Introduction to Machine Learning Project
Group: 6

Manjeet 24AI06007, Souvik Karmakar 24AI06015
Anirban Dey 24AI06019, Swarnava Bose 24CS06014, Dinesh Soni 24RA06001

0) Problem definition :

In the real estate market, accurate price forecasting is crucial for investors, buyers, and sellers. Factors that impact home values include location, size, condition, and market trends. This project aims to develop a robust machine learning model that predicts home values based on a set of attributes from the dataset.

The primary goal of this Project is to develop an precise and dependable forecasting algorithm that estimates property prices based on historical data. Through adherence to the assessment metric, Root mean Squared Error (RMSE), the model aims to lower prediction errors and achieve a competitive score on the Kaggle hierarchy.

1) Data preparation and analysis

Data Cleaning :-

The dataset contains missing and outlier values that needed to be addressed before model training. The following steps were undertaken:

1. Outlier detection and removel

Using scatterplots and Z-scores, several outliers were identified and removed to prevent distortion in the model's predictions. Some key findings:

• Houses with extremely high LotArea values (above 55,000) or GrLivArea (above 4,400) significantly impacted the target variable, SalePrice.

• Specific IDs were flagged as outliers and excluded from the dataset.

2. Handling Missing Values:

• Features such as Alley, Fence, MasVnrType, FireplaceQu, and GarageCond contained missing categorical data. Missing values were replaced with a place-

holder value ("No") to indicate the absence of the feature.

• Numerical columns such as MasVnrArea and LotFrontage were filled with 0, indicating no measurable value.

• GarageYrBlt, highly correlated with YearBuilt, was analyzed and appropriately filled to maintain consistency.

Exploratory Data Analysis (EDA)

Exploratory analysis was conducted to understand the relationships between various features and the target variable (SalePrice). The following insights were derived:

1. Numerical Features:

• Strong correlations were observed between OverallQual (overall material and finish quality) and SalePrice.
• Features such as LotArea and YearBuilt showed moderate relationships, with some outliers affecting the trends.

2. Categorical Features:

• Using boxplots, categorical features such as Alley, Fence, MasVnrType, and FireplaceQu were found to have distinct group effects on SalePrice.
• GarageType and GarageCond categories showed variability in prices, emphasizing the importance of garage-related features.

Example :1

The Alley feature in the dataset represents the type of alley access to a property. It is a categorical feature with values like Grvl (Gravel), Pave (Paved), and missing values representing properties without alley access.
Upon analyzing the dataset, it was found that the Alley feature had 1369 missing values out of 1460 rows.

$\Longrightarrow$ Solution: Instead of dropping the feature or the rows with missing values, we imputed the missing values with a placeholder value, No, to indicate that the property lacks alley access. This approach ensures retention of the feature and preservation of all data points.

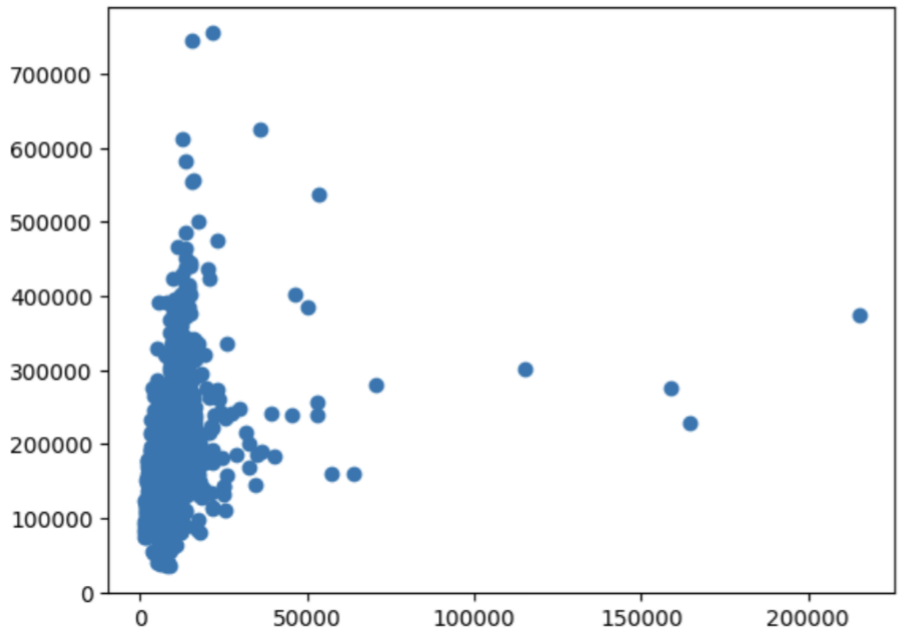
Example :2

The LotArea feature represents the total area of the property in square feet. This numerical feature is critical for determining house prices, as larger properties typically correlate with higher values.
Upon analyzing the LotArea feature, several extreme outliers were identified

that could distort the relationship between LotArea and SalePrice. Using scatterplots and statistical methods :

```
plt.scatter(x='LotArea', y='SalePrice', data=train_df)
```

```
[9]: <matplotlib.collections.PathCollection at 0x7b2f3a0e6bf0>
```



Outliers in the LotArea feature were identified using scatterplots and Z-scores. Properties with unusually large areas, specifically those exceeding 55,000 square feet, were flagged as extreme outliers. For example, rows with Id values 250, 314, 336, and 707 showed abnormally high LotArea values that distorted the relationship with SalePrice. These outliers were removed by filtering them out using their Id values.

2) Machine Learning model used

This project employs a comprehensive ensemble of machine learning algorithms to predict house prices. To attain better predictive performance, the pipeline preprocesses the data, trains several models, optimizes hyperparameters, and applies sophisticated ensemble strategies like voting and stacking. A thorough description of the models and methodology used is provided below.

1. Individual Models :

a) Linear Regression:

Linear regression served as a baseline model. It established the fundamental relationship between features and SalePrice. While simplistic, it helped identify potential feature engineering opportunities. This model gave moderate performance limited by its inability to capture non-linear relationships.

b) Random Forest Regressor:

A robust ensemble method that uses decision trees. Random Forests are adept at handling non-linearity and feature interactions. max_depth and n_estimators are the hyperparameters tuned for better results.

```
RFR = RandomForestRegressor(random_state=13)
```

+ Code + Markdown

[143]:
```
param_grid_RFR = {
    'max_depth': [5, 10, 15],
    'n_estimators': [100, 250, 500],
    'min_samples_split': [3, 5, 10]
}
```

[144]:
```
rfr_cv = GridSearchCV(RFR, param_grid_RFR, cv=5, scoring='neg_mean_squared_error', n_jobs=-1)
```

c) XGBoost Regressor:

An efficient gradient boosting algorithm widely used for structured data. It focuses on reducing bias and variance through iterative improvements.Achieved highly competitive performance with excellent feature importance insights.

d) Gradient Boosting Regressor:

This algorithm builds trees sequentially, optimizing for errors from previous iterations very effectively with strong predictive power.

d) Ridge Regression :

A linear model with L2 regularization to address multicollinearity and overfitting. It was tuned for various regularization strengths and solvers.

2. Ensemble Techniques :

To enhance predictive accuracy, ensemble methods were used to combine the strengths of individual models.

a) Voting Regressor :

The Voting Regressor averaged predictions from the top-performing models (Gradient Boosting Regressor, XGBoost, and Ridge), weighted based on their performance. Improved over individual models due to reduced variance.

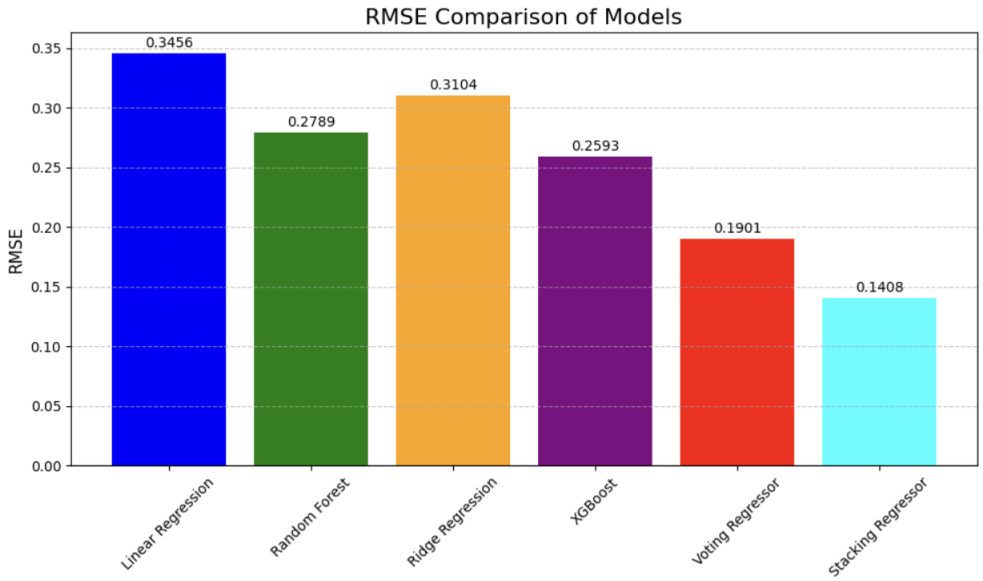
b) Stacking Regressor :

A sophisticated ensemble technique where predictions from multiple models (Gradient Boosting Regressor, XGBoost and Random Forest) were used as input for a final estimator (Voting Regressor). This hierarchical approach leveraged complementary strengths of models.

3. Approach to Stacking :

Five machine learning models (Linear Regression, Random Forest Regressor, Ridge Regression, and XGBoost Regressor, Ridge Regression) were trained on the dataset.
The Voting Regressor, which combines predictions from the base models using a weighted average. This design ensures that the strengths of both voting (stability) and stacking (non-linear combinations) are utilized. The Stacking Regressor, with the Voting Regressor as its final estimator, combines the predictions from all base models to make the final prediction.

2) Result analysis



The findings reveal a notable enhancement in model performance as we transitioned from individual models to ensemble methods. Among the individual models, XGBoost recorded the lowest RMSE of 0.2593 e8, highlighting its proficiency in identifying complex, non-linear relationships. The Voting Regressor further reduced the RMSE to 0.1901 by integrating the strengths of the individual models through weighted averaging, thereby minimizing variance.

Ultimately, the Stacking Regressor achieved the most favorable RMSE of 0.1408 by utilizing hierarchical combinations of predictions, effectively capturing the interactions among models. This underscores the efficacy of ensemble methods in improving predictive accuracy. The substantial performance improvements validate the effectiveness of the stacked ensemble strategy.

3) Future directions:

Advanced models like deep neural networks, convolutional neural networks, and Bayesian regression may capture more complex relationships. Additional ensemble techniques, such as custom weighted blending or meta-model stacking, could also boost performance. Automated hyperparameter tuning with Bayesian optimization and integrating AutoML frameworks might streamline model optimization.

4) Conclusion :

This project successfully predicted housing prices using various machine learning techniques, including preprocessing, feature engineering, and model ensembles. The stacking and voting ensembles, leveraging algorithms like XGBoost achieved superior accuracy by capturing complex relationships across features. This approach demonstrated that combining diverse models can significantly enhance predictive performance. While the final model produced promising results, future work could further improve accuracy by exploring deep learning models, refined hyperparameter tuning, and additional feature engineering. Incorporating explainability tools would also increase model transparency, making it more practical for real-world applications in housing market analysis.