Project Report

Neural Network Model for House Price Prediction

Alex Yogesh

A report submitted in part fulfilment of the degree of

Program: BCA

Supervisor: Ms. Jyoti

Department
School of Engineering & Sciences
GD Goenka University, Gurgaon

26th April, 2025

Declaration by the Student

I hereby grant permission to GD Goenka University to publish my submitted project/research work for academic purposes. I understand that the published copy may be accessible to the public through the university's library, digital repositories, and other relevant platforms. This consent encompasses both print and electronic formats.

By signing this consent form, I give my consent to publish my dissertations as described below:

Student Name: Alex Yogesh

Student Affiliation: GD Goenka University

Title of work: Neural Network Model for House Price Prediction

Date: 26th April 2025

Place: GD Goenka University

Student Signature:

Declaration from Supervisor

I hereby grant permission to GD Goenka University to publish my submitted dissertations//research papers/academic content written/supervised/guided/co-authored for academic purpose. I understand that the published copy may be accessible to the public through the university's library, digital repositories, and other relevant platforms. This consent encompasses both print and electronic formats.

By signing this consent form, I give my consent to publish my work as described below:

Faculty Name: Ms. Jyoti

Faculty Affiliation: GD Goenka University

Title of work: Neural Network Model for House Price Prediction

Date: 26th April 2025

Place: GD Goenka University

Faculty Signature:

Acknowledgement

I would like to express my deepest gratitude to everyone who contributed to the successful completion of this project.

First and foremost, I extend my sincere thanks to my supervisor, **Ms. Jyoti**, for her consistent guidance, encouragement, and constructive feedback throughout the development of this work. Her insights were instrumental in shaping this project from the ground up.

I also wish to acknowledge the support of my mentor, whose valuable suggestions and technical advice helped in refining the implementation, especially in transitioning the machine learning model to a fully functional web application.

Special thanks to the open-source community — particularly the developers of **TensorFlow, Keras, Scikit-learn, Pandas, Flask**, and other essential libraries. Their contributions and well-documented resources made it possible to understand, train, and deploy a real-world machine learning model effectively.

I am also thankful to the platforms and resources that provided access to real estate data for Delhi and Gurugram, which allowed this project to move beyond theory and into practical application.

Finally, I would like to thank my family and friends for their constant motivation and support throughout this journey. Their encouragement kept me inspired to learn, build, and complete this project with dedication.

Thank you all!

Table of Contents

Abstract

The prediction of house prices plays a vital role in the real estate industry, directly impacting buyers, sellers, investors, and financial institutions. This project presents a practical implementation of a house price prediction system using a real-world dataset sourced from properties in **Delhi and Gurugram**. Unlike traditional academic datasets, this project emphasizes **regional market relevance and user accessibility**.

The machine learning model is built using **TensorFlow and Keras**, with preprocessing handled by **Pandas** and **Scikit-learn**. Key features such as area, number of bedrooms and bathrooms, property age, type, location, and city are processed using one-hot encoding and normalized using a **StandardScaler**. A neural network is then trained to capture the complex nonlinear relationships between these features and property prices.

Beyond model development, this project also includes the creation of a **Flask-based web application**, allowing users to input property details and instantly receive an estimated price. The application features a user-friendly interface and demonstrates real-time prediction capabilities, bringing machine learning closer to real-world usage.

The model's performance is evaluated using **Mean Absolute Error (MAE)** and shows strong generalization on unseen data. This project illustrates the potential of deploying machine learning models in real estate scenarios and highlights the value of integrating data science with web development for building intelligent applications.

Future enhancements may include increasing dataset diversity, integrating location intelligence (e.g., maps, coordinates), and deploying the app to the cloud for broader public access. This work demonstrates a complete end-to-end solution — from data preprocessing and model training to user interaction — bridging the gap between data science and practical deployment.

Project Specification

Objective
The primary goal of this project is to build and deploy a deep learning model capable of predicting **house prices in Delhi and Gurugram** using real-world property data. In addition to model training, the project focuses on delivering a **web-based application** using Flask to allow users to input property details and obtain live price estimates.

Technical Specifications

- **Programming Language:** Python 3.11
- **Development Environment:** Visual Studio Code
- **Libraries Used:**
  - **TensorFlow & Keras**: Model building and training
  - **Scikit-learn**: Data preprocessing, scaling, and evaluation
  - **Pandas & NumPy**: Data handling and manipulation
  - **Flask**: Web application framework
  - **Joblib**: Saving and loading scaler
  - *(Optional: Matplotlib for exploratory analysis during dataset creation)*

Hardware Requirements

- **Recommended System**: Laptop/PC with minimum **Intel i5 9th Gen**, **8GB RAM**, and **GTX 1650 GPU**
- **GPU Usage**: Not mandatory — model trains comfortably on CPU for the dataset size
- **OS Compatibility**: Windows 10/11 (tested), also compatible with macOS and Linux

Installation Instructions

1. **Install Python 3.11 and VS Code**
2. **Create a virtual environment**:
   `python -m venv .venv`
3. **Activate the virtual environment**:
   - Windows: `.\.venv\Scripts\activate`
   - macOS/Linux: `source .venv/bin/activate`
4. **Install dependencies**:
   `pip install tensorflow pandas numpy scikit-learn flask joblib`

Project Workflow

- **Dataset Preparation**
  - Real estate data collected from property listings in **Delhi and Gurugram**
  - Features include: `Area_sqft`, `Bedrooms`, `Bathrooms`, `Age`, `Property Type`, `Location`, and `City`
  - Applied **one-hot encoding** to handle categorical variables
  - Used **StandardScaler** to normalize numeric features
- **Model Development**
  - Developed a **Neural Network** using TensorFlow & Keras
  - Model architecture includes multiple dense layers with ReLU activation
  - Used `Mean Squared Error (MSE)` as loss and `Mean Absolute Error (MAE)` as a performance metric
  - Performed **train-test split** for evaluation
- **Model Saving and Inference**
  - Trained model saved as `.keras` file
  - Scaler saved with **Joblib**
  - **Feature column structure saved** to ensure consistent input format during inference

- **Web Application Integration**
  - Built using **Flask**
  - Includes a user interface with input fields for all required features
    Displays predicted price in a styled HTML result page
  - Routes:
    - `/` → Input form
    - `/predict` → Processes form, scales input, returns price
- **User Interaction**
  - Inputs property details
  - Instantly receives estimated price
  - Allows multiple predictions in one session

Chapter 1: **Introduction**

The ability to accurately predict house prices is a critical component in real estate analytics, impacting decision-making for buyers, sellers, developers, and financial institutions. Traditional valuation approaches — often based on human intuition or basic regression models — can fall short in capturing the non-linear, dynamic nature of real estate markets, especially in growing urban areas like **Delhi and Gurugram**.

With the rise of **machine learning and deep learning**, artificial neural networks now offer a sophisticated alternative, capable of learning complex patterns in large datasets and delivering highly accurate predictions. This project leverages such techniques to develop a house price prediction system that goes beyond theoretical models and delivers a **fully functional, interactive web application**.

The project utilizes a **real-world dataset** sourced from property listings across Delhi and Gurugram, encompassing key features such as area (sqft), number of bedrooms and bathrooms, property age, location, city, and type (house, apartment, villa). After preprocessing and normalizing the data, a deep neural network is trained using **TensorFlow and Keras** to model the relationship between these features and the target variable — **Price (INR)**.

Why This Project Stands Out

🔹 **Use of Real Data**
Unlike many academic projects that use the Boston dataset, this project is based on **actual listings from the Indian real estate market**, making its predictions more meaningful and relevant.
🔹 **Web-Based Deployment**
The model is not just theoretical — it's integrated into a **Flask-based web application** that allows users to input property details and receive real-time price estimates.
🔹 **Deep Learning Model**
A **multi-layer neural network** is designed to capture complex relationships between features. It offers improved performance over traditional linear regression models.
🔹 **Full Feature Pipeline**
Includes **one-hot encoding, feature scaling**, model saving, and column alignment to ensure consistent predictions even in production environments.
🔹 **Practical Use Case**
The final application can be useful for home buyers, realtors, and analysts looking to get quick, data-driven property evaluations.

By combining machine learning with modern web technologies, this project delivers a complete end-to-end solution — from data collection and model training to deployment and user interaction. It demonstrates the powerful role artificial intelligence can play in transforming the real estate sector.

Chapter 2: **Research Gaps**

Despite significant advancements in machine learning for real estate prediction, there remain critical gaps that this project aims to address through practical implementation and modern techniques:

🔹 **Overreliance on Academic or Outdated Datasets**
Many studies in this domain still use small or outdated datasets such as the Boston Housing dataset. While useful for learning, they lack real-world complexity. This project overcomes that limitation by using a **custom dataset based on real property listings from Delhi and Gurugram**, offering more relevant insights for local housing markets.

🔹 **Underutilization of Deep Learning for Real Estate**
Traditional models like linear regression or decision trees dominate most implementations. Deep learning — despite its ability to model complex, non-linear interactions — is often dismissed due to fears of overfitting or computational cost. This project demonstrates that with proper tuning and preprocessing, **neural networks can be applied effectively to real-world real estate data**.

🔹 **Lack of End-to-End Deployment in Prior Work**
Most existing models stop at prediction accuracy without addressing how such models could be delivered to end-users. This project **bridges that gap by deploying the trained model through a Flask web application**, enabling real-time interaction with predictions based on user input.

🔹 **Missing Scaler and Feature Alignment Handling**
Previous studies often overlook the technical challenges of applying a trained model to new data — especially when working with encoded features. This project implements **persistent scaler saving**, **column alignment**, and a complete preprocessing pipeline, ensuring that the model remains usable across sessions and environments.

🔹 **Limited Focus on User Experience**
Many house price prediction tools lack user-friendly interfaces. This project incorporates a clean, responsive **web UI** that enhances accessibility and demonstrates how machine learning can be productized for end-users.

🔹 **Evaluation Limited to MAE/MSE**
Traditional metrics such as Mean Absolute Error (MAE) or Mean Squared Error (MSE) are often used in isolation. This project expands on that by also considering the model's **R-squared score**, enabling a more holistic understanding of its predictive capability.

By addressing these key research gaps, the project not only improves predictive accuracy but also ensures practical usability and relevance — making it a more complete solution for modern real estate analytics.

Chapter 3: **Project Objectives**

The primary objective of this project is to develop a **deep learning-based house price prediction system** using real-world property data from **Delhi and Gurugram**, and to make it accessible to users through a **web-based application**. The project combines artificial intelligence and web development to deliver a practical, scalable tool for real estate valuation.

🔹 Specific Objectives

1. Develop a Neural Network Model
- Design and implement a multi-layered neural network using **TensorFlow and Keras**
- Optimize the model through **hyperparameter tuning** for improved performance and reduced error

2. Leverage Real-World Data
- Use a **custom dataset** based on real listings from Delhi and Gurugram to ensure the model reflects real market behavior
- Include key property features such as area, number of bedrooms and bathrooms, property type, age, location, and city

3. Improve Data Preprocessing
- Normalize numeric features using **StandardScaler** for better training convergence
- Apply **one-hot encoding** to categorical variables
- Save the preprocessing pipeline (including the scaler and expected columns) for consistent predictions

4. Build a Web Application
- Create a **Flask-based web interface** for easy user interaction
- Allow users to input property details and receive instant predicted prices in INR
- Design a clean and user-friendly front-end for seamless interaction

5. Enable Model Reusability and Deployment
- Save the trained model in `.keras` format
- Load and use the model for fast predictions without re-training
- Ensure column alignment between training and prediction data

6. Evaluate Model Performance
- Measure prediction accuracy using **Mean Absolute Error (MAE)** and **R-squared score**
- Validate the model with unseen test data to assess generalization

7. Highlight Feature Relevance
- Identify which features (e.g., area, location, property type) have the most impact on predictions
- Ensure the input form reflects those important features for accurate results

By achieving these objectives, this project not only enhances prediction accuracy using deep learning, but also delivers a real-time, user-friendly tool that bridges the gap between machine learning and real-world application in the real estate domain.

Chapter 4: **Project Hypothesis**

This project is based on the hypothesis that **deep learning models**, particularly **artificial neural networks (ANNs)**, can deliver highly accurate house price predictions by learning complex patterns from real-world property data. Additionally, it posits that such models can be successfully deployed in a **web-based environment** for practical use in the real estate industry.

Primary Hypothesis

A well-structured and optimized neural network model, trained on real estate data from Delhi and Gurugram, will outperform traditional regression-based models in predicting house prices and can be effectively deployed as a web application for end users.

Supporting Hypotheses

1. Real Data Enhances Generalization
Using actual property listings from Delhi and Gurugram, with detailed features like property type, location, and number of rooms, enables the model to capture real market trends more effectively than academic datasets.

2. Feature Scaling Improves Learning Efficiency
Applying scaling techniques such as **StandardScaler** on numerical attributes improves model convergence and training stability.

3. Hyperparameter Tuning Impacts Model Performance
Adjusting neural network parameters — including the number of hidden layers, learning rate, and activation functions — significantly boosts model accuracy and generalization.

4. Neural Networks Handle Complex Relationships
Unlike linear regression, ANNs can capture **non-linear feature interactions**, such as the impact of location on price depending on other factors (like property type or area), resulting in more robust predictions.

5. Model Persistence Enables Practical Usage
By saving the trained model and preprocessing components (like scalers and column templates), the solution becomes reusable and scalable, suitable for integration into live systems without the need for re-training.

6. Deployment Enhances Accessibility
Deploying the model in a **Flask web application** allows real users to input property details and receive real-time price estimates, showcasing the practical viability of deep learning in real estate applications.

By testing these hypotheses, this study validates the utility of **deep learning models not just in theory, but in real-world deployment**, marking a step forward in applying AI to real estate analytics.

Chapter 5: **Research Methodology**

This chapter outlines the systematic approach followed in this project, detailing the processes involved in data collection, preprocessing, model development, training, evaluation, and deployment. The methodology ensures an end-to-end real-world implementation — from raw dataset to an interactive web application.

5.1 Data Collection

The dataset used in this project is a custom real estate dataset sourced from public property listings in **Delhi and Gurugram**. It contains thousands of records with relevant features that influence house pricing, such as:

- **Area_sqft**: Size of the property in square feet
- **Bedrooms**: Number of bedrooms
- **Bathrooms**: Number of bathrooms
- **Age_of_Property**: Property age in years
- **Property_Type**: Type of property (Apartment, Villa, etc.)
- **Location**: Locality or neighborhood
- **City**: Delhi or Gurugram
- **Price_INR**: Target variable representing the property price in Indian Rupees

The dataset reflects current real estate market dynamics and serves as the foundation for building a regionally relevant predictive model.

5.2 Data Preprocessing

To ensure data quality and optimal model performance, several preprocessing steps are carried out:

🔹 **One-Hot Encoding:**

Categorical variables such as `Property_Type`, `Location`, and `City` are encoded using **one-hot encoding**, enabling the model to interpret non-numeric features.

🔹 **Feature Scaling:**

Numerical features like area, bedrooms, and bathrooms are scaled using **StandardScaler** to standardize the distribution and enhance training convergence.

🔹 **Column Alignment:**

The feature column structure used during training is saved to ensure that any new data passed to the model matches the trained schema exactly.

🔹 **Data Splitting:**

The dataset is split into training (80%) and testing (20%) sets using `train_test_split` from Scikit-learn to evaluate model performance on unseen data.

5.3 Model Development

A **deep learning model** is built using **TensorFlow and Keras**, following this architecture:

- **Input Layer**: Number of neurons matches the number of processed features
- **Hidden Layers**:
    - Layer 1: 128 neurons, ReLU activation
    - Layer 2: 64 neurons, ReLU activation
- **Output Layer**: 1 neuron for regression output (predicted price)

🔹 **Optimization:**

- **Optimizer**: Adam
- **Loss Function**: Mean Squared Error (MSE)
- **Metric**: Mean Absolute Error (MAE)

Hyperparameters such as batch size and epochs are tuned through experimentation.

5.4 Model Training

The model is trained for **100 epochs** with **validation data** to monitor performance and overfitting. The training process includes:

- Shuffling of training data
- Batch-wise updates to minimize the MSE loss
- Real-time evaluation on test data using MAE

The training history confirms steady convergence and reduced validation error across epochs.

5.5 Model Evaluation

Model performance is assessed using the following metrics:

- **Mean Absolute Error (MAE)**: Measures the average deviation from actual prices
- **R-Squared (R^2) Score**: Measures how well the model explains the variance in property prices.

🔹 **Visualization:**

Graphs such as actual vs. predicted price scatter plots are generated to visually confirm prediction accuracy.

5.6 Model Deployment

To ensure usability beyond development:

- The trained model is saved in `.keras` format
  The `StandardScaler` and column template are saved using **Joblib**
- These components are reused during inference, ensuring consistency

A **Flask-based web application** is developed to allow users to:

- Input property details
- Receive instant price estimates
- Interact with the model in real-time through a clean UI

Routes `/` and `/predict` handle form display and prediction logic respectively.

Chapter 6: **Results & Discussion**

This chapter presents the performance outcomes of the trained deep learning model and evaluates its effectiveness in predicting house prices for Delhi and Gurugram. The results are analyzed using standard regression metrics, followed by model comparisons, visual interpretation, and key insights.

6.1 Model Performance Metrics

After conducting several training iterations and fine-tuning the hyperparameters, the best-performing neural network model was selected based on its performance on the validation set.

Matrix	Value
MAE	₹26.7 lakhs
MSE	₹8.74×10^{14}
R^2	0.79

- **MAE (Mean Absolute Error)**: The model has an average error of about ₹26.7 lakhs, which is acceptable given the large price variations in the dataset.

- **MSE (Mean Squared Error)**: Reflects penalization of larger errors — useful for identifying outlier predictions.

- **R-squared (R^2)**: A score of **0.79** indicates that the model explains 79% of the variance in house prices, showcasing its strong generalization ability.

These results confirm that the model learns underlying market patterns effectively and delivers high-quality predictions on unseen property data.

6.2 Visualization of Results

A scatter plot of **actual vs. predicted** prices was generated to visually assess the model's performance:

- Most points align closely with the diagonal line (ideal predictions), indicating that the model is generally accurate.
- Minor deviations exist in the upper range (luxury properties), where the model slightly underpredicts due to lower representation of high-priced listings in the dataset.
- The plot highlights a **strong positive correlation**, reinforcing the model's reliability.

(Visualization was performed during experimentation but not included in the live Flask app for simplicity.)

6.3 Model Comparison

To validate the performance of the neural network, it was benchmarked against traditional regression models:

Model	MAE	R^2 Score
Linear Regression	₹35.4 lakhs	0.63
Decision Tree Regressor	₹29.1 lakhs	0.70
Neural Network (ours)	₹26.7 lakhs	0.79

- The **neural network outperformed** both Linear Regression and Decision Trees, particularly in generalization.
- It handled the **non-linear interactions** between features (e.g., the combined effect of location and property type) more effectively than traditional models.

6.4 Discussion on Findings

- The model's performance validates the potential of **deep learning for real estate price prediction**, especially with regional datasets.
- **Hyperparameter tuning** played a crucial role in optimizing accuracy and reducing training error.
- Despite strong results, there's room for improvement — especially in higher price ranges and for outlier properties.
- The successful **deployment through Flask** further demonstrates the real-world viability of the solution.

Key Insight: Neural networks offer not just higher accuracy but also flexibility for scaling, retraining, and deployment in production settings.

Future Scope

Expand the dataset with more listings across multiple cities to improve prediction diversity.

Integrate location-based intelligence (e.g., proximity to schools, metro stations, or amenities) to enhance prediction accuracy.

Deploy to cloud platforms (e.g., Render, Heroku, or AWS) for public access and real-time API integration.

Incorporate user feedback in future versions for personalized estimation.

Chapter 7: **Conclusion**

7.1 Summary of Findings

This project successfully developed and deployed a deep learning-based house price prediction model using real-world property data from **Delhi and Gurugram**. By combining artificial neural networks with essential data preprocessing steps, the model demonstrated strong performance in predicting house prices based on features such as area, number of bedrooms and bathrooms, location, property type, age, and city.

The neural network was trained using TensorFlow and Keras, with preprocessing powered by Pandas and Scikit-learn. StandardScaler normalization and one-hot encoding of categorical variables ensured model stability and accuracy. The trained model achieved competitive performance metrics including a low Mean Absolute Error (MAE) and a high R-squared (R^2) score, indicating strong generalization on unseen data.

The final solution was deployed using **Flask**, transforming the trained model into an **interactive web application** that allows users to input property details and instantly receive estimated prices.

7.2 Key Takeaways

- The use of **real estate data from Delhi and Gurugram** greatly enhanced the relevance and applicability of the model compared to traditional academic datasets.
- **Neural networks significantly outperformed** traditional regression models like Linear Regression and Decision Trees in prediction accuracy.
- **Preprocessing techniques** such as scaling and one-hot encoding were crucial for model convergence and stability.
- Saving the trained model, scaler, and expected column format made the solution **reusable and deployable** in real-world applications.
- Deployment via Flask enabled **user-friendly interaction**, making the system accessible even to non-technical users.

7.3 Limitations

While the project achieved substantial results, a few limitations were noted:

- The model had slightly **higher prediction error for luxury properties**, likely due to fewer samples in the upper price range.
- The dataset was focused only on Delhi and Gurugram. The model might not generalize well to other regions without retraining.
- Location-specific factors such as proximity to metro stations, malls, or school districts were not explicitly included and may improve accuracy.
- External influences like economic trends, inflation, or policy changes were not factored into the model.

7.4 Future Work

To enhance and scale this project, the following future developments are recommended:

- **Expand the dataset** across multiple Indian cities or states to improve generalizability.
- **Integrate location intelligence**, such as GPS coordinates, Google Maps APIs, or distance to amenities.
- **Deploy the application to the cloud** using services like Render, Heroku, or AWS for public access.
- **Develop a mobile-friendly version** of the app to enable on-the-go use by homebuyers, investors, and real estate agents.
- Explore **advanced deep learning models**, such as attention mechanisms or autoencoders, for more sophisticated feature interactions.
- Include **live data sources** (e.g., listing APIs) for real-time prediction updates.

Final Remarks

This project demonstrates a **complete, end-to-end implementation** of a real estate price prediction system — from dataset creation and model training to deployment and real-time prediction.

By combining **deep learning with web technologies**, the solution bridges the gap between data science and user-centered applications, offering a powerful tool for modern real estate analytics. With continued development and data integration, such systems have the potential to revolutionize how property valuation is done in India and beyond.

Bibliography

Belsley, D. A., Kuh, E., & Welsch, R. E. (1980). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. John Wiley & Sons.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep Learning*. *Nature*, 521(7553), 436–444.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Scikit-learn Developers. (2024). *Scikit-learn: Machine Learning in Python*. Retrieved from https://scikit-learn.org

TensorFlow Developers. (2024). *TensorFlow: An Open Source Machine Learning Framework*. Retrieved from https://www.tensorflow.org

Keras Developers. (2024). *Keras Documentation*. Retrieved from https://keras.io

Flask Developers. (2024). *Flask Documentation*. Retrieved from https://flask.palletsprojects.com

Pandas Developers. (2024). *Pandas Documentation*. Retrieved from https://pandas.pydata.org

Real Estate Data Source (2025). *Delhi and Gurugram Property Listings Dataset*. Self-curated from open listings and property platforms. *(Used for model training in this project.)*

Appendix: Code Implementation

The complete implementation of the trained model, data preprocessing pipeline, and Flask web application and datasets are available on GitHub:

🔗 https://github.com/alexstephen2025/Projects