CAPSTONE PROJECT

"Prediction of House SalePrice with Advanced Machine Learning Models."

Abhinav Thupili (811234108)

CAPSTONE PROJECT

TABLE OF CONTENTS

Abhinav Thupili (811234108)

CAPSTONE PROJECT

I. Introduction:

A home buyer's biggest fear is understanding the sale price of a house, particularly when factoring in all the essential amenities required for a comfortable living experience. In today's intricate real estate landscape, this concern is being addressed through a revolutionary approach: advanced machine learning models. These models capitalize on historical data encompassing property specs, neighborhood attributes, and market trends to offer a data-driven perspective on valuation. The exploration "Predicting House Sale Prices with Advanced Machine Learning Models" delves into the methodologies and advancements underpinning this intersection of real estate and data science. This discourse highlights the mechanics of machine learning algorithms while emphasizing feature engineering, data preprocessing, and model evaluation for reliable and accurate sale price predictions.

II. Problem Statement:

Accurately determining house sale prices while considering crucial amenities and features remains a continuous challenge in real estate. Traditional methods often need to pay more attention to the intricate interplay of property attributes, local factors, and market trends, resulting in misguided decisions and discontent. This project aims to leverage the dataset from the Kaggle competition to address this issue and develop a robust predictive model that accounts for these variables, thereby enhancing the accuracy of house price predictions.

III. Dataset Overview:

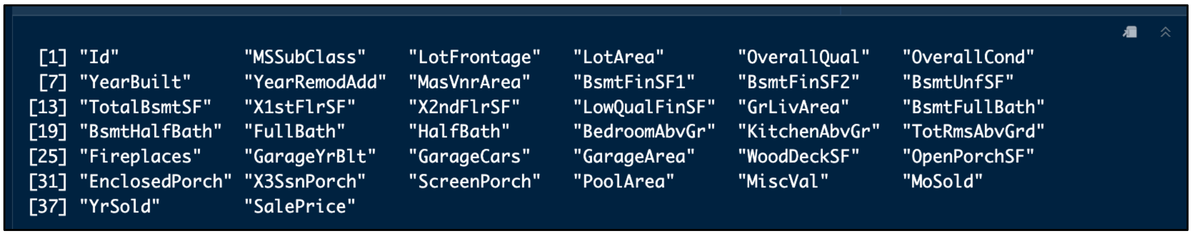
- In this section, we provide an in-depth overview of the dataset that serves as the foundation of our project. We comprehensively understand the information by examining the dataset's structure, variables, and underlying trends. This exploration allows us to identify key features, potential challenges, and areas for further investigation. Through a meticulous analysis of the dataset's characteristics, we lay the groundwork for informed decision-making throughout the project, ensuring that our predictive models are built on a solid and well-informed basis. There are two datasets, train.csv, and test.csv, with both datasets consisting of 1460 observations and 81 attributes each. The dataset consists of attributes like

- **LotArea:** Lot size in square feet.
- **Street:** Type of road access.
- **BldgType:** Type of dwelling.
- **HouseStyle:** Style of dwelling.
- **OverallQual:** Overall material and finish quality.
- **OverallCond:** Overall condition rating.
- **MoSold:** Month Sold.
- **YrSold:** Year Sold.
- **SaleType:** Type of sale.
- **SaleCondition:** Condition of sale.

IV. Data Preparation & Exploration:

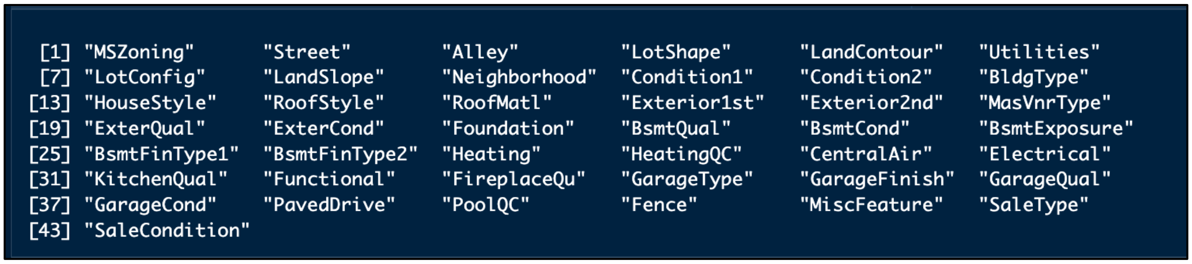
- In this phase, we delve into the intricate process of preparing and exploring the dataset. This critical step lays the foundation for accurate house SalePrice predictions by ensuring the data is clean, relevant, and appropriately structured. Through meticulous data cleaning, transformation, and feature engineering, we aim to enhance the quality of input for our machine-learning models. Additionally, we will explore the dataset's characteristics and identify patterns, outliers, and correlations, providing insights to guide our modeling decisions. This phase is integral to the success of our endeavor, bridging the gap between raw data and actionable insights for effective prediction models.

- The dataset consists of both categorical and numerical values. We can observe that the dataset has 43 categorical variables and 38 numerical variables.



```
 [1] "Id"            "MSSubClass"    "LotFrontage"   "LotArea"       "OverallQual"   "OverallCond"  
 [7] "YearBuilt"     "YearRemodAdd"  "MasVnrArea"    "BsmtFinSF1"    "BsmtFinSF2"    "BsmtUnfSF"    
[13] "TotalBsmtSF"   "X1stFlrSF"     "X2ndFlrSF"     "LowQualFinSF"  "GrLivArea"     "BsmtFullBath" 
[19] "BsmtHalfBath"  "FullBath"      "HalfBath"      "BedroomAbvGr"  "KitchenAbvGr"  "TotRmsAbvGrd" 
[25] "Fireplaces"    "GarageYrBlt"   "GarageCars"    "GarageArea"    "WoodDeckSF"    "OpenPorchSF"  
[31] "EnclosedPorch" "X3SsnPorch"    "ScreenPorch"   "PoolArea"      "MiscVal"       "MoSold"       
[37] "YrSold"        "SalePrice"    
```

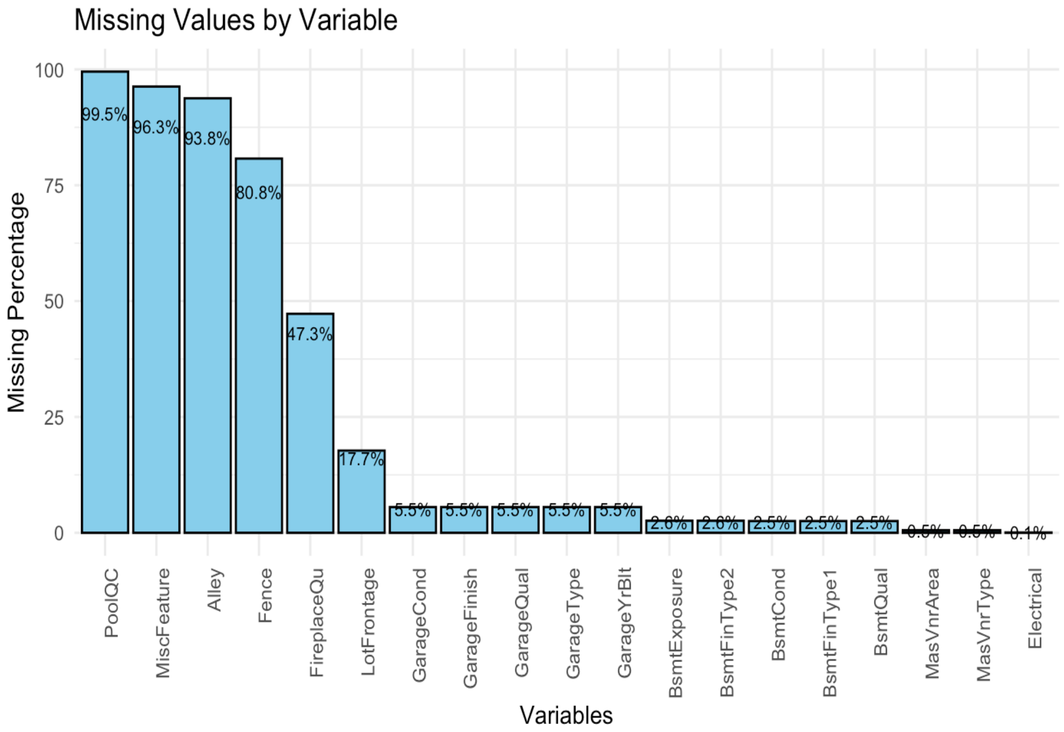
Numerical Variables



```
 [1] "MSZoning"      "Street"        "Alley"         "LotShape"      "LandContour"   "Utilities"    
 [7] "LotConfig"     "LandSlope"     "Neighborhood"  "Condition1"    "Condition2"    "BldgType"     
[13] "HouseStyle"    "RoofStyle"     "RoofMatl"      "Exterior1st"   "Exterior2nd"   "MasVnrType"   
[19] "ExterQual"     "ExterCond"     "Foundation"    "BsmtQual"      "BsmtCond"      "BsmtExposure" 
[25] "BsmtFinType1"  "BsmtFinType2"  "Heating"       "HeatingQC"     "CentralAir"    "Electrical"   
[31] "KitchenQual"   "Functional"    "FireplaceQu"   "GarageType"    "GarageFinish"  "GarageQual"   
[37] "GarageCond"    "PavedDrive"    "PoolQC"        "Fence"         "MiscFeature"   "SaleType"     
[43] "SaleCondition"
```

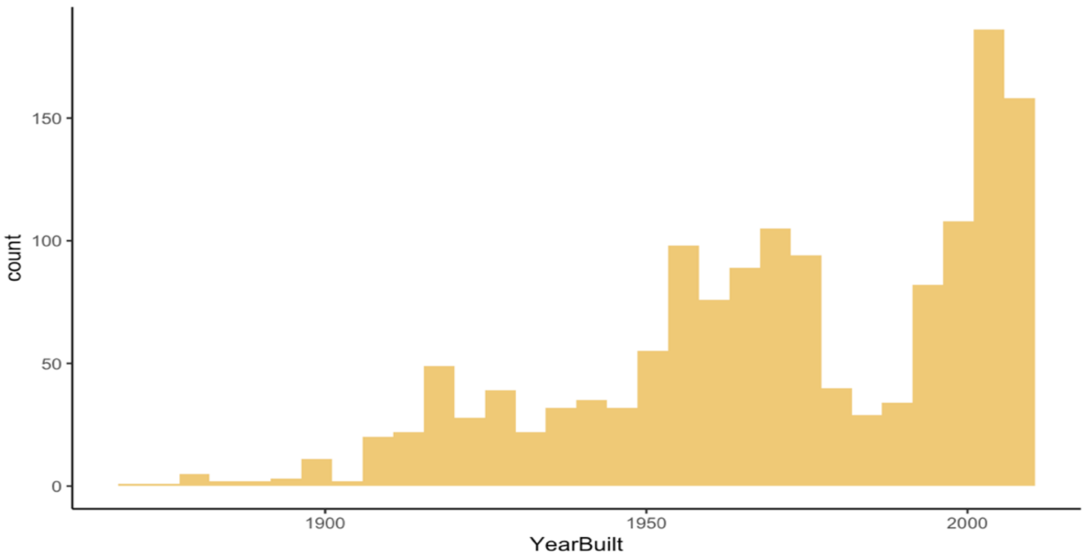
Categorical variables

- **Missing Values:** The accurate analysis of any dataset hinges on the quality and completeness of the data. However, missing values are common occurrences that impede our ability to derive meaningful insights. We can see below that this dataset contains 19 attributes with missing values ranging from 0.1%-99.5%. There are multiple ways to handle missing values, like the imputation method, deletion, and mean or median substitution. I opted for the deletion method, i.e., removing the attributes with missing values.

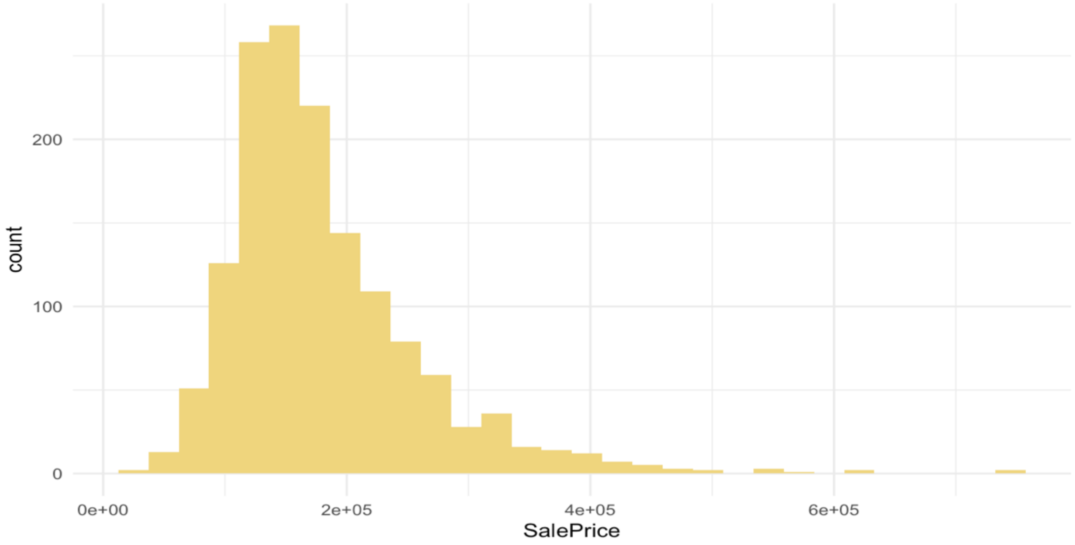


- **Exploratory Data Analysis:**

  - A quick overview shows that the dataset has records of houses built from 1870-2010.



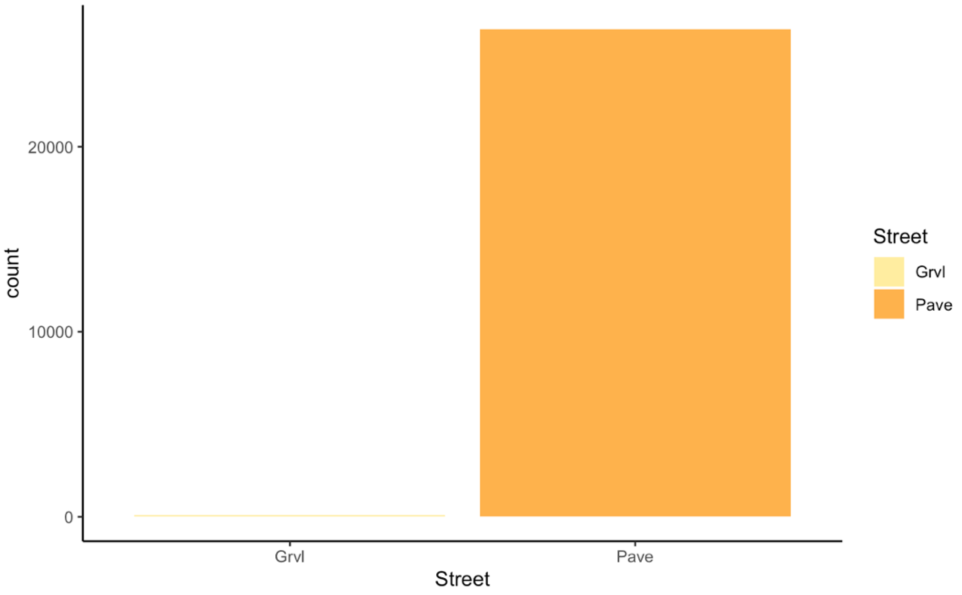
- An overview of the SalePrice column showed that the data is right-skewed.



Histogram of “SalePrice”

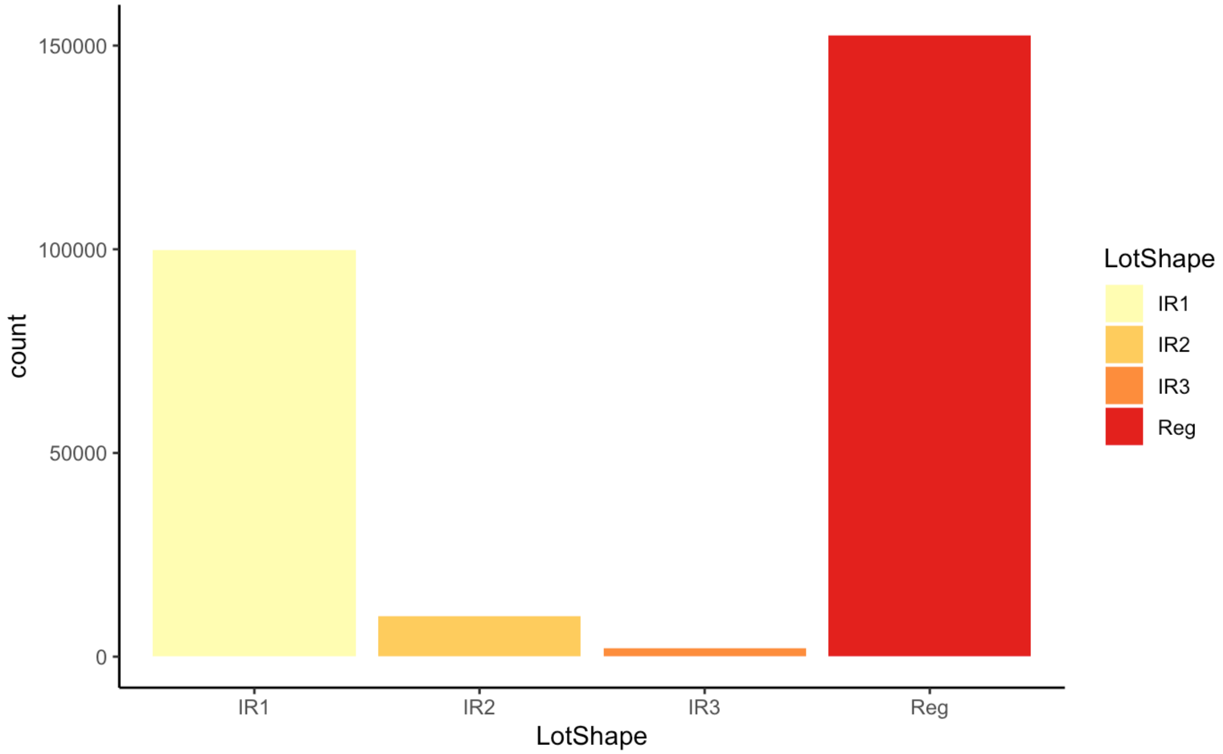
- **Categorical variables:** The categorical variables represent characteristics like colors, regions, or types that can't be quantified, like numerical data. A key data preprocessing step is properly encoding categorical variables to be usable by machine learning algorithms. Proper categorical variable handling is vital for maximizing predictive power. Various methods of handling categorical variables involve Label encoding, Dummy variable encoding, ordinal encoding, etc. I opted for dummy variable encoding, where this creates binary columns, but we exclude one category, avoiding multicollinearity issues.

- After removing the missing values, this dataset contains 25 categorical variables with multiple levels. I used a bar chart to identify and select only important categorical variables. Each categorical attribute was checked against the "SalePrice" column to see if show any significance. Once these variables have been identified, only these attributes are converted into factors. Then they are converted into dummy variables using the "mlr" package in R. Below is a code snippet of how a categorical attribute is analyzed. We can observe from below that the "Street" attribute has two levels, "Grvl" and "Pave." The SalePrice is higher for houses with "Pave" and less for "Grvl." This signifies that "street" is directly proportional to "SalePrice." The same process was applied to all the categorical attributes to identify the important attributes which can help predict the "SalePrice" accurately.

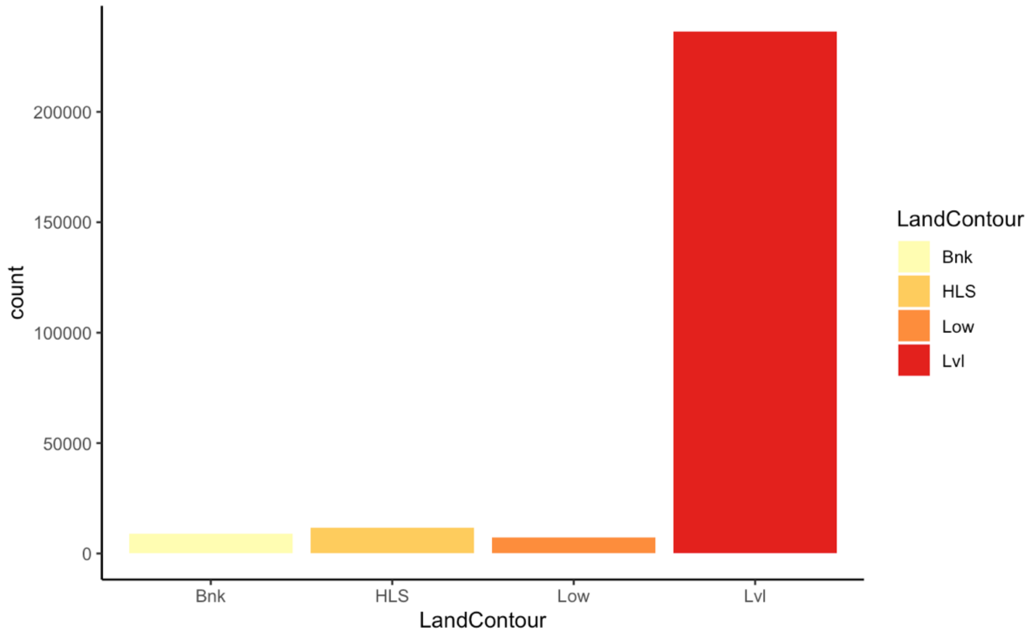


Significance of "Street" with "SalePrice"

CAPSTONE PROJECT

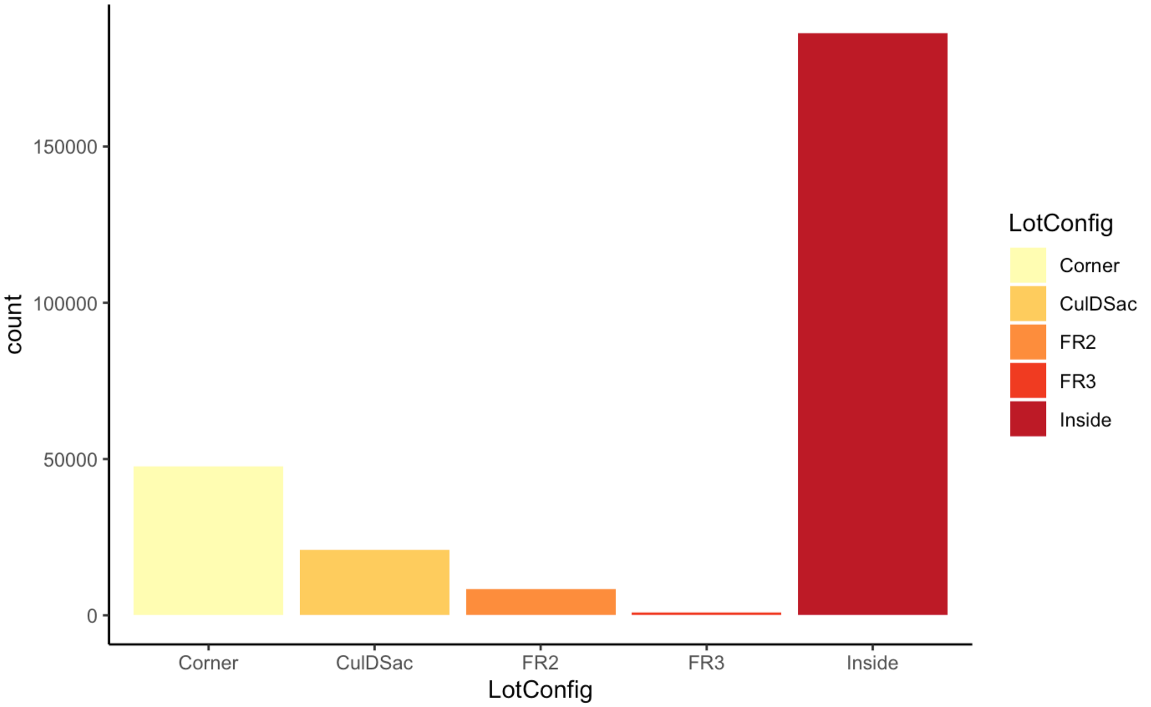


Significance of “LotShape” with “SalePrice”

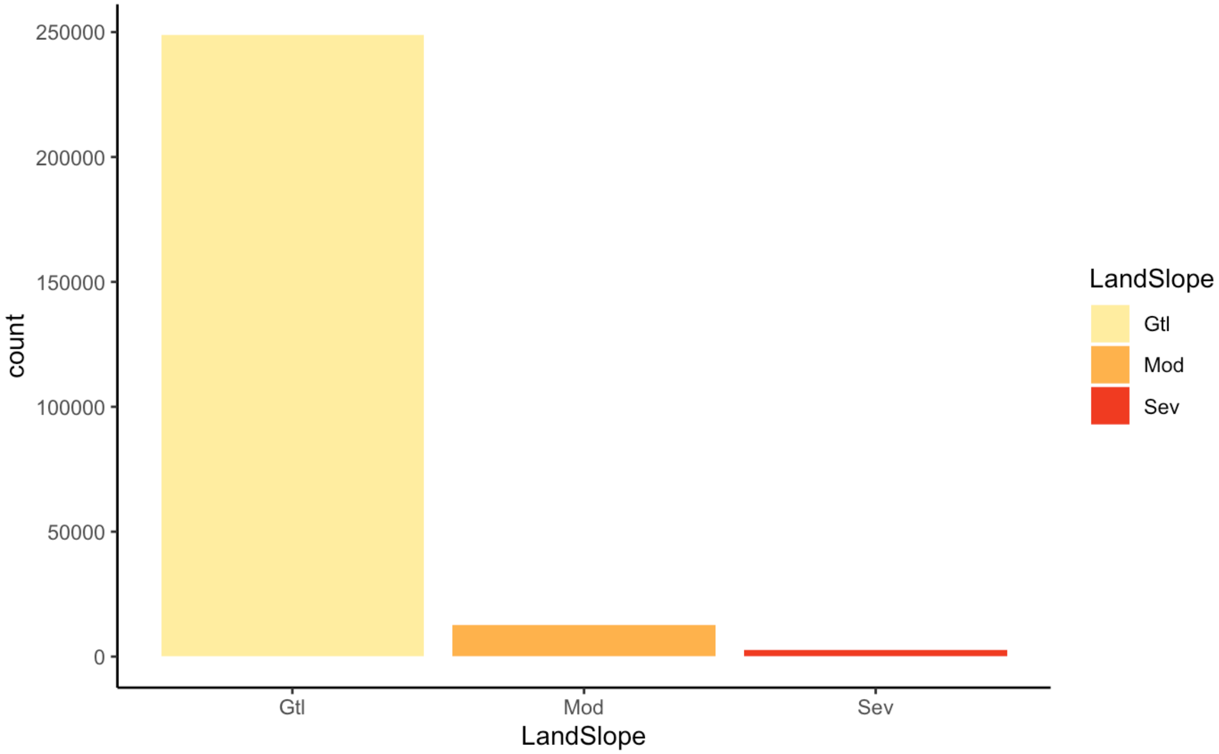


Significance of “LandContour” with “SalePrice”

CAPSTONE PROJECT

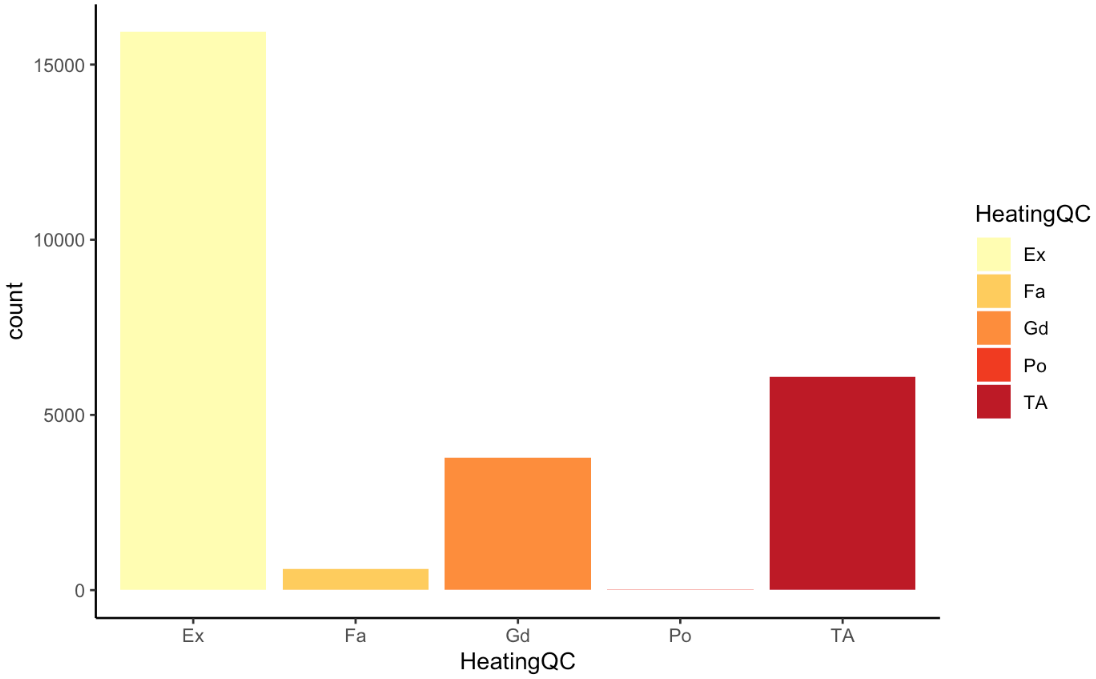


Significance of “LotConfig” with “SalePrice”

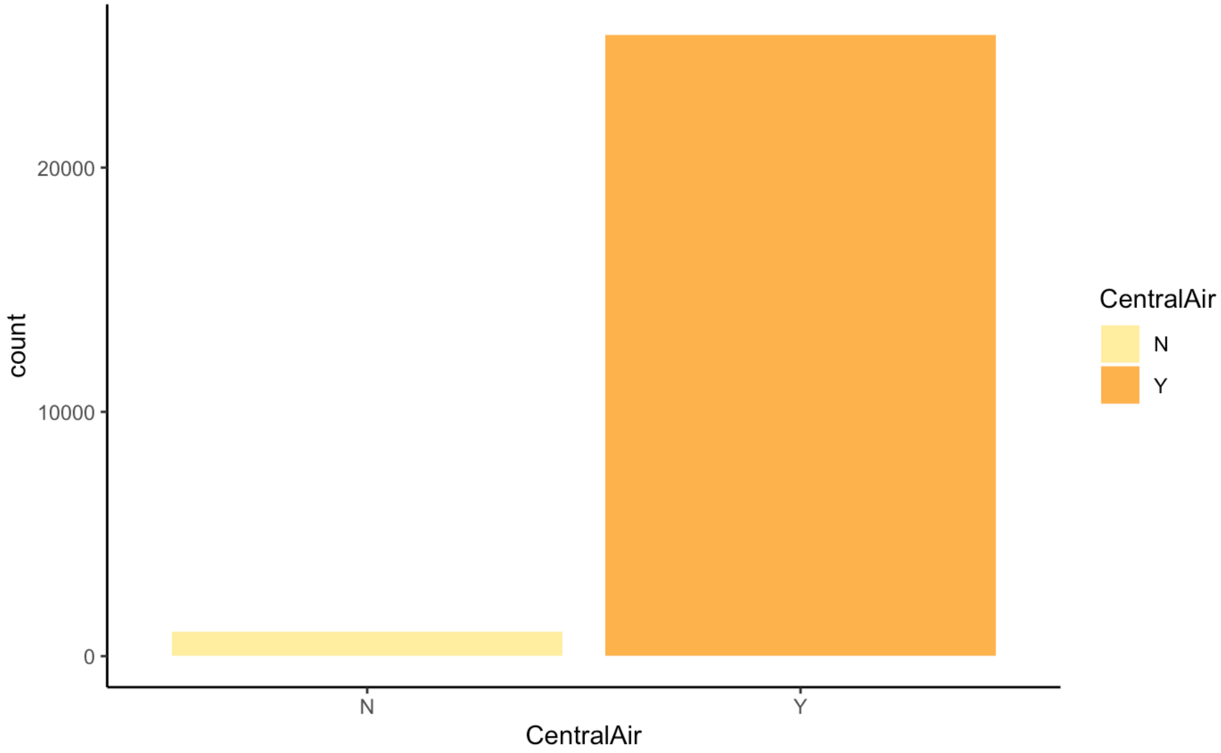


Significance of “LandSlope” with “SalePrice”

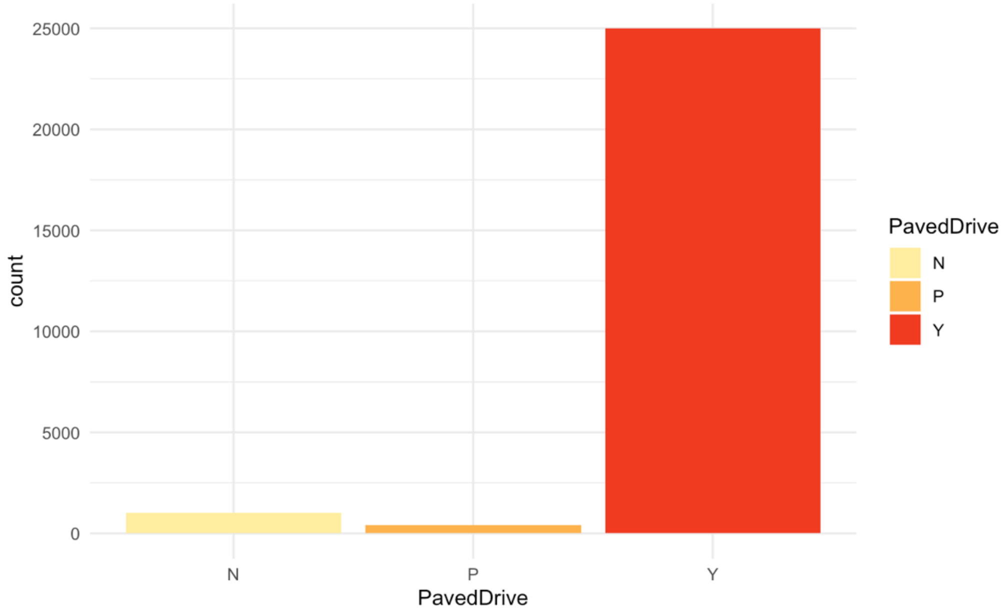
Abhinav Thupili (811234108)



Significance of “HeatingQC” with “SalePrice”

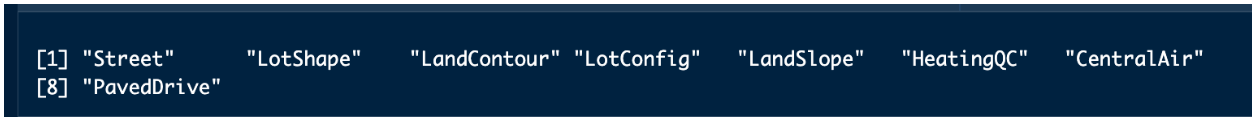


Significance of “CentralAir” with “SalePrice”



Significance of “PavedDrive” with “SalePrice”

- From the above graphs, all eight attributes have significant importance with SalePrice Once all the categorical attributes were analyzed; we were left with eight categorical attributes with significance with “SalePrice.” Below is the code snippet of the final categorical attributes used for the model.

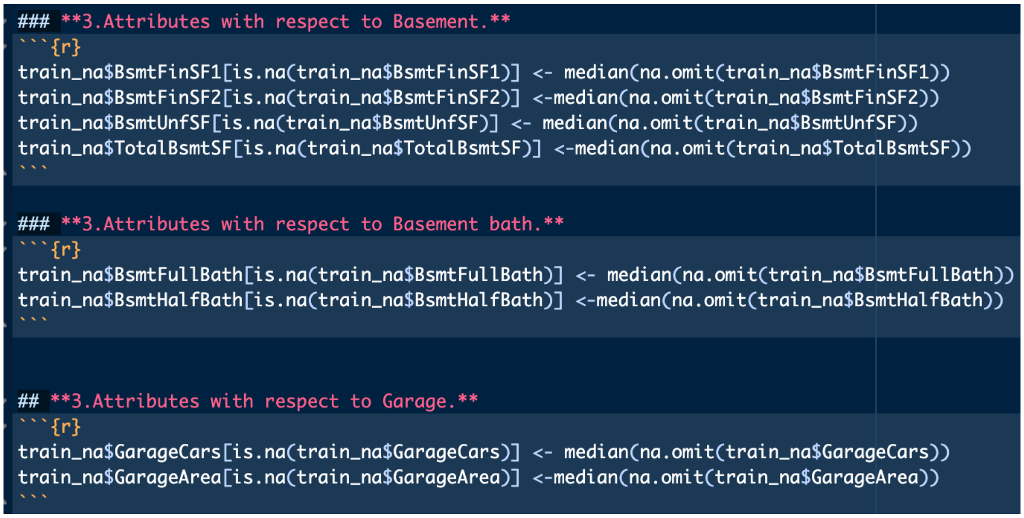


```
[1] "Street"      "LotShape"     "LandContour" "LotConfig"    "LandSlope"    "HeatingQC"    "CentralAir"
[8] "PavedDrive"
```

Final Categorical Varibales

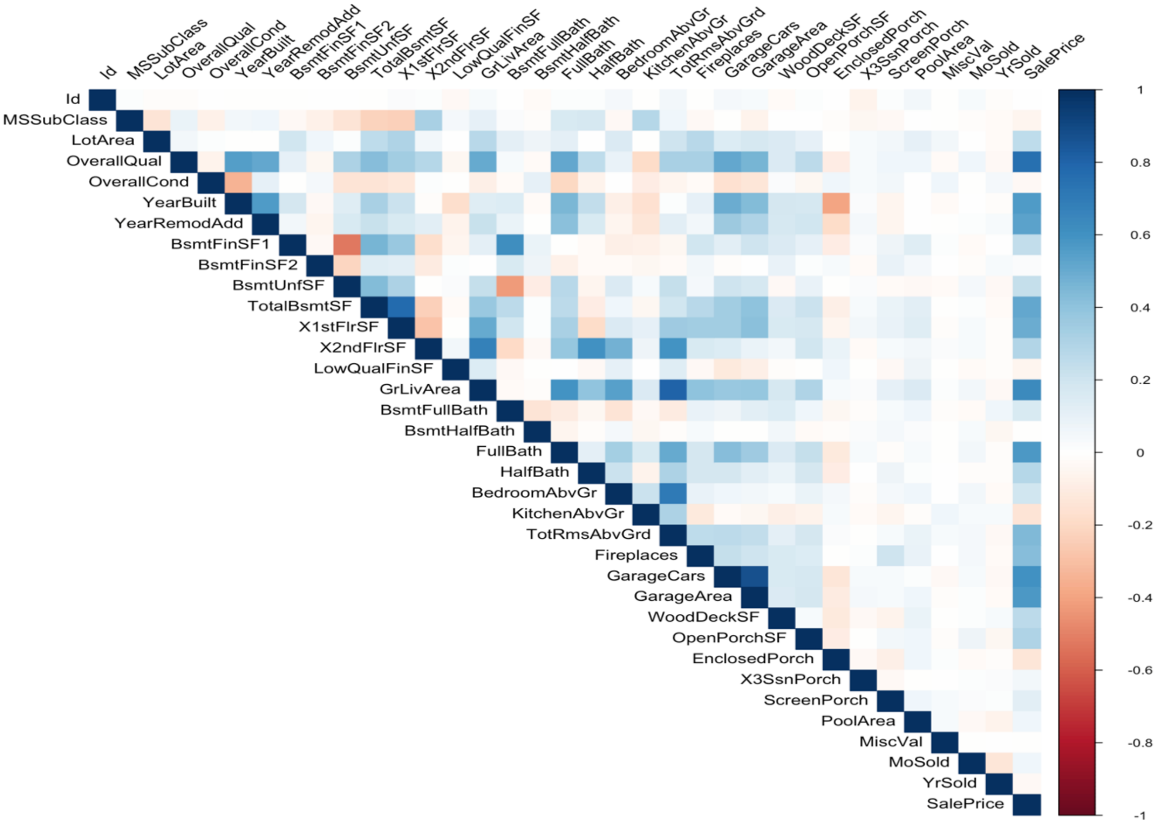
- **Numerical Attributes:** Numerical attributes provide a quantitative dimension to our dataset, allowing us to work with values that can be measured and computed. These attributes can be further categorized into two types:
  - Continuous Numerical Attributes: These variables can take any real value within a certain range. Examples include age, income, and temperature. They often involve measurements and offer a wide range of possible values.

- Discrete Numerical Attributes: These variables can only take specific, distinct values. Examples include the number of bedrooms in a house, the count of items, or the floor number of a building. These values are typically whole numbers and don't have a continuum like continuous variables.
- Remembering that numerical transformation should be applied solely to the training dataset, not the test dataset, is paramount. The test dataset remains unseen, making it crucial to refrain from applying numerical transformations. This practice helps ensure that our predictive models are evaluated on real-world, untouched data during testing, thereby maintaining the integrity of our evaluation process and providing accurate insights into the model's performance on new, unseen inputs. The separation between training and test data safeguards against unintentional data leakage and ensures that our model's generalization capabilities are thoroughly assessed.
- Before further analysis, a meticulous check for any missing values (NAs) within the numerical attributes is essential. In cases where NAs are identified, we employ the median imputation method to replace these missing values comprehensively.
- As a robust central tendency measure, the median is well-suited for imputation as it is less sensitive to outliers than the mean. By replacing NAs with the median, we ensure that the imputed values align with the overall distribution of the data, maintaining its statistical integrity. Below is a code snippet of how values are replaced using the median.

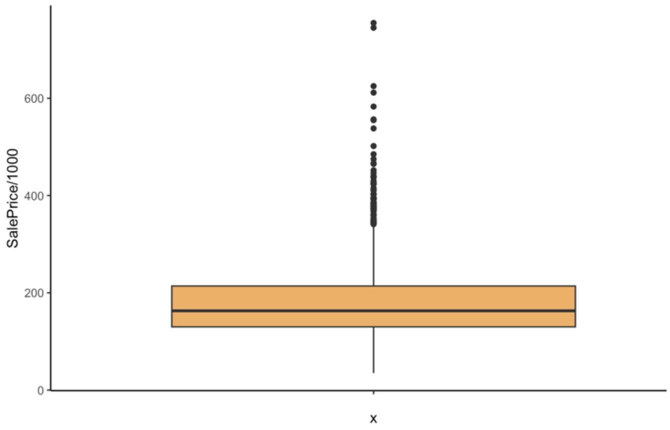


~~~
### **3.Attributes with respect to Basement.**
```{r}
train_na$BsmtFinSF1[is.na(train_na$BsmtFinSF1)] <- median(na.omit(train_na$BsmtFinSF1))
train_na$BsmtFinSF2[is.na(train_na$BsmtFinSF2)] <-median(na.omit(train_na$BsmtFinSF2))
train_na$BsmtUnfSF[is.na(train_na$BsmtUnfSF)] <- median(na.omit(train_na$BsmtUnfSF))
train_na$TotalBsmtSF[is.na(train_na$TotalBsmtSF)] <-median(na.omit(train_na$TotalBsmtSF))
~~~

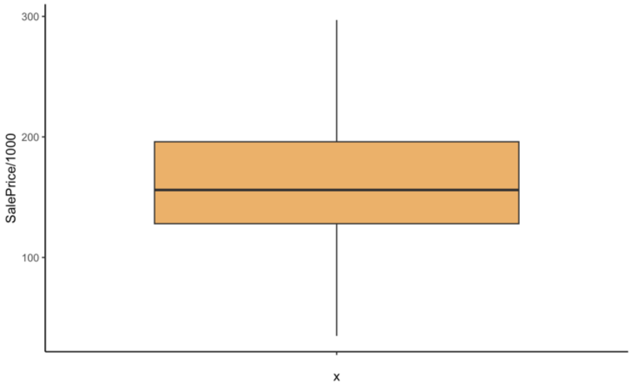
3.Attributes with respect to Basement bath.
```{r}
train_na$BsmtFullBath[is.na(train_na$BsmtFullBath)] <- median(na.omit(train_na$BsmtFullBath))
train_na$BsmtHalfBath[is.na(train_na$BsmtHalfBath)] <-median(na.omit(train_na$BsmtHalfBath))
```

3.Attributes with respect to Garage.
```{r}
train_na$GarageCars[is.na(train_na$GarageCars)] <- median(na.omit(train_na$GarageCars))
train_na$GarageArea[is.na(train_na$GarageArea)] <-median(na.omit(train_na$GarageArea))
```
```

Using Median to Replace NAs

- Below are representation correlations of all numerical attributes in the dataset.



Correlation of all numerical attributes with SalePrice
```

- **Understanding Outliers:** Outliers, those distinct data points that deviate from the norm, exert a notable impact on data analysis and modeling. They may arise due to data entry errors, measurement inconsistencies, or inherent data variability. Detecting outliers amalgamates visual cues and statistical techniques, thoughtfully aligned with the analytical context. In this exploration, two indispensable analytical tools step forward: the boxplot and the “boxplot. stats” functions. The boxplot visually segments data into quantiles, effectively unveiling anomalous points that significantly diverge from the data's central tendencies. Additionally, the “boxplot. stats” function imparts vital numerical insights—comprising median, interquartile range, and more—quantifying the magnitude of these deviations.
    - Removing outliers rows using the “SalePrice” column is important as the model is trained to predict this for the test set.
    - Below we can observe the difference before and after removing the outliers from the dataset.



Before removing Outliers



After removing Outliers

- Before moving to the model phase, the final step is creating dummy variables using the “mlr” package that converts the categorical variables into numerical values for processing through the model.
- Finally, the training dataset was split into train and validation sets in the ratio of 60:40, respectively, to evaluate model performance.

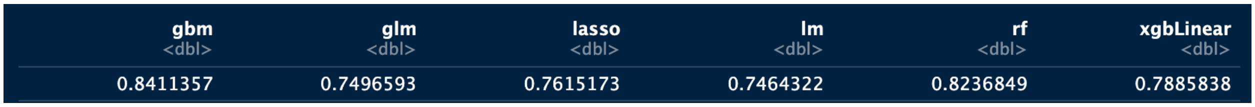
## V. Model Selection:

The model selection is a strategic framework for building predictive models that accurately capture the patterns and relationships within the data. This phase involves selecting appropriate algorithms, tuning their parameters, and evaluating model performance. A well-defined modeling approach enhances the model's predictive power and ensures its effectiveness in making accurate predictions. I used multiple machine algorithms on the current train dataset to find the correct algorithm to check which would provide the best performance.

The following algorithms were used to predict the “SalePrice”:

1. **GBM (Gradient Boosting Machine)**: Gradient Boosting is an ensemble learning technique that combines the predictions of multiple weak learners (typically decision trees) to create a strong predictive model. It's effective for complex relationships in data and can handle both numeric and categorical features.
2. **GLM (Generalized Linear Model)**: Generalized Linear Models are a broad class of models that include linear regression as a special case. They can handle various distribution families and are suitable for cases where the response variable follows a different distribution than the normal distribution assumed by linear regression.
3. **Lasso Regression (Lasso)**: Lasso Regression is a linear regression technique that includes L1 regularization, which encourages sparsity in the coefficients. It's useful for feature selection and can help prevent overfitting by reducing the impact of irrelevant features.
4. **LM (Linear Regression)**: Linear regression is a simple and interpretable algorithm that models the relationship between the dependent variable and one or more independent variables. It's effective when the relationship between the variables is approximately linear.
5. **RF (Random Forest)**: Random Forest is an ensemble learning algorithm that builds a collection of decision trees and combines their predictions to improve accuracy and control overfitting. It's suitable for both regression and classification tasks.
6. **xgbLinear (XGBoost Linear)**: XGBoost is a popular gradient-boosting framework. The "xgbLinear" variant of XGBoost uses linear base learners instead of decision trees, which can be useful when dealing with high-dimensional data and complex interactions.

- Each model was evaluated based on R-squared values. The R-squared value of a regression model quantifies how well the model's predictions align with the observed data. This value ranges between 0 and 1, where 0 signifies that the model does not explain any of the variability in the data, and 1 indicates a perfect alignment between the model's predictions and the actual data points.



| gbm<br>&lt;dbl&gt; | glm<br>&lt;dbl&gt; | lasso<br>&lt;dbl&gt; | lm<br>&lt;dbl&gt; | rf<br>&lt;dbl&gt; | xgbLinear<br>&lt;dbl&gt; |
|---|---|---|---|---|---|
| 0.8411357 | 0.7496593 | 0.7615173 | 0.7464322 | 0.8236849 | 0.7885838 |

- From what we can observe, GBM (Gradient Boosting Machine), RF (Random Forest), and xgbLinear (XGBoost Linear) had R-squared values of more than 0.80. It Is evident that GBM (Gradient Boosting Machine) is the most suitable model for predicting the "SalePrice."
- Further, the model was modified using Hyper-parameters to find optimal performance on the training dataset.
- Hyper-parameters applied are as follows:

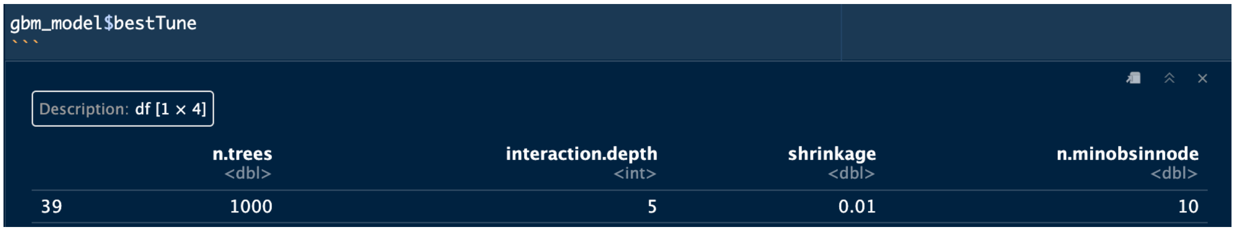
    1.n.trees=c(100,500,1000),

    2.shrinkage=c(0.1,0.05,0.01),

    3. interaction.depth=1:5,

    4. n.minobsinnode=c(10,15,20))

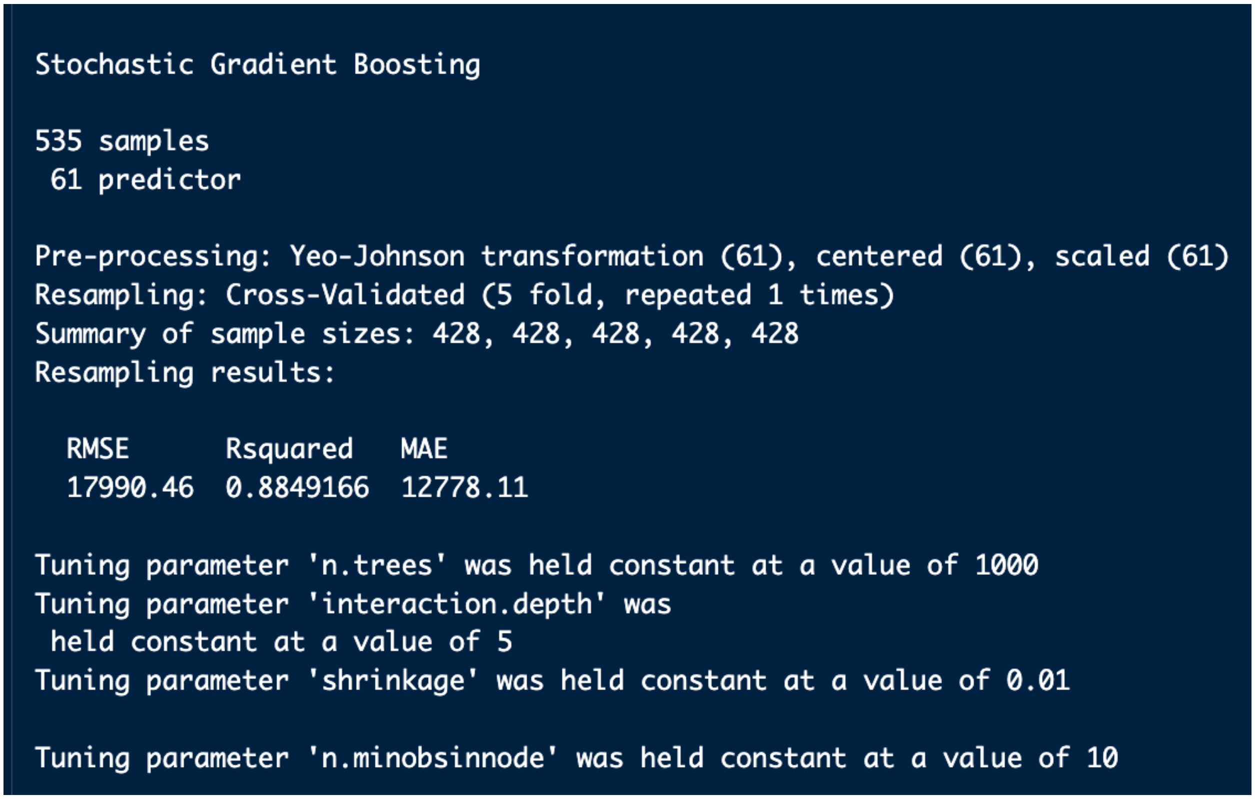
- Using the GBM model, the best hyper-parameters returned to achieve maximum performance are shown below:



```
gbm_model$bestTune
```
```

Description: df [1 × 4]

	n.trees <dbl>	interaction.depth <int>	shrinkage <dbl>	n.minobsinnode <dbl>
39	1000	5	0.01	10
```

## VI. Performance Evaluations:

- The model's performance was validated on the dataset using hyper-parameters.



```
Stochastic Gradient Boosting

535 samples
 61 predictor

Pre-processing: Yeo-Johnson transformation (61), centered (61), scaled (61)
Resampling: Cross-Validated (5 fold, repeated 1 times)
Summary of sample sizes: 428, 428, 428, 428, 428
Resampling results:

  RMSE      Rsquared   MAE
  17990.46  0.8849166  12778.11

Tuning parameter 'n.trees' was held constant at a value of 1000
Tuning parameter 'interaction.depth' was
 held constant at a value of 5
Tuning parameter 'shrinkage' was held constant at a value of 0.01

Tuning parameter 'n.minobsinnode' was held constant at a value of 10
```

- The final R-squared value shows that 0.8849166, the model explains 88% of the variance.

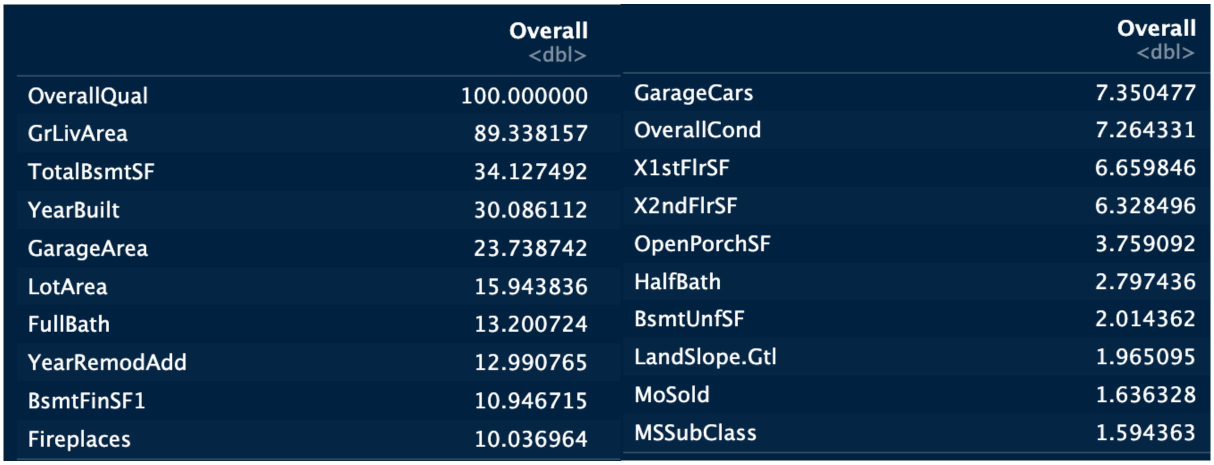
- RMSE (Root Mean Squared Error) measures the standard deviation of the residuals (prediction errors). It takes the square root of the average of the squared residuals. This has the effect of weighting larger errors more heavily since squaring larger values gives even greater weight.

- MAE (Mean Absolute Error) measures the average magnitude of the residuals. It is the simple average of the absolute values of the residuals. This weights all errors equally rather than emphasizing larger errors.

- RMSE and MAE are expressed in dollars of “SalePrice,” where the Root Mean Squared error is $17990.46, and the Mean Absolute Error is $12778.11.

VII. Insights & Conclusion:

- Insights from 20 top variables returned from the model are as follows:



	Overall <dbl>
OverallQual	100.000000
GrLivArea	89.338157
TotalBsmtSF	34.127492
YearBuilt	30.086112
GarageArea	23.738742
LotArea	15.943836
FullBath	13.200724
YearRemodAdd	12.990765
BsmtFinSF1	10.946715
Fireplaces	10.036964
GarageCars	7.350477
OverallCond	7.264331
X1stFlrSF	6.659846
X2ndFlrSF	6.328496
OpenPorchSF	3.759092
HalfBath	2.797436
BsmtUnfSF	2.014362
LandSlope.Gtl	1.965095
MoSold	1.636328
MSSubClass	1.594363

INSIGHTS:

- Out of 43 variables that were selected for modeling, which consisted of eight categorical variables and 33 numerical variables(excluding “Id” and “SalePrice”), the model returned 20 variables which consisted of 19 numerical variables and one categorical variable.

- We can observe that the house's OverallQual – (Rates the overall material and finish) plays an important role in assessing the SalePrice, which could help a potential buyer or seller assess the Sale Price based on the house's overall quality.

- GrLivArea–(Above grade (ground) living area square feet), TotalBsmtSF–(Total square feet of basement area), YearBuilt, GarageArea, LotArea, FullBath, BsmtFinSF1–(Type 1 finished square feet), YearRemodAdd – (Remodel date), Fireplaces signifies model performance is predictions of “SalePrice” is closer to real-world scenarios where all these variables play a crucial role in assessing the sale price of the House.

- Out of the 20 top variables, the bottom variables are OpenPorchSF – (Open porch area in square feet), HalfBath, BsmtUnfSF – (Unfinished square feet of basement area), LandSlope – (Slope of property), MoSold – (Month sold), MSSubClass– (Identifies the type of dwelling involved in the sale) signify very less importance can be given when it comes to these attributes when assessing the sale price of the House.

CONCLUSION:

In summation, our modeling efforts culminate with an R-squared of 0.8849166, indicating that the model explains approximately 88% of the variance in the data. It exemplifies a commendable capacity to encapsulate the underlying relationships.

The Root Mean Squared Error (RMSE) of $17990.46 provides insight into the standard deviation of prediction errors, quantifying the magnitude of deviations from actual "SalePrice" values. Meanwhile, the Mean Absolute Error (MAE) of $12778.11 offers a balanced perspective by measuring average error magnitude irrespective of size.

With the RMSE and MAE expressed in "SalePrice" dollars, these metrics are critical for evaluating predictive accuracy. The RMSE highlights large discrepancies, while the MAE comprehensively considers all error magnitudes.

Taken together, the strong R-squared and reasonable RMSE and MAE signify robust model performance in predicting "SalePrice." The results underscore proficient generalization and reliable estimation abilities, establishing a firm foundation for real-world application.

In conclusion, our rigorous evaluation validates the efficacy of the proposed approach for property valuation modeling and its potential to enable confident, data-driven decision-making.

References:

1. Source of the dataset.
2. https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab
3. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
4. https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e