AUTUMN - 2024

Department of Computer Sc. & Engineering,
School of Electrical and Computer Sciences,
IIT Bhubaneswar

HOUSE PRICE PREDICTION USING ADVANCED REGRESSION TECHNIQUES

KAGGLE PROJECT REPORT

TEAM - IML_GROUP_7_IITBBS

1. Chandan Keshari (24CS06022)
2. Subham Sanket Rout (24CS06017)
3. Ishan Chauhan (24CS06013)
4. Naveen Kumar (24CS06011)
5. Deep Mandal (24CS06007)

UNDER GUIDANCE

Dr. Ashwini Nanda
Dr. Narayana Darapaneni

TABLE OF CONTENTS

PROBLEM STATEMENT

In real estate, knowing what affects house prices is important for buyers and sellers. While homebuyers often picture their dream homes based on personal likes like design, size, and location, many factors actually impact property values. This project will explore these factors using a detailed dataset with 79 features related to residential properties in Ames, Iowa.

The main goal is to predict the final sale price of homes based on these different features. Important factors include the age of the property, its size, the number of bedrooms and bathrooms, and more subtle details like construction quality and neighborhood amenities. Traditional price estimation methods often miss these details, which can create gaps between what buyers expect and the actual market.

This project aims to create a strong predictive model using advanced regression methods to analyze historical housing data and predict prices accurately. By using different machine learning techniques, this research will clarify how various features relate to house prices and help stakeholders make better decisions.

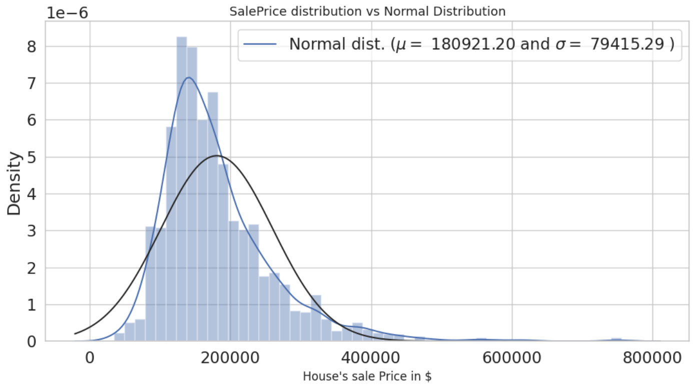
DATA PREPARATION

OVERVIEW OF DATA

- **Source:** The dataset provided from the Kaggle that contains housing data from the state of Ames, Iowa.
- **Shape:**
    - Number of rows in training dataset: ***1460***
    - Number of features in training dataset: ***80*** (including 1 Target Value)

EDA & VISUALIZATION

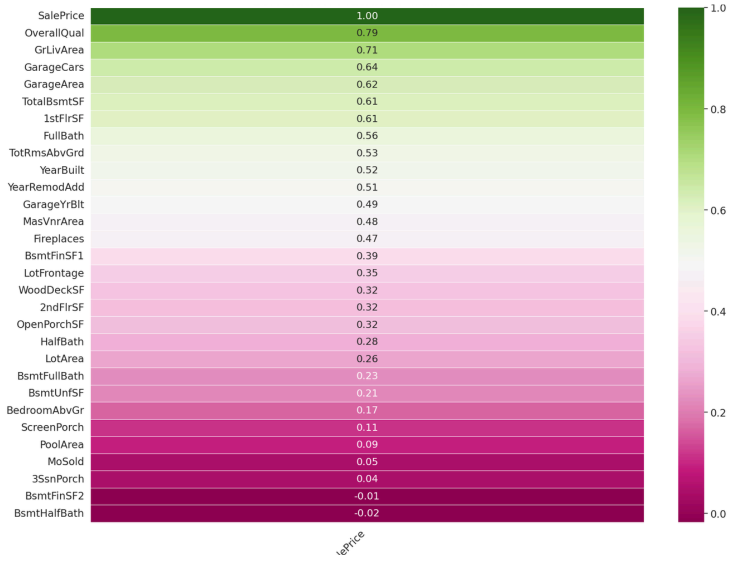
Before working with data, it is important to understand it. A key step in this process is Exploratory Data Analysis (EDA). EDA uses visualizations and statistical analysis to help us understand the data better and see how different features relate to each other. Now, let's look at our target variable and see how the other features affect it.



Skewness values between -0.5 and 0.5 are acceptable, and kurtosis values should be between -2 and 2. The plot shows a highly right-skewed distribution, and the Shapiro test for normality also confirms this with a very small p-value.

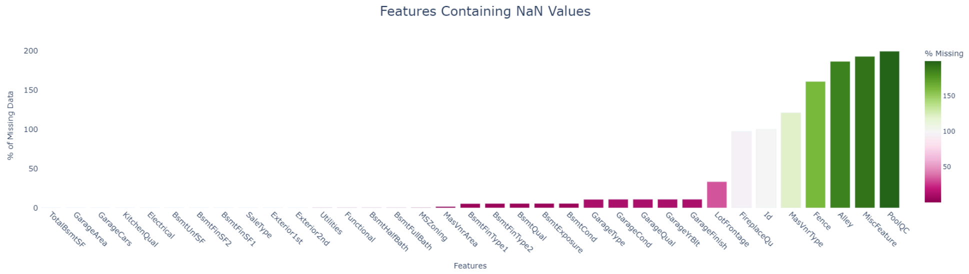
The correlation matrix is the best way to see all the numerical correlation between features.

- **Correlation Among Features and Target Variable:** Based on our analysis from the correlation among features & target variable, we have found that out of 79 features, these are the top 20 features which have the most influence on the target variable '***SalePrice***'.



DATA PREPROCESSING & CLEANING

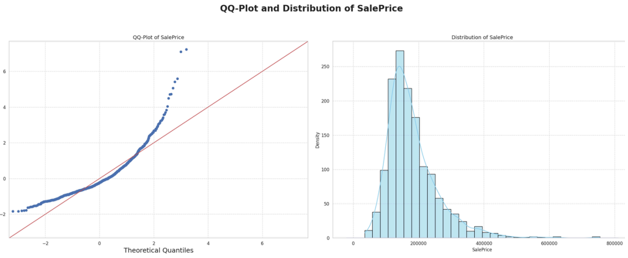
- **Dealing with NaN Values:**
  - First, we computed the % of missing values(NaN) for each features in the dataset. For some of the categorical features, we imputed them with appropriate values by observing the dataset.
  - For numerical features containing NaN values, we filled them using KNN regression.
  - Furthermore, we dropped those features whose % of NaN values exceeded 20%.
  - Removed useless variables that don’t provide meaningful information for modeling.



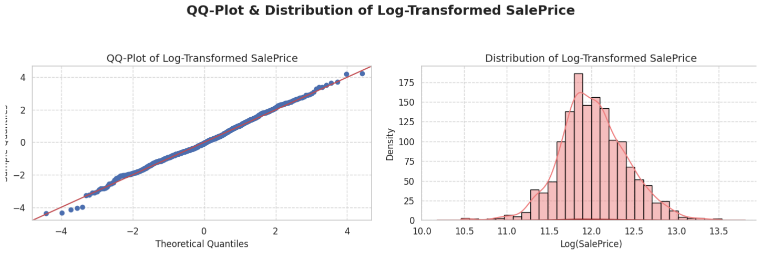
- **Detection & Removal of Outliers:** Outliers can skew analysis and negatively impact the accuracy of ML models, so we detected them using ***Tukey IQR*** method, and dropped the corresponding rows.

FEATURE ENGINEERING

- We have created several new features (‘***SqFtPerRoom***’, ‘***Total_Home_Quality***’, ‘***Total_Bathrooms***’, and ‘***HighQualSF***’) that enhances predictive power of our model by captures additional aspects of the data that might be relevant to the target variable, this also includes handling non-numeric data, creating dummy variables for One-Hot encoding, and handling skewed features by applying log transformation.
- We also applied log transformation to the target variable to make its distributions more symmetric & to stabilize variance, making it suitable for regression modeling.



SalePrice Before Log Transformation

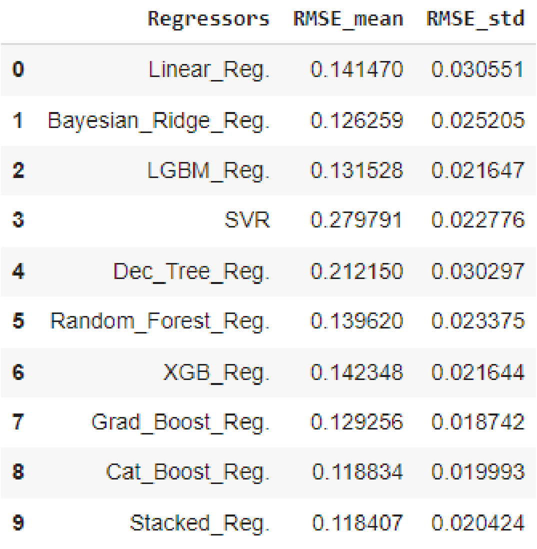


SalePrice After Log Transformation

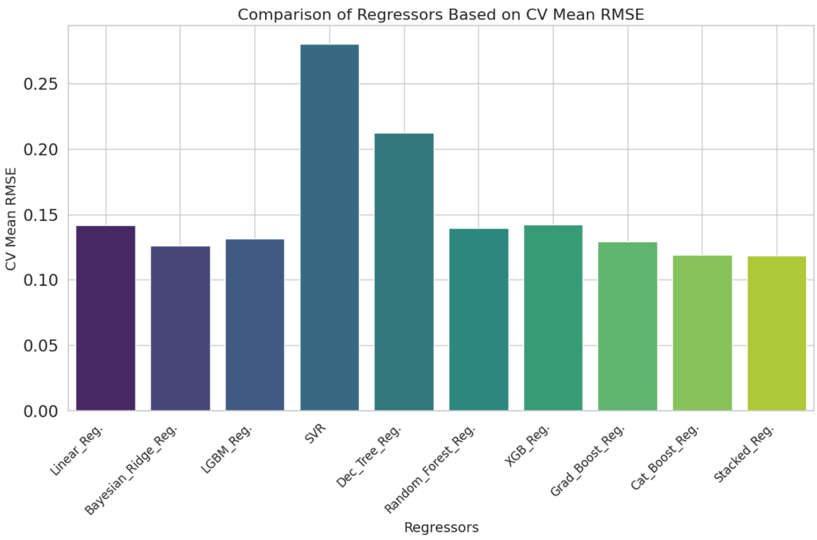
REGRESSION MODEL TRAINING

TRAINING DATA ON VARIOUS REGRESSION MODELS

- **10-Fold Cross Validation:** We have used this to assess the generalization performance of multiple regression models. This helps to mitigate overfitting by ensuring that the model's performance is not dependent on a single train-test split, providing a more robust estimate of how well the model will perform on unseen data.
- **Model Selection & Evaluation:** We trained the following regression models on our preprocessed training data,
    - Linear Regression
    - Bayesian Ridge Regression
    - LightGBM Regressor (LGBM)
    - Support Vector Regression (SVR)
    - Decision Tree Regressor
    - Random Forest Regressor
    - XGBoost Regressor (XGB)
    - Gradient Boosting Regressor
    - CatBoost Regressor
    - Stacked Regressor
- **Performance Comparison Among Models:** Below table gives the information of the mean RMSE across the 10-fold and Standard deviation of RMSE of the different regression model used.



	Regressors	RMSE_mean	RMSE_std
0	Linear_Reg.	0.141470	0.030551
1	Bayesian_Ridge_Reg.	0.126259	0.025205
2	LGBM_Reg.	0.131528	0.021647
3	SVR	0.279791	0.022776
4	Dec_Tree_Reg.	0.212150	0.030297
5	Random_Forest_Reg.	0.139620	0.023375
6	XGB_Reg.	0.142348	0.021644
7	Grad_Boost_Reg.	0.129256	0.018742
8	Cat_Boost_Reg.	0.118834	0.019993
9	Stacked_Reg.	0.118407	0.020424

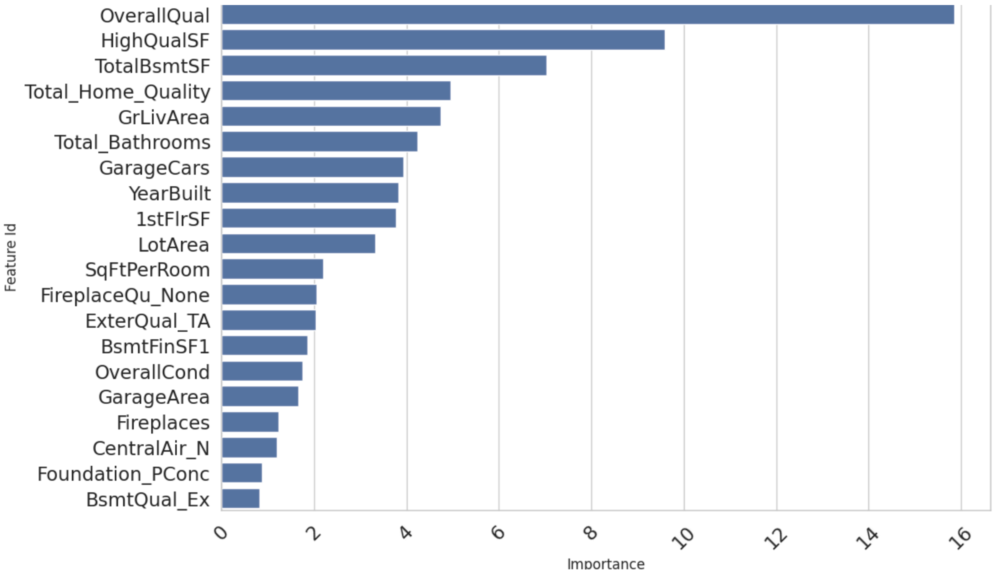


The RMSE measure is used here to quantify the prediction error and how a lower RMSE indicates a better performing model. Here, Standard deviation of RMSE is also used because it indicates the consistency of the models across different data splits, with smaller standard deviation suggesting more stable models.

By analyzing above data, we found that "***CatBoostRegressor***", has low RMSE & low standard deviation.

FINAL MODEL - CATBOOST REGRESSOR

- **Why CatBoost?:**
  - Among all the regression models used so for, we chose the "***CatBoostRegressor***" model because of it's low mean & standard deviation of RMSE.
  - CatBoost is often preferred for its handling of categorical variables without needing explicit encoding (like one-hot encoding) and its robustness in dealing with missing values.
  - It typically provides strong performance even with default hyperparameters, making it a good starting point for regression tasks.
- **Model Choice:** We chose the CatBoost Regressor since it is a powerful gradient-boosting algorithm that works well with both categorical and numerical features. CatBoost stands for "Categorical Boosting," as it automatically handles categorical data, reducing the need for manual encoding.
- **Model Training**:
  - The `***CatBoostRegressor()***` model is initialized and trained on the training data (`***X_train***` and `***y_train***`) using the `***fit()***` method.
  - During training, the model is evaluated on the validation data (`***eval_set = (X_val, y_val)***`). This allows the model to track its performance on unseen data during training, helping to avoid overfitting.
  - The `**verbose=0**` parameter suppresses the output of training details, making the training process more concise and less cluttered in the output.
- **Most Important Features:** Below are the top 20 most important features for our model, which help us understand how the model works on unseen data to make its final prediction.



RESULT ANALYSIS

- After analyzing the house sale price data, our findings confirm that while the dataset approximates a ***normal distribution***, there is some skewness, which we mitigated using log transformations.

- Applying a ***10-fold cross-validation*** on multiple regression models allowed us to identify models with strong performance.

- In particular, the ***CatBoost Regressor*** models yielded lower RMSE (Root Mean Squared Error) values, suggesting a better fit for the dataset. These transformations and model selections improved prediction accuracy and reduced the prediction error variance.

- Through our data preprocessing and transformation steps, we addressed the initial data distribution's skewness and handled missing values as well as outliers to enhance data quality. Data visualization of feature correlations provided insights into underlying data patterns.

- In cross-validation, ensemble models, specifically the CatBoost Regressor model, outperformed other models by capturing diverse data patterns. This approach indicates that model ensembles may be preferable for datasets with complex relationships, as they offer greater stability and accuracy in predictions.

- K**aggle Score:** Our submission of '.csv' file containing predictions of 'SalePrice' of test data, on Kaggle obtained a score of ***0.04987.***

This analysis applies to predictive modeling in other domains where variance in data distribution and feature relationships can affect model performance.

CONCLUSION & FUTURE DIRECTION

CONCLUSION

- The prediction of house sale prices using various data preprocessing techniques and advanced regression models provided valuable insights into the dataset’s characteristics and the effectiveness of different models.

- By addressing data skewness, handling missing values and outliers, and utilizing feature engineering, we prepared a robust dataset that enabled us to achieve more accurate predictive models.

- Cross-validation revealed that ensemble models, particularly CatBoost Regressor, performed best in terms of minimizing RMSE and providing stable, consistent results.

- This highlights the benefit of ensemble methods in capturing complex patterns and minimizing predictive errors, making them suitable for real-world applications in housing price prediction.

FUTURE DIRECTIONS

For future analysis, several improvements can be explored to further enhance model accuracy and adaptability.

- Firstly, experimenting with additional feature engineering techniques, such as incorporating location-based features or economic indicators, may improve model performance.

- Secondly, employing more sophisticated hyperparameter tuning methods, such as grid search or Bayesian optimization, could further optimize model parameters.

- Additionally, exploring deep learning techniques, such as neural networks or hybrid models, may reveal even more nuanced patterns in the data.

- Finally, integrating these models into a live prediction system with regular updates could provide continuous, real-time predictions, making the models more applicable to dynamic real estate markets.