# CSC-6605: Lab 4 Report

**Authors:** Ben Paulson, Sydney Balboni

**Professor:** Dr. Nowling

**Date:** March 10th, 2025

## Introduction

In this project, we developed a machine learning model for predicting house prices by analyzing data from the house\_price\_prediction\_service database. The dataset contained information on house sales, including physical attributes (bedrooms, bathrooms, square footage), location data, and neighborhood characteristics (schools, population). we implemented a comprehensive model development process, from exploratory data analysis to deployment-ready production models.

## Model Development and Selection

### Feature Engineering

We began by examining the raw data and creating derived features to improve prediction accuracy:

1. **Temporal features**: Extracted year, month, day of week, and day of year from sale dates
2. **Property characteristics**:
3. Age of house (year - yr\_built)
4. Renovation status (binary indicator)
5. Total square footage (living + lot)
6. Price per square foot
7. Bedroom to bathroom ratio
8. Total rooms (bedrooms + bathrooms)
9. **Neighborhood metrics**:
10. School density (schools per 10,000 people)
11. **Log transformations**:
12. Applied log transformations to price, square footage, and population to normalize distributions
13. **Categorical bins**:
14. Created age categories (New, Recent, Mid, Old, Very Old)
15. Created size categories (Tiny, Small, Medium, Large, Mansion)

The feature engineering process significantly improved model performance by capturing non-linear relationships and creating more informative representations of the raw data.

### Model Selection

We evaluated several regression models using time-series cross-validation to respect the temporal nature of the data:

A table with numbers and a number of figures

AI-generated content may be incorrect.

The Random Forest model showed the best performance by a significant margin. Given these results, we selected Random Forest for hyperparameter tuning.

### Hyperparameter Tuning

We performed grid search with time-series cross-validation to optimize the Random Forest model, exploring:

1. Number of estimators (trees): 50, 100, 200
2. Maximum depth: None, 10, 20, 30
3. Minimum samples to split: 2, 5, 10

The best parameters were:

1. n\_estimators: 200
2. max\_depth: 30
3. min\_samples\_split: 2

After tuning, the model achieved an RMSE of $16,711.64, a MAPE of 1.41%, and an R² of 0.9966 - a significant improvement over all other approaches.

### Evaluation and Visual Analysis

The figure below shows the Actual vs. Predicted values for the tuned Random Forest model:

A graph with a line and a red dot

AI-generated content may be incorrect.

The strong alignment along the diagonal indicates excellent predictive performance. The model captures the variance in house prices across the full range, with very few outliers.

Feature importance analysis revealed that location-based features (latitude, longitude), property size (sqft\_living), and school-related metrics were the most influential predictors - aligning with real estate domain knowledge.

## Pipeline Implementation

We implemented a scikit-learn pipeline to encapsulate the entire prediction workflow, from raw data to predictions. The pipeline includes:

1. Feature engineering (custom transformer)
2. Feature selection (dropping non-predictive columns)
3. Preprocessing (imputation, scaling, one-hot encoding)
4. Model prediction (Random Forest Regressor)

The complete pipeline structure is shown below:

A screenshot of a computer program

AI-generated content may be incorrect.

We verified that the pipeline matched the performance of my manually developed model from Part II:

1. **RMSE:** $16,999.67 (vs. $16,711.64 in Part II)
2. **MAPE:** 1.43% (vs. 1.41% in Part II)
3. **R²:** 0.9966 (identical to Part II)

This confirms that the pipeline correctly implements all preprocessing and modeling steps.

**Production Model**

For production deployment, we trained the final pipeline on the complete dataset (no train/test split) to leverage all available data. We saved the trained model using the dill library:

A close-up of a computer code

AI-generated content may be incorrect.

The saved model file (models/house\_price\_model\_20250312\_150411.pkl) contains the complete pipeline, including all preprocessing steps and the trained Random Forest model.

We verified the saved model by loading it and making a prediction:

A close-up of a text

AI-generated content may be incorrect.

**Conclusion**

The house price prediction model achieved excellent performance with an RMSE of approximately $17,000, a MAPE of 1.43%, and an R² of 0.9966. The Random Forest algorithm proved significantly better than linear approaches for this task, likely because it can capture complex non-linear relationships between features and house prices.

The scikit-learn pipeline provides a robust implementation that handles all preprocessing steps and ensures consistency between development and production. The saved model is ready for deployment in a production environment.