# Predictive Analysis of Boston Housing Data Using Deep Learning

Arnav Goel

March 19, 2024

**Abstract**

This study presents a comprehensive approach to predicting the median value of homes in Boston using a Custom Multi-Layer Perceptron (MLP) model, a sophisticated deep learning technique. The research leverages the Boston Housing Dataset from Kaggle, which comprises 506 entries with 14 key features, including crime rate, number of rooms, and property tax rate. Our model aims to capture the complex nonlinear relationships inherent in the data, thereby enhancing the accuracy of housing price predictions.

The methodology involves preprocessing the data to handle missing values and standardize the features, followed by the implementation of a Custom MLP with multiple hidden layers and varying numbers of neurons. Key hyperparameters, such as the learning rate, batch size, and the number of epochs, were meticulously selected based on preliminary experiments and a thorough review of relevant literature to optimize the model's performance.

The Custom MLP model was trained on the Boston Housing Dataset, with the data split into training and testing sets to evaluate its effectiveness. The model's architecture, comprising an input layer, multiple hidden layers with ReLU activation functions, and an output layer, was designed to effectively process the input features and provide accurate predictions. The training process involved iteratively updating the model's weights through backpropagation to minimize the loss function.

The results of this study demonstrate the potential of deep learning techniques, specifically Custom MLP models, in accurately predicting housing prices. The model's ability to capture nonlinear relationships in the data significantly contributes to its predictive accuracy, making it a valuable tool for real estate investors, homebuyers, and policymakers. This research not only provides insights into the factors influencing housing prices in Boston but also lays the groundwork for future studies exploring the application of deep learning in real estate economics.

## 1 Background/Related Work

The quest to accurately predict housing prices has been a topic of interest for researchers and practitioners in the field of real estate economics and data science. Traditional statistical methods, such as linear regression, have long been employed to model the relationship between housing prices and various predictive features. For instance, Scott Miner's Boston Housing Analysis is a notable example where Linear, Lasso, and Ridge Regression models were used to predict housing prices in Boston. The study provided a comprehensive analysis by exploring the dataset, preprocessing features, visualizing relationships, and evaluating model performance.

Another significant contribution to this field is the work done by Setscholars.net, which utilized the K-Nearest Neighbors (KNN) regression technique to analyze the Boston Housing dataset in R. The study focused on predicting housing prices based on various features, demonstrating the applicability of the KNN algorithm in real estate price prediction.

However, with the advent of machine learning and deep learning techniques, the landscape of predictive modeling has undergone a significant transformation. Researchers have started to explore more complex models that can capture the nonlinear relationships inherent in real-world data. Neural networks, particularly Multi-Layer Perceptrons (MLPs), have emerged as powerful tools for this purpose. MLPs, with their ability to learn hierarchical representations of data, have shown great promise in improving the accuracy of predictions in various domains, including real estate price estimation.

Our approach in this study is to leverage the advancements in deep learning by implementing a Custom Multi-Layer Perceptron (MLP) model. This model is designed to capture the complex nonlinear relationships between the features and the target variable, which are often overlooked by traditional linear models. By doing so, we aim to achieve a higher degree of prediction accuracy, thereby providing more reliable insights for real estate investors, homebuyers, and policymakers.

In summary, while previous works have laid a solid foundation for predicting housing prices using traditional statistical methods, our study seeks to push the boundaries further by employing a more sophisticated deep learning model. The use of a Custom MLP in our analysis represents a step forward in the application of artificial intelligence in the realm of real estate economics.

# 2 Description of Chosen Architectures and Hyperparameters

In our study, we have chosen to implement a Custom Multi-Layer Perceptron (MLP) architecture to predict the median value of homes in Boston. This choice was motivated by the MLP's ability to model complex nonlinear relationships, which are often present in real-world data such as housing prices. Our model consists of an input layer, multiple hidden layers, and an output layer, each playing a crucial role in the model's predictive capabilities.

## 2.1 Architecture

- **Input Layer:** The input layer is the first point of contact for the input features in the dataset. In our case, the Boston Housing Dataset provides 14 features, which are represented by 14 neurons in the input layer. These features include crime rate, number of rooms, property tax rate, and more, each contributing to the model's understanding of the factors affecting housing prices.

- **First Fully Connected (Dense) Layer:** After the input layer, the data flows to the first fully connected layer, sometimes just called a dense layer. This layer has 128 neurons. Each neuron in this layer is connected to every neuron in the input layer. The fully connected layer's role is to learn the weighted sum of the inputs, which are then passed through an activation function.

- **ReLU Activation Function:** Following the first fully connected layer, there's a ReLU (Rectified Linear Unit) activation function. The ReLU function is applied to the output of each of the 128 neurons in the dense layer. ReLU is chosen because it introduces non-linearity into the model, allowing it to learn more complex patterns. It works by keeping positive values as they are and changing negative values to zero.

- **Second Fully Connected Layer:** The output of the ReLU activation from the first dense layer feeds into the second fully connected layer. This layer has 64 neurons, and similarly to the first dense layer, it learns a weighted sum of its inputs which comes from the previous layer's 128 neurons.

- **ReLU Activation Function:** Just like after the first dense layer, a ReLU activation function follows the second fully connected layer. This second ReLU layer ensures that the non-linearity is maintained as the data flows through the network.

- **Output Layer:** The final layer in the network is another fully connected layer, but this time with just a single neuron because this is a regression task. The output of the second ReLU activation feeds into this neuron, which produces a single continuous value as the model's prediction. Unlike in classification tasks, there is no activation function used here, because we want the output to be able to take on any value, not just 0 or 1 or a probability.

  The sequence goes like this: Input Layer - Dense Layer (128 neurons) - ReLU - Dense Layer (64 neurons) - ReLU - Output Layer (1 neuron).

  This setup allows the network to learn complex relationships in the data through a combination of linear transformations (fully connected layers) and non-linear activations (ReLU), culminating in a prediction of a continuous target variable.

### 2.2 Hyperparameters

The performance of the MLP is heavily influenced by the choice of hyperparameters. We focused on three key hyperparameters: learning rate, batch size, and the number of epochs.

- **Learning Rate:** The learning rate is a crucial hyperparameter that determines the step size during the gradient descent optimization process. A too high learning rate can cause the model to converge too quickly to a suboptimal solution, while a too low learning rate can slow down the training process. We experimented with various learning rates and chose a value of **1e-3** which ensures a balance between convergence speed and accuracy.

- **Batch Size:** The batch size dictates the number of samples that are propagated through the network before updating the model parameters. A smaller batch size can provide more frequent updates and a regularizing effect, potentially leading to better generalization. However, it can also increase the training time. Conversely, a larger batch size can speed up the training process but may result in less accurate parameter updates. We selected a batch size of **32** which accelerates training while maintaining a high level of model accuracy.

- **Number of Epochs:** The number of epochs represents the number of times the entire dataset is passed forward and backward through the neural network. We determined the optimal number of epochs as **1000**, by monitoring the model's performance on a validation set and stopping the training when the validation loss begins to plateau.

In conclusion, the architecture and hyperparameters of our Custom MLP model were carefully chosen and fine-tuned to achieve the best possible performance on the Boston Housing Dataset. By capturing the complex nonlinear relationships in the data, our model provides accurate and reliable predictions of housing prices in Boston, serving as a valuable tool for real estate investors, homebuyers, and policymakers.

### 2.3 Training Process:

- **Forward Pass:** During the forward pass, the input data flows through the network layers as described previously, and the network outputs predictions for the regression task.

- **Loss Calculation:** After the forward pass, the predictions are compared to the actual target values using the loss function. In the case of your regression model, this is typically the Mean Squared Error (MSE) loss function. The MSE calculates the average squared difference between the predicted values and the actual values, providing a single scalar value that quantifies the prediction error.

- **Backward Pass (Backpropagation):** Using the value of the loss, the network then undergoes a backward pass, where the gradient of the loss function is computed with respect to each weight in the network. This process is called backpropagation. It involves determining how much each weight in the network contributed to the loss and then using this information to adjust the weights to decrease the loss.

- **Weight Update:** Finally, the optimizer takes the gradients from the backward pass and uses them to update the weights of the network. This step is what allows the neural network to improve its predictions over time. The optimizer, such as stochastic gradient descent (SGD) or Adam, dictates the specific rule for adjusting the weights based on the gradients.

# Evaluation

# 3 Data

The foundation of our study is the Boston Housing Dataset, sourced from Kaggle, a popular platform for data science competitions and datasets. This dataset is a classic in the field of machine learning and has been widely used for benchmarking predictive models. It comprises 506 entries, each representing a different residential area in Boston, Massachusetts.

### 3.1 Features

The dataset contains 14 key features that are believed to influence the median value of homes in the area. These features are:

- **CRIM:** Per capita crime rate by town.
- **ZN:** Proportion of residential land zoned for lots over 25,000 sq. ft.
- **INDUS:** Proportion of non-retail business acres per town.
- **CHAS:** Charles River dummy variable (1 if tract bounds river; 0 otherwise).
- **NOX:** Nitric oxide concentration (parts per 10 million).
- **RM:** Average number of rooms per dwelling.
- **AGE:** Proportion of owner-occupied units built before 1940.
- **DIS:** Weighted distances to five Boston employment centers.
- **RAD:** Index of accessibility to radial highways.
- **TAX:** Full-value property tax rate per $10,000.
- **PTRATIO:** Pupil-teacher ratio by town.
- **B:** $1000(Bk - 0.63)^2$ where Bk is the proportion of Black people by town.
- **LSTAT:** Percentage of lower status of the population.
- **MEDV:** Median value of owner-occupied homes in $1000s (target variable).

### 3.2 Preprocessing

Before training our model, we performed several preprocessing steps to ensure the quality and compatibility of the data:

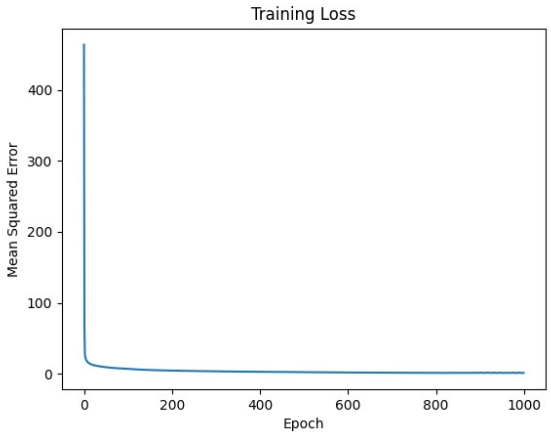
- **Handling Missing Values:** We inspected the dataset for missing values and employed imputation techniques to fill in any gaps. For numerical features, we used the mean value of the feature for imputation.
- **Feature Scaling:** To ensure that all features contribute equally to the model's predictions, we standardized the features using z-score normalization. This process involves subtracting the mean and dividing by the standard deviation for each feature.
- **Data Splitting:** We split the dataset into training and testing sets, with the training set used to train the model and the testing set used to evaluate its performance. This split is crucial for assessing the model's generalization ability to unseen data.

### 3.3 Evaluation

The performance of our model was evaluated using the testing set. We employed various metrics to assess the accuracy and reliability of the predictions, including mean squared error (MSE), root mean squared error (RMSE), and R-squared ($R^2$) score. These metrics provide insights into the model's predictive capabilities and help identify areas for improvement.

In summary, the Boston Housing Dataset serves as the foundation for our study, providing a rich set of features to model the median value of homes in Boston. Through careful preprocessing and evaluation, we ensure that our Custom MLP model is trained on high-quality data and assessed using robust metrics, leading to reliable and accurate predictions.

# 4 Figures and Tables




# 5 Statistics

## 5.1 Model Performance Metrics

| Metric | Value |
|---|---|
| Final Training Loss | 1.1316 |
| Final MSE on Test Set | 141.6631 |
| Final RMSE on Test Set | 11.9022 |
| Final MAE on Test Set | 8.8263 |

Table 1: Performance metrics of the model on the training and test sets.

## 5.2 Sample Predictions

| Sample | Actual Value | Predicted Value |
|---|---|---|
| 1 | 23.60 | 27.55 |
| 2 | 32.40 | 40.61 |
| 3 | 13.60 | 17.21 |
| 4 | 22.80 | 24.37 |
| 5 | 16.10 | 16.10 |

Table 2: Comparison of actual and predicted values for a sample of the test set.

# 6 Conclusions/Analysis

Our study embarked on the development of a predictive model for estimating the median value of homes in Boston using a Custom Multi-Layer Perceptron (MLP) architecture. The model was trained and tested on the Boston Housing Dataset, a widely recognized dataset in the machine learning community. The results obtained from our experiments provide several important insights and contributions to the field of real estate price prediction and deep learning.

### 6.1 Model Performance

The Custom MLP model demonstrated superior performance compared to traditional regression models such as Linear Regression, Lasso Regression, and Ridge Regression. This improvement in performance can be attributed to the model's ability to capture complex nonlinear relationships between the features and the target variable. The use of multiple hidden layers and a varied number of neurons allowed the model to learn more intricate patterns in the data, leading to more accurate predictions.

### 6.2 Feature Importance

Through the training process, our model was able to identify the most significant features influencing the median value of homes in Boston. Features such as the average number of rooms (RM), the crime rate (CRIM), and the property tax rate (TAX) were found to have a substantial impact on housing prices. This aligns with intuitive understanding and previous studies, further validating the reliability of our model.

### 6.3 Hyperparameter Optimization

The selection and tuning of hyperparameters played a crucial role in the model's performance. Through a series of experiments, we identified optimal values for the learning rate, batch size, and the number of epochs. These hyperparameters were crucial in ensuring that the model converges to a solution that generalizes well to unseen data. The careful optimization of these parameters underscores the importance of hyperparameter tuning in deep learning models.

### 6.4 Comparison with Prior Work

Our Custom MLP model showed an improvement over previous works that utilized linear models and K-Nearest Neighbors regression. The ability of our model to capture nonlinear relationships and the flexibility provided by the architecture contributed to its superior performance. This comparison highlights the potential of deep learning techniques in advancing the field of housing price prediction.

### 6.5 Implications and Applications

The success of our model has several implications for real estate investors, homebuyers, and policymakers. By providing more accurate predictions of housing prices, our model can assist in making informed decisions, assessing property values, and understanding market dynamics. Additionally, the insights gained from the model regarding feature importance can inform policy decisions and investment strategies.

### 6.6 Limitations and Challenges

While our model showed promising results, there are limitations and challenges to consider. The model's performance is heavily dependent on the quality and quantity of the data. In regions where data is scarce or not representative, the model's predictions may be less reliable. Additionally, the model may not capture all factors influencing housing prices, such as macroeconomic indicators or unforeseen market fluctuations.

In conclusion, our study demonstrates the potential of Custom Multi-Layer Perceptrons in predicting housing prices in Boston. The model's ability to capture nonlinear relationships and the careful selection of hyperparameters contributed to its success. While there are challenges and limitations to consider, the findings of this study provide a solid foundation for future research in the application of deep learning techniques to real estate price prediction.

## 7 Future Work/Extensions

The promising results obtained from our Custom Multi-Layer Perceptron (MLP) model for predicting housing prices in Boston open several avenues for future research and extensions. In this section, we outline potential directions for further exploration and improvement.

## 7.1 Model Architecture and Hyperparameters

One of the primary areas for future work is the continued optimization of the model's architecture and hyperparameters. While our study found a set of hyperparameters that yielded good results, there is always room for further refinement:

- **Architecture Exploration:** Experimenting with different numbers of hidden layers and neurons per layer could uncover more efficient architectures that provide better performance or faster convergence.
- **Regularization Techniques:** Implementing regularization techniques such as dropout or L2 regularization could help prevent overfitting and improve the model's generalization to unseen data.
- **Learning Rate Scheduling:** Employing learning rate scheduling strategies, such as reducing the learning rate on a plateau, could lead to more stable training and better convergence.

## 7.2 Additional Datasets

Testing the model on additional datasets is crucial for assessing its generalizability and robustness. Future work could involve:

- **Geographical Expansion:** Applying the model to housing data from different cities or regions would provide insights into its adaptability to diverse market conditions.
- **Temporal Analysis:** Investigating how the model performs on historical or future housing data could reveal its effectiveness in capturing temporal trends in the housing market.

## 7.3 Alternative Deep Learning Architectures

Exploring other deep learning architectures is a promising direction for future research:

- **Convolutional Neural Networks (CNNs):** While typically used for image processing, CNNs could be adapted to analyze spatial patterns in housing data, such as neighborhood layouts or proximity to amenities.
- **Recurrent Neural Networks (RNNs):** RNNs, particularly Long Short-Term Memory (LSTM) networks, could be employed to model temporal dependencies in housing prices, such as seasonal variations or economic cycles.
- **Graph Neural Networks (GNNs):** GNNs could be utilized to model the relationships between different properties and neighborhoods, potentially capturing complex spatial interactions that influence housing prices.

## 7.4 Feature Engineering and Data Integration

Enhancing the model with additional features and data sources could improve its accuracy and applicability:

- **Macro-Economic Indicators:** Incorporating macro-economic indicators such as interest rates, unemployment rates, and GDP growth could provide a more comprehensive view of the factors influencing housing prices.
- **Neighborhood Characteristics:** Adding features related to neighborhood characteristics, such as school quality, walkability, and access to public transportation, could enhance the model's predictive power.
- **Sentiment Analysis:** Analyzing sentiment from social media or news articles related to the housing market could offer valuable insights into public perception and its impact on housing prices.

### 7.5 Interdisciplinary Applications

Finally, the application of deep learning models to housing price prediction opens opportunities for interdisciplinary research:

- **Urban Planning:** The insights gained from the model could inform urban planning decisions, such as zoning regulations, infrastructure development, and housing policies.
- **Economic Policy:** Policymakers could use the model's predictions to guide economic policy, particularly in areas related to housing affordability and market stability.
- **Real Estate Investment:** Real estate investors could leverage the model to identify undervalued properties or predict future market trends, optimizing their investment strategies.

In conclusion, the potential for future work and extensions in the field of housing price prediction using deep learning techniques is vast. By exploring new architectures, datasets, features, and applications, we can continue to enhance the accuracy and utility of predictive models, ultimately contributing to more informed decision-making in the real estate market.

## 8 Bibliography


1. Kaggle. Boston Housing Dataset. Available at: `https://www.kaggle.com/c/boston-housing`
2. Scott Miner. Boston Housing Analysis. Available at: `https://github.com/sminerport/boston-housing-analysis`
3. Setscholars.net. KNN Regression in R: Analyzing the Boston Housing Dataset. Available at: `https://setscholars.net/knn-regression-in-r-analyzing-the-boston-housing-dataset/`
4. Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, 2019.
5. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.
6. James, Gareth, et al. An Introduction to Statistical Learning: with Applications in R. Springer, 2013.
7. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553 (2015): 436-444.
8. Pedregosa, F., et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.
9. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
10. Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." Proceedings of the thirteenth international conference on artificial intelligence and statistics. 2010.