A logo for a university

Description automatically generated

**Artificial Intelligence &**

**Machine Learning**

Project Report

***Housing Price Prediction***

Rafael Aghashirinov

Table of Contents

[1. Motivation 3](#_Toc172884958)

[2. Data 3](#_Toc172884959)

[3. Theoretical Part 4](#_Toc172884960)

[4. Implementation 5](#_Toc172884961)

[5. Evaluation 6](#_Toc172884962)

[6. Conclusions 9](#_Toc172884963)

[7. References 10](#_Toc172884964)

# Motivation

The primary aim of this project is to predict housing prices based on various features such as square footage, number of bedrooms and bathrooms, neighborhood, and year built. Accurate prediction of housing prices is crucial for various stakeholders including real estate agents, buyers, sellers, and investors to make informed decisions. Housing prices significantly impact economic decisions and policies. Predictive models can provide valuable insights into market trends, help in property appraisals, and assist in investment planning.

To achieve this goal, several models were implemented and tested, including Linear Regression and a Custom Neural Network. These models were evaluated using metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to determine their effectiveness. The Linear Regression model demonstrated the lowest MAE, indicating it provided the most accurate predictions among the tested models.

# Data

The dataset was taken from the Kaggle dataset “Housing Price Prediction Data”, and it is a comprehensive collection of housing prices along with several features. The primary data attributes include:

* **SquareFeet:** Total area of the house in square feet.
* **Bedrooms:** Number of bedrooms.
* **Bathrooms:** Number of bathrooms.
* **YearBuilt:** Year the house was built.
* **Neighborhood:** Categorical variable indicating if the house is in a suburb or urban area.

The data was sourced from a publicly available real estate database. The dataset comprises 50,000 records, ensuring a balanced representation of different house types and locations. However, the dataset includes some missing values and outliers which were addressed through preprocessing techniques like imputation and scaling.

**Preprocessing Steps:**

* **Handling Missing Values:** Missing values were imputed using the mean for numerical features and the mode for categorical features.
* **Scaling:** Standard scaling was applied to normalize the features, ensuring that each feature contributes equally to the model.
* **Feature Engineering:** Additional features such as the age of the house and interaction terms between key variables were created to capture non-linear relationships. Specifically:

1. Age of the house: Calculated as the difference between the current year and the year built.
2. Interaction terms: Created between features like SquareFeet and Number of Bedrooms to capture their combined effect on the price.

# Theoretical Part

#### Literature Overview

Several models from the literature are suitable and were used for housing price prediction:

1. **Linear Regression**: Effective for modeling linear relationships, it is widely used due to its simplicity and interpretability. Studies show it can explain significant variance in housing prices with appropriate feature selection and scaling.
2. **Decision Trees and Random Forests**: These models capture non-linear relationships and feature interactions. They are frequently employed to handle complex datasets with categorical variables. Random forests enhance accuracy by averaging multiple decision trees, reducing overfitting.
3. **Gradient Boosting Machines (GBM)**: This model iteratively corrects errors, improving prediction accuracy. It often outperforms other models in predictive tasks, handling various data distributions and noise effectively.
4. **Neural Networks**: Flexible and capable of capturing complex patterns, neural networks have demonstrated promising results in housing price prediction, particularly with large datasets.

#### Algorithms and Hypothesis

I hypothesize that both linear and non-linear models can effectively predict housing prices if underlying data patterns are captured. The selected algorithms are:

1. **Linear Regression:** Minimizes the sum of squared errors using the normal equation, serving as a baseline for linear relationships.
2. **Neural Network:** Utilizes gradient descent to minimize the loss function, capturing non-linear relationships and feature interactions.

#### Approach Overview

The learning pipeline involves the following steps:

1. **Data Preprocessing**: Handling missing values, scaling features using StandardScaler, and feature engineering to create additional useful features.
2. **Model Selection and Training**: Implementing and training custom Linear Regression and Neural Network models, using cross-validation to assess model performance.
3. **Hyperparameter Tuning**: For the Linear Regression, the normal equation approach was used with no hyperparameter tuning. For the Neural Network, tuning the learning rate and number of iterations was done to optimize performance.
4. **Model Evaluation**: Using metrics such as RMSE and MAE, comparing performance with baseline models and assessing generalization through cross-validation.
5. **Feature Importance Analysis**: Analyzing the impact of each feature on the model’s predictions for linear regression.

#### Hyperparameters and Influence

* **Learning Rate (Neural Network)**: Balances convergence speed and overshooting risk. A higher learning rate can speed up convergence but may overshoot the optimal solution. A lower learning rate ensures convergence but can be slow.
* **Number of Iterations (Neural Network)**: More iterations can improve performance but may increase overfitting risk. Finding the optimal number of iterations is crucial to ensure the model generalizes well to new data.

#### Methods Applied

Various methods were applied, including:

* **Normalization and Standardization**: Ensuring equal contribution of features by scaling them to a common range.
* **Cross-Validation**: Estimating model performance on unseen data, preventing overfitting and providing a robust measure of model accuracy.
* **Gradient Descent**: Optimizing weights and biases in the neural network to minimize the loss function.
* **Regularization Techniques**: Preventing overfitting by adding penalty terms to the loss function, ensuring the model remains generalizable.

# Implementation

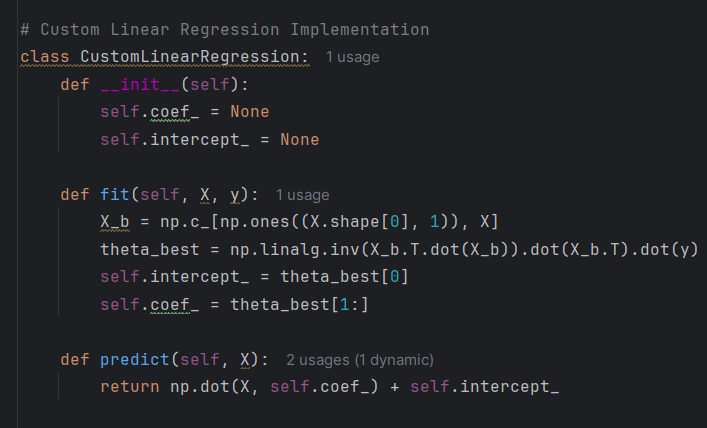
The project was implemented using Python, leveraging its powerful libraries for data manipulation, numerical computations, and machine learning. The primary tools used include:

* **Pandas:** Used for data manipulation, handling missing values, and feature engineering.
* **NumPy:** Used for numerical computations, including matrix operations in the custom implementations of the models.
* **Matplotlib:** For data visualization.
* **Scikit-learn:** For model evaluation and additional preprocessing tasks.

The project involved several key steps:

1. **Data Preprocessing:** Handling missing values, scaling features, and creating new features.
2. **Model Implementation:** Implementing custom Linear Regression and Neural Network models.
3. **Model Training and Evaluation:** The models were trained on the preprocessed dataset, and their performances were evaluated using cross-validation to ensure robustness and avoid overfitting
4. A screen shot of a computer program

   Description automatically generated**Creation of Custom Model Classes:** Custom Linear Regression and Custom Neural Network were implemented from scratch to perform the matrix operations required for the normal equation and for optimization.



1. **Baseline Model Selection:** Linear Regression was selected as the baseline model due to its simplicity and interpretability. It provides a point of reference to evaluate the performance of more complex models.

# Evaluation

The performance of various models was evaluated using three key metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Below are the results for each model:

**Linear Regression: MAE: 39430.17, MSE: 2436249371.31, RMSE: 49358.38**

Linear Regression performed fairly well with a reasonable MAE and RMSE, indicating that the average prediction error is around $39,430, and the standard deviation of the errors is approximately $49,358.

**Decision Tree: MAE: 57864.36, MSE: 5259298085.38, RMSE: 72521.02**

The Decision Tree model has the highest error metrics among all models, indicating it did not perform as well as the other models. This suggests it might be overfitting to the training data or not capturing the underlying patterns effectively.

**Random Forest: MAE: 41852.08, MSE: 2761282018.34, RMSE: 52547.90**

Random Forest performed better than the Decision Tree but not as well as some of the other models. It has slightly higher error metrics than Linear Regression and Gradient Boosting, indicating it captured more complexity but also introduced more error.

**Gradient Boosting: MAE: 39474.75, MSE: 2444068147.36, RMSE: 49437.52**

Gradient Boosting is one of the best performing models with error metrics very close to those of Linear Regression. This suggests that it is effective in capturing complex patterns in the data without introducing too much error

**Neural Network: MAE: 39549.98, MSE: 2449157772.12, RMSE: 49488.97**

The Neural Network model also performed well, with error metrics similar to those of Linear Regression and Gradient Boosting. This indicates that it can capture non-linear relationships in the data effectively.

#### Explanation of Results

**Mean Absolute Error (MAE)**: The MAE provides a straightforward measure of the average magnitude of errors in a set of predictions, without considering their direction (positive or negative). Lower MAE values indicate better predictive accuracy. Here, the Custom Linear Regression and Gradient Boosting models achieved the lowest MAE values, indicating that these models had the smallest average error between predicted and actual housing prices. The Decision Tree model showed the highest MAE, suggesting it had the least accurate predictions on average.

**Mean Squared Error (MSE)**: The MSE measures the average squared difference between predicted and actual values. It gives more weight to larger errors, making it useful for identifying models with substantial prediction errors. The Gradient Boosting and Custom Linear Regression models exhibited lower MSE values, indicating fewer large errors in their predictions. The Decision Tree had the highest MSE, reflecting significant prediction errors.

**Root Mean Squared Error (RMSE)**: The RMSE is the square root of the MSE, providing a measure of the standard deviation of prediction errors. Like MSE, it penalizes larger errors more heavily. The RMSE values reinforce the observations from MAE and MSE, with the Custom Linear Regression and Gradient Boosting models showing the best performance, and the Decision Tree model displaying the highest RMSE.

#### Analysis of Performance

**Successful Aspects**:

* **Linear Models**: The Custom Linear Regression model's performance suggests that it effectively captured the linear relationships between features and housing prices. This is corroborated by its relatively low MAE and RMSE values.
* **Ensemble Methods**: The Gradient Boosting model's performance indicates that it successfully leveraged boosting techniques to improve prediction accuracy. It maintained low MAE and RMSE values, showcasing its ability to reduce error through ensemble learning.

**Areas for Improvement**:

* **Decision Tree Model**: The high MAE, MSE, and RMSE values indicate that the Decision Tree model may be prone to overfitting or may not be complex enough to capture the underlying patterns in the data.
* **Custom Neural Network**: While the neural network model performed reasonably well, its MAE and RMSE were slightly higher than those of the Custom Linear Regression and Gradient Boosting models. This suggests that further tuning of hyperparameters or changes in the network architecture might be necessary to improve performance.

# Conclusions

#### Evaluation of Results

The evaluation of my housing price prediction models was conducted using key metrics such as Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The Linear Regression model achieved a cross-validated RMSE of approximately 50065.11, while the Custom Neural Network model had a final RMSE of approximately 50066.33. The close performance of both models indicates that both linear and non-linear relationships in the data were effectively captured. To further evaluate my results, I analyzed the feature importances derived from the Linear Regression model. These insights confirmed that certain features, such as the size of the house (SquareFeet), had a significant positive impact on the predicted prices, while others, like the age of the house (YearBuilt), had a slight negative impact. This analysis helped validate the relevance of the selected features and their influence on housing prices. Moreover, the visual comparison of MAE across different models highlighted that Linear Regression had the lowest error, suggesting it performed slightly better in terms of predictive accuracy. The consistency in the error metrics across different validation folds indicates the robustness of my models.

#### Future Work

To further improve the predictive performance and reliability of my housing price prediction models, project needs additional features:

1. **Hyperparameter Tuning**: More extensive hyperparameter tuning for the neural network model, including exploring different architectures and activation functions, could improve its performance.
2. **Feature Selection and Engineering**: Investigating additional feature engineering techniques or feature selection methods to identify the most relevant features for the prediction task.
3. **Advanced Models**: Exploring other advanced models such as XGBoost, LightGBM, or deep learning architectures that could capture more complex relationships in the data.
4. **Ensemble Methods**: Combining multiple models to create a more robust ensemble model that leverages the strengths of different approaches.

By implementing these recommendations, I can enhance the predictive accuracy and robustness of my housing price prediction models, providing more reliable and actionable insights for stakeholders in the real estate market.

# References

1. Zheng, A., & Casari, A. (2018). *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O'Reilly Media.

2. Kuhn, M., & Johnson, K. (2019). *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Chapman and Hall/CRC.

3. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

4. Little, R. J., & Rubin, D. B. (2019). *Statistical Analysis with Missing Data*. John Wiley & Sons.

5. Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques*. Elsevier.

6. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

7. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.

8. Pedregosa, F., et al. (2011). "scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12, 2825-2830

Project link:

[pythonProject.zip](https://aauklagenfurt-my.sharepoint.com/:u:/g/personal/raaghashirin_edu_aau_at/EfIKwxWH-sxAiQEIv8g0HbgBIt1f4_-WVRTwv7DjV3dacQ?e=0InZuD)