UNIVERSITY OF MARYLAND

ROBERT H. SMITH SCHOOL OF BUSINESS

BUDT 758J
Final Project Report
Spring 2024

Team Name: Cloud Commanders UMD
Gautam Bhatia, Gunjan Suriya, Keva Chawla, Marcelo Chavarria, Tanmay Sakharkar

Team member names and contributions

a) Gautam:
- Data cleaning
- Feature Engineering
- Random Forest Model
- Lasso Model
- Knn Model
- Ridge Model
- XGBoost Model
- Decision Tree Model
- Report Writing

b) Gunjan:
- Data cleaning
- Feature Engineering
- Additional feature creation
- XGBoost Model
- AdaBoost Model
- Random Forest
- Report writing

c) Keva:
    - Data Cleaning
    - Exploratory Data Analysis
    - Feature engineering
    - Learning curve chart for XGBoost
    - Fitting Curve chart fort XGBoost
    - Boost Model Tuning
    - Report Writing

d) Marcelo:
    - Data Cleaning
    - Exploratory Data Analysis
    - Feature engineering
    - Code References
    - Comments explaining the code
    - Report writing

e) Tanmay:
    - Data Cleaning
    - Exploratory Data Analysis
    - Feature engineering
    - Code References
    - Comments explaining the code
    - Report writing
    - Project Learning

Table of Contents

Section 1: Business Understanding

Executive Summary:

In the dynamic real estate market, data-driven decision-making is pivotal for accurately assessing property values. Our project's primary objective is to develop a robust prediction model that leverages comprehensive data from the Ames Housing dataset to forecast the final sales price of residential homes. This model can significantly aid home buyers, sellers, real estate agents, and developers by providing precise pricing insights, enhancing negotiation strategies, and optimizing investment decisions.

To achieve this goal, we will employ advanced regression techniques such as Ridge, Lasso, Knn, Random Forest, Decision Trees, AdaBoost, and XGBoost. The evaluation metric used will be the Root Mean Squared Error (RMSE), which quantifies the average prediction error. By minimizing RMSE, we aim to create a model that delivers highly reliable price predictions.

These predictions along with EDA, can help property owners, brokers as well as buyers understand different metrics needed to procure housing at affordable prices. At the same time, the house owners can change different variations such as garage space, decorations,etc to increase house prices.

Section 2: Methods Used

The process began by exploring the data and identifying any missing values in the dataset. The next step was to consider how we wanted to deal with any missing information, which included options such as filling in 'NA', 0 or the mode of the column. Following this initial step, ordinal and categorical dictionaries were created, with the respective variable names serving as keys and their associated categories/levels as values. Additionally, the numerical variables were stored in a list, which allowed for proper organization and management of the different types of data present in the dataset. Next, we applied one-hot encoding to create different columns for each category in the original categorical variable. We also encoded the ordinal variables to map each value to a number and replaced the original column with the encoded one. Finally, we standardized the numeric variables.

Table 1: Feature names and a brief description of the data cleaning process

ID	Feature Name	Variable Type	Data Cleaning
1	MSZoning	Categorical	1. Used one-hot encoding to create different columns for each category in the original categorical variable 2. Gave binary values 0-1 based on the present category in the data point 3. Deleted the original categorical variable column from the train dataframe For missing values in: 1. Alley, MasVnrType, GarageType, MiscFeature : Filled with 'NA' 2. MasVnrType: Filled with 'None' 3. MSZoning, Electrical, Utilities, SaleType, Exterior1st, Exterior2nd : Filled with the mode of variable
2	Street	Categorical	
3	Alley	Categorical	
4	LotShape	Categorical	
5	LandContour	Categorical	
6	Utilities	Categorical	
7	LotConfig	Categorical	
8	Neighborhood	Categorical	
9	Condition1	Categorical	
10	Condition2	Categorical	
11	HouseStyle	Categorical	
12	RoofStyle	Categorical	
13	RoofMatl	Categorical	
14	Exterior1st	Categorical	
15	Exterior2nd	Categorical	
16	MasVnrType	Categorical	
17	Foundation	Categorical	
18	Heating	Categorical	

19	Electrical	Categorical	
20	GarageType	Categorical	
21	MiscFeature	Categorical	
22	SaleType	Categorical	
23	SaleCondition	Categorical	
24	MSSubClass	Categorical	
25	BldgType	Ordinal	1. Encoded the ordinal variables to map each category to a number as ordinal variables have an order/ rank to them 2. Replaced the original column with the encoded column For missing values in: 1. BsmtQual, BsmtFinType1, BsmtFinType2, BsmtCond, BsmtExposure, GarageFinish, GarageQual, GarageCond, MasVnrType, GarageType, MiscFeature, PoolQC, Fence : Filled with 'NA' 2. KitchenQual, Functional : Filled with mode of variable
26	OverallQual	Ordinal	
27	OverallCond	Ordinal	
28	ExterQual	Ordinal	
29	ExterCond	Ordinal	
30	BsmtQual	Ordinal	
31	BsmtCond	Ordinal	
32	BsmtExposure	Ordinal	
33	BsmtFinType1	Ordinal	
34	BsmtFinType2	Ordinal	
35	HeatingQC	Ordinal	
36	CentralAir	Ordinal	
37	KitchenQual	Ordinal	
38	Functional	Ordinal	
39	FireplaceQu	Ordinal	
40	GarageFinish	Ordinal	
41	GarageQual	Ordinal	
42	GarageCond	Ordinal	
43	PavedDrive	Ordinal	
44	PoolQC	Ordinal	
45	Fence	Ordinal	
46	LandSlope	Ordinal	
47	LotFrontage	Numerical	
48	LotArea	Numerical	
49	YearBuilt	Numerical	

50	YearRemodAdd	Numerical	1. Performed standard normalization on Numerical variables to improve predictions by eliminating scaling issues. For missing values in: 1. LotFrontage, BsmtFinSF1, BsmtFinSF2, BsmtUnfSF BsmtExposure, TotalBsmtSF, BsmtHalfBath, BsmtFullBath, GarageYrBlt, GarageCars, GarageArea: Filled with 0
51	MasVnrArea	Numerical	
52	BsmtFinSF1	Numerical	
53	BsmtFinSF2	Numerical	
54	BsmtUnfSF	Numerical	
55	TotalBsmtSF	Numerical	
56	1stFlrSF	Numerical	
57	2ndFlrSF	Numerical	
58	LowQualFinSF	Numerical	
59	GrLivArea	Numerical	
60	BsmtFullBath	Numerical	
61	BsmtHalfBath	Numerical	
62	FullBath	Numerical	
63	HalfBath	Numerical	
64	BedroomAbvGr	Numerical	
65	KitchenAbvGr	Numerical	
66	TotRmsAbvGrd	Numerical	
67	Fireplaces	Numerical	
68	GarageYrBlt	Numerical	
69	GarageCars	Numerical	
70	GarageArea	Numerical	
71	WoodDeckSF	Numerical	
72	OpenPorchSF	Numerical	
73	EnclosedPorch	Numerical	
74	3SsnPorch	Numerical	
75	ScreenPorch	Numerical	
76	PoolArea	Numerical	
77	MiscVal	Numerical	
78	MoSold	Numerical	
79	YrSold	Numerical	

In order to get better predictions, we engineered 7 new features relevant to the sale price of a house based on our domain knowledge. These helped enhance our model performance.

1. **'Baths'**: Calculated by summing the number of full baths and half baths from both the main level and the basement of the house. Half baths are weighted by multiplying them by 0.5 since they count as half of a full bath.
2. **'TotalSquareFeet'**: Represents the total square footage of various areas of the house, including the first floor, second floor, basement, above-ground living area, garage, wood deck, open porches, enclosed porches, three-season porches, screen porches, and pool area.
3. **'TotalPorchArea'**: Combines the square footage of open porches, enclosed porches, and three-season porches thus giving overall porch area.
4. **'RoomsabvgradeFunc'**: Represents the interaction between the number of rooms above grade and the functional status of the home to see the effect of quality with quantity.
5. **'HouseAge'**: Represents the age of the house, calculated by subtracting the year the house was built from the current year (2024).
6. **'BedroomToRoomRatio'**: Represents the ratio of bedrooms to total rooms above grade in each house, calculated by dividing the number of bedrooms above grade by the total number of rooms above grade.
7. **'TotalOutdoorArea':** Combines the areas of various outdoor amenities and structures, such as the decks, porches (open, enclosed, three-season and screened), and the pool areas.

These newly created features capture various aspects of the houses that may contribute to predicting sale prices. They consider factors such as the size, age, functionality, and composition of the houses, which are likely important in modeling their characteristics.

Section 2A: Exploratory Data Analysis (EDA)

To understand the structure and quality of our data better, we conducted EDA on select features. The analysis helped identify patterns and trends.

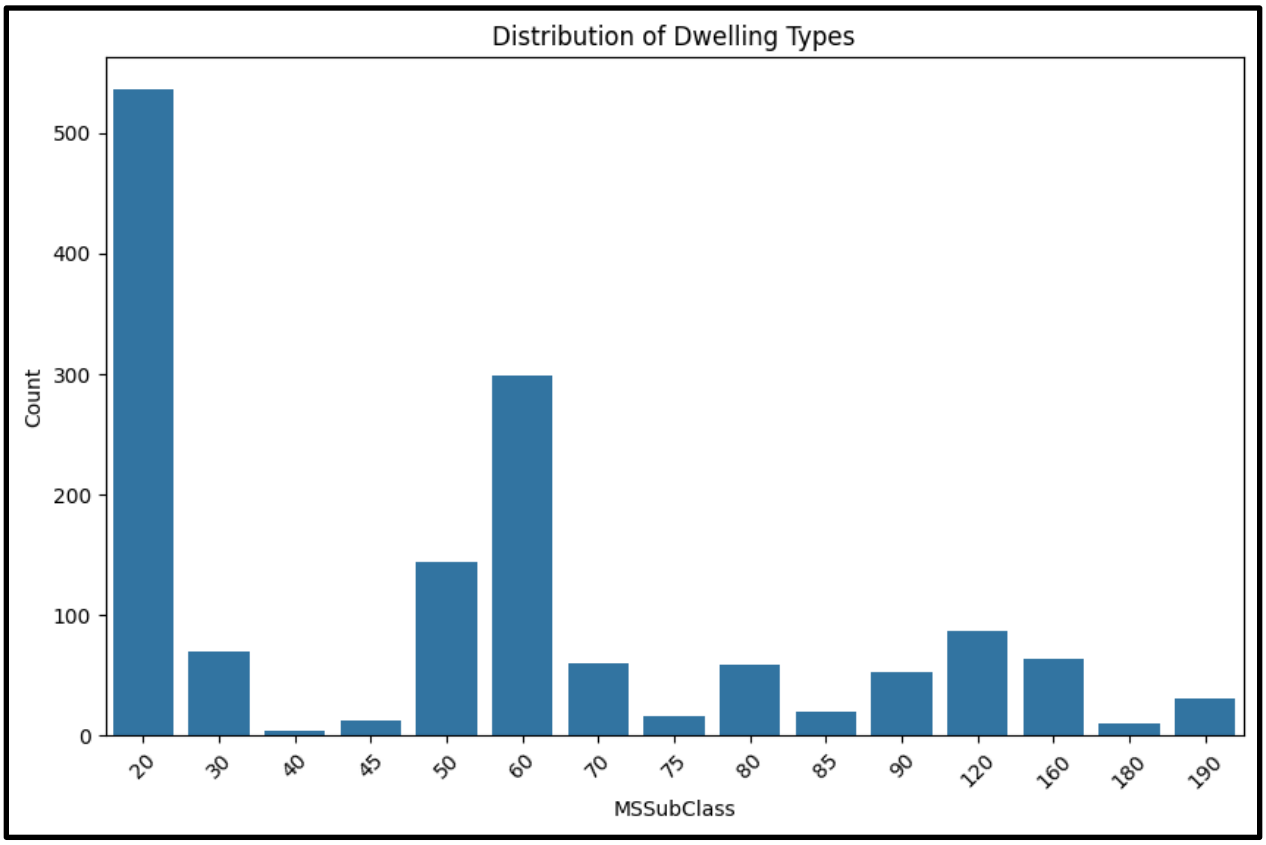


Figure 1

Distribution of Dwelling Types for Train Data

This chart illustrates the distribution of dwelling types based on MSSubClass. The most prevalent dwelling type is class 20 (1-STORY 1946 & NEWER ALL STYLES), accounting for over 500 properties. Following behind is class 60 (2-STORY 1946 & NEWER), and the third most abundant is class 50 (1-1/2 STORY FINISHED ALL AGES). This distribution suggests that the dataset mostly comprises newer homes built in 1946 and onwards.

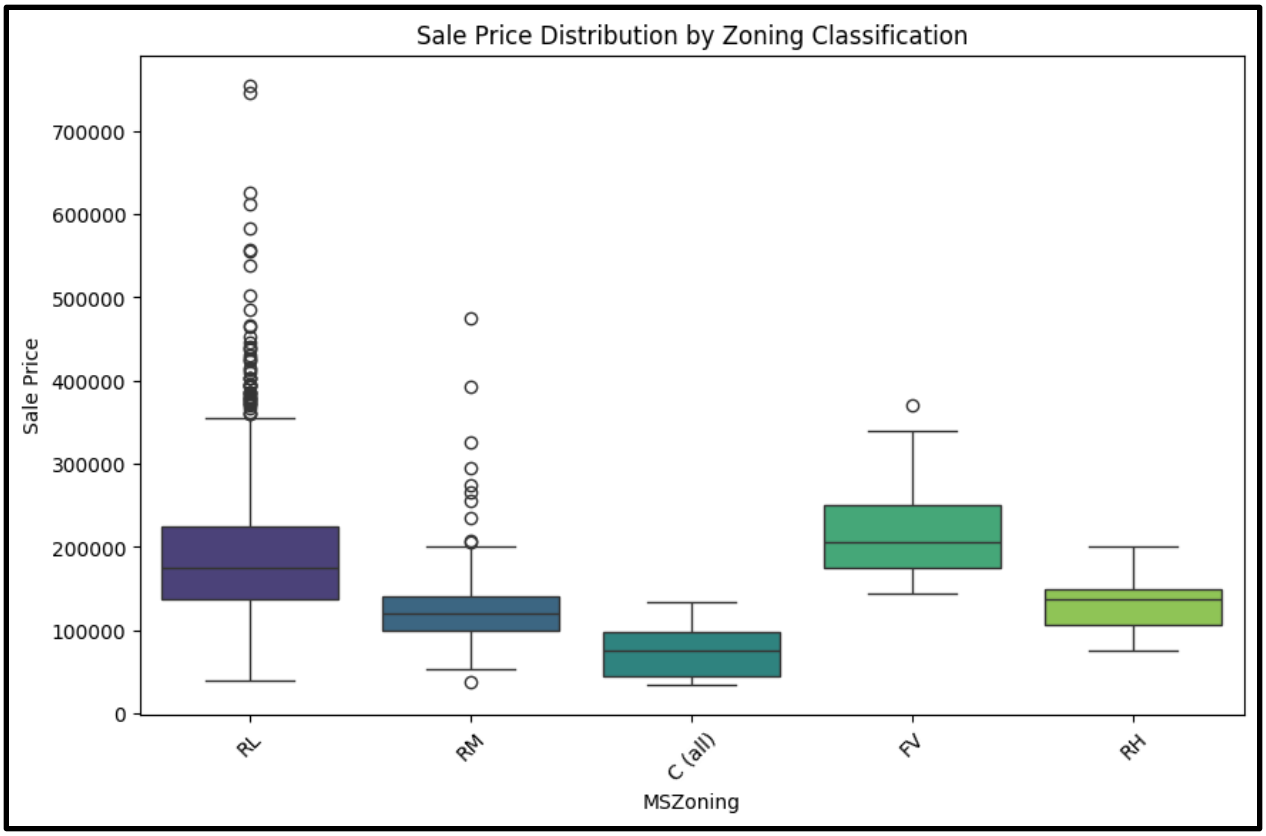


Figure 2

Sale Price vs Zoning Classification

The box plot provides insights into the median and quartile sales prices across various zoning classifications. 'RM' zoning shows the highest median sales price, indicating relatively higher property values within this category. Although 'RL' zoning also has a high median sales price, it contains some high outliers, suggesting potential data skewness. 'C' and 'FV' zoning classifications have the lowest median sales prices among the categories examined.

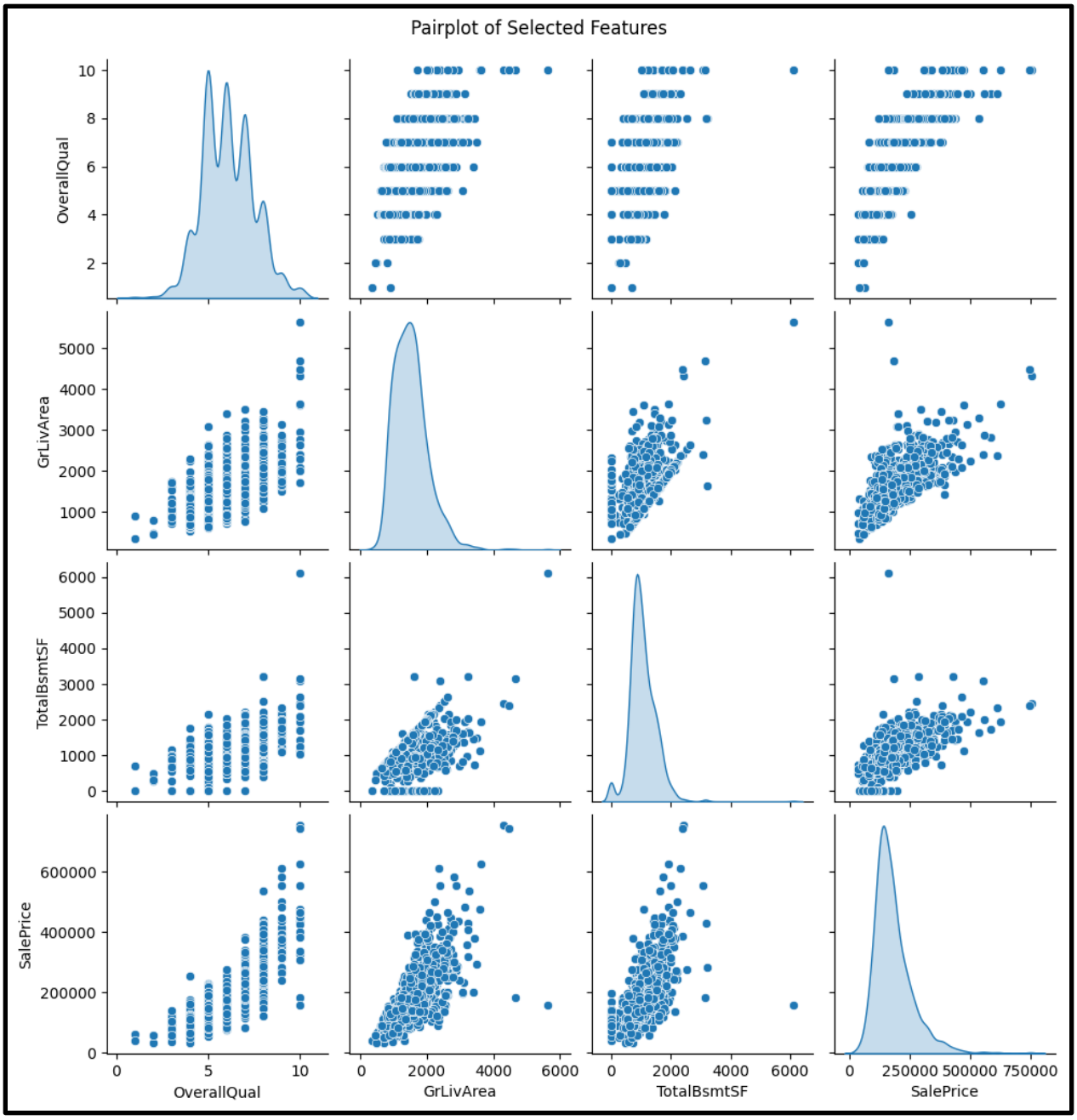


Figure 3
Pairplot

Pairplot from the selected features of OverallQual (rating the overall material and finish of the house), GrLiveArea (above-ground living area in square feet), TotalBsmtSF (total square feet of basement area), and SalePrice.

When we examine how the variables correlate with Sales Price specifically we find some interesting trends. We can see that there seems to be a positive correlation between the GrLivArea (ground living area) and the sales price. This may indicate that properties with larger ground living

areas tend to sell for higher prices. Additionally, we also see a positive correlation between OverallQual (the overall quality of the house) and the price of the house. This indicates that properties in better overall condition tend to sell for a higher price.

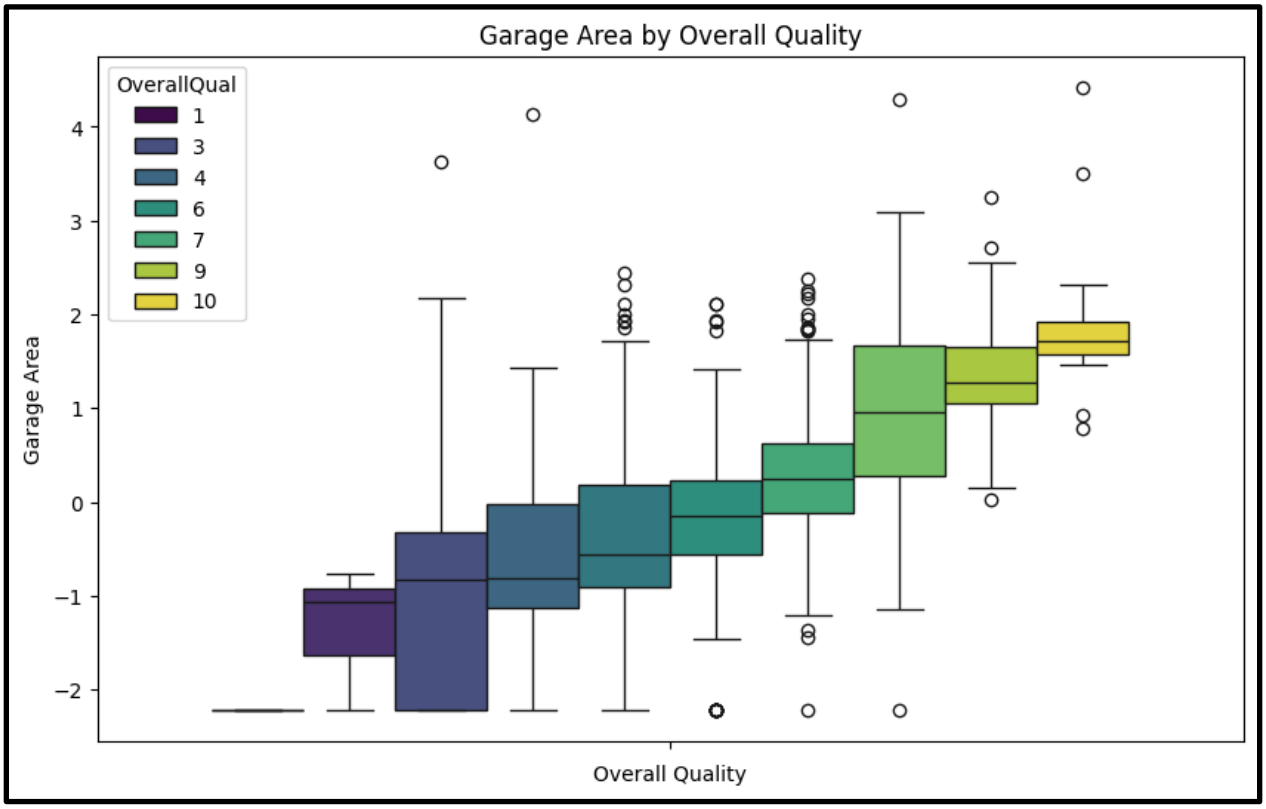


Figure 4
Garage Area by Overall Quality

The boxplot of the Garage area by the Overall Quality of the Properties illustrates a positive relationship between the quality of properties and their garage areas. This can indicate that higher quality homes are larger and can have more spacious garage areas. Also, the size of the garage might contribute to the overall quality rating of a home, serving as a factor in assessing its overall quality and pricing.

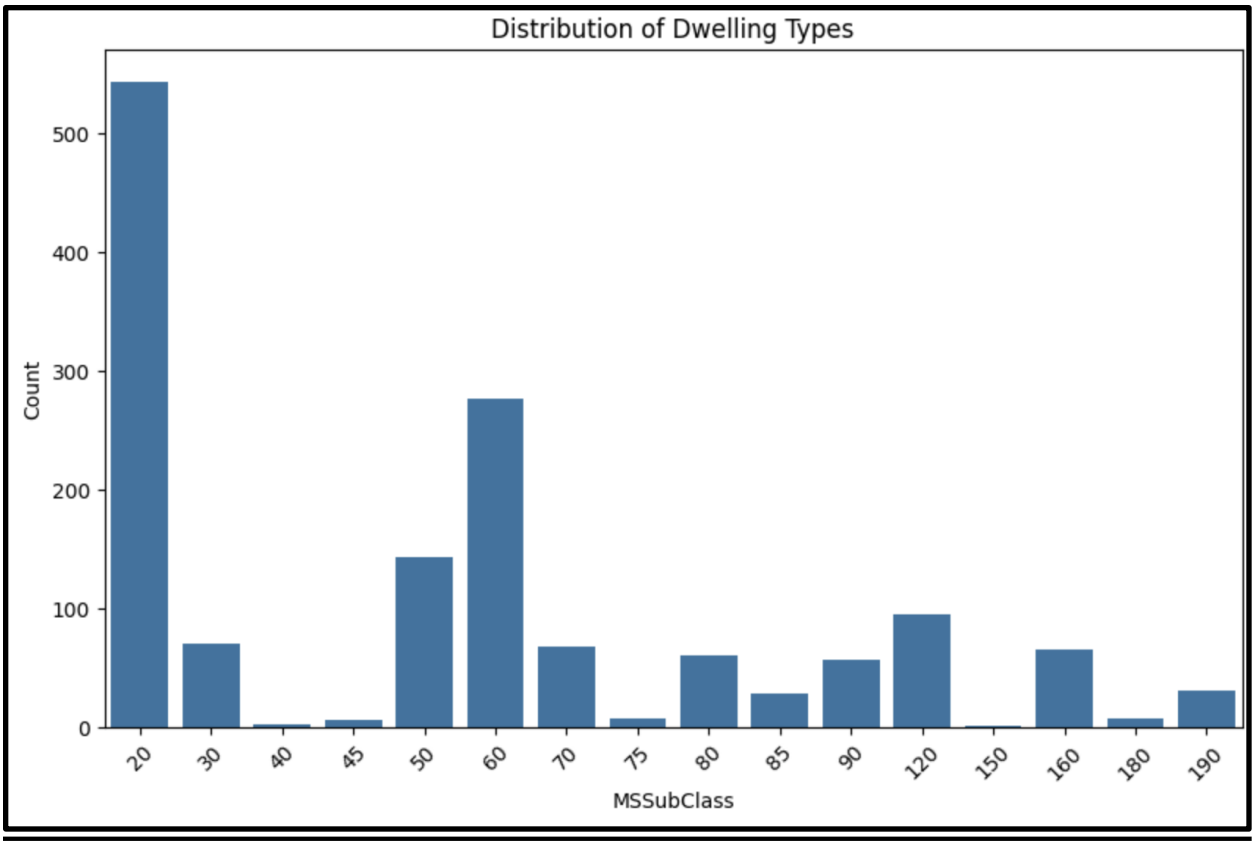


Figure 5

Distribution of Dwelling Types for Test Data

We perform this on the test set. We can see that in the test set, 1-STORY 1946 & NEWER ALL STYLES type of dwelling has the most count followed by 2-STORY 1946 & NEWER and 1-1/2 STORY FINISHED ALL AGES. We can see that 1-story listings which are around 1946 and older are in the majority by a huge margin compared to all other dwelling types.

Section 2: Results and Reports

We trained Lasso, Ridge, XGBoost, AdaBoost, Random Forest, Knn and Decision Tree. We compared the RMSE performance of each model against each other to find the model that gives the best result (lowest RMSE). In each model, we used a simple 80% train, and 20% validation split to gauge its performance.

The Lasso model was produced with a very small alpha set to 0.0001. After splitting and fitting the model to our training data, we evaluated our lasso model to our validation data. The model produced an RMSE of 0.01027 on the validation data.

The Ridge model was initialized with a regularization parameter (alpha) set to 0.1, which helps to control overfitting by penalizing large coefficients. After splitting and fitting the model to our training data, we evaluated the model on our validation data. The model produced an RMSE of 0.01066 on the validation data.

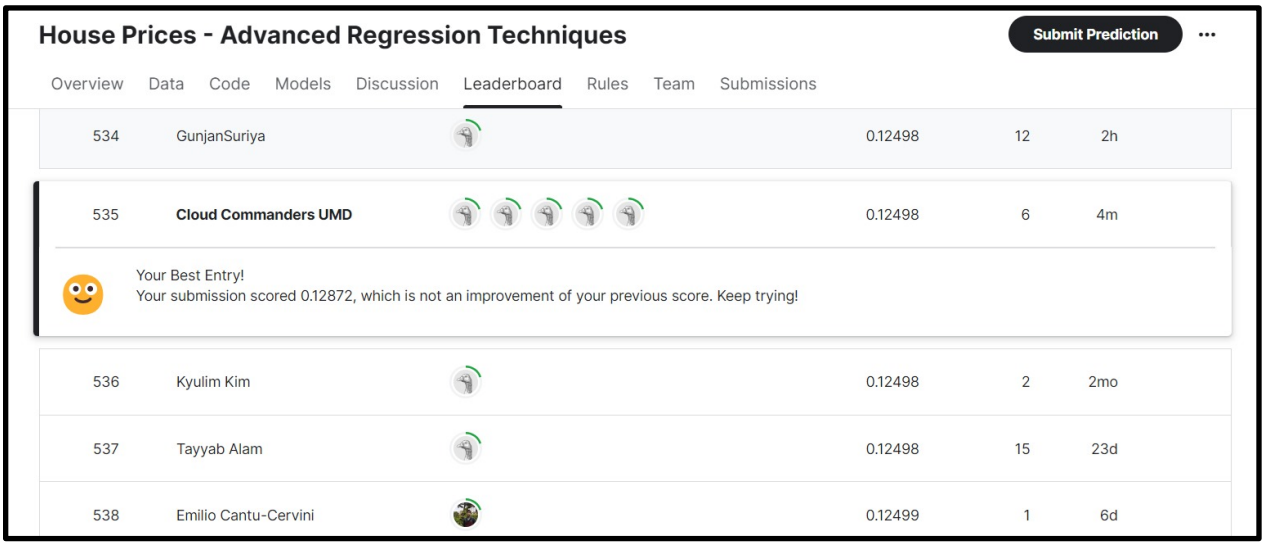
The XGBoost model was optimized using a grid search to be able to identify the best hyperparameters. In the parameter grid, we included values ranging from 100 to 1500, a learning rate ranging from 0.01 to 0.1, and a max depth ranging from 3 to 7. We also did a 5-fold cross-validation after finding that the model gave us a low RMSE of 0.011144 which turned out to be the lowest on the validation data. This model ended up being our best model and the result that we submitted on Kaggle. The results for this model can be seen below.

The AdaBoost model was produced with a grid search approach to find the optimal hyperparameters. A grid search with a 5-fold cross-validation was then executed to find the optimal parameter with learning rates of 0.1, 0.5, and 1, and n estimators of 50, 100, and 150. After fitting the model to our training data, we evaluated the model on our validation data. The model produced an RMSE of 0.0149 with the best learning rate being 1 and the n estimator at 150.

The Random Forest model was produced with the hyperparameter set of bootstrap qualing to False, a minimum value of 2 nodes required to split the node, 100 trees, and a setting to 3 samples for leaf nodes. After splitting and fitting the model to our training data, we evaluated the model on our validation data and got an RMSE value of 0.01652.

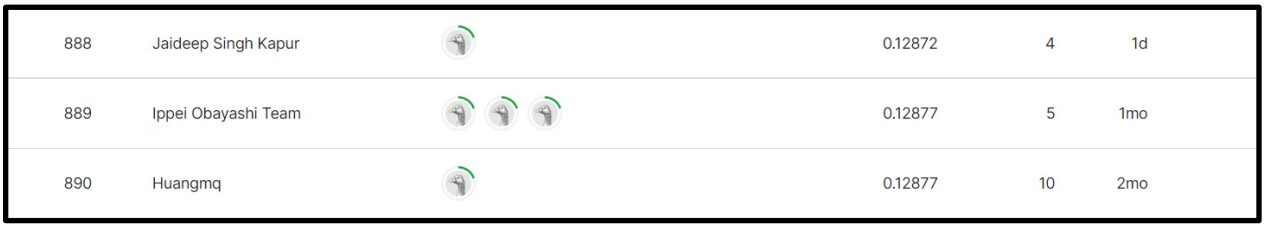
The Knn model was produced with 5 neighbors (n_neighbors=5). After splitting and fitting the model to our training data, we evaluated the model on our validation data. The model produced an RMSE of 0.01489.

The Decision Tree model was produced using a maximum depth of 5. After splitting and fitting the model to our training data, we evaluated our model on the validation data. The model produced an RMSE of 0.0178 on the validation data.



We tried making predictions using an advanced regressor: Stacking regressor which averaged values across multiple other models to make it final prediction. (Ridge, Lasso, XGBoost) which gave us the test RMSE of 0.12498 and **ranked us at 535**. However, due to our limited understanding of the stacking regressor model, we decided to use the **XGBoost model** with a **RMSE of 0.12872** which would **rank us at 888**.

Below is the screenshot of the RMSE of 888 ranked team/participant (0.12872**)** in the competition for your reference. Our latest submission was XGBoost but it didn't take our team rank back to 888 due to a better submission before



.

XGBoost gave us a clear understanding of feature importance and fitting curve. Since we intended to prioritize feature understanding and how different variables affected our target variable, we have used XGBoost as our final and winning model.

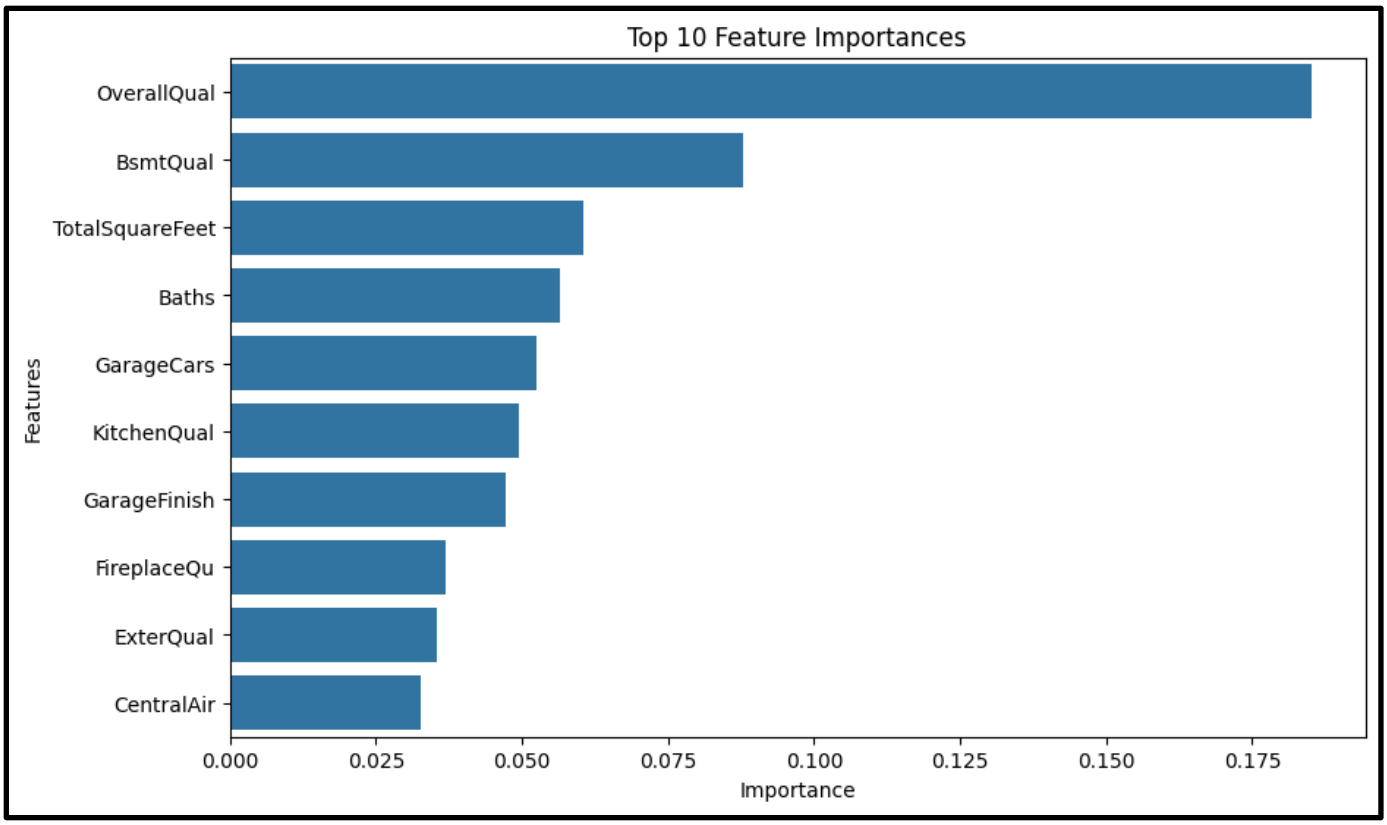


Figure 6
Variable Importance Plot

This graph shows the top 10 features ranked in terms of importance from our winning model (XGBoost). It is worth noting that 4 out of the top 10 features are related to some sort of aspect of the quality of the property. In addition two of the features, TotalSquareFeet and Baths, are features that we created.

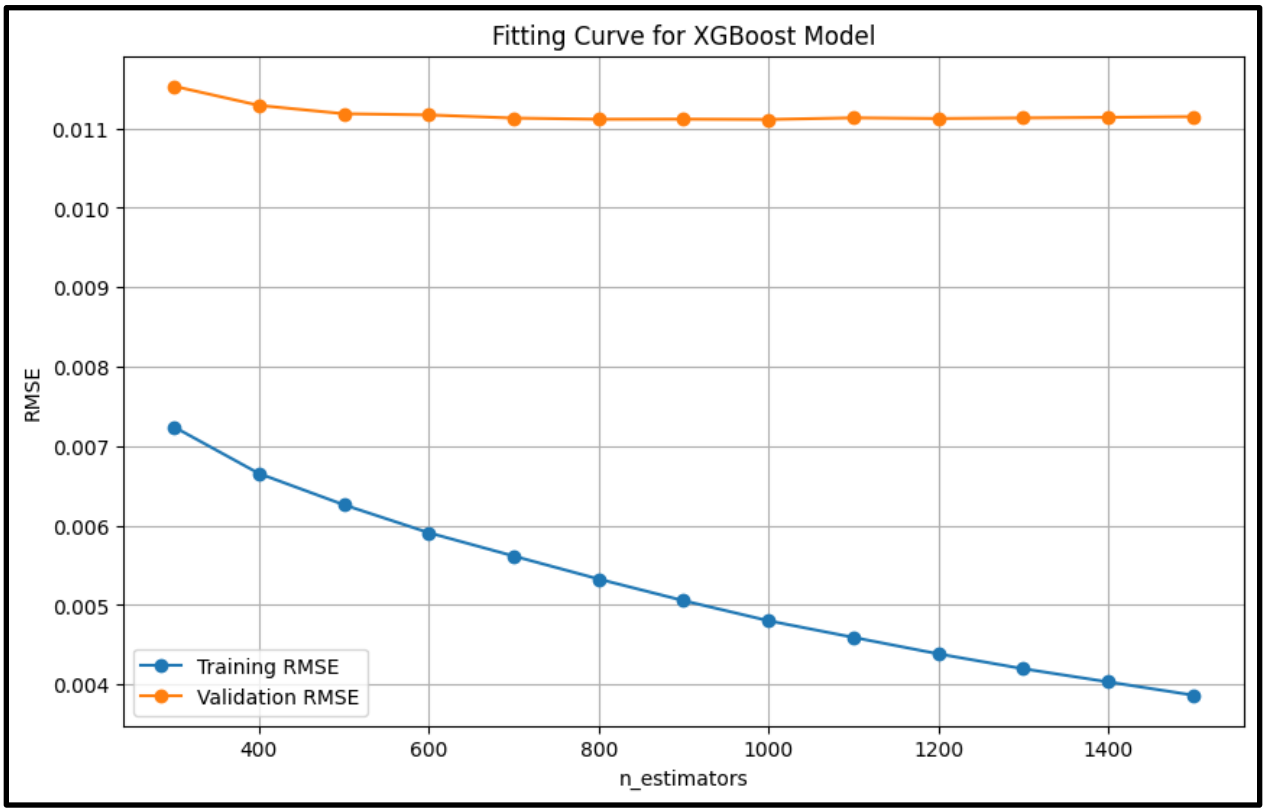


Figure 7
Fitting Curve for XGBoost Model

Fitting curve for the winning XGBoost model. It plots the RMSE by the number of estimators in the model. As you can see in the chart, the training RMSE consistently decreases as the number of estimators increases. However, the validation RMSE reaches its lowest value at about 1000 estimators before showing a slight increase with further increments in the number of estimators.

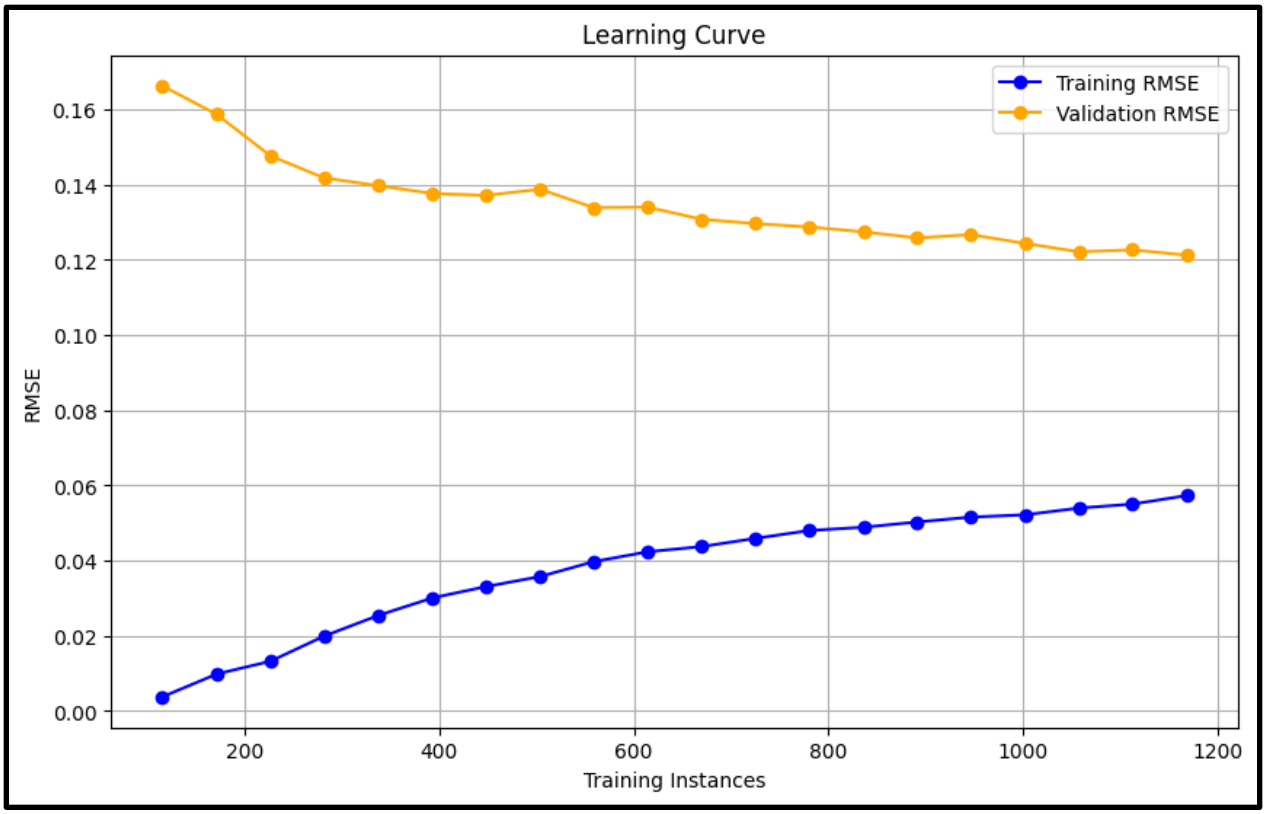


Figure 8
Learning Curve for XGBoost Model

The learning curve for the winning XGBoost model. It shows how this model accurately predicts the target variable in the training set. However, as the number of training instances increases, the number of errors in the training set also increases. With more training instances, the model predicts the target variable more accurately in the validation data, resulting in a decrease in generalization error.

Section 3: Learning

Machine learning has developed quite significantly over the years with the introduction of various new libraries like Sci-Kit Learn and Numpy, etc. By taking part in this competition and working on this project we were able to learn different Models that can be used to predict House Prices and learn the importance of RMSE (Root Mean Squared Error) while making those predictions to evaluate the overall performance of the model. We learned how to train different models like Ridge, Lasso, Random Forest, and Boosting (XGBoost and AdaBoost) on the training data provided and make predictions on the test data provided.

We learned how to perform feature engineering and data cleaning, its impact on the performance of certain models and how to tune them to incorporate different features and their importance for a particular model

For Ridge, we learned how a ridge model works by giving less importance to features by assigning a Complexity parameter to features which make the model more complex. This ideally reduces the importance of those features and helps to focus only on the main features.

For Lasso, we learned how a Lasso model ideally works for feature selection as it practically drops variables which are not important to the model. This model helped us to understand which models we should be focusing on.

For Random Forest, we learned how random forest helps to increase generalization performance by producing various trees using different sets of variables to reduce collinearity amongst trees and taking an average prediction of all values from all trees generated.

For Boosting, we learned how to use two types of boosting methods, namely XGBoost and AdaBoost. For our competition, we found XGBoost to be the most useful as it had the lowest RMSE value on the validation dataset that we created. This also helped us to increase our rank in the competition standings by incorporating other supplementary techniques along with XGBoost which helped us reach a higher rank.

XGBoost works on the principle of repetitively running the same model on the dataset and learning continuously by assigning a higher weightage to values that were misclassified in the previous run of the model.

Overall, we learned how to effectively use different models for making predictions and also studied the theory behind them. Mainly the functioning of these models, the different scenarios in which these models flourish the most and the effectiveness of each model based on their performance on the validation dataset.