Calibrated Supervised Learning for Housing Price Prediction: A Probabilistic Inference Approach

John Doe
Department of Computer Science
Western University
London, Canada
11111@uwo.ca

Christopher Betancur
Department of Computer Science
Western University
London, Canada
cbetancu@uwo.ca

Xiaowei Feng
Department of Computer Science
Western University
London, Canada
xfeng282@uwo.ca

***Abstract*—This report explores how supervised learning can be combined with uncertainty-aware postprocessing to predict housing prices more reliably. Using the California Housing Prices dataset from Kaggle [2], we first build a transparent baseline with linear regression on standardized features. We then train a feedforward neural network (ReLU, Adam optimizer [3]) to model nonlinear relationships that the linear model cannot capture. To quantify how confident the system is in its point forecasts, we apply conformal prediction [5] to produce calibrated prediction intervals that require minimal distributional assumptions. All models are implemented in Python using the Scikit-learn ecosystem [4]. The project illustrates a central CS3346 idea: accurate predictors are useful, but predictors that can also report well-calibrated uncertainty are more actionable.**

***Index Terms*—supervised learning, regression, Bayesian inference, calibration, uncertainty quantification**

I. Introduction

Artificial Intelligence (AI) systems learn patterns from data to approximate complex functions and make predictions about real-world quantities. Within the CS3346: *Artificial Intelligence* framework, this project falls under the module *Intelligence from Data*, which focuses on how machines learn from examples, generalize to unseen cases, and reason under uncertainty. The project demonstrates these ideas through a practical supervised learning task: predicting housing prices using real-world socioeconomic data.

The California Housing dataset [2] is a classic benchmark for regression analysis, containing 13 features that describe various neighborhood attributes such as average rooms per dwelling, air pollution levels, and proximity to employment centers. Predicting the median home value from these attributes illustrates key challenges in supervised learning, including bias–variance trade-offs, feature relevance, and model interpretability. Although this dataset is relatively small, it remains widely used for educational purposes due to its simplicity and clear variable relationships.

To evaluate different modeling strategies, we employ both linear and nonlinear methods. Linear regression provides an interpretable baseline, while regularized variants (Ridge and Lasso) improve stability by penalizing overfitting. For nonlinear modeling, we utilize neural network are applied to capture complex feature interactions and reduce variance. These models are implemented using the Scikit-learn library [4], which offers standardized APIs and reproducible workflows.

While point predictions are valuable, real-world decision-making often requires an understanding of uncertainty. Therefore, we extend the analysis with probabilistic techniques that estimate confidence around predictions. Specifically, conformal prediction [5] is used to construct calibrated intervals that guarantee a desired coverage probability without assuming specific data distributions. This aligns with the *probabilistic inference* component of CS3346, emphasizing that intelligent systems should not only predict but also measure their confidence.

Overall, this study aims to integrate core AI principles—learning from data, reasoning under uncertainty, and ensuring interpretability—within a compact yet meaningful application. The remainder of this paper is organized as follows: Section II outlines the modeling methodology; Section III describes implementation details; Section IV presents experimental results and discussion; and Section V concludes with key takeaways and future directions.

II. Methodology

A. Dataset

We used the publicly available *California Housing Prices* dataset from Kaggle [2]. Each instance corresponds to a California district and includes socioeconomic and geographic attributes such as median income, housing characteristics, latitude/longitude, and the target variable *median_house_value*, which we aim to predict. After loading the dataset, we removed rows with missing or clearly invalid entries to ensure that the downstream models were trained on clean data. Features (inputs) and the target (output) were then separated into $\mathbf{X}$ and y, respectively.

B. Feature Standardization

Most of the input attributes are continuous and measured on different scales (e.g., income vs. longitude). To prevent features with large numeric ranges from dominating the learning process, we standardized all continuous features using the z-score transformation:

$$x' = \frac{x - \mu}{\sigma}, \tag{1}$$

where μ and σ are the sample mean and standard deviation computed from the training data. The same statistics were applied to the validation and test splits to avoid data leakage.

C. Train–Validation–Test Split

To assess generalization fairly, we randomly partitioned the dataset into three disjoint subsets: 70% for training, 15% for validation, and 15% for testing. A fixed random seed was used so that the split is reproducible. The training set was used to fit model parameters; the validation set was used to monitor performance and to select hyperparameters (e.g., number of hidden layers, number of neurons, learning rate); and the test set was held out until the end for final reporting.

D. Baseline: Linear Regression

As a first model, we trained an ordinary least squares linear regressor on the standardized features to predict *median_house_value*. This baseline captures only linear relationships between the housing attributes and the target and serves as a reference point for judging whether a more expressive model is warranted.

E. Neural Network Regressor

Our main model was a feedforward neural network for regression. The network took the standardized feature vector as input, passed it through one or more fully connected hidden layers with ReLU activations, and produced a single scalar output representing the predicted house value. The network was trained to minimize the mean squared error (MSE) loss using the Adam optimizer [3]. Training was performed on the 70% training split, while the 15% validation split was used for early stopping and hyperparameter selection. All models were implemented in Python using the scikit-learn library [4].

F. Evaluation

After training, we evaluated both the linear regression baseline and the neural network on the held-out 15% test set. We reported standard regression metrics such as mean squared error (MSE) and mean absolute error (MAE) to determine whether the nonlinear neural network provided a meaningful improvement over the linear baseline.

III. Implementation

IV. Results and Evaluation

V. Conclusion

References

[1] D. Harrison and D. L. Rubinfeld, "Hedonic housing prices and the demand for clean air," *Journal of Environmental Economics and Management*, vol. 5, no. 1, pp. 81–102, 1978.

[2] F. Soriano, "California Housing Prices (Data + Extra Features)," Kaggle dataset, 2022. [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/california-housing-prices-data-extra-features

[3] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations (ICLR)*, San Diego, CA, USA, 2015.

[4] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] r13xc9
arXiv preprint arXiv:2107.07511, 2021.