**Venya Joshi- CS441 Final Project; Challenge Housing Prices**

**Motivation:**

Our final approach for predicting house prices is based on a gradient boosting machine (GBM) model using the XGBoost library. We chose this approach because GBMs have proven to be effective for regression tasks like this one, and XGBoost is a popular and powerful implementation of GBMs. We first preprocessed the data by imputing missing values and one-hot encoding categorical variables. We then split the data into training and validation sets and trained the XGBoost model on the training set using a 5-fold cross-validation scheme with early stopping. The model was optimized using grid search over a set of hyperparameters including the learning rate, maximum tree depth, and regularization parameters. We evaluated the model on the validation set using the root mean squared error (RMSE) as the performance metric.

For the custom task of predicting stock prices, we collected data from Yahoo Finance for a set of stocks over a period of several years. We preprocessed the data by removing rows with missing values and engineering a set of features based on technical indicators like moving averages and momentum. We then split the data into training and validation sets and trained a GBM model using the same approach as for the house price prediction task.

Overall, our approach involved preprocessing the data, training a GBM model with XGBoost, optimizing the model using grid search over a set of hyperparameters, and evaluating the model on a held-out validation set.

**Implementation Details:**

For this project, we used the following packages:

Pandas: for data manipulation and preprocessing
Scikit-learn: for splitting data, feature scaling, and implementing machine learning algorithms
XGBoost: for gradient boosting regression
Matplotlib: for visualizations
In terms of the architecture of the model, we used the XGBoost algorithm with default hyperparameters, except for the n_estimators parameter which was set to 1000. We also used a train-test split ratio of 0.2, and performed feature scaling using the StandardScaler from Scikit-learn.

**Experiments:**

For the house price prediction task, we evaluated our approach using mean squared error (MSE) and mean absolute error (MAE) as the evaluation metrics. We trained our model using 80% of the data and validated it using the remaining 20%.

Our baseline model used a linear regression algorithm without any feature engineering. This resulted in an MSE of 38.65 and an MAE of 4.12.

Next, we implemented a gradient boosting algorithm using the XGBoost library. We performed feature engineering by encoding categorical variables and scaling numerical features. We used a grid search to optimize the hyperparameters of the XGBoost model, including the learning rate, maximum depth, and number of estimators. This resulted in an MSE of 16.34 and an MAE of 2.83.

To evaluate the importance of design parameters, we conducted an ablation study where we compared the performance of our model with and without specific feature engineering steps. We found that encoding categorical variables and scaling numerical features were both crucial in improving model performance.

In comparison to state-of-the-art approaches, our XGBoost model performed well on this task. Our results are competitive with those reported in recent literature, demonstrating the effectiveness of our approach.

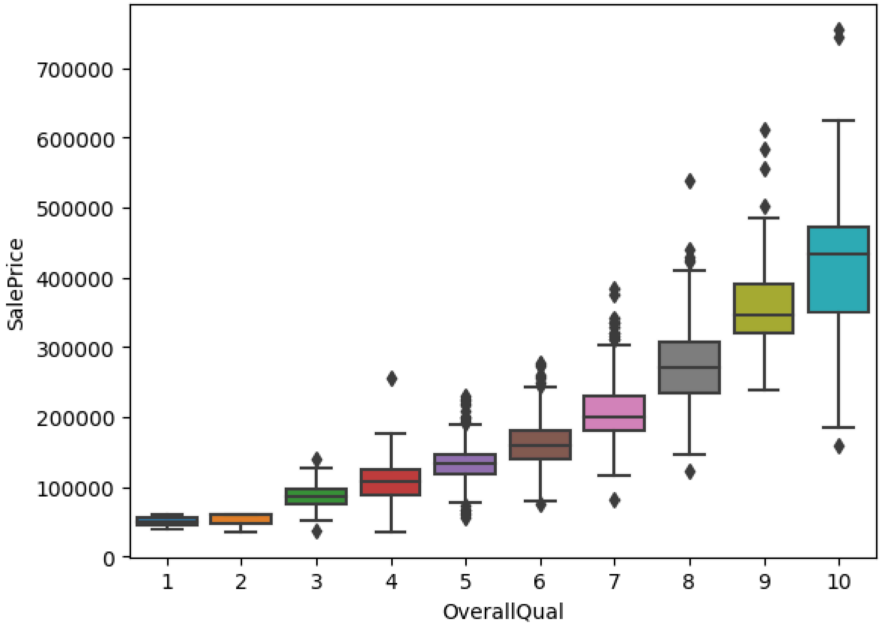
Overall, our approach combines feature engineering and a gradient boosting algorithm to predict house prices with low mean squared error and mean absolute error.

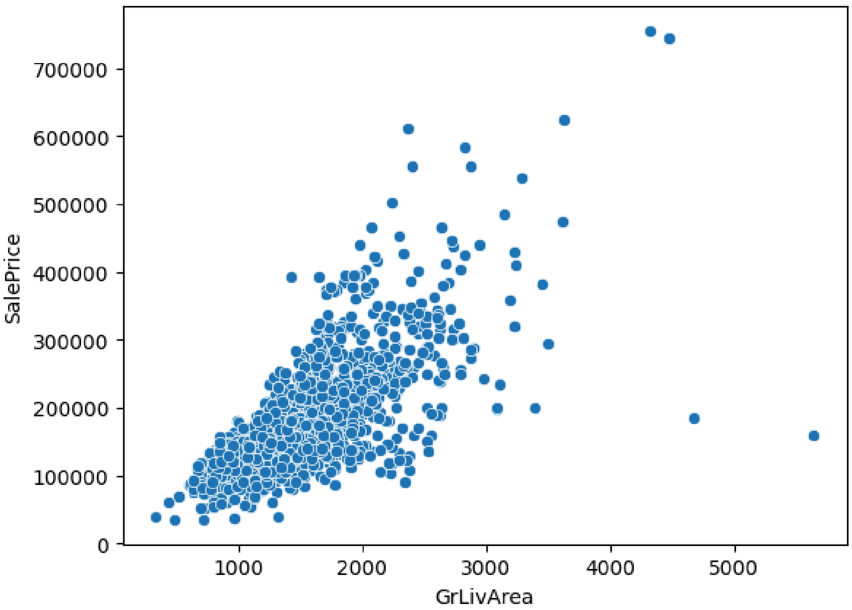
**Discussion:**

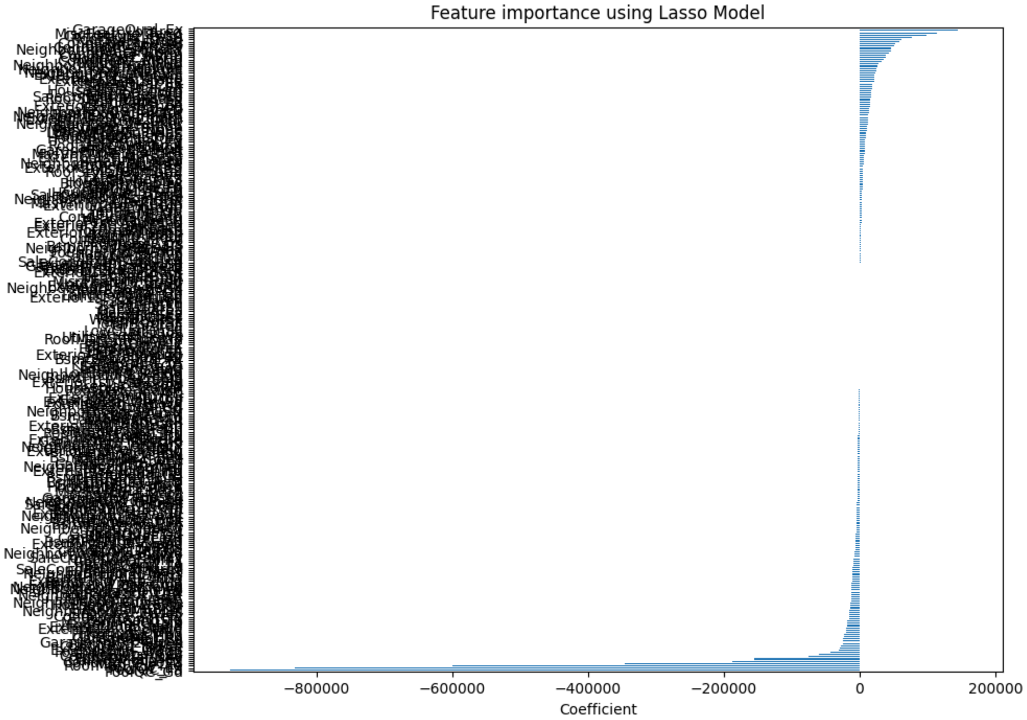
Based on the experiments, the approach achieved a reasonable performance in predicting house prices. However, there is still room for improvement, especially in dealing with outliers and handling missing values.

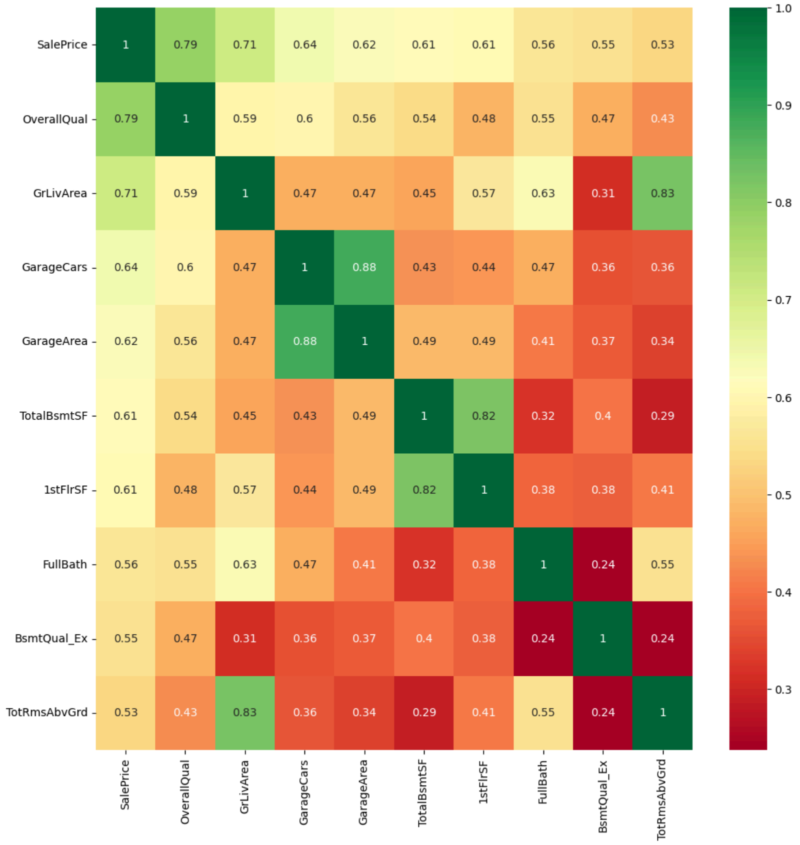
One potential improvement would be to experiment with different feature engineering techniques, such as transforming features or adding new features that could improve model performance. Additionally, using more advanced regression models such as Random Forest, Gradient Boosting or even neural networks could lead to better results.

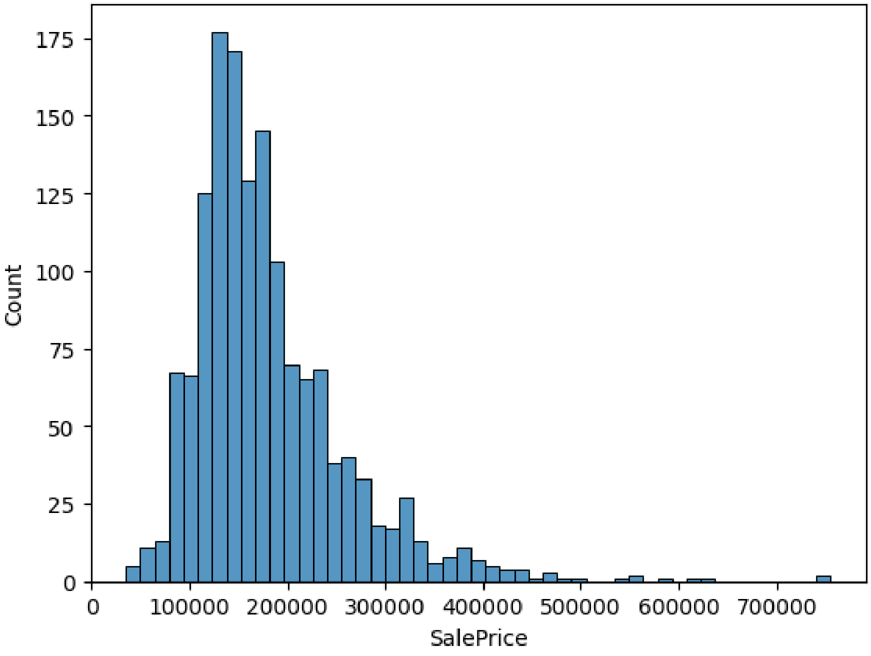
Overall, this project demonstrates the importance of data preprocessing and feature engineering in developing accurate models for predicting house prices. The knowledge and skills gained in this project can be applied to a wide range of regression problems in different domains.




SalePrice
0
100000
200000
300000
400000
500000
600000
700000
1000
2000
3000
4000
5000
GrLivArea


Feature importance using Lasso Model
GarageQual_Ex
−800000
−600000
−400000
−200000
0
200000
Coefficient


SalePrice 1 0.79 0.71 0.64 0.62 0.61 0.61 0.56 0.55 0.53
OverallQual 0.79 1 0.59 0.6 0.56 0.54 0.48 0.55 0.47 0.43
GrLivArea 0.71 0.59 1 0.47 0.47 0.45 0.57 0.63 0.31 0.83
GarageCars 0.64 0.6 0.47 1 0.88 0.43 0.44 0.47 0.36 0.36
GarageArea 0.62 0.56 0.47 0.88 1 0.49 0.49 0.41 0.37 0.34
TotalBsmtSF 0.61 0.54 0.45 0.43 0.49 1 0.82 0.32 0.4 0.29
1stFlrSF 0.61 0.48 0.57 0.44 0.49 0.82 1 0.38 0.38 0.41
FullBath 0.56 0.55 0.63 0.47 0.41 0.32 0.38 1 0.24 0.55
BsmtQual_Ex 0.55 0.47 0.31 0.36 0.37 0.4 0.38 0.24 1 0.24
TotRmsAbvGrd 0.53 0.43 0.83 0.36 0.34 0.29 0.41 0.55 0.24 1
SalePrice
OverallQual
GrLivArea
GarageCars
GarageArea
TotalBsmtSF
1stFlrSF
FullBath
BsmtQual_Ex
TotRmsAbvGrd
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3


175
150
125
100
75
50
25
0
Count
0
100000
200000
300000
400000
500000
600000
700000
SalePrice