SIVARAMAN LAKSHMIPATHY

Kaggle competition

Programming language used: Python
Environment: Jupyter
Kaggle competition link:
https://www.kaggle.com/c/house-prices-advanced-regression-techniques
Dataset: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

Use-case description:

- The competition is centered around the prediction of house sale prices.
- Two sets of data, for training models and to test the model, are provided.
- Submissions are evaluated based on root-mean-square-error (RMSE) values.

Environment details:

- The project utilizes the Python programming language.
- Jupyter environment is utilized and the corresponding notebook is attached in its original (executable) form as well as a PDF form.

Steps to execute the submission:

1. Activate the Jupyter environment with the following packages installed:
    a. Pandas
    b. Numpy
    c. Sklearn (scikit-learn)
2. Load the Python notebook (.ipynb) and run the cells sequentially to observe the results.

Steps involved in solving the challenge:

1. Data Acquisition
2. Data cleaning and pre-processing
3. Data transformations
4. Model generation and evaluations
5. Final results

1. Data Acquisition:
The data is obtained from Kaggle using the following link: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

The following observations are obtained:

1. The size of the training data is (1460, 81) => 1460 rows and 81 attributes.
2. The size of the test data is (1459, 80) => 1459 rows and 80 attributes.

2. Data cleaning and pre-processing:

1. Dropping columns: The target column is extracted separately and dropped in the dataset. Additionally, the column ‘Id’ is dropped in both the training and testing datasets.

2. Identifying the categorical and numerical columns from the data description.
3. The percent of rows with missing/null values is calculated for each column. All columns with the percent value greater than a threshold (50%) are identified to be dropped from the dataset.

3. Data transformations:

The training and testing datasets are transformed as described below:

1. All columns with the percent of rows with missing/null values greater than the threshold are dropped.
2. All the categorical columns are converted into separate columns for each categorical value per category using a one-hot encoding. The original columns are dropped from the datasets.
3. The missing values are filled in using the mode and median values of the corresponding columns.
4. Additionally, the following transformations are applied:
    a. The age of the house at the time of sale is calculated and added as a new column.
    b. The years since the remodel is calculated and added as a new column.
    c. The age of the garage is also calculated similarly and added as a new column.
5. Finally, the year of sale is converted to string values in order to avoid it being treated as a numerical value and weighted accordingly.

4. Model generation and evaluations:

The target column consists of sale prices of the houses. This leads to the problem being labeled a regression problem. Thus, regressor models are generated and evaluated by submitting the predictions in Kaggle.

1. SVM with RBF kernel: The SVM overfits the data but performs poorly on the test data.

2. Random forest: The random forest regressor produced the second best score.

3. Gradient Boosting: An ensemble method, generated poor results initially.

4. Gradient boosting after hyperparameter tuning using GridSearch: Performed the best amidst all the generated models.

5. Lasso regression: It is an advanced regression technique which also produced good results.

Model	Kaggle score
SVM with RBF kernel	0.44840
Random forest	0.15896
Gradient Boosting	0.62641
Gradient Boosting with hyperparameter tuning using GridSearch	0.15225
Lasso regression	0.18872

5. Final results:

The best performing model is the Gradient Boosting model (with hyperparameter tuning) with a Kaggle score of **0.15225**.