What makes a house high quality?

Barış Deniz Sağlam
Informatics Institute
Middle East Technical University
Ankara, Turkey
e155841@metu.edu.tr

***Abstract*—This paper investigates two research questions on Ames Housing Dataset:**

1) **Can we predict overal quality (*OverallQual*) of houses?**
2) **Which features are good contributor and predictors of overall quality?**

A random forest regression model is trained for this purpose and Shapley values are used to explain the trained model. We found that external quality and built year features are good predictors for overall quality of the house.

***Index Terms*—Regression, Random Forest, Model Explainability, Shapley Values, SHAP**

I. Introduction

Ames Housing Dataset [1] consists of the 2930 house sale records in Ames, Iowa from 2006 to 2021. With 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables [1], it's a sufficiently rich dataset to experiment with machine learning algorithms. We investigate two research questions in this paper:

1) Can we predict overal quality (*OverallQual*) of houses?
2) Which features are good contributors or predictors of overall quality?

For the first research question, we train and tune linear regression models, and random forest models. Then, we apply model explainability techniques to understand the contribution of each feature.

A. Literature Search

Compared to the classical software, machine learning models are much more complex and mostly black boxes. However, it's often needed to understand and explain why a machine learning model made an inference in production systems. In GDPR, it's required that every individual has the right to obtain an explanation for an algorithmic assessment. For example, a bank is obligated to provide an explanation to a customer for a loan application rejected. As machine learning models have been gaining popularity in the recent years, explainability of them have been becoming more important. Hence, many studies have been conducted on model explainability. The approaches can be split into two categories [2]:

- Explainable models
- Model-agnostic interpretation methods

The first approach is to train explainable models at the first place, such as decision trees. Although this approach sounds reasonable, it excludes many black-box models that perform well in practice such as deep learning models. Whereas, with the model agnostic approach, any machine learning model can be explained with various techniques proposed.

Model agnostic methods can be further split into two categories: global and local explainers. Local explanation methods are capable of providing an explanation for any data instance. Whereas, global explanation methods provide a single explanation for the model such feature importances.

One of the main global model explanation methods is permutation feature importance method introduced by Breiman (2001) [3] for random forest models and made model agnostic by Fisher, Rudin, and Dominici (2019) [4]. The method estimates feature importances in a trained model with such an algorithm [2]:

1) Estimate the model error with the original dataset.
2) For each feature j, permute the values in the feature and estimate the model error with the predictions made with this modified data. The importance of the feature j, is the difference between the new error and the original error.

Intuitively, when a feature is important for the model, model performance degrades when the values of that feature is permuted.

A well-known local model-agnostic explanation method is Shapley value method, introduced by by Shapley (1953) [5] as a technique for coalitional game theory. It estimates individual player's performance when the game is not individualistic. The method achieves this by measuring the performances of the different coalitions of players. The idea is that if a player performs well, the coalitions, i.e. teams, that contain the player also should perform well. The method is applied in machine learning by treating the features as the players and the model inference as the game. It forms different sets of features and measures the model performance. Then, for each feature, it averages all the scores of the feature set the feature was included. Since the method can be applied to both a single data point and a full dataset, it can generate both local and global explanations. A useful property of Shapley values is that they are additive such that their summation is equal to the model's prediction for a data point. The asymptotic runtime of original Shapley algorithm is exponential with the number of features, hence, it's too slow. Lundberg and Lee (2017) [6] proposed a variation of the method, SHAP, with polynomial time for tree-based models.

II. Methods

A. Data quality

Before analyzing the dataset, we inspect the description of the dataset to identify each variable's data type and split them into three types: numerical, ordinal, and nominal. The data type of a variable is important when choosing encoding and imputation methods for that variable. We also determine the order of categories for ordinal variables whose order cannot be inferred by *pandas* library [7]. We perform basic data quality checks on variables to ensure that there're no inconsistent values according to the definition of each variable. For instance,

- Any date variable cannot be negative.
- Month cannot be out of range [1,12].
- Area cannot be negative.
- The sale year must be greater or equal to all dates related the building such as house built year, garage built year.

Any inconsistent value is treated as missing and erased from the dataset.

There are varying number of missing values for variables in the dataset. Any row in the training dataset which misses the target variable (*OverallQual*) is dropped since it cannot be used for training. For the majority of the categorical variables, there is a category defined representing missing value (*NA* or *None*) in the description of the dataset. The missing values in those categorical variables are filled with the corresponding missing value defined. After imputing the missing values in such categorical variables, there are two variables left in the training set that have missing values: *LotFrontage* and *MasVnrArea*. From Fig. 1, it's inferred that the missingness mechanisms for both variable are MCAR since there is no correlation between their missingness. When there is no category defined for missing value, the most frequent value is used to impute missing values in the categorical variables. For numerical variables, the mean imputation is used.

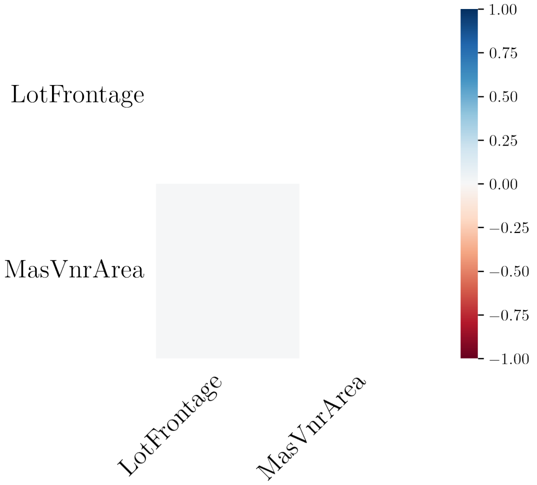

Fig. 1. missingno [8] heatmap for missing values

As it's seen from Fig. 2 and Fig. 3, while *FullBath*, *GarageArea*, *GarageCars*, and *TotRmsAbvGrd* have roughly symmetrical distributions, the rest of the numerical variables are skewed. The numerical variables are scaled to eliminate scale differences among variables. The numerical variables with symmetrical distribution, i.e. the ones with skewness value less than 0.5, are standardized. Whereas, the numerical variables with skewed distribution, i.e. with skewness value greater than 0.5, are scaled with min-max scaling.

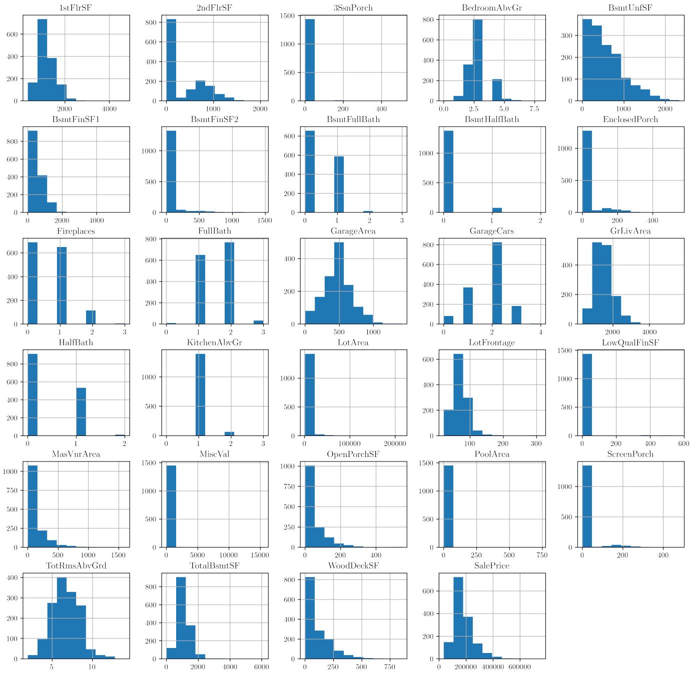

Fig. 2. Histogram of numerical variables

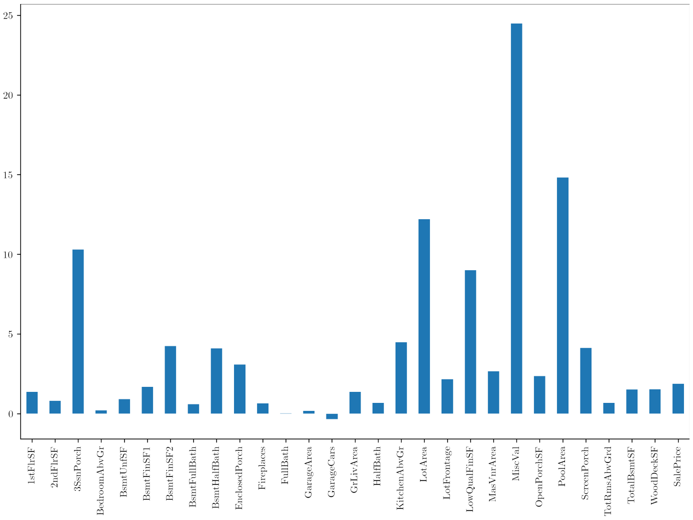

Fig. 3. Skewness of numerical variables

In all preprocessing steps, only the statistical features of training set are used to prevent information leakage from training to evaluation.

B. Dimensionality reduction and feature selection

Since the target variable is *OverallQual*, we remove *SalePrice* and *SaleCondition* features from the dataset to prevent information leakage. Similarly, we ignore *YrSold* and *MoSold* featuers since the model's predictions should be independent of the sale date.

To achieve a smaller and simpler model, we apply dimensionality reduction and feature selection on the dataset. For numerical variables, principal component analysis is performed and as it's seen from the Fig. 4, 3 components cover 90% of the variance in the numerical variables of dataset. Hence, we reduce the all numerical variables to 3 dimension.

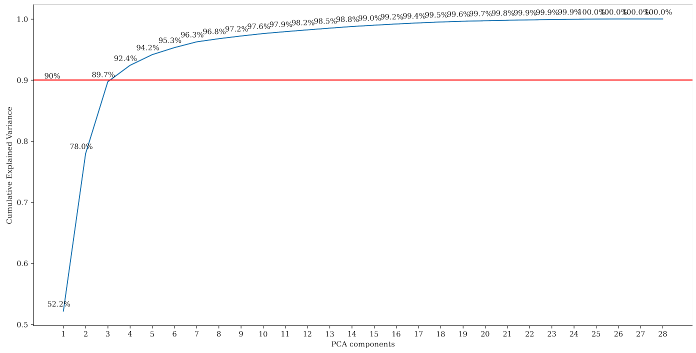

Fig. 4. PCA - Cumulate Explained Variance

Similarly, for ordinal and nominal variables, we calculate global feature importances with SHAP method. We use TreeExplainer from *shap* library [9] on the trained model with the validation dataset to compute Shapley values per data point per feature. Then, we calculate global feature importances by taking absolute mean of Shapley values per feature across all data points. As it's seen from Fig. 5, 17 categorical features cover 90% of the contributions of all categorical features.

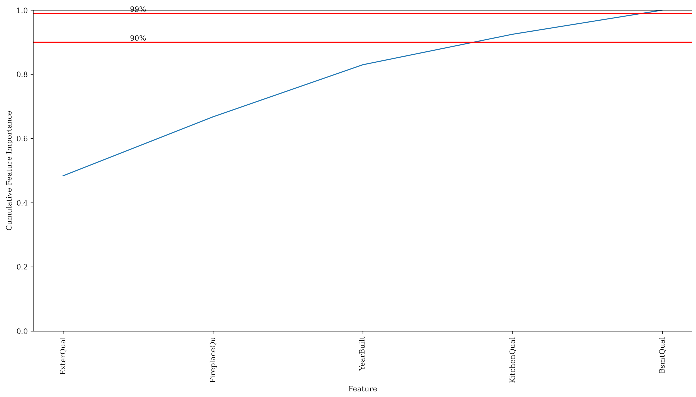

Fig. 5. SHAP - Cumulative global feature importance

Therefore, we only use the most important ones of the categorical variables listed in Table I to train our model.

C. Model training and evaluation

The dataset is split into train and validation sets with 80:20 ratio. The test set is not used for training or model selection.

TABLE I
SHAP - Global Feature Importance

variable	% feature importance
ExterQual	0.288264
YearBuilt	0.188572
FireplaceQu	0.139039
KitchenQual	0.035111
BsmtQual	0.033722
HouseStyle	0.032869
Neighborhood	0.031666
OverallCond	0.020790
MasVnrType	0.020446
BsmtFinType1	0.019180
MSSubClass	0.019014
RoofStyle	0.014213
Exterior1st	0.013521
YearRemodAdd	0.012969
GarageType	0.010599
GarageYrBlt	0.009502
Foundation	0.008329

A random forest model is a collection of decision trees under the hood. It averages the predictions of all decision trees to make a prediction. This reduces the variance in the model error. We train random forest models with 5-fold cross validation and grid search of hyper-parameter space with *scikit-learn* library [10]. Due to constraints on the computation resources available for the research, 18 different hyper-parameter configurations are used for random forest regressor. We also train a linear regression model as a baseline. R^2 and mean squared error metrics are used to evaluate and compare models' performances during training.

R^2 metric is defined as

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{1}$$

Hence, R^2 score is:

- 1 for a perfect model, always predicting the correct value
- 0 for a regression model that always predicts the sample mean
- negative for worse models

Hence, R^2 score is a well-suited metric for regression problems.

The random forest regressor clearly outperforms linear regressor by achieving 0.743 R^2 score and 0.412 MSE on the validation set.

TABLE II
Model scores on train and validation sets

Model	Set	R^2 (higher is better)	MSE (lower is better)
Random Forest	Train	0.945	0.087
	Validation	0.734	0.412
Linear Regressor	Train	0.665	0.562
	Validation	0.564	0.750

D. Model explanation

Shap method provides local explanations to data points as well. It's possible to explain a model's prediction for a data point with Shapley values. A SHAP feature contribution graph starts from the expected value for prediction and builds up the prediction by adding each feature's contribution. For example, for the data point with OverallQual=10, the contribution of each feature can be seen from the Fig. 6. A high external quality and a late built year features make most contribution for this house's overall quality.

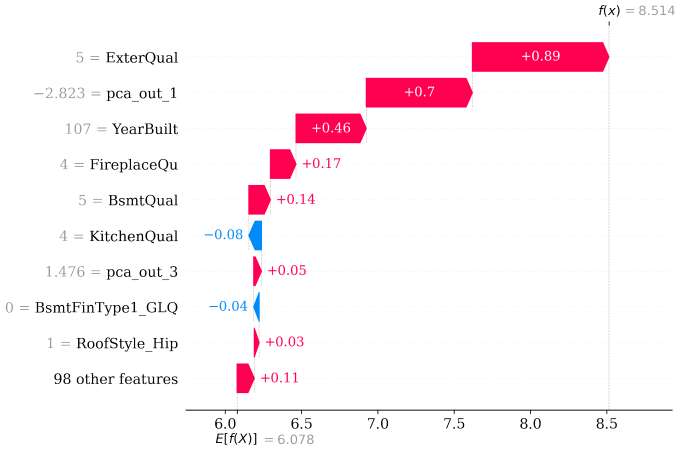

Fig. 6. SHAP - Feature contributions to the prediction

Similarly, for the data point with OverallQual=3, the contribution of each feature can be seen from the Fig. 7. Low external quality, fireplace quality, overall condition, and being an old building contributes negatively to overall quality.

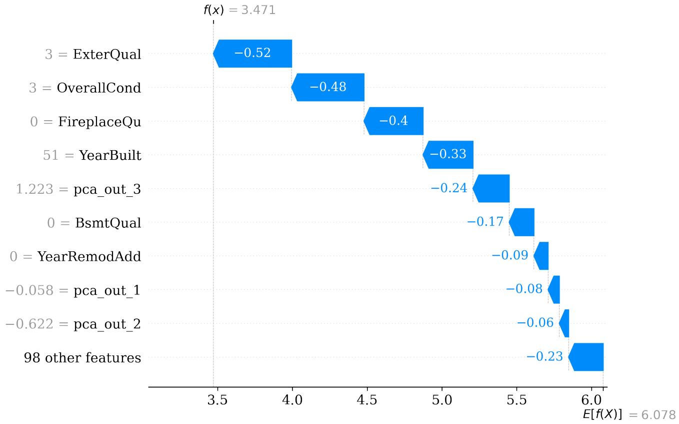

Fig. 7. SHAP - Feature contributions to the prediction

Fig. 8 shows how different values for each feature affect Shapley values. It implies that external quality is the most important attribute for overall quality, which is quite reasonable for housing market. The wider a variable's different values are divided in terms of Shapley values, the more important the feature is for the model. Hence, external quality, built year, and fireplace quality provide the strongest signals to predict overall quality.

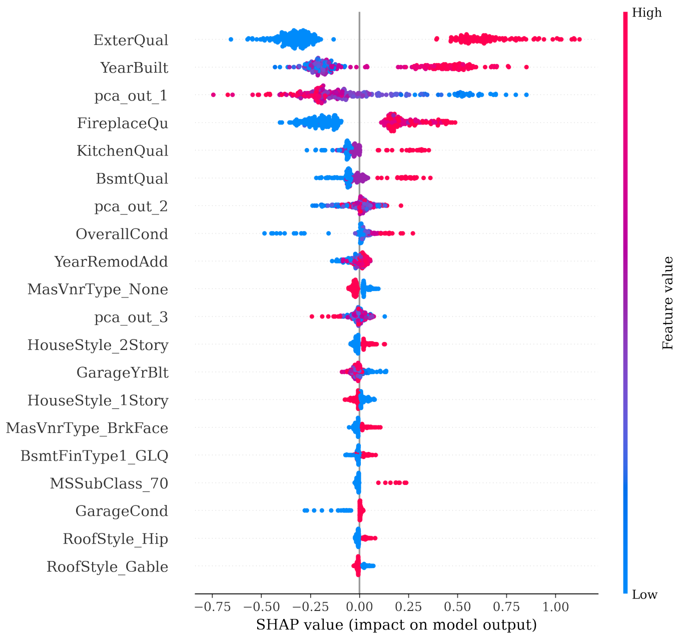

Fig. 8. SHAP Summary

III. Conclusion

In this paper, we analyzed Ames Housing Dataset and trained a regression model to predict overall quality of the houses. We leveraged PCA and SHAP methods for dimensionality reduction and feature selection respectively. We demonstrated that SHAP method provides reasonable global and local explanations for the random forest regressor trained.

References

[1] D. De Cock, "Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project," *Journal of Statistics Education*, vol. 19, no. 3, 2011. [Online]. Available: www.amstat.org/publications/jse/v19n3/decock.pdf

[2] C. Molnar, *Interpretable Machine Learning*. Github, 2019.

[3] L. Breiman, "Random Forests," *Machine Learning 2001 45:1*, vol. 45, no. 1, pp. 5–32, oct 2001. [Online]. Available: https://link.springer.com/article/10.1023/A:1010933404324

[4] A. J. Fisher, C. Rudin, and F. Dominici, "All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously," *undefined*, 2019.

[5] L. Shapley, "A value fo n-person games," *Ann. Math. Study28, Contributions to the Theory of Games, ed. by HW Kuhn, and AW Tucker*, pp. 307–317, 1953.

[6] S. M. Lundberg, P. G. Allen, and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: https://github.com/slundberg/shap

[7] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 51–56.

[8] A. Bilogur, "Missingno: a missing data visualization suite," *Journal of Open Source Software*, vol. 3, no. 22, p. 547, feb 2018. [Online]. Available: https://joss.theoj.org/papers/10.21105/joss.00547

[9] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature Machine Intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.