Determining the Optimal Regression Model for Amen Housing Dataset

Koki Yamanaka
COMP 4980, Special Topics: Machine Learning
Code link
December 9th 2023

ii. Data description

Ames House Price dataset is required from Kaggle. It contains 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa with the goal of predicting the selling price. Our focus will be using the train set 'train.csv' due to the complexity.

Datatype

There are 4 kinds of data type, binary, numeric, categorical, time series. It comprises binary, numeric, categorical, and time series. Binary data, such as the presence of a fireplace, is less common overall. Numeric variables fall into two categories based on diversity—some, like Garage Living and Masonry Veneer areas, exhibit diverse values, while others, like Garage Cars and fireplace count, have less diversity. Categorical data is expressed in both string and numeric forms, representing details like house materials and neighborhood information. Time series data includes features like the year the house was built. On volume, train.csv with file size of 449 KB consists of 1460 rows with 79 features.

Scope

To streamline the analysis of the complex 79-feature dataset, we implement a three-step column reduction process. Firstly, we exclude columns with a string data type, narrowing our focus to numerical data. Secondly, we generate a correlation matrix with the target variable and eliminate correlations below 0.32. Lastly, we discard columns without significant meaning, such as Overall Quality, resulting in a refined set of 15 features for further analysis.

Data Characteristics

Fifteen variables are categorized into three groups, including Time Series (YearBuilt, YearRemodAdd, GarageYrBlt), Continuous Numeric (LotFrontage, YearBuilt, YearRemodAdd, MasVnrArea, BsmtFinSF1, TotalBsmtSF, 1stFlrSF, GrLivArea, WoodDeckSF with a range of [0, 6110]), and Discrete Numeric (FullBath, TotRmsAbvGrd, Fireplaces, GarageCars with a range of [0, 14]). Data quality is high with minimal missing values (1 row), and the dataset is deemed ideal for machine learning due to its diverse data types and numerous features, making it applicable for house price prediction with multiple model exploration.

iii. Data analysis

Distributions & descriptive statistics

Figure 1 illustrates the distribution of each feature, showcasing a mix of left or right-skewed patterns, while some adhere to a normal distribution. Noteworthy outliers, such as the occurrence around 1500 on the x-axis for GarageArea, are observed. To enhance comparability, each feature is subsequently normalized using MinMax scaler, and basic statistics are provided. Notably, time variables like YearBuilt (0.218), YearRemodAdd (0.344), and GarageYrBlt (0.218) exhibit higher standard deviations compared to other features, indicating greater dispersion of data points from the mean. The non normal distribution suggests a violation of linear regression assumptions, prompting consideration of Tree algorithms and MLP for improved model performance.

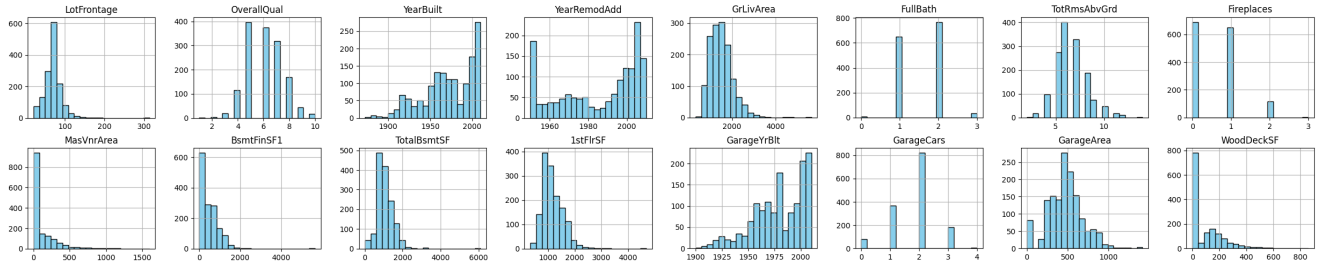

Figure 1 : Distribution of features for Housing Data

Correlation analysis

Figure 2 of the last row, features exhibit diverse correlations with SalePrice. Notably, YearBuilt and GarageYrBlt show a high correlation of 0.78.

	LotFrontage	OverallQual	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	TotalBsmtSF	1stFlrSF	GrLivArea	FullBath	TotRmsAbvGrd	Fireplaces	GarageYrBlt	GarageCars	GarageArea	WoodDeckSF	SalePrice
LotFrontage	1.000000	0.234196	0.117598	0.082746	0.179283	0.215828	0.363358	0.414266	0.368392	0.180424	0.320146	0.235755	0.064324	0.269729	0.323663	0.077106	0.334901
OverallQual	0.234196	1.000000	0.572323	0.550684	0.410238	0.239666	0.537808	0.476224	0.593007	0.550600	0.427452	0.396765	0.518018	0.600671	0.562022	0.238923	0.790982
YearBuilt	0.117598	0.572323	1.000000	0.592855	0.314745	0.249503	0.391452	0.281986	0.199010	0.468271	0.095589	0.147716	0.780555	0.537850	0.478954	0.224880	0.522897
GarageCars	0.269729	0.600671	0.537850	0.420622	0.363778	0.224054	0.434585	0.439317	0.467247	0.469672	0.362289	0.300789	0.482534	1.000000	0.882475	0.226342	0.640409
GarageArea	0.323663	0.562022	0.478954	0.371600	0.372567	0.296970	0.486665	0.489782	0.468997	0.405656	0.337822	0.269141	0.476498	0.882475	1.000000	0.224666	0.623431
WoodDeckSF	0.077106	0.238923	0.224880	0.205726	0.159349	0.204306	0.232019	0.235459	0.247433	0.187703	0.165984	0.200019	0.220623	0.226342	0.224666	1.000000	0.324413
SalePrice	0.334901	0.790982	0.522897	0.507101	0.475241	0.386420	0.613581	0.605852	0.708624	0.560664	0.533723	0.466929	0.470177	0.640409	0.623431	0.324413	1.000000

Figure 2: Correlation matrix depicting features (first 3 rows) and target (last 4 rows)

Figure 3 presents scatter plots of features against the target variable. A comparison with the correlation matrix yields insights. The scatter plot for YearBuilt to BsmtFinSF1 shows a scattered pattern with no distinct correlation. TotalBsmtSF to GrLivArea features exhibit a positive linear correlation with heteroskedasticity. Features from FullBath to GarageYrBlt, with 4-5 distinct values, display lower correlation. Beyond that range, features demonstrate high correlation. In conclusion, features with distinct values and clear linear relationships tend to exhibit higher correlation. Conversely, features with random scattered data present challenges in correlating with the target variable.

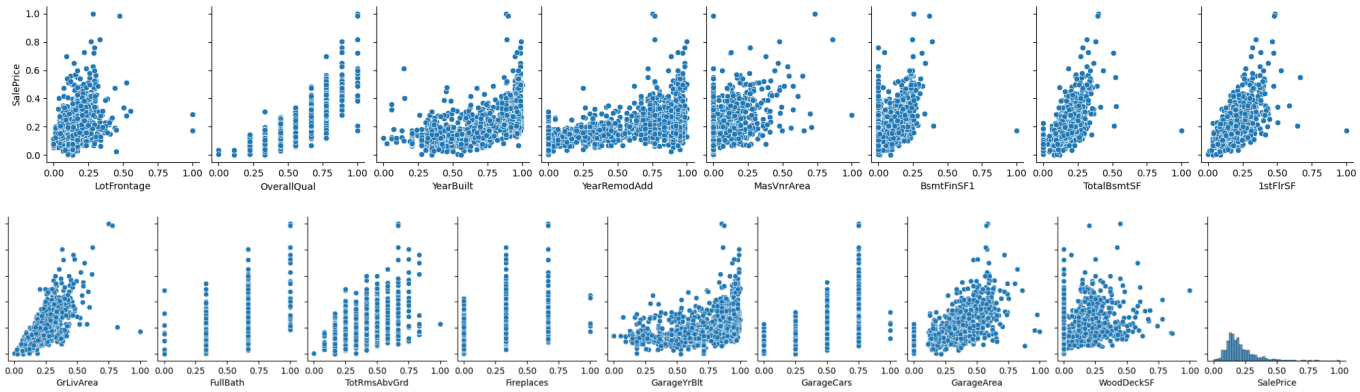

Figure 3: Scatter Plots of Features vs. Target Variable

iv. Data exploration

PCA

Before PCA, features are normalized with MinMaxScaler, revealing that 10 components capture 96% variance. The first component, explaining 47% variance, suggests data complexity can be simplified, possibly through an effective linear combination of features. Visualization of data and feature contributions in a PCA biplot has been created, which implies high eigenvalues for time-related features, hinting at their significant influence on the target variable. Notably, YearBuilt and GarageYrBlt closely aligned in vectors with high magnitude suggests a strong relationship or redundancy between them.

Decision Tree

In Figure 4, five key insights emerge. Firstly, GrLivArea is split twice, suggesting nuanced relationships with house prices are crucial for prediction. Secondly, GarageCars, with distinct values, notably correlates well, especially around 2.5 cars. Thirdly, the TotalBsmtSF feature displays a subtle upward trend after the value of 1036. This implies that such trends become crucial for understanding relationships in features when dealing with randomly dispersed data. Fourth, the most right leaf node produces an extreme value of $69,000 for the target variable, with only two samples, indicating potential overfitting. However, other leaves are separated with a good sample size. Overall, distinct features like garage count, specific area measurements, and construction year are deemed important, suggesting the value of testing each separately in models.

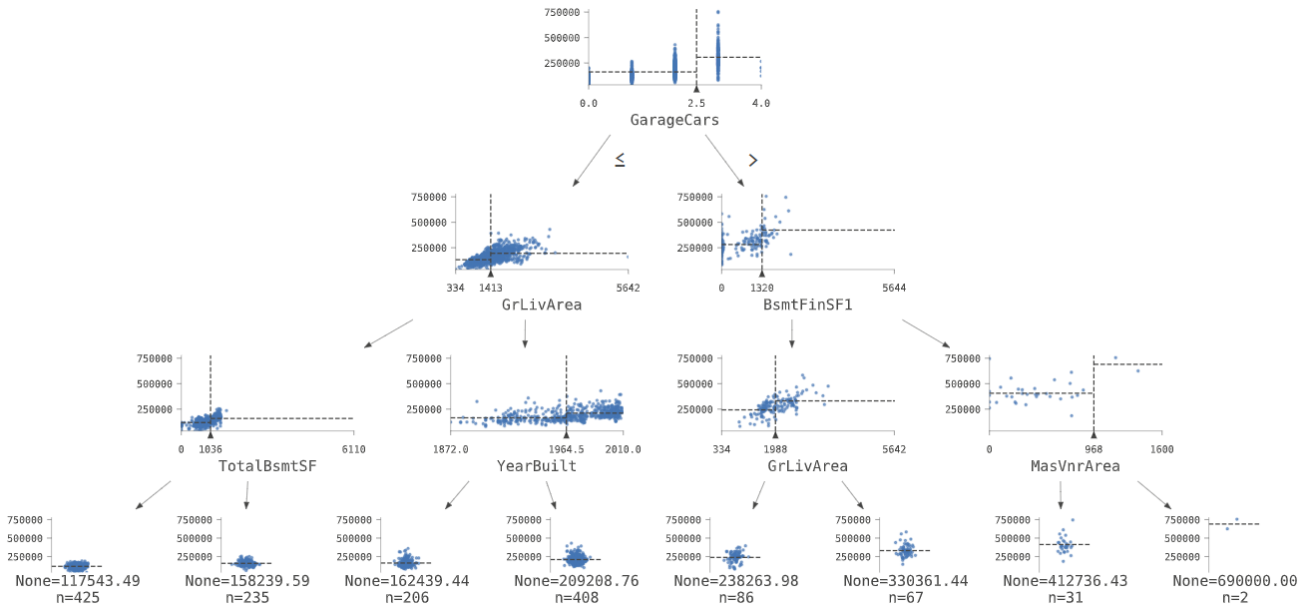

Figure 4 Decision Tree with max_depth=3 and 15 features.

v. Experimental method

Hypothesis Setup

We chose 5 algorithms RandomForest, AdaBoost, GradientBoosting, Multiple Linear regression, MLP, SVM(includes nonlinear/linear) to test on our dataset. We anticipate Gradient Boosting (Gboost) to outperform other algorithms due to the dataset's non-smooth target functions. Neural networks struggle with irregular functions (e.g., aggregated ball shapes correlations seen in Figure 3), and other algorithms may not handle non-linear data effectively. While SVM with a kernel trick excels at high-dimensional mapping, its lack of sequential improvement, unlike Gboost, might limit its performance on intricate, non-linear patterns.

Experiment Setup

Our experiment comprises 3 stages: Initially, we assessed RandomForest, AdaBoost, GradientBoosting, and Multiple Linear Regression to identify the best-performing model, with GradientBoosting emerging as the top choice. In the second stage, we compared GradientBoosting and MLP by testing all 15 features with a grid search (cv=5) while aiming to identify optimal parameters. In the final stage, we employed GridSearchCV on GradientBoosting, MLP, and SVM(nonlinear/linear), focusing on the 5 most explainable feature ('GarageCars','GrLivArea','TotalBsmtSF','YearBuilt','MasVnrArea'). Also, this process uses the optimal parameters determined in stage 2. Now, Let's dive into the details.

Stage	No.	Model	Parameters	Param tested	Test score (non cv)	Mean Test Score (cv)
1	1	RandomForest	N_estimator	100	0.804	-
	2	AdaBoost	N_estimators	50	0.786	-
	3	GradientBoost	N_estimators, max depth	50, 3	0.821	-
	4	Linear Regression	-	-	0.764	-
2	5	Gradient Boost	N_estimators	150, 100, 300, 1000	-	0.846, 0.846, 0.845, 0.837
	6		learning_rate	0.1, 0.2, 0.3, 0.6, 0.01	-	0.84, 0.83, 0.1, 0.79, 0.72
	7		Max_depth	3,5,12,18	-	0.84, 0.83, 0.81,0.71
	8	MLP	Hidden_layer	(100,100), (1000,1000), (2000,)	-	0.75, 0.74, 0.71
	9		Hidden_layer	(200,200), (300,200,100), (900,900,900)	-	0.75, 0.74, 0.72
	10		Hidden_layer	(150,100,50,30),(120,100,80,40,1), (200,100,50,30,20)	-	0.79, 0.79, 0.75

Table 1.0 : Result of Stage 1 (Comparing 4 models) and Stage 2 (Hyperparameter tuning on Gboost & MLP)

Note: ‘cv’ refers to 5 fold cross validation, ‘non cv’ refers to no cross validation performed.

Stage 1

In Table 1.0, GradientBoost outperforms model 1,2,4, indicating that Random Forest and AdaBoost, improving based on assigning different data regions , may not be well-suited for the aggregated data in Figure 3. The high coefficient (0.3893) in Linear Regression and the importance feature plot of Gradient Boost underscore the significance of the "GrLiveArea" feature in predicting the outcome.

Stage 2

Testing Gradient Boost with N_estimators set at 150 demonstrated enhanced performance, while default settings for learning_rate and max_depth proved optimal. Our systematic approach to MLP hyperparameter tuning focuses on hidden layers unfolded in five steps. Initially, we addressed convergence of gradient descent hurdles by min-max scaling the target variable, as large values like $69000 resulted in a negative R^2. Secondly, extending max_iter to 1200 from the default 200 ensured improved outcomes. Thirdly, Adjusting neuron count in configurations like (50,50) or (30,20) lacked significance. Exploring parameters akin to model 8 proved unfruitful. At last, increasing hidden layers, as in model 9, led to the discovery of the efficacy of layer-wise reduction, exemplified by (150,100,50) in model 10. In addition to hidden layers, we tested all available activation functions from sci-kit learn, 3 diverse alpha values from 0.001 to 0.01 and diverse optimization solver, but defaults seems to be the best.

vi. Results and analysis

Final Method

Stage 3 is our final method and the results are displayed in Table 2.0. After Stage 1 analysis, decision tree, and PCA, it became evident that numerous features were unnecessary for precise predictions. To streamline, we tested only the pivotal features 'GarageCars,' 'GrLivArea,' 'TotalBsmtSF,' 'YearBuilt,' and 'MasVnrArea,' employing them in Gradient Boosting, MLP, and both linear and non-linear Support Vector Machines for Mixmax Scaled normalized house price prediction. Each model underwent a 5-fold grid search with parameters detailed in Table 2.0. The mean test score of R^2 was obtained through this 5-fold process.

Performance

Table 2.0 below highlights the superior performance of Gradient Boosting with an R^2 of 0.7877, outshining the other two models. The optimal parameters, identified from all combinations, include N_estimators = 100, learning_rate = 0.1, and max_depth = 5. Similarly, MLP mirrors this structure, achieving an R^2 of 0.7877 with optimal parameters (200, 100, 50, 30, 20) and an adaptive learning rate.

No.	Model	Parameter grid	Params	Best Parameter	Mean test score
11	GradientBoost	N_estimators, learning_rate, max_depth	[100,150,200], [0,1], [1,2,3,5]	100, 0.1, 5	0.7877
12	MLP	Hidden layers, learning_rate	(150,100,50,30), (200,100,50,30,20), (120,100,80,40,10), 'adaptive'	(200,100,50,30,20), adaptive	0.7544
13	SVM	Kernel, C, gamma	['linear', 'rbf', 'poly'], [0.1,0.5,0.9], [0.1, 'auto']	Rbf, 0.9 auto	0.6910

Table 2.0 : Result of Stage 3 (Comparison of 3 models on 5 unique features)

Patterns

Table 2.0 revealed four prominent patterns as seen in Figure 3 scatter plot: (1) few distinct values as seen in 'GarageCars', (2) lively positive relationships accompanied by heteroscedasticity in 'GrLivArea' and 'TotalBsmtSF', (3) aggregated data points resembling an area under the curve of exponential functions, (e.g.'YearBuilt'), and (4) randomly scattered points, exemplified by 'MasVnrArea'.

Key Observations

MLP Shortcoming: The MLP with 0.6910 test score struggles with non-smooth data patterns, especially evident in features like patterns (1), (3), and (4) due to its probabilistic splitting approach. SVM Limitation: non linear Rbf SVM with 0.7544 test score encounters difficulties with patterns like (3) and (4) in high-dimensional spaces, making separation challenging. Gradient Boost Success: Highest test score among all, Gradient Boost excels with its simplistic, deterministic split approach, effectively identifying crucial points distinguishing house prices across patterns (1) to (4). MLP Nuance vs. Gradient Boost Simplicity: MLP's nuanced focus on low-level details, like YearBuilt, may lead to overfitting, while Gradient Boost's high-level perspective avoids overfitting by emphasizing key patterns. Data Conclusions: Few unique features notably impact house prices, with critical points that are not immediately apparent at first glance. An example is the significant increase in house prices for newer builds post-0.75 on the x-axis in the YearBuilt scatter plot (Figure 3). These critical points necessitate simple splits, as opposed to the complex splits observed in Table 2.0. For instance, MLP with a high number of neurons, hidden layers, and a high penalty (0.9) in SVM do not yield favorable results.

Applications

Our model's practical application extends to anticipating machine aging in manufacturing, exemplified by Kuka robotic arms used in Tesla car production. By converting 'YearBuilt' into the machine's age, we can predict potential failures in aging machinery. Additionally, correlating the size of a robot (square feet) and discrete number of power batteries with the likelihood of a machine crash enhances predictive maintenance.

vii. References

1. Nabil M. Decision Trees vs. Neural Networks. [Internet]. Available from: https://medium.com/@navarai/decision-trees-vs-neural-networks-ff46f47ce0a0; Jul 21.
2. Jackson P. COMP4980 Lecture Notes and Notebooks. Dec 19, 2023.
3. Seralouk. Feature/Variable importance after a PCA analysis. [Internet]. Available from: https://stackoverflow.com/questions/50796024/feature-variable-importance-after-a-pca-analysis; Jun 13, 2018.
4. Grinsztajn L, Oyallon E, Varoquaux G. Why do tree-based models still outperform deep learning on tabular data? [Internet]; Jul 18, 2022.
5. Ben Fraj M. In-depth Parameter Tuning for Gradient Boosting. [Internet]. Available from: https://medium.com/all-things-ai/in-depth-parameter-tuning-for-gradient-boosting-3363992e9bae ; Dec 24, 2017.