Introduction

The goal of the project is to predict house rent prices based on several features based on house characteristics. Additionally, it is to evaluate the effectiveness of different machine learning models, specifically a neural network (NN) and a logistic regression (LR) model, and compare their performance using various metrics

Analysis

The dataset consists of house rent prices along with various features that influence rent. These are: BHK: Number of bedrooms, Size: House size in square feet, Area Type: Urban or rural areas, City: Location of the property, Furnishing Status: Whether the house is furnished, Tenant Preferred: The tenant types preferred by the landlord, Bathroom: Number of bathrooms
The rent prices (target variable) are in Indian Rupees. After loading the data, the features were split into continuous and categorical variables for further processing. There were additional columns (like date posted & point of contact) that were removed due to irrelevance.

Methods

Before training the models, the data was split into training (80%) and testing (20%) sets. The continuous features were standardized using a StandardScaler, while the categorical features were transformed using one-hot encoding. The make_column_transformer was used to automate this transformation for both training and testing datasets. A Keras sequential neural network model was constructed with the following layers:

- Input layer: Matching the number of input features.
- Three dense layers: These layers had 16, 8, and 4 neurons, each activated by ReLU and regularized with L1 and L2 penalties.
- Output layer: A single neuron for predicting rent.

The model was compiled using the Adam optimizer, mean_squared_error as the loss function, and mean_absolute_error as a performance metric. It was trained for 20 epochs with a batch size of 64 for the model's regularization. The model's architecture allowed it to learn complex patterns in the data. I intensively tested many combinations of layer sizes, activations, regularizations, etc and this was the best combination found.

For comparison, a logistic regression model was also implemented using the standard fit, predict, and analyze, with the same preprocessing as the NN. This model serves as a baseline, allowing us to understand how a simpler algorithm performs in comparison to the neural network.

Results:

For the results I chose to use the Mean Absolute Error and Root Mean Squared Error metrics to determine the model's performance since the predicted values of rent were continuous. The results for the neural network (NN) and logistic regression (LR) models on both training and testing data are as follows:

Metric	Train NN	Test NN	Train LR	Test LR
MAE	21,938.51	21,086.92	15,647.81	14,353.20
RMSE	77,410.24	57,362.51	70,946.04	43,970.82

From the metrics, the logistic regression model outperforms the neural network in terms of all evaluation metrics. The logistic regression model achieved lower MAE and RMSE on both the training and testing datasets. This suggests that the simpler linear model is more effective for this particular dataset. The neural network, while able to learn from the data, exhibited higher errors and could be prone to overfitting, especially since the gap between the training and testing errors is not significantly large. However, the neural network model may benefit from further tuning that I may have overlooked, such as adjusting the architecture or hyperparameters like dropout rates, regularization strengths, or epochs. However, from the various combinations of dropout rates, number of neurons per layer, additional layers and regularizers, these were giving the best consistently low metrics.

Reflection:

One of the conceptual things I learned from this project is the importance of starting with simpler models like logistic regression before moving on to more complex models like neural networks. Although neural networks can capture more intricate relationships in the data, they require careful tuning and often more data to generalize well. In this case, logistic regression, being a more interpretable and simpler model, achieved better performance with this relatively small dataset.