House Price Prediction Using Machine Learning: Hendricks, Hamilton, Tippecanoe of Indiana, United States

Joel Klein
Indiana University
joeklein@iu.edu

Jongwon John Lee
Indiana University
leejojo@iu.edu

Katherine Danielle Hartley
Indiana University
hartleyk@iu.edu

Nikhil Jaywant
Indiana University
njaywant@iu.edu

Siddhant Meshram
Indiana University
sidmeshr@iu.edu

Srikanth R Bolishetty
Indiana University
sbolishe@iu.edu

Abstract

The proposed article identifies factors such as property locations and characteristics that drive residential housing prices. We aim to gain insights from housing price models to use in economic development, urban planning, financial services, logistics, and industrial development. To do so, we developed a predictive model for housing prices that provides an understanding of the most important factors driving property prices. In order to do this, we leveraged data sources such as Sales Disclosure Forms (STATS Indiana), Geocoding (ArcGIS with the help of IU Polis Center), property characteristic features (Melissa Data - Intrinsic), and property location-based features (Niche.com Ratings, Google Places API, School Ratings - Extrinsic).

***Keywords*—** House Price Prediction, Random Forest, XGBoost, LightGBM, Gradient Boosting, CatBoost, Shapley Additive Explanations

1 Introduction

1.1 Objectives

Using six years of transaction housing sales data gathered by the Spring 2022 team, the Fall 2022 purpose is to identify the key factors that have predicted price differences for these residential properties over time. Key research questions are:

- How accurate can the team get in predicting residential housing prices?
- Are prices driven mostly by intrinsic residential characteristics such as square footage, bedrooms, bathrooms, etc.?
- How much of an impact does location have?
- How much of an impact do certain nearby non-residential properties such as transportation hubs, airports, and auto shops have on housing prices?

This research will guide future property development for residential uses. The insights help develop knowledge of pricing models used extensively in economic development, urban planning, financial services, logistics, and industrial development.

1.2 Scope

The scope of analysis for this project was restricted to single-family residential properties. Due to the limited features available from SDF (Sales Disclosure Form) data and the cost of purchasing housing characteristics data from a third party, the scope was further revised to focus on Hamilton, Hendricks, and Tippecanoe counties in the state of Indiana. With SDF data and Redfin housing listings data, we discovered the features that affect housing prices have different characteristics by county. Thus, the quality of the analysis will decrease if we keep the research scope as the state of Indiana. We decided to purchase carefully selected features of Melissa Data in the selected counties. Then, we will be adding features using SDF Data, Google Places API, and School District Data. We have acquired Redfin's housing listings data by counties which consists of basic features but the volume is relatively small.

The scope of analysis for this project was restricted to single-family residential properties sold between 2016 and 2020 in the state of Indiana. Due to the limited features available from SDF (Sales Disclosure Form) data and the cost of purchasing housing characteristics data from a third party, the scope was further revised to focus on Hamilton, Hendricks, and Tippecanoe counties in

the state of Indiana. With SDF data and Redfin housing listings data that were publicly available, we discovered the features that affect housing prices have different characteristics by county. A refined county scope enabled the Data Science team to dive deeper in features affecting housing prices in the three counties.

The homes are scoped from the SDF data as follows:

- Property class code: 510 (single family residential on platted)
- Counties: Hamilton, Hendricks, Tippecanoe (3 most populous candidates, also top in new constructions)
- Years: 2016-2020 (2015 might be a little dated)
- Sales Price > 0
- Assessed Value > 0
- For properties sold more than once, take the most recent sale in data (don't want to include the same house twice)
- log(Sales Price) between 2.5 and 97.5 percentile (removes homes of high and low values)
- log(Assessed Value) between 2.5 and 97.5 percentile (removes homes of high and low assessed values)

2 Background

Residential property prices are driven by a mix of property characteristics and property location. This project aims to identify the key factors driving property prices over time, with some emphasis on property location. Property sales disclosure records are required in 39 states, including Indiana. The Indiana University's Kelley School of Business Indiana Business Research Center is the custodian of the State of Indiana's sales disclosure records. Beginning in 1993, the Indiana legislature required a Sales Disclosure Form to be used by local governments to assist with the study of fair market value and determining the true tax value of properties in Indiana. In 2008, the legislature required a revision to the form that would allow taxpayers to use the form as an application for certain deductions, and the definition of "conveyance" document was revised to include documents for compulsory transactions and partitions of land.

Diverse methods ranging from regression models, tree-based algorithms, support vector regression, bagging-based approaches, and neural networks have been applied to house price prediction problems exploring a wide range of factors. Attempts were made to add versatility to the data set such as using natural language processed data and demographic information on top of standard features of the house and geographical data. Limsombunchai (Limsombunchai, 2004) and Hong, et al. (Hong and Kim, 2020) compare the traditional hedonic model for house price prediction with machine learning models such as artificial neural networks and Random Forest using house price data in New Zealand and South Korea. They conclude the hedonic model performs worse than their proposed approaches. Phan proposed that the combination of Stepwise and Support Vector Machine (SVM) showed the highest performance in predicting Melbourne City's house prices (Phan, 2018). Phan concluded the number of bedrooms, distance to the Central Business District (CBD), latitude and longitude-based location, and type of houses are the most critical factors. According to Macpherson and Silman, demographic factors such as the level of the Hispanic population, in comparison to other races and ethnicities, showed a positive correlation to house prices with house price data in Tampa and Orlando, Florida (Macpherson, 2001). Perez et al. compared algorithms such as Linear Regression, Regression Trees, Random Forest, and Bagging Regressor using the features of Colombia's house prices leveraged with natural language processing (J. I. Perez and Correa, 2020). They concluded that the Random Forest and Bagging Regression performed better than the regression trees and linear regression; they also found that area, the number of bathrooms, and the age of the property were relevant features. Ozdemir used housing data from Iowa and concluded CatBoost showed the highest performance among other methods such as Random Forest, Gradient Boosting, XGBoost, and Light GBM (Ozdemir, 2022). Ozdemir revealed among 80 factors; living area, overall quality, and neighborhood were the most important features for predicting house prices.

3 Data Description

3.1 Sales Disclosure Forms (SDF)

Indiana state law (IC 6-1.1-5.5) requires the filing of a sales disclosure form (SDF) whenever real property is sold. These data are used by assessors in the determination of the annual market-based adjustments of assessed property values. The last six years of property sale transaction data was extracted from Indiana's public data utility resources (Stats Indiana). Residential properties which were part of the scope were filtered from this dataset. The following features from the SDF were included in the modeling research: Parcel Number, Property Street, Property City, Property State, Property Zip, Conveyance Data, and Sales Price.

3.2 Redfin Data

The team also acquired Redfin's housing listings data by counties which consist of basic features but the volume is relatively small. This data was used to re-scope the project but not considered in scope for the final analysis.

3.3 Melissa Data

The team purchased carefully selected features of Melissa Data for the scoped homes in the selected counties. Melissa Data is a third-party data aggregator with an address database recognized by USPS. The database includes residential property characteristics (200+ features) data such as the number of bedrooms and bathrooms, building area, built year, and many more. The team shortlisted the housing characteristic features to approximately twenty based on EDA, SME, and Sponsor Feedback. The following features from Melissa data were included in the modeling research: Neighborhood Code, Plumbing Fixtures Count, Rate, Parking Space Count, Tax District, Pool Area, Buildings Count, Building Area, Shed Area, Lot Acres, Roof Material, Attic Area, Fireplace, Basement Area, Partial Bath Count, Property Use, Bedrooms Count, Year Built, Rooms Count, Property City, Stories Count, Property Zip, and Construction.

3.4 Niche.com

Additional subjective city rankings published by Niche.com were added to the dataset to adjust for location in predicting housing prices. These are indexes of more specific public data sources which are simple and can effectively improve model accuracy. The following features from the Niche.com rankings were included in the modeling research: Public School, Good for Families, Jobs Outdoor Activities, Nightlife, Diversity, Health & Fitness, and Commute.

3.5 ArcGIS (geocoding)

Geocodes were manually obtained for each property using arcGIS Pro. Geocodes are required to extract place features from the Google Places API. The latitude and longitude values were not included in the modeling exercise. 99.82% of homes had a match to a geocoded address. There are homes that either didn't have a geocode match or there was a tie in the match for address and returned default result.

3.6 Google Places API

Extrinsic, location-based features were extracted from the Google Places API. Features such as the number of places within a certain x-mile radius of a home, average place ratings for the respective place types, and the distance to the nearest airport were considered in evaluating the impact of location on price. 21 place types were identified to add to the analysis: Bars, Bus stations, Car repair shops, Churches, Doctors, Gas stations, Gyms, Hindu temples, Hospitals, Laundromats, Light rail stations, Mosques, Night clubs, Parks, Pharmacies, Shopping malls, Supermarkets, Synagogues, Train stations, Transit stations, and Airports.

4 Methods

After identifying the candidate features for building the predictive housing price model, several machine learning models were fitted and tested to identify which feature sets predict housing prices with the highest level of accuracy. The team trained models in an iterative approach adding specific feature sets to determine if the additional feature sets improved model performance. There were three sets of features used in training and evaluating the housing price model:

- Base model: SDF & Melissa Data Features
- Base with City Rankings model: SDF, Melissa, Niche.com Data Features
- Base with City Rankings & Places model: SDF, Melissa, Niche.com, Google Places Data Features

For each feature set, the input data and model target is preprocessed, and the model is tuned, trained, selected, evaluated, and interpreted via the following pipeline (Figure 1):

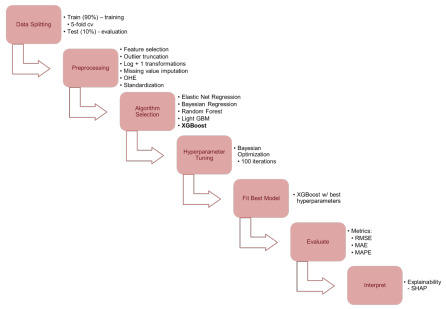

Figure 1: Methods

(details for each step are outlined below) The best model for each feature set is reported in the results section.

4.1 Data Splitting

The full data set is split such that 90% of the data is used for training and 10% of the data is used for testing. Data is split via stratified sampling across the target price variable. The purpose of the test set is to have an unbiased set of unseen data that the final model can be evaluated on to determine its performance results. The target price is binned into 50 equal-sized bins across its distribution and the sampling is stratified across these bins such that an equal representation of houses based on price is in the training and testing set.

The training data set is subsequently split such that 90% of the data is used for training and 10% of the data is used for validation. The purpose of the training set is to train the model while the validation set is for informing the best model selection and estimating evaluation.

4.2 Data Pre-Processing

Prior to model training, the raw features are transformed appropriately to assure the models can be effectively trained. There are several preprocessing steps applied to the raw data prior to fitting:

1. Target transformation: There is a heavy right skew with the target housing price variable (Figure 2).

   A log + 1 transformation is applied to the training data such that the loss function does not place more penalty on incorrect predictions on homes with a higher magnitude in price (Figure 3).

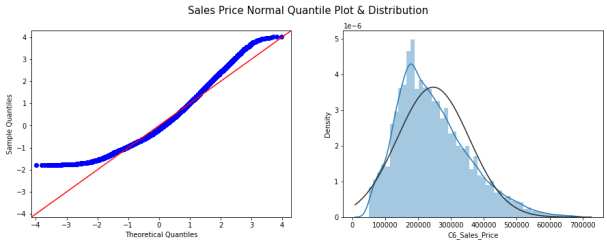

Figure 2: Distribution

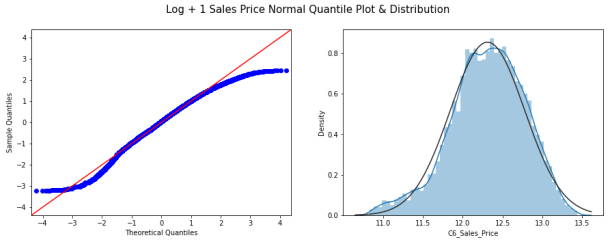

Figure 3: Log + 1 Distribution

2. Input Feature Selection: The input features are selected from the full training data set for each respective model below:
   - Base model: SDF & Melissa Data Features
   - Base with City Rankings model: SDF, Melissa, Niche.com Data Features
   - Base with City Rankings & Places model: SDF, Melissa, Niche.com, Google Places Data Features
3. Input Outlier Truncation: There are extreme outliers for the area lot size feature. Winsorization is applied to truncate the top 1% of values to the 99th percentile.
4. Input Missing Value Imputation: All missing values for numeric columns are imputed with the median. All missing values for categorical variables are imputed with "unknown". These decisions were elected because there are few missing values and this will not heavily affect the performance results of the fitted model. This approach should be revisited in the future if the model is used for a different purpose.
5. Input Log + 1 Transformations: All area/square footage features are heavily right-skewed, so a log + 1 transformation is applied to make these distributions more normal.

6. Input One Hot Encoding: Machine learning models can not handle categorical variables unless they are represented as a series of one hot encoded numerical vectors. This transformation is applied and the "unknown" category is dropped to avoid collinearity problems when fitting the model.
7. Input Standardization: To assure and interpret feature importances appropriately, all features must be transformed to be on the same scale prior to model training. All variables are standardized.

The resulting processed data set is passed into the training process.

4.3 Algorithm Selection

These are multiple steps during model training to determine the best model for predicting housing prices with the feature set. The first step selects the best candidate algorithm for further hyper-parameter tuning in the next step. There were 5 candidate algorithms considered:

- Linear Regression
- Bayesian Regression
- Random Forest (Ho, 1995)
- Light GBM (Ke et al., 2017)
- XGBoost (Chen and Guestrin, 2016)

It was initially expected that the XGBoost algorithm would generate the best evaluation results due to nonlinear relationships, interaction, collinearity, missingness, and outliers in the input features, and this was later confirmed (see results).

To identify the best model

1. Apply k-fold cross-validation with 5 folds to evaluate the tuning results. The algorithm with the lowest average Root Mean Squared Error (RMSE) across the folds is selected as the winner.
2. To identify the best model for each algorithm, hyper-parameter tuning is applied:
   (a) Set candidate hyperparameter ranges for each algorithm
   (b) Perform search using bayesian optimization with 10 iterations to identify best candidate hyperparameters for each algorithm across the range of candidates.
   (c) Bayesian optimization is a sequential search design strategy (using Bayes Theorem) for optimizing a loss function using results from previous search iterations. It is a State of the Art approach that eliminates the negative computationally intensive consequences of both random and grid search.
3. Select the best model for further hyper-parameter tuning.

4.4 Hyper-parameter Tuning

After the best algorithm is selected, a more detailed hyper-parameter search is performed using 50 iterations of bayesian optimization for identifying the best hyper-parameter combination for the selected algorithm. Hyper-parameter tuning is a computationally intensive process. Thus, the refined, detailed hyper-parameter search is applied only to one algorithm.

4.5 Fit Best Model

Once the best hyper-parameter combination is identified for the winning algorithm, the model is fit on the entire training data set and evaluated initially using the validation data set.

4.6 Evaluation

The model performance is evaluated using three common regression error functions:

- Root Mean Squared Error (RMSE): Square root of the sum of squared difference between the values that are fitted by the model, and actual price values that are divided by the number of historical points.
- Mean Absolute Error (MAE): average absolute difference between the values fitted by the model, and the actual price values.
- Mean Absolute Percentage Error (MAPE): average absolute percent difference between the values fitted by the model, and the actual price values.

These calculations are performed both before and after the predictions and the target price values are transformed back to the original actual price scale.

4.7 Interpretability

After the final model is identified, SHAP values (Shapley Additive Explanations) are computed from the tree-based model to interpret both

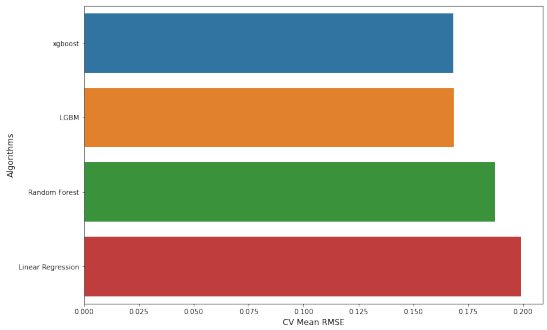

Figure 4: RMSE, MAE, MAPE

the global and local variable importances on the model predictions. SHAP is based on cooperative game theory and is a State of the Art model agnostic approach to explain machine learning model predictions and increase transparency and interpretability. SHAP computes the individual contribution, SHAP value, of each feature on the model prediction for each home.

5 Results

5.1 Best Model

The performance of the models with the CV Mean RMSE for the test set is shown below. In our case, the XGBoost algorithm has the lowest RMSE; thus it is the most accurate prediction among the selected models. We believe it is due to the data set's nonlinear relationships, interaction, collinearity, missingness, and outliers. XGBoost seems to be predicting rare cases better along with its generally high performance.

5.2 Model Performance

The modeling was performed on three different iterations of the dataset. The first model (Base) was a preliminary analysis to understand the base features of SDF and Melissa data. By evaluating using only Melissa data we can get a baseline for improvements. The second model (Base + Niche) incorporated the Niche school rating data. The final model included the google places API dataset with the base and Niche. By evaluating the model at each step, we could conclude that the additional features reduced our error and provided more precise results.

The following table shows the improvement in error metrics as new features were available for analysis:

The addition of Niche data improved the RMSE by 1.614% while adding the Places data improved

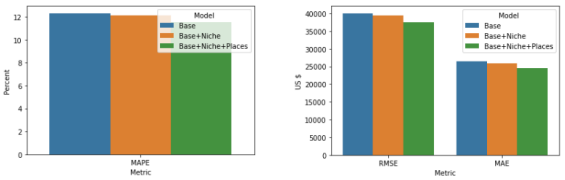

Figure 5: RMSE, MAE, MAPE

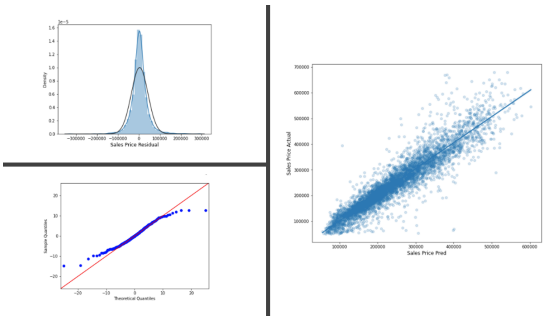

Figure 6: Base

RMSE by 4.802%. MAE also decreased by 5.164% with the Places data over a 2.051% improvement of Niche over Base. This concludes the features provided by the Places dataset is imperative to model accuracy. The MAPE score showed similar improvement over error as new features were added (1.786% with Niche, and 4.711% adding Places).

5.3 Evaluation of Predicted Values vs Actual

Base Model: It has a higher correlation at lower sales prices with deviation occurring at higher sales prices. This is highlighted in the QQ plot with high kurtosis. The density plot suggests the tailing is due to the concentration of data around the median values.

Base + Niche Model: Similarly to our base model, the predictive values had high correlation to actual median home prices with similar tailing to base model.

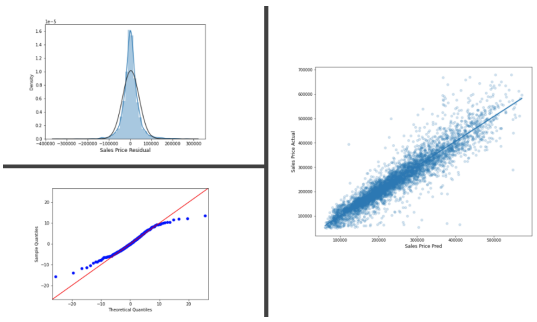

Figure 7: Base+Niche

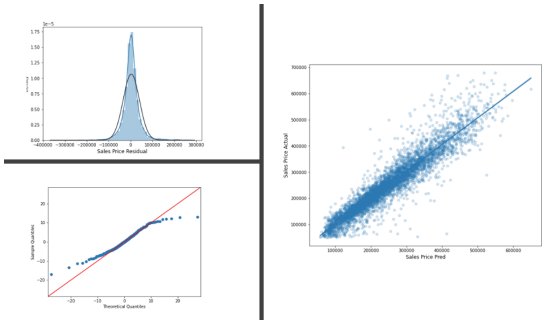

Figure 8: Base+Niche+Place

Base + Niche+ Places: Again, the final model had high correlation of predicted prices, particularly in the $100K to $300K home prices. The extreme home prices resulted in tailing on both ends of the distribution. The density plot suggests the tailing is due to the high concentration of data in the median range, and less data at the extremes.

SHAP Values: The most important features were mainly from the Melissa data such as the property size of area buildings. This shows that a larger property size would predict a larger sales price. As we go down the list of features descending in importance, we can also see that low values of Selected_YR_2020 had little effect on the predictive value, but the higher values showed high SHAP values. We can see that the impact of the features that were created to control the house price difference due to the inflation of the house prices (e. g. Selected_YR_2020) has less impact as we add more features. We can also see that proximity to health and fitness had an impact on model output.

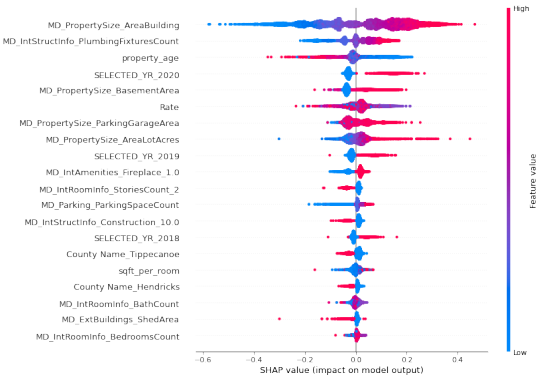

Figure 9: Beeswarm Plot by XGBoost - Base

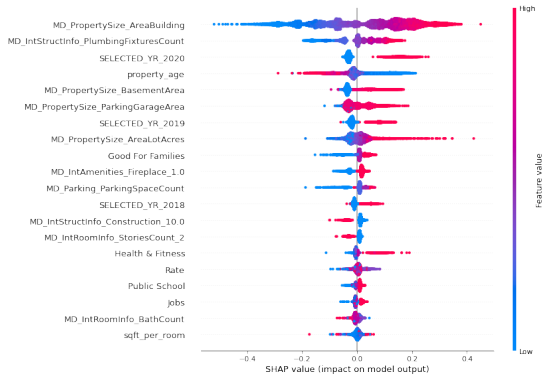

Figure 10: Beeswarm Plot by XGBoost - Base + Niche

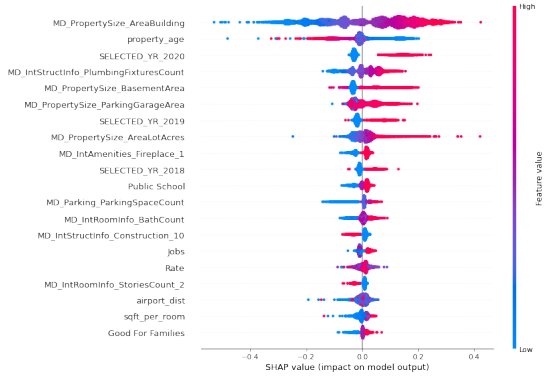

Figure 11: Beeswarm Plot by XGBoost - Base+Niche+Place

6 Conclusion and Discussion

Housing price prediction is a crucial part of the continuously growing real estate industry because stakeholders highly value accurate predictions. This project focuses on developing an accurate machine-learning model for predicting housing sales prices by experimenting with tree-based algorithms. To do so, we leveraged data sources from Sales Disclosure Forms, geocoding, property characteristic features, and property location-based features (Niche.com Ratings, Google Places API, School Rating). Our research revealed that XGBoost had the best performance among other considered models. Among 66 features, property size, property age, plumbing fixtures count, basement size, garage area, lot size, fire place, public school, parking space count, bath count, construction, jobs, rate, stories count, and airport distance were considered important to predict sales price of houses.

XGBoost algorithm performs best due to non-linear relationships, interaction, collinearity, miss-

ingness, and outliers. City rating data from Niche.com had a negligible effect in reducing overall error, but city rankings combined with location data improved performance. The model suffers from limited observations in higher price ranges. The model does not predict homes over 500,000 with acceptable accuracy. Purchasing more data will improve this drastic performance decline.

In the future, we recommend conducting experiments and methods such as rebalancing data (upsampling train data via SMOTE (Synthetic Minority Oversampling Technique)), further outlier analysis & removal, reducing property scope further, removing features with low importance & retraining models, adding additional location-based features such as proximity to shopping centers, entertainment, etc., purchasing more data (currently 30k sales records) for additional counties in the State of Indiana, and supplementing existing features with house image data. Moreover, acquiring funding to purchase more housing data and extend the predictive analysis to the entire state of Indiana is crucial. Some other considerations include identifying feature sets and indexes representing economic, mortgage rates, and housing conditions in the pandemic time period to effectively account for additional volatility in housing prices during this time frame.

References

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

Choi H. Hong, J. and W.-sung Kim. 2020. A house price valuation based on the random forest approach: the mass appraisal of residential property in south korea. 24(3):140–152.

F. Gonzalez J. I. Perez and J. C. Correa. 2020. Modeling of apartment prices in a colombian context from a machine learning approach with stable-important attributes. 87(22):63–72.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30:3146–3154.

V. Limsombunchai. 2004. House price prediction: Hedonic price model vs. artificial neural network. *New Zealand Agricultural and Resource Economics Society Conference*.

Sirmans G.S. Macpherson, D.A. 2001. Neighborhood diversity and house-price appreciation. 22:81–97.

Ozancan Ozdemir. 2022. House price prediction using machine learning: A case in iowa.

The Danh Phan. 2018. Housing price prediction using machine learning algorithms: The case of melbourne city, australia. In *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*, pages 35–42.