Available online at http://www.advancedscientificjournal.com
http://www.krishmapublication.com
IJMASRI, Vol. 2, issue 12, pp.792- 798, Dec. -2022
https://doi.org/10.53633/ijmasri

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY ADVANCED SCIENTIFIC RESEARCH AND INNOVATION (IJMASRI)

ISSN: 2582-9130 **IBI IMPACT FACTOR 1.5** **DOI: 10.53633/IJMASRI**

RESEARCH ARTICLE

HOUSING PRICE PREDICTION

Deep Bansal

Department of Information Technology, Maharaja Agrasen Institute of Technology, Rohini, Delhi
Email: deepvinaybansal@yahoo.com

Abstract

House Price Index (HPI) is commonly used to estimate the changes in housing prices. House prices strongly depend on factors such as location, area, population, it requires other information also to predict individual housing price. There has been a large number of papers adopting machine learning approaches to predict housing prices accurately, but they rarely take into account the performance of the models. This paper will validate multiple techniques in model implementation on regression and provide a better result for housing price prediction.

Keywords: Housing Price Prediction; Machine Learning; Stacked Generalization

Introduction

House is one of human life's most essential needs, along with other fundamental needs such as food, water, and much more. Demand for houses grew rapidly over the years. Existing prediction models usually use single model to predict the prices. The prediction accuracy of this type model is not up to the mark when datasets are noisy. In this paper we have considered various intrinsic parameters as well as external parameters. Several studies explored this problem but most of them compared the models' performance but did not consider combination of different machine learning models. We used the "House prices- Advanced Regression Techniques" dataset fetched from Kaggle. By applying several methods on this dataset, we could validate the performance of each individual approach. The lowest Root Mean Squared Logarithmic Error (RMSLE) is 0.076 on the data set, which belong to Stack Generalization Ensemble method.

Methodology

Data Collection

The dataset used in this project was from Kaggle website. Our dataset contains 81 attributes and 1460 observations. All the attributes are taken into account for house price prediction. Selling price is a dependent variable on several other independent variables. Dataset has 38 numerical and 43 categorical attributes which we will have to convert into dummy variables. First step was describing the futures in table.

future	description
• MSSubClass	The building class
• MSZoning	The general zoning classification
• LotFrontage	Linear feet of street connected to property
• LotArea	Lot size in square feet
• Street	Type of road access
• Alley	Type of alley access
• LotShape	General shape of property
• LandContour	Flatness of the property
• Utilities	Type of utilities available
• LotConfig	Lot configuration
• LandSlope	Slope of property
• Neighborhood	Physical locations within Ames city limits
• Condition1	Proximity to main road or railroad
• Condition2	Proximity to main road or railroad (if a second is present)
• BldgType	Type of dwelling
• HouseStyle	Style of dwelling
• OverallQual	Overall material and finish quality
• OverallCond	Overall condition rating
• YearBuilt	Original construction date
• YearRemodAdd	Remodel date
• RoofStyle	Type of roof
• RoofMatl	Roof material
• Exterior1st	Exterior covering on house
• Exterior2nd	Exterior covering on house (if more than one material)
• MasVnrType	Masonry veneer type
• MasVnrArea	Masonry veneer area in square feet
• ExterQual	Exterior material quality
• ExterCond	Present condition of the material on the exterior
• Foundation	Type of foundation
• BsmtQual	Height of the basement
• BsmtCond	General condition of the basement
• BsmtExposure	Walkout or garden level basement walls
• BsmtFinType1	Quality of basement finished area
• BsmtFinSF1	Type 1 finished square feet
• BsmtFinType2	Quality of second finished area (if present)
• BsmtFinSF2	Type 2 finished square feet
• BsmtUnfSF	Unfinished square feet of basement area
• TotalBsmtSF	Total square feet of basement area
• Heating	Type of heating

Table 1

future	description
• MSSubClass	The building class
• MSZoning	The general zoning classification
• LotFrontage	Linear feet of street connected to property
• LotArea	Lot size in square feet
• Street	Type of road access
• Alley	Type of alley access
• LotShape	General shape of property
• LandContour	Flatness of the property
• Utilities	Type of utilities available
• LotConfig	Lot configuration
• LandSlope	Slope of property
• Neighborhood	Physical locations within Ames city limits
• Condition1	Proximity to main road or railroad
• Condition2	Proximity to main road or railroad (if a second is present)
• BldgType	Type of dwelling
• HouseStyle	Style of dwelling
• OverallQual	Overall material and finish quality
• OverallCond	Overall condition rating
• YearBuilt	Original construction date
• YearRemodAdd	Remodel date
• RoofStyle	Type of roof
• RoofMatl	Roof material
• Exterior1st	Exterior covering on house
• Exterior2nd	Exterior covering on house (if more than one material)
• MasVnrType	Masonry veneer type
• MasVnrArea	Masonry veneer area in square feet
• ExterQual	Exterior material quality
• ExterCond	Present condition of the material on the exterior
• Foundation	Type of foundation
• BsmtQual	Height of the basement
• BsmtCond	General condition of the basement
• BsmtExposure	Walkout or garden level basement walls
• BsmtFinType1	Quality of basement finished area
• BsmtFinSF1	Type 1 finished square feet
• BsmtFinType2	Quality of second finished area (if present)
• BsmtFinSF2	Type 2 finished square feet
• BsmtUnfSF	Unfinished square feet of basement area
• TotalBsmtSF	Total square feet of basement area
• Heating	Type of heating

Table2

DATA PREPROCESSING

First, we checked and removed any outliers that are present in our dataset. It is most important step in data pre-processing as it can affect a regression model by pulling our estimated regression line further away from the true population regression line. We used scatterplot technique to remove two potential outliers from are pipeline but still, we need to make our model robust to outliers. We used Sale Price and GrLivArea for checking outliers. Next, we did was to check for skewness in Target variable 'Sale Price'. The target variable has original skewness of 1.879 and

was positively left skewed therefore we normalized the distribution and attained final skew of 0.121 using box-cox.

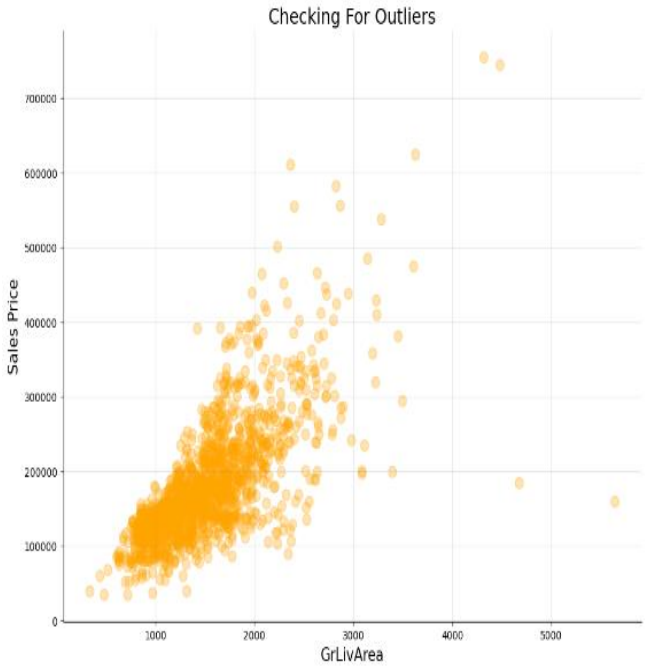

Fig. 1 Checking for outliers

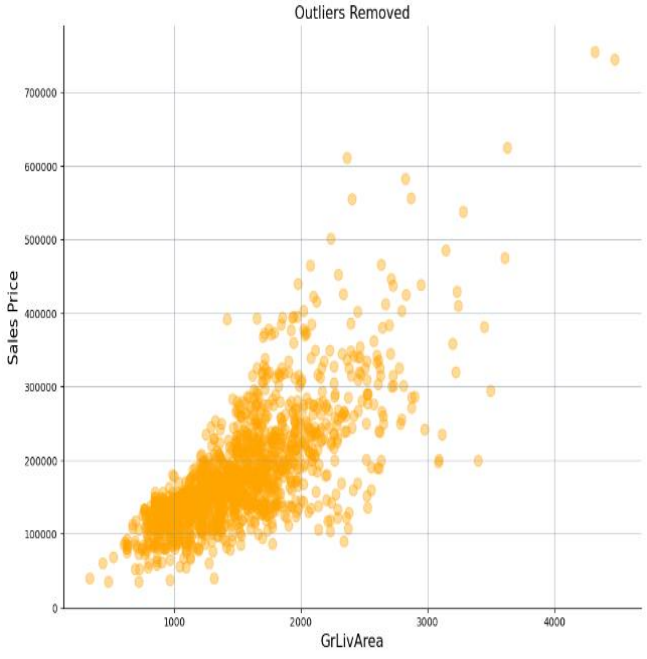

Fig. 2: After removing outliers

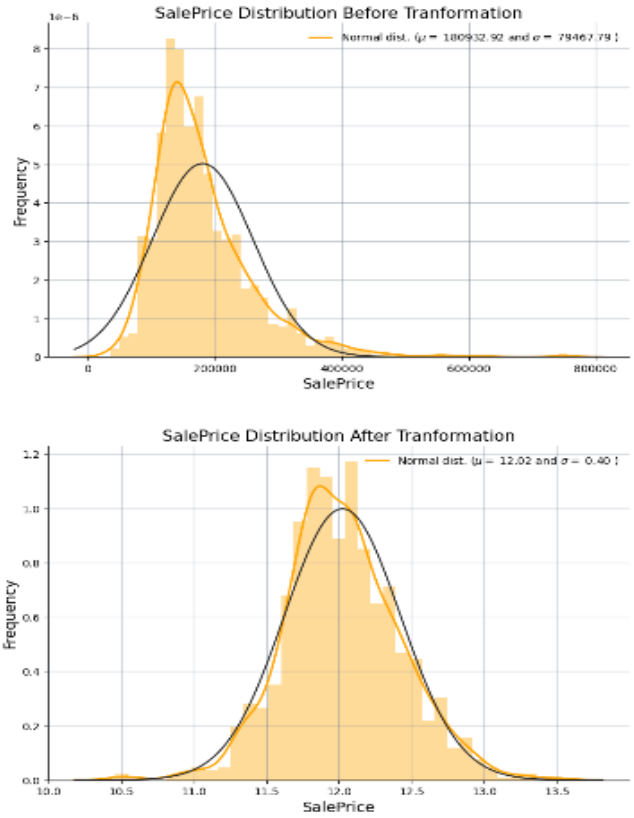

Fig. 3

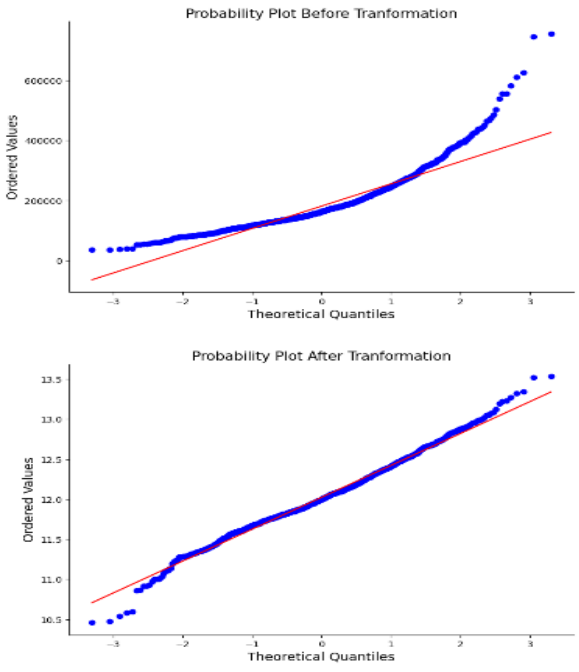

Fig. 4

Further we handled missing and Nan values in our dataset. Few of the attributes such as PoolQC, MiscFeature, Alley, Fence etc. had Nan values or empty fields which were replaced with None. LotFrontage was filled using median LotFrontage of the neighborhood. GarageYrBlt, GarageArea and GarageCars replaced missing data with 0.

MODEL SELECTION

Before building models, the data should be processed accordingly so that the models could learn the patterns more efficiently. After being processed, the dataset included 81 features. Function used is Root Mean Squared Logarithmic Error (RMSLE). This function is illustrated as follow:

$$RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2}$$

Lasso Regression

Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. Additionally, the lasso regression technique employs variable selection, which leads to the shrinkage of coefficient values to absolute zero. The accuracy of this model on the dataset came out to be **0.1116**.

Elastic Net Regression

Elastic net linear regression is combination of both the lasso and ridge techniques to regularize regression models. Ridge utilizes an L2 penalty and lasso uses an L1 penalty. With elastic net, you don't have to choose between these two models, because elastic net uses both the L2 and the L1 penalty. The accuracy of this model on the dataset came out to be **0.1116** not different from lasso regression.

Gradient Boosting Regression

Gradient Boosting is a machine learning technique that uses regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. The accuracy of this model on the dataset came out to be **0.1166**.

- Set learning_rate = 0.05
- Set n_estimators = 3000
- Set min_samples_leaf = 15

Kernel Ridge Regression

Kernel ridge regression (KRR) combines ridge regression (linear least squares with l2-norm regularization) with the kernel trick. It thus learns a linear function in the space induced by the respective kernel and the data. For non-linear kernels, this corresponds to a non-linear function in the original space. The accuracy of this model on the dataset came out to be **0.1152**.

EXTREME GRADIENT BOOSTING (XGBOOST)

XGBoost is a tree boosting scalable machine learning system. The scalability of XGBoost is due to several major systems and algorithmic optimizations including a novel tree learning algorithm for handling sparse data and a theoretically justified weighted quantile sketch procedure enabling instance weight handling in approximate tree learning. After tweaking the XGBoost model multiple times, we set our parameter to the following:

- Set learning_rate = 0.05
- Set n_estimators = 2200
- Determined the optimal tree specific parameters min_child_weight = 1.7817, subsample = 0.5213, colsample_bytre = 0.4603
- Set requlariza tion parameter: reg_lambda = 0.8571, reg_alpha = 0.4640, gamma = 0.0468

The model performed with a high accuracy where the RMSLE of the training set is around **0.1161.**

LIGHT GRADIENT BOOSTING MACHINE (LIGHTGBM)

LightGBM is a gradient boosting framework that uses a tree-based learning algorithm. LightGBM has faster training speed with lower memory usage compare to XGBoost. There are also detailed differences in modelling making them outstanding among different gradient boosting models. In this paper, we utilized the LGBMRegressor from lightgbm open-source package.

The optimal parameters for the model are listed as follows:

- *Set learning_rate* = 0.05
- *Set n_estimators*=720
- *Set optimal trees specific parameters: num_leaves*=5

When applying the LightGBM method, the model achieved a decent accuracy with the loss of **0.1164** on the dataset.

STACKED GENERALIZATION

Stacked Generalization or "Stacking" for brief is associate ensemble machine learning formula. It involves combining the predictions from multiple machine learning models on the constant dataset, like bagging and boosting. The good thing about stacking is that it will harness the capabilities of a spread of well-performing models on a classification or regression task and build predictions that have higher performance than any single model within the ensemble. The design of a stacking model involves two or a lot of base models, usually observed as level-0 models, and a meta-model that mixes the predictions of the base models, observed to as a level-1 model.

- **Level-0 Models (*Base-Models*):** Models fit on the training data and whose predictions are compiled.
- **Level-1 Model (*Meta-Model*):** Model that learns the way to best combine the predictions of the bottom models.

Model Stacking: Average Based Mostly Model Category

We begin with this easy approach of averaging base models. We have a tendency to build a replacement class to increase scikit-learn with our model and conjointly to leverage encapsulation and code apply (inheritance). Averaged based Model Class Score: 0.1087 (Std: 0.0076). It appears even the simplest stacking approach very improve the score. This encourages us to go further and explore a less simple stacking approach.

Model Stacking: Adding a Meta Model

In this approach, we have a tendency to add a meta-model on averaged base models and use the out-of-folds predictions of those base models to train our meta-model.

The procedure, for the training part, may be described as follows:

Split the total training set into two disjoint sets (here train and holdout).

Train many base models on the first part (train)

Test these base models on the second part (holdout).

Use the predictions from holdout fold (called out-of-folds predictions) as the inputs, and therefore correct responses (target variable) as the outputs to train a higher-level learner called meta-model.

The first 3 steps are done iteratively. If we take for example a 5-fold stacking, we have tendency to first split the training data into five folds. Then we are going to do five iterations. In every iteration, we train each base model on four folds and predict on the remaining fold (holdout).

So, we will be sure, after 5 iterations , that the entire data is used to get out-of-folds predictions that we will then use as new feature to train our meta-model

For the prediction part , We average the predictions of all base models on the test data and used them as meta-features on which, the final prediction is done with the meta-model.

After this all the Results were put in an Ensemble to get the Final Results. Final Ensemble Score: **0.076**

Result

Many iterations of performance tuning were done to find optimal solution. Lasso Regression,

Elastic Net Regression, Gradient Boosting Regression, XGboost, LightGBM, and Stack Generalization are shown and compared. The results regarding them are shown in Table 3.

Model Used	RMSLE Value
Lasso	0.1116
Kernel Ridge	0.1152
ElasticNet	0.1116
Gradient	0.1166
XGBoost	0.1161
LightGBM	0.1164
Stack Generalization Ensemble	0.076

Table 3

The best results are shown by stack generalization with RMSLE value 0.076. Also Lasso and ElasticNet performed equally on the dataset. Gradient Boosting, XGBoost and LightGBM didn't do much well. Stack Generalization performed much well due to

- K- Fold cross-validation as it obtains best performance of each model
- Coupling of various models

Conclusion

In the current proposal, we gave a model to predict house prices based on given dataset. The dataset was obtained from Kaggle website which has 81 attributes. The dataset was preprocessed to remove any outliers and then few attributes with empty fields were replaced with appropriate values so that our data doesn't deviate. Then box-cox transformation was applied to normalize the dataset and remove skewness. The paper has compared the RMSLE values of Lasso regression, Elastic Net Regression, Gradient Boosting, Extreme Gradient Boosting, Light Gradient Boosting, and Stack generalization. Lasso and Elastic net performed equally in terms of score and stack generalization came out to be most optimal. The base models in stack generalization were Elastic Net regression, Gradient Boosting, and Kernel Ridge Regression, and meta model is lasso regression. The XGBoost and LightGBM are decent methods when comparing accuracy, but their time complexities are the best, especially Light GBM.

Reference

1. Breiman, L. (2019). Random Forests. SpringerLink.https://doi.org/10.1023/A:1010933404324 (accessed September 11, 2019).
2. Quang Truong, Minh Nguyen, Hy Dang, Bo Mei. (2019). International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI2019) Housing Price Prediction via Improved Machine Learning Techniques.
3. Darshan Sangani, Kelby Erickson and Mohammad Al Hasan. (2017). "Predicting Zillow Estimation Error Using Linear Regression and Gradient Boosting", *IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 530-534.
4. Rakesh Kumar Saini. (2019)." Data Mining tools and challenges for current market trends", Journal (IJSRNSC) Vol.7, Issue.2, pp.1114, Apr-2019. https://doi.org/10.26438/ijsrnsc/v7i2.11104.
5. Atharva Chouthai, Mohammed Athar Rangila, Sanveed Amate, Prayag Adhikaari, Vijay Kukre. (2019)." House Price prediction Using Machine Learning", IRJET, Vol.6, Issue:03, Mar 2019.
6. Thuraiya Mohd, Suraya Masrom, Noraini Johari. (2019)." Machine Learning Housing Price Prediction in Petaling Jaya, Selangor, Malayasia", IJRTE, ISSN:2277-3878, Vol.8, Issue-2S11, Sept 2019, Blue Eyes Intelligence Engineering & Science Publication, DOI:10.35940/ijrte. B1084.0982S1119.
7. Chen, T and Guestrin, C. (2016). (2016). Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 16 2016. doi:10.1145/2939672.2939785.
8. DMLC. (2019) xgboost. GitHub. https://github.com/dmlc/xgboost (accessed June 1, 2019).

9. Ke, G., Meng, Q. Finley, T. Wang, T. Chen, W. Ma, W. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems 30 2017:3146–54.
10. Microsoft. Light GBM. GitHub. https://github.com/microsoft/LightGBM (accessed June 1, 2019).

*****