1] Q. Truong, M. Nguyen, H. Dang, and B. Mei, “Housing Price Prediction via Improved Machine Learning Techniques,” Procedia Computer Science, vol. 174, pp. 433–442, 2020, doi: https://doi.org/10.1016/j.procs.2020.06.111.

[2] P.-Y. Wang, C.-T. Chen, J.-W. Su, T.-Y. Wang, and S.-H. Huang, “Deep Learning Model for House Price Prediction Using Heterogeneous Data Analysis Along With Joint Self-Attention Mechanism,” IEEE Access, vol. 9, pp. 55244–55259, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3071306.

[3] J. Manasa, R. Gupta, and N. S. Narahari, “Machine Learning based Predicting House Prices using Regression Techniques,” IEEE Xplore, Mar. 01, 2020. https://ieeexplore.ieee.org/abstract/document/9074952/ (accessed Aug. 09, 2021).