Disclaimer of Liability

The material and information contained on this website is for general information, reference, and self-learning purposes only. You should not rely upon the material or information on the website as a basis for making any academic, business, legal or any other decisions. You should not copy any material or information on the website into any of your academic, business, legal or any other non-private usages. ZHANG Wengyu will not be responsible for any consequences due to your violations.

Whilst ZHANG Wengyu endeavours to keep the information up to date and correct, ZHANG Wengyu makes no representations or warranties of any kind, express or implied about the completeness, accuracy, reliability, suitability or availability with respect to the website or the information, products, services or related graphics contained on the website for any purpose. Any reliance you place on such material is therefore strictly at your own risk.

ZHANG Wengyu will not be liable for any false, inaccurate, inappropriate or incomplete information presented on the website.

Although every effort is made to keep the website up and running smoothly, due to the nature of the Internet and the technology involved, ZHANG Wengyu takes no responsibility for and will not be liable for the website being temporarily unavailable due to technical issues (or otherwise) beyond its control or for any loss or damage suffered as a result of the use of or access to, or inability to use or access this website whatsoever.

Certain links in this website will lead to websites which are not under the control of ZHANG Wengyu. When you activate these you will leave ZHANG Wengyu's website. ZHANG Wengyu has no control over and accepts no liability in respect of materials, products or services available on any website which is not under the control of ZHANG Wengyu.

To the extent not prohibited by law, in no circumstances shall ZHANG Wengyu be liable to you or any other third parties for any loss or damage (including, without limitation, damage for loss of business or loss of profits) arising directly or indirectly from your use of or inability to use, this site or any of the material contained in it.

THE HONG KONG POLYTECHNIC UNIVERSITY

Department of Computing

Group Project Report

COMP4433 Data Mining and Data Warehousing
Group ***Pangolin***

Name:	Student ID:
JIANG Yiyang	**xxx**
ZHANG Wengyu	**xxx**

Nov. 2023

Contents

1 Exploratory Data Analysis

There are 1460 instances of training data and 1460 of test data. Total number of attributes equals 81, of which 36 is numerical, 43 categorical + Id and SalePrice.

1.1 Dataset Overview

The dataset is composed of 1460 entries, each representing a unique house with 81 features. These features encompass a wide range of information, including, but not limited to, physical attributes, quality assessments, and geographical location, culminating in the target variable SalePrice.

1.2 Distribution of Target SalePrice Variable

The distribution of the SalePrice is notably **right-skewed**, suggesting that while most homes are priced below the average, there is a number of homes with prices significantly above the average, which may represent luxury or outlier properties. Examining the shape of the distribution of these variables indicates that several are skewed, particularly SalePrice and LotArea. Such skewness can have implications for statistical modeling and may necessitate transformations to meet model assumptions. Additionally, the presence of outliers, especially in the upper ranges of SalePrice and LotArea, suggests that luxury homes or unusually large properties could disproportionately influence mean values and variance estimates.

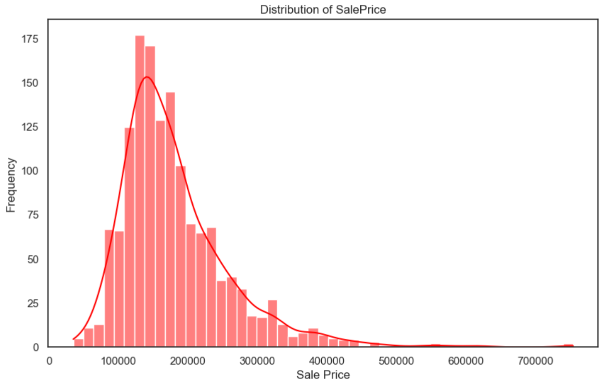


Figure 1: Distribution of Target SalePrice Variable

1.3 Missing Data Analysis

An important step in exploratory data analysis is assessing the extent of missing data, as this can significantly affect the performance of predictive models. Missing data can arise for various reasons, such as errors in data collection, non-response in surveys, or the absence of certain features in some observations (e.g., a house without a pool).

Features like PoolQC, MiscFeature, Alley, and Fence have high percentages of missing values, which may suggest that these are not applicable to all houses in the

Feature	Missing %	Feature	Missing %
PoolQC	99.52%	FireplaceQu	47.26%
MiscFeature	96.30%	LotFrontage	17.74%
Alley	93.77%	GarageType	5.55%
Fence	80.75%	GarageYrBlt	5.55%
GarageFinish	5.55%	BsmtExposure	2.60%
GarageQual	5.55%	BsmtFinType2	2.60%
GarageCond	5.55%	BsmtFinType1	2.53%
BsmtCond	2.53%	BsmtQual	2.53%
MasVnrArea	0.55%	MasVnrType	0.55%
Electrical	0.07%		

Table 1: Percentage of missing values for each feature

dataset (for example, not all houses have pools or alleys). For such features, it might be reasonable to assume that the absence of a record could indicate the absence of the feature. Conversely, features with a low percentage of missing values, such as Electrical, are likely missing due to data recording errors. Handling missing data requires careful consideration—techniques range from imputation (filling in missing values) to exclusion (removing features or observations with missing values) or using models that can accommodate missing data. The choice of technique will depend on the nature of the data and the intended analysis or predictive modeling approach.

1.4 Correlated Numerical Feature Analysis on Sale Price

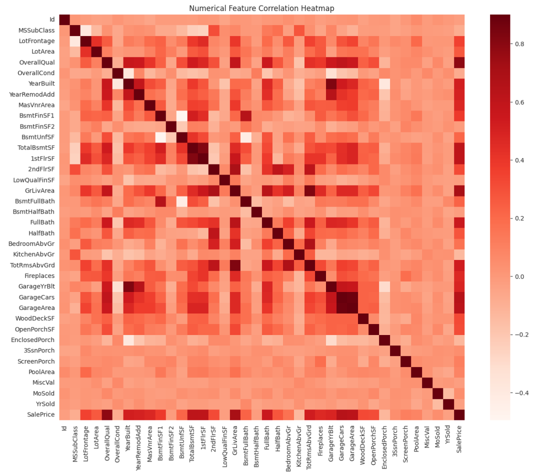


Figure 2: Numerical Feature Correlation

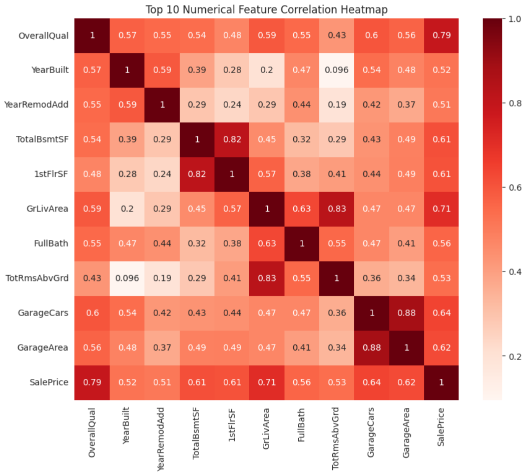


Figure 3: Top 10 Numerical Feature Correlation

In analyzing the dataset, several features demonstrate a noteworthy correlation with the sale price of houses. These features are pivotal in predicting house prices and are visually represented in Figures 2 and 3.

Overall Quality (OverallQual): As shown in Figure 2, the overall quality of a house has a strong positive correlation with its sale price, indicating that houses with higher quality ratings tend to sell for more, which is expected since buyers are typically willing to pay a premium for superior quality.

Above Ground Living Area (GrLivArea): Figure 3 highlights that the total living area above ground is positively correlated with sale price, confirming that larger homes tend to command higher prices.

Size of Garage (GarageCars and GarageArea): Both the capacity of the garage, in terms of the number of cars it can hold, and its size in square footage are significantly correlated with sale price as seen in Figure 2. A spacious garage is a valuable feature for many buyers.

Total Basement Area (TotalBsmtSF): The size of the basement also plays a crucial role in the house's value, with a positive correlation between basement size and sale price depicted in Figure 3.

Year of Construction and Remodel (YearBuilt and YearRemodAdd): Newer and recently remodeled houses tend to fetch higher prices, which can be attributed to modern amenities and reduced maintenance, as illustrated in Figure 2.

Bathrooms and Rooms (FullBath and TotRmsAbvGrd): The number of bathrooms and total rooms are indicative of the size and functionality of a house, which in turn reflects on the sale price, as shown in Figure 3.

First Floor Square Footage (1stFlrSF): The area of the first floor is particularly important for many homebuyers, with a larger first floor correlating with a higher sale price, as can be seen in Figure 2. This could be due to preferences for larger main-level spaces or the need for accessibility.

1.5 Descriptive Statistics Summary

The descriptive statistics provide a comprehensive summary of the dataset's central tendencies, dispersion, and shape of the distribution of the dataset's features. Here we discuss the summary statistics of some key features in detail:

- SalePrice: The sale prices of homes in the dataset exhibit considerable variability. The mean sale price is around $180,921, with a median of $163,000, indicating a right-skewed distribution as the mean is higher than the median. The standard deviation is substantial at $79,442, signaling a wide range of sale prices from as low as $34,900 to as high as $755,000. This range suggests the dataset includes a diverse set of properties, from modest homes to luxury estates.

- LotArea: The lot size of the properties also shows a broad distribution, with the average being 10,516 square feet. The median lot size is 9,478.5 square feet, which, being lower than the mean, indicates a right-skewed distribution. The maximum lot size is a sprawling 215,245 square feet, which points to the presence of large estates within the dataset.

- OverallQual: Reflecting the overall material and finish quality of the homes, this feature has a mean value of 6.1 on a scale from 1 to 10. The majority of homes have a quality rating of 5 to 7, suggesting a moderate level of quality in the dataset's homes. However, the presence of homes with a rating of 10 indicates that there are high-quality, premium homes as well.

- YearBuilt: The average year of construction is 1971, but the houses in the dataset range from those built in 1872 to newer constructions in 2010. The standard deviation of approximately 30 years indicates a mix of older historical homes and newer constructions. Newer homes could potentially fetch higher prices due to updated amenities and structural standards.
- GarageCars: The number of cars that can be parked in the garage has a mean value of 1.77 with a standard deviation of 0.75, suggesting that most homes have space for one or two cars. Notably, some homes can accommodate up to four cars, which may be a feature of higher-end properties.
- GarageArea: Complementing the GarageCars feature, the area available for garage space has an average of 472.98 square feet. The data range from no garage space at all (0 square feet) to substantial garage spaces of up to 1,418 square feet, accommodating larger or multiple vehicles, which is a desirable feature for many home buyers.
- GrLivArea: This feature represents the above-grade (ground) living area in square feet. The mean living area is 1,515.46 square feet, with a maximum of 5,642 square feet, pointing to some very large properties. This feature is likely a significant factor in the sale price, as larger living areas are often associated with higher prices.

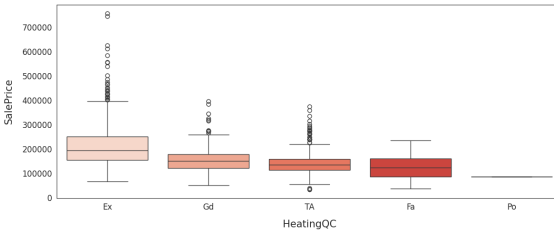


Figure 4: Heating Quality and Condition vs SalePrice

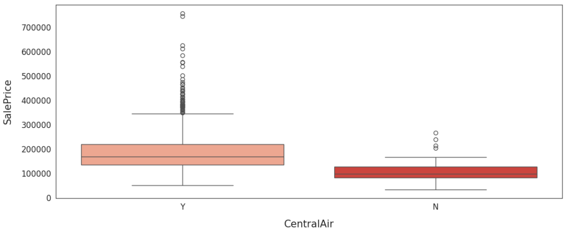


Figure 5: Central Air Conditioning vs SalePrice

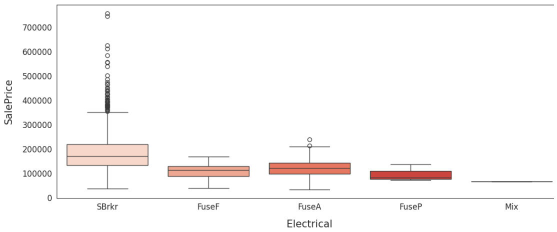


Figure 6: Electrical System vs SalePrice

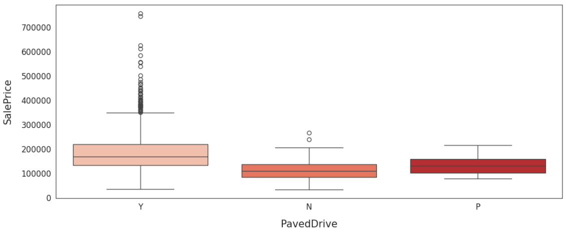


Figure 7: Paved Driveway vs SalePrice

1.6 Categorical Feature Analysis on Sale Price

In the pursuit of understanding the relationship between the categorical features and sale prices of houses, we have identified four categorical variables of interest based on their boxplot distributions: HeatingQC, CentralAir, Electrical, and PavedDrive. These features were selected due to their discernible impact on the median sale price, as well as the presence of higher maximum sale prices within certain categories. This suggests a potential positive correlation between these specific features and higher property values.

2 Feature Engineering

2.1 Optimizing Target Variable Distribution

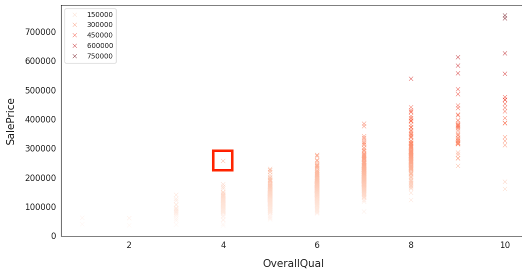


Figure 8: OverallQual vs SalePrice

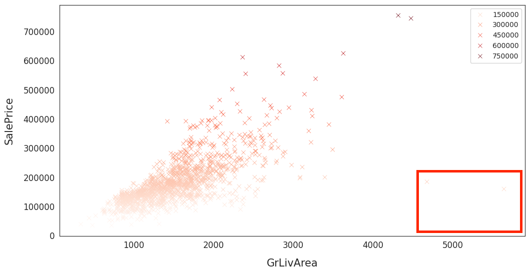


Figure 9: GrLivArea vs SalePrice

The decision to apply a logarithmic transformation to the SalePrice variable using `np.log1p` is well-justified based on the insights from the EDA. Here's how it relates to the earlier findings:

- Right-Skewed Distribution: The EDA indicated that SalePrice was right-skewed. Such a skewness could lead to challenges in modeling since many algorithms assume a normal distribution of the target variable. The log transformation helps to mitigate this by normalizing the distribution and reducing the skewness, as previously discussed.

- High-Value Outliers: Our analysis of the boxplots revealed that higher levels of **OverallQual** (overall material and finish quality) as can be seen in Figure 8 and larger **GrLivArea** (above grade (ground) living area square feet) as can be seen in Figure 9 are linked to increased maximum sale prices. These outliers have the potential to disproportionately influence the performance of regression models. By applying a log transformation, we reduce the relative difference between these outliers and the rest of the data, which can enhance the model's ability to generalize better across varied data points.

- Linear Relationships: The log transformation can enhance the linear relationship between features and the target variable. Since we've observed certain features that have a strong relationship with SalePrice, transforming the target variable could help in uncovering more linear relationships, which is especially beneficial if linear regression techniques are to be used.

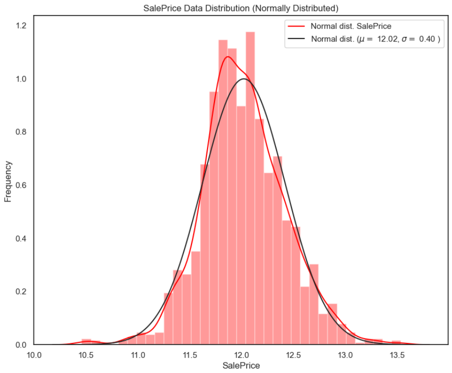


Figure 10: SalePrice Data Normal Distribution

In essence, the log transformation of SalePrice is a preparatory step that aligns with the EDA findings, aiming to optimize the dataset for the modeling phase. It's an effort to meet the assumptions of many predictive modeling techniques, potentially leading to more accurate and robust models.

2.2 Handling missing values

Before addressing missing values, certain numerical features like MSSubClass, YrSold, and MoSold were converted to string data type. This is because these features, despite being represented by numbers, are categorical in nature. Their numerical representation doesn't necessarily denote any ordinal relationship.

- For features where a 'NA' value indicates the absence of that feature (e.g., PoolQC for houses without a pool), we filled missing values with 'None'.
- For garage-related features, missing values were filled with 'None' for categorical columns and 0 for numerical columns, signifying the absence of a garage.
- Basement-related features had their missing values replaced with 'None', indicating the absence of a basement.
- The feature LotFrontage had its missing values replaced with the median LotFrontage of their respective neighborhoods. This is based on the assumption that houses in the same neighborhood would have similar frontage.
- The Functional feature's missing values were filled with 'Normal'.
- For features like Electrical, KitchenQual, Exterior1st, Exterior2nd, and SaleType, missing values were filled with the mode of the respective columns.
- The MSZoning classification was inferred from the MSSubClass — the type of dwelling involved.

Finally, any remaining missing values in numerical features were filled with 0, and in categorical features were filled with 'None'.

2.3 Addressing Skewness in Feature Distribution

Skewness in the distribution of data for numeric features can pose a challenge for predictive modeling as many algorithms assume normally distributed data. Skewness measures the asymmetry of the probability distribution of a real-valued random variable about its mean. In our preprocessing steps, we identified features with a skewness above 0.5 as highly skewed and targeted them for transformation.

Initially, we considered the **Box-Cox** transformation, which is a versatile tool for normalizing skewed distributions and can handle positive values. We also evaluated the **Yeo-Johnson** transformation, which extends the Box-Cox method to support zero and negative values. However, upon comparative analysis, we found the **log1p** function, which applies $\log(1+x)$ to each element, to be the most effective for our dataset.

The superior performance of the **log1p** transformation in our context can be attributed to its simplicity and its particular suitability for handling features with a long tail distribution. It effectively mitigates the impact of extreme values, which can distort the predictive model's performance. By applying the **log1p** transformation, we achieved a more symmetric distribution for our features, enhancing the predictive accuracy of our models. The distributions of the transformed features align more closely with the assumptions of our modeling techniques, thereby improving their ability to discern patterns and make accurate predictions.

The revised subsection of the report integrates the practical findings with the theoretical considerations, providing a clear rationale for the choice of the **log1p** transformation over the other methods.

2.4 Features Enhanced

In the realm of predictive modeling, particularly for house price prediction, the preprocessing of data is a critical step that enhances model performance and predictive accuracy. Our recent preprocessing pipeline consisted of several key stages: feature elimination, feature creation, feature transformation, dummy variable encoding, dimensionality reduction, and outlier removal. Below we detail each stage and its impact on the final dataset.

2.4.1 Feature Elimination

The initial step in our preprocessing involved the removal of features deemed less influential or redundant for predicting house prices. Specifically, `Utilities,` `Street`, and `PoolQC` were dropped. These features either showed little variance across observations or contained a large number of missing values, thereby providing negligible predictive power for our models.

2.4.2 Feature Creation and Transformation

To capture more detailed aspects of the houses that could potentially affect their prices, we engineered new features through combinations and transformations of existing ones:

- `YrBltAndRemod`: This feature encapsulates both the age of the house and its latest renovation, recognizing that newer or recently remodeled homes might fetch higher prices.
- `TotalSF`: By summing the basement, first, and second-floor square footage, we constructed a comprehensive metric of the total living space, which is a significant factor in home valuation.
- `Total_sqr_footage`: This aggregates the liveable area, excluding the basement, to account for the primary living spaces.
- `Total_Bathrooms`: A count of all bathrooms, weighing full and half baths differently, reflects the functionality and convenience of the home.
- `Total_porch_sf`: The combined area of all porch types contributes to the curb appeal and outdoor living space, potentially influencing buyer interest.

  Additionally, we simplified the dataset by converting some numerical features into binary ones, indicating the presence or absence of certain amenities like pools, second floors, garages, basements, and fireplaces.

2.4.3 One-Hot Encoding

To accommodate the categorical nature of some variables, we employed one-hot encoding, which transforms categorical variables into a format that can be provided to ML algorithms. This process increased the feature space, resulting in a matrix of dimensions (2917, 333), where each column represents a unique attribute or category.

2.4.4 Outlier Removal

In a critical step to ensure model robustness, we removed outliers identified from previous analyses. These rows were dropped based on their extreme values in either features or the target variable, which could skew model training and lead to overfitting.

The preprocessing stage has meticulously transformed the raw data into a refined dataset primed for machine learning. Our dataset now consists of **1453** observations, each described by **331** features engineered to encapsulate the multifaceted nature of house valuation. This extensive preprocessing not only aids in mitigating the issues of non-linearity, high-dimensionality, and outlier sensitivity but also ensures that the most relevant and informative features are utilized for predicting house prices.

3 Model Training

The model training phase is a crucial step in our predictive analysis, aimed at forecasting house prices with high precision. This report outlines the approaches and methodologies adopted in the training of various regression models, their ensemble, and the final prediction strategy.

3.1 Cross-Validation Strategy

To assess the performance and generalizability of our models, we employed a 10-fold cross-validation strategy. This method involves partitioning the training dataset into ten subsets, training the model on nine subsets, and validating it on the remaining one. This process is repeated ten times, ensuring that each subset serves as the validation set once. Cross-validation helps in mitigating the overfitting risk and provides a more robust evaluation of the model's predictive power.

3.2 Individual Model Training

We trained several individual models, each with its strengths and peculiarities:

1. Ridge Regression: Utilizes L2 regularization to prevent overfitting and is less sensitive to outliers.

2. LASSO Regression: Employs L1 regularization promoting sparsity in the model coefficients, thus performing feature selection.

3. Elastic Net: Combines L1 and L2 regularization to leverage the benefits of both Ridge and LASSO.

4. Support Vector Regression (SVR): Provides a flexible decision boundary using kernel functions.

5. Gradient Boosting Regressor: Builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions.

6. LightGBM: A gradient boosting framework that uses tree-based learning algorithms, known for its efficiency and speed.

7. XGBoost: An optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable.

   These models are equipped with robust scalers and are tuned with a range of hyperparameters to find the optimal settings.

3.3 Ensemble Techniques

Stacking: we used a StackingCVRegressor that combines the individual models using a meta-regressor. In this case, the meta-regressor is an XGBoost model, which was trained to optimally combine the predictions from all base regressors. Stacking helps in blending different types of models, taking advantage of their unique aspects.

Blending: after training, each model holds a certain degree of bias towards the training data. To mitigate this and enhance the prediction robustness, we applied a blending strategy. We took the predictions from each model and combined them, assigning different weights to each model's predictions to achieve the best performance on the training data.

3.4 Training and Scoring

The models were trained on the feature matrix derived from the training data, and their performance was evaluated using the RMSLE (Root Mean Squared Logarithmic Error) score. This metric is particularly suited for our regression problem as it penalizes underestimates and overestimates equally and is robust to the effects of the scale of the data.

3.5 Prediction and Submission

We leveraged an ensemble of sophisticated regression models to predict housing prices on the test dataset. Each model's predictions were weighted and combined using a blending approach, resulting in a composite prediction that capitalizes on the strengths of each individual model. The ensemble prediction was then prepared for submission:

- Sale Price Prediction: A composite prediction was generated for the test set using the blended model approach. The blend_models_predict function integrated outputs from various individual models.
- Inverse Transformation: The logarithmic scale predictions were transformed back to the original price scale using the np.expm1 function and then rounded down to the nearest whole number to comply with the competition's submission requirements.

We addressed potential outliers in the predicted prices: extreme values were identified using the 0.005 and 0.995 quantiles of the predicted prices. Prices below the 0.005 quantile were considered extremely low and were scaled up by a factor of 0.77. Prices above the 0.995 quantile were considered extremely high and were scaled down by a factor of 1.1. This step ensures that our submission is not unduly penalized for the presence of potential outliers, which could be a result of overfitting or anomalous data.

3.6 Comparative Analysis of Regression Models

In our recent experiment, we evaluated various regression models to predict housing prices. The objective was to understand the impact of log transformation on feature normalization and its subsequent effect on the model's performance. The table below summarizes our findings. The model performances with our log-transformed features (Ours) were compared with their counterparts using non-transformed features (Ours*). The results suggest a clear trend: models trained on log-transformed data consistently outperformed those trained on raw data, across all model types.

The Blending approach, which combines predictions from multiple models, yielded the best performance with a score of **0.11885**. This ensemble method not only improved prediction accuracy but also demonstrated the benefits of feature transformation—emphasized by the increased score to 0.12436 when log transformation was not applied.

Method	Ours	Ours*
Ridge Regression	0.12191	0.12510
Lasso	0.12275	0.12811
Elastic	0.12378	0.12811
SVR	0.12264	0.13049
Gradient	0.12744	0.12939
LightGBM	0.12541	0.12588
XGBoost	0.12102	0.13183
Stacking	0.12041	0.13439
Blending	**0.11885**	0.12436

Table 2: Performance Comparison of Regression Models

Stacking, another ensemble technique, showed significant performance degradation without log transformation, jumping from 0.12041 to 0.13439. This suggests that the stacking model was particularly sensitive to the distribution of the features.

Individual models like Ridge Regression, Lasso, and SVR showed a moderate increase in scores when the features were not log-transformed, indicating their robustness to non-normalized data but still confirming the advantages of normalization.

In conclusion, our investigation highlights the effectiveness of feature normalization via log transformation and the superiority of ensemble methods, particularly blending, in predicting housing prices with higher accuracy. This study emphasizes the need for preprocessing steps and the selection of appropriate models to improve prediction performance.

4 Discussion

4.1 The most useful feature engineering effort

4.1.1 Data Transformation

- Normalization: Addressing skewness in the distribution of the features through transformations like Box-Cox or np.log1p is a crucial step. It helps in stabilizing variance and making linear models assumptions (like normality of errors) more valid, which can be particularly beneficial for models like Lasso and Ridge regression that assume normally distributed errors.

- New Features: Creation of new features based on existing data, particularly those that aggregate multiple related features into more holistic ones. Examples include TotalSF (total square footage), Total_Bathrooms, and Total_porch_sf. These features likely capture more nuanced aspects of a house's size and quality than the individual components alone. By combining multiple related variables into a single feature, the models can better understand the underlying patterns that contribute to a house's price. These engineered features can significantly impact the model's predictive power because they relate more directly to potential buyers' considerations when evaluating a house.

4.2 Works done for the best improvement

In the case of multiple submissions, the best improvement (not necessarily the best absolute performance) seems to come from the blend of models and the subsequent post-processing of predictions. By combining model predictions, the code is likely leveraging the strengths of each model and mitigating their individual weaknesses. This ensemble approach often yields better generalization on unseen data, as different models may capture different aspects of the data.

Specifically, the improvement came from adjusting the final blended predictions to account for the competition's evaluation metric, which is sensitive to the error in the log of the price. By fine-tuning the predictions with a focus on the distribution's extremes, our team could better align predictions with the price scale's nuances. This kind of post-processing can effectively push the score up significantly on the leaderboard, especially in the tail ends of the distribution where large errors can disproportionately affect the RMSLE.

5 Further Exploration: MLP and Transformer

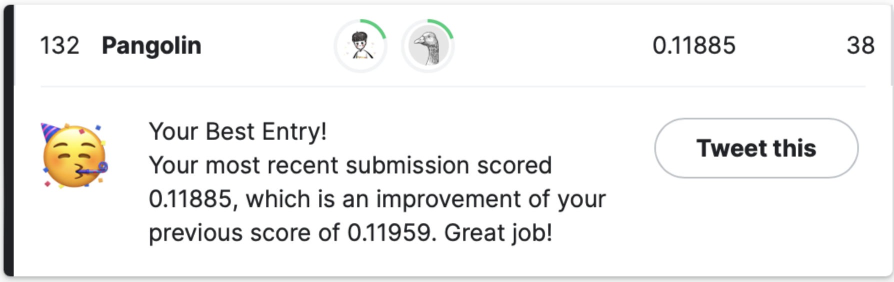
In our quest for the most accurate predictive model, we ventured beyond traditional techniques and experimented with advanced neural network architectures, namely Multilayer Perceptron (MLP) and Transformers. Although these models are celebrated for their success in various domains, our experiments yielded scores of **0.13117** for the MLP and **0.12983** for the Transformer, which did not surpass the performance of our linear models.

Our housing price dataset, while rich in features, is limited in the number of observations, a scenario where simpler models often have the edge. They tend to generalize better and avoid overfitting, a common pitfall when complex models encounter sparse datasets. MLPs and Transformers, with their numerous parameters and deep architectures, are particularly prone to overfitting. They excel in environments where massive amounts of data are available to learn from, which allows them to finely tune their parameters without latching onto noise or spurious patterns.

The suboptimal performance of complex MLP and Transformer structures in our case underscores a fundamental principle of machine learning: data availability constrains model choice. Where data is scarce, the intricate architectures designed to harness vast datasets become a liability rather than an asset, leading to models that perform well on training data but fail to generalize to unseen examples.

This experience has been a clear reminder of the need to adapt our modeling approach to the size and nature of the dataset at hand. It has affirmed the value of parsimonious models in situations where data is limited and has reinforced the importance of feature engineering and domain expertise. As we move forward, these insights will serve as a guide, helping us to better navigate the trade-offs between model complexity and data availability in pursuit of the most predictive and robust models for our projects.

6 Leaderboard Score



132 Pangolin 0.11885 38

Your Best Entry!
Your most recent submission scored 0.11885, which is an improvement of your previous score of 0.11959. Great job!

Figure 11: Leaderboard Score

References

[1] Erik Bruin. House prices: Lasso, xgboost, and a detailed eda. https://www.kaggle.com/code/erikbruin/house-prices-lasso-xgboost-and-a-detailed-eda.

[2] Lavanya Shukla. How i made top 0.3 https://www.kaggle.com/code/lavanyashukla01/how-i-made-top-0-3-on-a-kaggle-competition.