House Price Prediction using Catboost Regression and Genetic Algorithm for Feature Selection

Sreenivasan R S, Sunena Rose M V, Tarun S K, and Karumuri Hari

***Abstract*—With ever-increasing housing prices, there is a demand for developing an optimal method to predict housing prices. Traditional Price Prediction models are found to be sub-optimal in price prediction as they could not grasp the non-linear relationships that exist between variables. Gradient Boosting methods like the Catboost regression model can effectively represent the nonlinear relationships in housing price prediction. However, with an increase in the dimensionality of features, model interpretation becomes more cumbersome and computationally more expensive. Genetic Algorithms can discover relevant features and eliminate irrelevant ones during the evolutionary process, allowing feature selection on a subset of the data increasing model interpretability and training efficiency. A feature and housing value dataset is collected Kaggle for King County, Seattle. Four different machine learning algorithms such as Ridge Regression, XGBoost Regression, Catboost Regression model and the G-Catboost regression model which incorporates a genetic algorithm for feature selection to reduce model complexity. The data obtained is cleaned and an exploratory data analysis is done where the distribution of housing prices with respect to area, and the correlation between different variables with the price is observed. The top 20 features from the feature selection algorithm are taken into consideration and then fed to the CatBoost Regression algorithm. The results obtained show that the discussed G-Catboost algorithm performs better than the existing algorithms.**

***Index Terms*—Machine Learning; Genetic Algorithms; Catboost Regression; Housing Price Prediction;**

I. Introduction

THE demand for housing is growing annually, and price fluctuations in this market indicate how the nation's economy is doing. Precise real estate forecasting has gained significance for the country's economy and attracted the attention of numerous academics[1]. Different comparative works have been made in the field that has used regression prediction for housing prices, for example Hasan et al, [2] have used Artificial neural networks, Hedonic regression, and nearest neighbor regression for house price prediction There are a total of 3114 and 11 variables that are used, including the house's location, age, credit availability, size (square meters), number of rooms, number of bathrooms, floor, number of stories of the building, distance to the city center, and heating system. The information pertains to Adana's four districts: Sarıçam, Yüreğir, Çukurova, and Seyhan. By these standards, there is no multicollinearity between independent variables.The outcome shows that compared to hedonic regression models, the Artificial Neural Network performs better. Srirutchataboon et al,[3] have applied a stacked ensemble learning model to predict the house price in Thailand. The stacking ensemble model framework with Convolutional Neural Network for feature extraction, ensemble models such as random forest, XGboost, AdaBoost, and simple linear regression technique to improve the performance. T. Zhen et al,[4] have used three fitting algorithms: multiple linear regression, decision tree regression, and XGboost algorithm to predict the house price in Chengdu. A total of 27961 which includes house address, total house price, building area housing, and other 36 fields of information. The XGboost performs better than the other two which prevents overfitting and provides an accuracy of 0.9251. Xiuyan et al, [5] have used ensemble learning-based models for predicting house prices in Miami. The algorithms used are random forest and XGboost which produced the of 0.9251. Cheng et al, [6] have used an optimization model known as particle swarm optimization with the XGboost algorithm that can automatically enhance the model's parameter adjustment procedure. The dataset is derived from Kaggle with a sum of 1461 records including 79 features. The metric, RMSE of the PSO-XGBoost model reaches 0.9887 and 0.1105 respectively. Rafea et al, [7] on the other hand have used twelve possible models including CatBoostRegressor, RandomForestRegressor, and K-NeighborsRegressor, with evaluation metrics such as Mean Squared Error, Mean Absolute Error, and Root Mean Square Error. It was concluded that the CatBoost Regression model performed best

While these models propose an optimized framework for prediction, an increased complexity in the model is met for the task of prediction which makes the model less interpretable. Feature selection is an approach that reduces the number of features needed by selecting a subset of the original inputs. This technique is frequently employed since it minimizes unnecessary inputs while keeping the original input interpretations [8]. Many different feature selection approaches exist to reduce model complexity. A recursive feature removal technique was presented by Guyon et al.[8] for gene selection in microarray-based cancer classification issues. Genetic Algorithms are another set of methods that have been used to tackle the problem of feature selection due to their nature for searching an exponential search space and display superiority compared to representative classical algorithms.[9] Thus using a GA for feature selection in addition to a gradient boosting algorithm such as CatBoost can reduce the number of features required, and improve model interpretability.

II. Algorithm Construction

A. Catboost Regression

Catboost Regression is a popular gradient-boosting framework developed by Yandex. Unlike other gradient-boosting methods, the Catboost Regressor uses symmetric binary decision trees for base predictors and has native handling for

categorical features. The principle of how the learning parameter occurs in Catboost is as follows.

Assume we look at a dataset of samples, where is the target and $x_k = (x_k^1 ..... x_k^m)$is a random vector of m characteristics, and the solution to $x_k \epsilon R$ can be binary or numeric. Training a function $FR^m \rightarrow R$ to minimize the predicted loss is the aim of the learning problem. Here, (x, y) is a test example taken from a random distribution P apart from the training set D. L acts as a smooth loss function [10]

The gradient boosting procedure here builds an interactive sequence of approximations greedily.$F^t : R^m \rightarrow R, t = 0, 1, ...$. with F^t obtained by the earlier estimation of $F^(t-1)$ in an additive manner.

$$F^t = F^(t-1) + \alpha h^t$$

Here is the step size, and h^t is picked to minimize the loss function

$$h^t = min(L(F^(t-1)+h)) = min(EL(y, F^(t-1)(x)+h(x))$$

A functional gradient descent ascertains the solution. The step h_t is picked such that $h^t(x)$ approximates to $-g^t(x, y)$ where

$$g^t(x, y) = (\delta L(y, s))/\delta x|(s = F^(t-1)(x))$$

The least squares approximation is the most popular approximation technique that is used here. As pointed before, A decision tree serves as Catboost's main predictor. The decision tree partitions the feature space into distinct areas by utilizing the values of numerous splitting attributes. Splitting attributes are of a binary nature that can recognize features sx^k that are greater than a given threshold t. This can be stated as $a = I_(x^k) > tx^k$ either numerical or binary features. The estimate of the response y is located at the last node in the tree. Thus, a decision tree may be expressed as

$$h(x) = \sum_{n=1}^{J} b_j \prod_{xR_j}$$

where R_j represents the disjoint region of the leaves of the tree.[10]

One of the more popular methods in dealing with categorical data is using one-hot encoding of the categorical features, however, this method is rather ineffective as there is a loss of data captured and increases the dimensionality of the data. Thus CatBoost uses a target-based statistical method called OrderedTBS motivated by online learning algorithms. However, unlike them which uses sliding window principles to capture the ordering, the strategy evolved by Catboost involves proposing an artificial "time" for ordering in offline settings. Namely, a random permutation represented by of the dataset is taken and along with $D_k = x_j : \sigma(j) < \sigma(k)$ as the training example. $D_k = D$ is taken for the test where D_k is the dataset.[10]

B. Genetic Algorithms

Genetic algorithms are stochastic algorithms that mimic natural evolution. This algorithm stands out the most since it keeps a collection of solutions inside a population. It has a process that selects for more fit chromosomes with each generation, just like in biological evolution. The chosen chromosomes go through processes like crossover and mutation to mimic the process of evolution.[8]

Kudo and Sklansky [11] presented a comparison between Genetic Algorithms and Sequential Forward Floating Search (SFFS) and found that GA's were most optimal with $D50$ where D represents the dimensions.

A search-based GA was designed to implement the feature selection function. A string of D binary digits is considered representing the features. The values 0 and 1 indicate eliminated and selected features, respectively. To locate a solution in a huge search space, the population is updated by a generational approach.[8]

Initial Population:

The generation of the initial population is done using a random_uniform function of binary

values [0,1] to represent the inclusion or exclusion of features.

Feature Evaluation:

For each individual in the population, the fitness is evaluated using a fitness function. The size value d is used as a constraint to force a feature subset to meet the specified subset size criteria. Chromosomes that violate this constraint are penalized. The definition of chromosome C's fitness is

$$fitness(C) = Y(X_C) - penalty(X_C)$$

Where X_C is the feature subset of C, and the penalty is defined below with penalty coefficient, w .

$$penalty(X_C) - w * ||X_c| - d|$$

Selection:

Selection involves picking out the chromosomes based on fitness and one that has a higher probability of survival. The chromosomes in our design are selected by a simple roulette-wheel method of selection.a non-linear function, $P(i) = q(1-q)^(i-1)$ is used to assign a probability of selection to the i^{th} chromosome after the chromosomes are first sorted non-increasingly in terms of fitness. A higher q value results in a more intense selection pressure.

Crossover and Mutation:

A crossover operation involves generating a new chromosome out of the two parents, which the mutation operator may have perturbed.

The m-point crossover operator alternately replicates each segment from each of the two parents and selects m-cutting points at random. Due to the possibility of differing amounts of occurrences of 1 in the swapped gene segments, the crossover could produce offspring that do not meet the subset size criterion.

Mutation is done in such a manner that it does not violate the subset size requirement, which carefully changes the control of conversion from 1-0 and 0-1. The algorithm of mutation is denoted below

1) Let N0 and N1 denote the quantity of 0-bits and 1-bits in the chromosome.

2) $p_i = p_m; p_0 = p_m.n_1/n_0$;
3) Produce a random number r within [0,1]
4) if(andconvert g to 0; else if($g = 0$ and $r < p_0$ convert g to 1
5) For each gene g in the chromosome, Repeat 3

Parameters:

Parameters are used to control the implementation of the algorithm. Tuning the values, the suitable ones for the dataset are given below.

TABLE I

Parameter	**Value**
Population	10
Mutation Rate	0.6
Generation	20
Penalty Factort	0.5

C. G-CatBoost Regressor

The training process of the G-Catboost Regressor involves initializing the genetic algorithm for feature selection before moving ahead with training the actual model based on the reduced dataset. The G-Catboost Regressor model can be seen in Fig. 1, where the steps are as follows

a) Initialize a random population of feature chromosomes that would act as the initial parents.

b) Evaluate the fitness value of the parents using the fitness function which would undergo a further rank-based selection process using the roulette wheel selection method. Feature importances f_i are tracked for each generation.

c) Crossover is done between the feature subset together with a mutation process.
d) Offspring created is checked for uniqueness, and if not present then added to the list of unique offspring.

e) A population update is done where the unique offspring is taken as the current population, and the loop continues until the stopping conditions for generations are met

f) The average importance score is calculated for the feature subset using

$$averageimportance = (\sum_{i=1}^{g} f_i)/(g * p)$$

where g = generation, p = population_size

g) The features are ranked based on the average importance out of which a subset containing the top 30 most important features is taken to train the Catboost Regression model

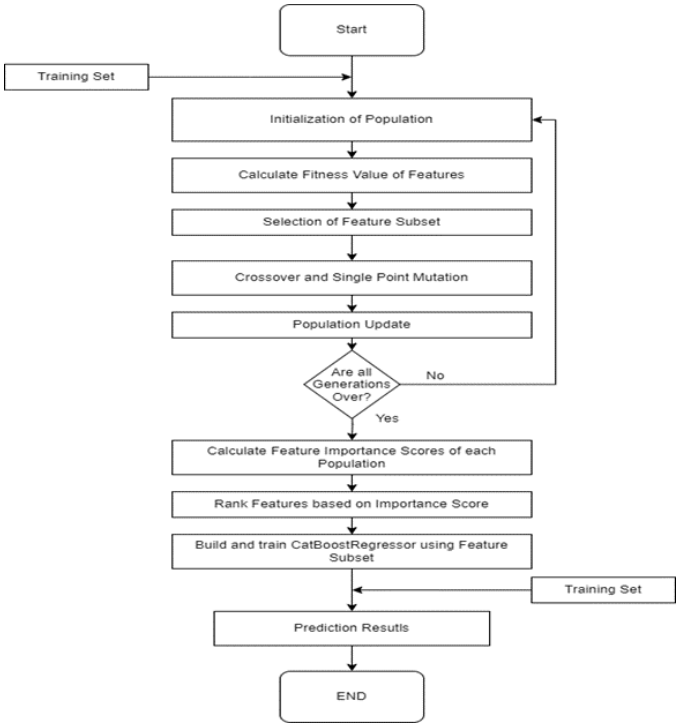

Fig. 1. Architecture of G-CatBoostRegressor Algorithm

TABLE II

No.	Features	No.	Features
1.	Bedrooms	27.	Condition
2.	Bathrooms	28.	RUR 1975
3.	sqft living	29.	LDC 1975
4.	floors	30.	HDC 2000
6.	waterfront	31.	Inhabited areas 1975
7.	view	32.	rel bai
8.	Year Built	33.	1975-1990 (SmoD)
9.	Grade	34.	1990-2000
10.	Year Renovated	35.	RUR 1990
11.	2000-2014	36.	Water Sf
12.	HDC 1990	37.	No BuiltUp
13.	rel ai	38.	Inhabited Aras 2000
14.	rel qhi	39.	LDC 2000
15.	Inhabited Areas 2014	40,	HDC 2000
16.	Br 1975	41.	RUR 2000
17.	RUR 2014	42.	HDC 2014
18.	rel hqll	43.	LDC 2014
19.	date	44.	HDC 1975
20.	sqft above	45.	Inhabited Areas 1990
21.	sqft basement	46.	population 1975
22.	sqft living-15	47.	population 2000
23.	sqft lot15	48.	population 2015
24.	LDC 1990	49.	gHM
25.	Latitude	50,	Longitude
26.	zipcode		

III. EXPERIMENTS AND ANALYSIS

A. Data and Preprocessing

The dataset for this paper has been obtained from Kaggle. The dataset consists of the property prices of King County, California. Cleaning was done with the removal of null and duplicated values. We take into consideration for 21,162 observations with more than twenty features which are a mixture of categorical and numerical features, displaying a good degree of nonlinearity. Feature Engineering was done to the original dataset and an additional 20 features were added. The list of features can be referred to from TABLE II

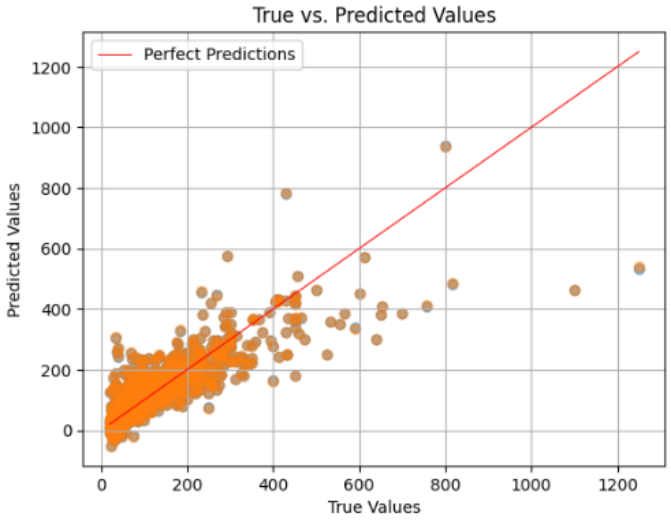

Fig. 2. Ridge Regression

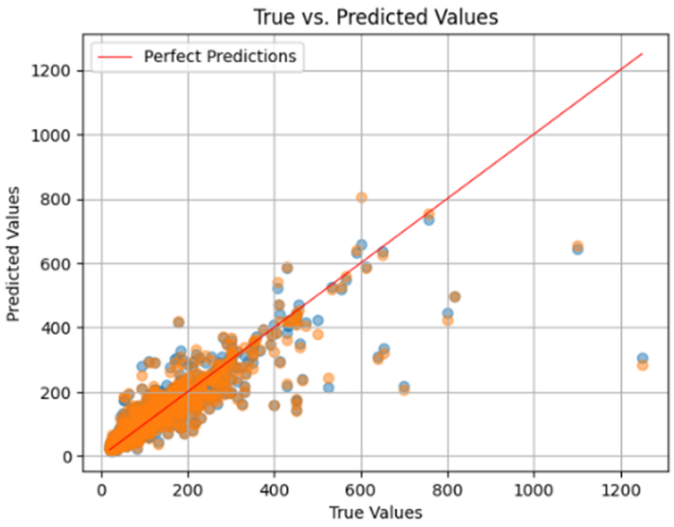

Fig. 3. XGBoost Regression

B. Results and Analysis

First, the value of house prices is predicted using the Ridge regression algorithm. The regularization is provided by L2 Norm with an alpha parameter of 1, and the loss function is the linear least squares function. A scatter plot made with the predicted and actual value of house price is illustrated in Fig 5 with X axis displaying the true value and Y axis displaying the predicted price of the house.

It can be viewed that the Ridge regression algorithm performs sub-optimally as the model can give predictions that are above the perfect prediction line, which indicates that the actual pricing of the house is much lower as compared to the predicted value. When the price goes beyond the 300 thousand mark however, we can notice that the price is on the lower end of the line which implies that the actual price is greater than the predicted price.

Given the non-linear relationship between the homes that might be inferred, XGBoost and Catboost are two integrated learning algorithms that we can take into consideration. XGboost is a gradient improvement algorithm based on an ensemble of weak classifiers, while Catboost is also a gradient improved algorithm, however, the performance of the Catboost algorithm is shown to be better as compared to XGBoost. Fig 6 and Fig 7 show the house prices predicted and actual values of XGBoost and Catoost regression with the X axis displaying the true value and Y axis displaying the predicted price of the house.

As can be observed from the graphs, both XGBoost and Catboost regressors seem to have a better performance as compared to Ridge Regression while Catboost has a better performance compared to XGBoost. We can see that the model can perform better up to the 400 thousand mark, however, the predictions thereby show values that are on the lower end of the line showing that the true values are still higher than the actual values of prediction.

Fig 8 shows the scatter plot for the G-Catboost algorithm and it can be seen that while the model doesn't seem to be

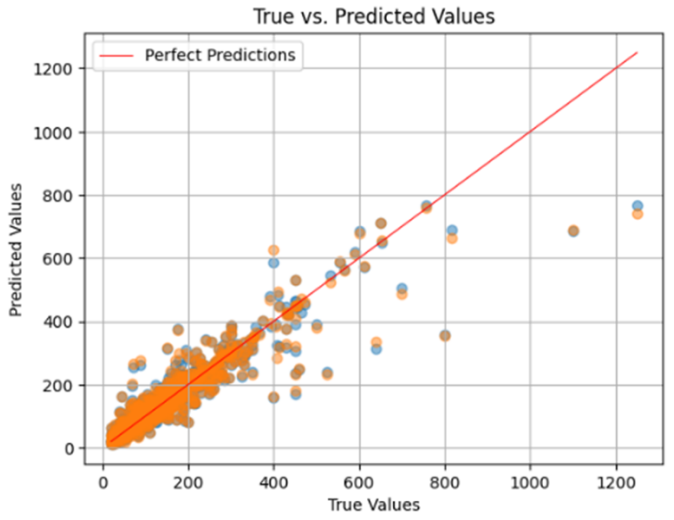

Fig. 4. Catboost Regression

TABLE III

No.	Features	No.	Features
1.	Area	12.	Bedrooms
2.	LDC_2014	13.	RUR 1975
3.	Water Sf	14.	LDC 1975
4.	view	15.	HDC 2000
6.	gHM	16.	Inhabited areas 1975
7.	Population 1975	17.	rel bai
8.	Year Built	18.	1990-2000 (SmoD)
9.	Inhabited $areas_2000$	19	$population_2000$
10.	Date	20.	Grade
11.	RUR 2014		

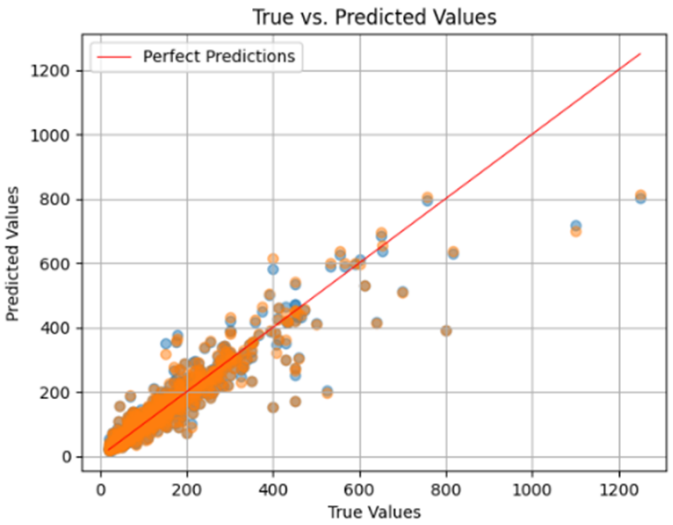

Fig. 5. G-Catboost Regression

TABLE IV

Model	**MSE**	R^2
Ridge Regression	2561.8620	0.7146
XGBoost	1509.5493	0.8448
CatBoost	845.4032	0.9152
G-CatBoost	803.3834	0.9173

performing better than the Catboost algorithm, it shows a slight improvement on the former with a reduced set of features being able to capture the information required. The results of evaluation using the proposed metrics is provided in TABLE IV.

It can be inferred from Table IV that the performance of CatBoost is superior to that of XGBoost while both perform better than Ridge Regression. However, the G-CatBoost algorithm shows a slightly better performance as compared to Catboost regression where the R^2 value of G-CatBoost increases from 0.9152 to 0.9173 and the MSE decreases from 845.4032 to 803.3834. The result shows that the Catboost regression model has an improved performance with a minimal set of features.

IV. Conclusion

In this study, the G-CatBoost regression algorithm is used to predict housing prices. The G-Catboost approach is used to tackle the dimensionality issue and lower model complexity, which in turn somewhat improves the prediction of house values due to a reduction in the number of irrelevant features. This technique targets the problem of feature set and model complexity optimization. Based on the sample data of houses in King County, Three models—Ridge, XGBoost, and Catboost—as well as the G-CatBoost model's prediction performance are examined. The experimental findings demonstrate that feature selection improves CatBoost's prediction performance, The R^2 and MSE of the G-Catboost model reaches 0.9173 and 803.3834 respectively. However, this work acknowledges that the Genetic algorithm used is a simple search-based algorithm, which can be improved using different hybrid genetic algorithms that can better battle the issue of high collinearity between the independent variables.

References

[1] H. Yıldırım, "Property value assessment using artificial neural networks, hedonic regression and nearest neighbors regression methods," vol. 7, pp. 387–404, 03 2019.

[2] L. L. Li, D. and H. Lv, "Prediction of china's housing price based on a novel grey seasonal model," *Mathematical Problems in Engineering*, 2021.

[3] G. Srirutchataboon, S. Prasertthum, E. Chuangsuwanich, P. Pratanwanich, and C. Ratanamahatana, "Stacking ensemble learning for housing price prediction: a case study in thailand," 01 2021.

[4] Z. Peng, Q. Huang, and Y. Han, "Model research on forecast of second-hand house price in chengdu based on xgboost algorithm," pp. 168–172, 10 2019.

[5] X. Wu and B. Yang, "Ensemble learning based models for house price prediction, case study: Miami, u.s," in *2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, pp. 449–458, 2022.

[6] C. Sheng and H. Yu, "An optimized prediction algorithm based on xgboost," in *2022 International Conference on Networking and Network Applications (NaNA)*, pp. 1–6, 2022.

[7] R. M. Almejrb, O. M. Sallabi, and A. A. Mohamed, "Applying c atboost regression model for prediction of house prices," in *2022 International Conference on Engineering MIS (ICEMIS)*, pp. 1–7, 2022.

[8] B. S. V. V. Guyon I, Weston J, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, pp. 389–422, 2002.

[9] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1424–1437, 2004.

[10] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," 2019.

[11] J. S. M. Kudo, "Comparison of algorithms that select features for pattern recognition," *Pattern Recognition*, vol. 33, pp. 25–41, 2000.