IE590 Machine Learning and its Applications: Project Report

Group 6

Decemeber 15, 2023

1 Problem Statement

1.1 Description of the problem

The problem at hand involves residential real estate data. Specifically, it revolves around predicting the final sale price of homes in Ames, Iowa, based on a comprehensive dataset comprising 79 explanatory variables. Each variable represents different aspects of the properties, including their characteristics, conditions, and attributes. The dataset is split into a training set (`train.csv`) to build predictive models and a test set (`test.csv`) to evaluate the model's performance. Only the training set consists of the Target outputs which is the Sales Price of the house. This problem is essentially a regression task, where the goal is to develop a model that accurately estimates the sale price of houses based on the provided features.

1.2 Research question or hypothesis

The primary research question or hypothesis is: ”How can the final sale price of homes in Ames, Iowa, be predicted based on features and characteristics using regression techniques?”

1.3 Justification of why the question is relevant

1. The precise evaluation of property costs is essential for both homeowners and real estate agents.

2. Homeownership is often the most significant financial decision individuals make. Predicting sale prices helps buyers make informed decisions and sellers set competitive prices.

3. Understanding the factors that influence property prices can provide valuable insights into the real estate market in Ames, Iowa, which can benefit real estate professionals, city planners, and policymakers.

4. Investors and developers can use accurate price predictions to identify properties with potential for appreciation and make informed investment choices.

1.4 Benefits of solving the problem

1. Home buyers and sellers can make more informed decisions about pricing and negotiations, leading to fairer transactions.

2. Accurate predictions can contribute to a more efficient and transparent real estate market, reducing the likelihood of overpricing or underpricing properties.

3. Investors can use predictive models to assess investment risks and make decisions that align with their financial goals.

4. Accurate pricing models can provide valuable insights into local real estate trends and contribute to a better understanding of the local economy.

5. Knowledge of property values can help city planners and policymakers make informed decisions about urban development and infrastructure.

2 Data

2.1 Source of the Data

The dataset for this project was originally compiled by Dean De Cock and is publicly available on Kaggle, a platform for data science competitions and machine learning projects. The dataset was created for educational and research purposes and is commonly used in machine learning competitions and exercises.

2.2 Data Description

The dataset consists of two files Train and Test. A total list of all the features are provided in Annexure 1.

2.3 Number of Observations

The number of observations (rows) in the training dataset (`train.csv`) is 1,460.

2.4 Number of Parameters

The dataset includes 79 explanatory variables (features) that describe various aspects of the residential properties in Ames, Iowa. These variables are used to predict the target variable, which is the sale price of the homes.

2.5 Type of Data

The dataset contains a mix of data types, including:

- Continuous data: Variables such as `LotFrontage` (linear feet), `LotArea` (square feet), and numerical measures of square footage, area, and counts.

- Categorical data: Variables like `MSZoning` (zoning classification), `Street` (type of road access), and other categorical attributes describing property characteristics.

- Ordinal data: Variables like `ExterQual` (exterior material quality) and other variables with ordered categories.
- Discrete data: Variables representing counts, such as the number of bedrooms (`Bedroom`), full bathrooms (`FullBath`), and other discrete features.
- Date data: Variables like `YearBuilt` (original construction date) and `YearRemodAdd` (remodel date) represent dates in time.

2.6 Combining Test and Train Data

Before diving into the Data Pre-processing we have combined the Training and Test data sets since we can apply all the pre-processing techniques in one go. While doing this we dropped the `Id`, and `SalePrice` in the Train dataset and the `Id` in the Test dataset.

2.7 Response variable

The response variable in the dataset is `SalePrice`, which represents the final sale price of each residential property in dollars. The goal of the project is to develop predictive models that can accurately estimate this sale price based on the provided explanatory variables.

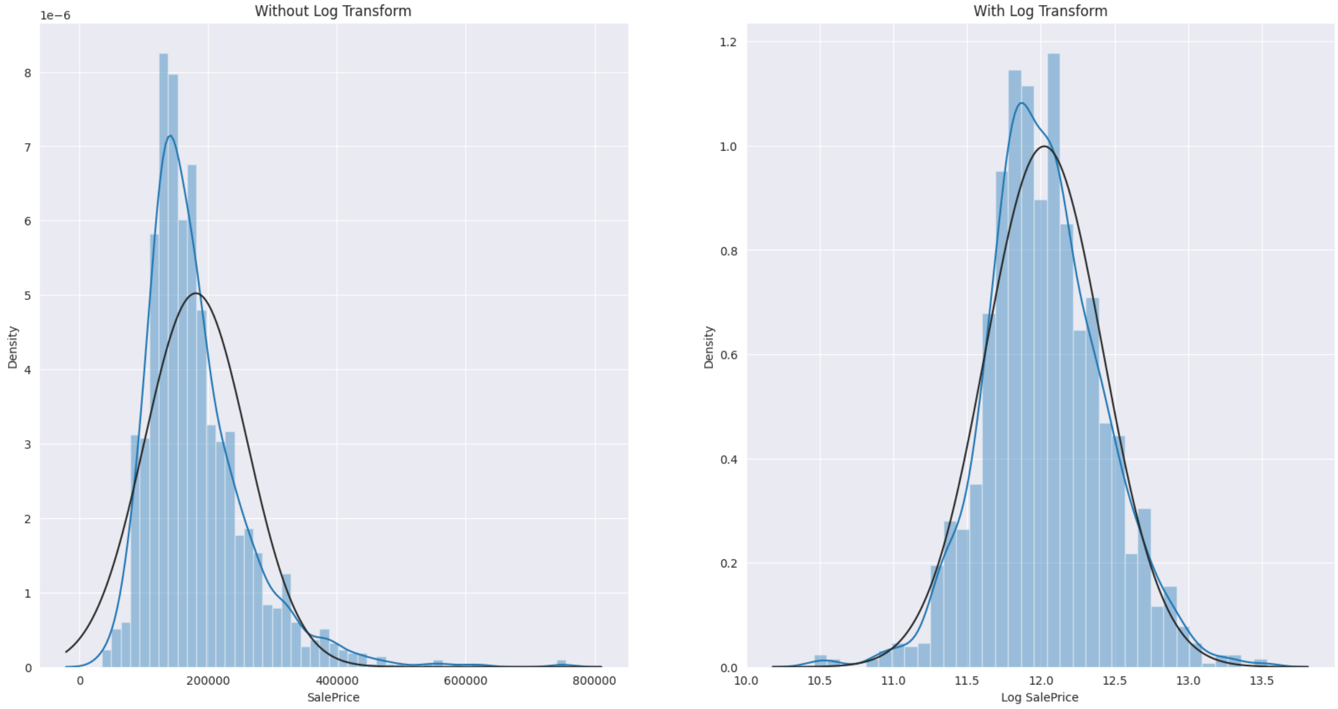


Figure 1: Log Transform of Sales Price

2.8 Missing Data Analysis

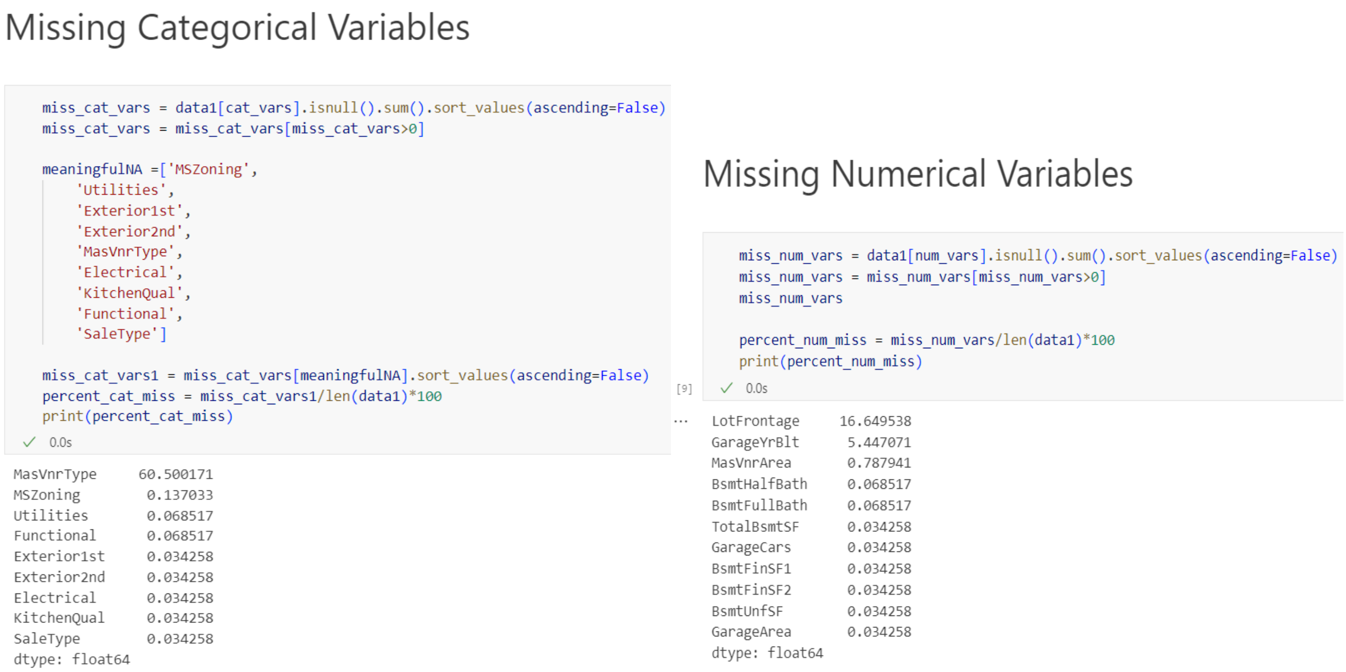
The data set has several missing continuous and categorical data points which vary in their interpretation. For example, consider the ”Alley” predictor. It has the following categorical arrangement:

Category	Description
Grvl	Gravel
Pave	Paved
NA	No alley access

Here the value ”NA” refers to there being no alley and does not indicate a missing data point. The null values were then replaced based on the type via data imputation.

2.8.1 Data Imputation for Missing Values

- Categorical data Type 1: Predictors such as `Alley`, `BsmtQual` (Along with other Basement related predictors) etc., where the NA values carry some meaning are replaced with ”None”.
- Categorical data Type 2: Predictors such as `MSZoning`, `Utilities` etc., where the NA values simply refer to missing data points have been replaced with the mode of the data.
- Numerical data: Variables like `LotFrontage`, `MasVnrArea`, and other numerical predictors which contain missing values had missing data points which were replaced using a K-Nearest Neighbours algorithm. A customized KNN imputation function was used to replace missing values uing surrounding data points as leverage.
- The percentage of missing data for these features are mentioned in Figure 2 below.



Missing Categorical Variables

```
miss_cat_vars = data1[cat_vars].isnull().sum().sort_values(ascending=False)
miss_cat_vars = miss_cat_vars[miss_cat_vars>0]

meaningfulNA =['MSZoning',
    'Utilities',
    'Exterior1st',
    'Exterior2nd',
    'MasVnrType',
    'Electrical',
    'KitchenQual',
    'Functional',
    'SaleType']

miss_cat_vars1 = miss_cat_vars[meaningfulNA].sort_values(ascending=False)
percent_cat_miss = miss_cat_vars1/len(data1)*100
print(percent_cat_miss)
```

```
MasVnrType     60.500171
MSZoning        0.137033
Utilities       0.068517
Functional      0.068517
Exterior1st     0.034258
Exterior2nd     0.034258
Electrical      0.034258
KitchenQual     0.034258
SaleType        0.034258
dtype: float64
```

(a) Missing Categorical Data

Missing Numerical Variables

```
miss_num_vars = data1[num_vars].isnull().sum().sort_values(ascending=False)
miss_num_vars = miss_num_vars[miss_num_vars>0]
miss_num_vars

percent_num_miss = miss_num_vars/len(data1)*100
print(percent_num_miss)
```

```
LotFrontage     16.649538
GarageYrBlt      5.447071
MasVnrArea       0.787941
BsmtHalfBath     0.068517
BsmtFullBath     0.068517
TotalBsmtSF      0.034258
GarageCars       0.034258
BsmtFinSF1       0.034258
BsmtFinSF2       0.034258
BsmtUnfSF        0.034258
GarageArea       0.034258
dtype: float64
```

(b) Missing Numerical Data

Figure 2: The Percentage of Data Imputed

2.9 Feature Engineering

Based on the raw data, the following new features are developed to enhance the model’s prediction performance:

- SqFtPerRoom: The proportion of ”GrLivArea” to all the rooms in the house.
- Total Home Quality: The sum of the ratings for ”OverallQual” and ”OverallCond.”
- Total Bathrooms: The weighted sum of the bathrooms that are there in the house overall including full and half baths.
- HighQualSF: Its the sum of the area on the first and second floors.

2.10 Skewness Correction

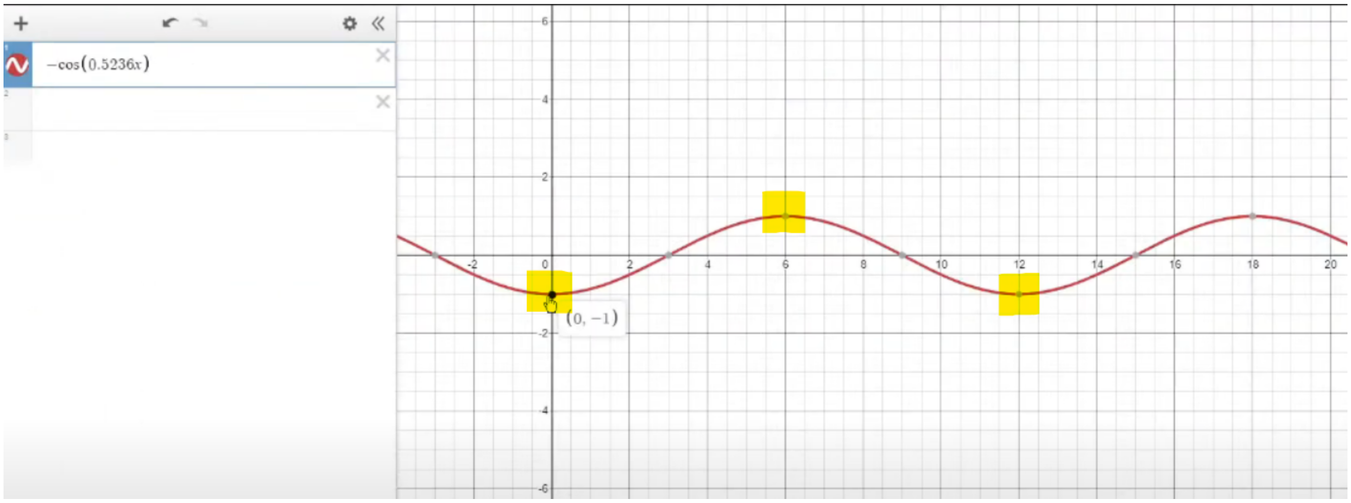
To make skewed numeric features more normally distributed, a log transformation is applied to columns which had an absolute skewness value exceeding 0.5.

2.11 Special Transformation for Cyclical Features

In the dataset the `MoSold` feature represents the month where the property was last sold in and is a cyclical variable. Simply considering this as a regualr categorical variable with levels from 1 to 12 does not capture the total essence of the information.

For example, We can define a relationship between December and January being colder months and June being a warmer month. To do this we can use a trigonometric function to align the values to the function to give them a cyclic relationship and closely relate levels 1 (January) and 12 (December) while putting 6 (June) on the polar opposite side of the spectrum.

To do this we use the function -cos(cx). Here C= 0.5236. We use desmos.com to find the optimum value of C so that the relationship between the month is defined based on our requirements.



$-\cos(0.5236x)$

Figure 3: Cyclical Transformation of MoSold

2.12 Encoding Categorical Data

The Categorical data has then been converted into numerical data and new columns were generated based on the various levels present in the particular categorical variable.

2.13 Standardization/Scaling

The dataset is standardized using the ‘StandardScaler‘ to ensure that features have a mean of zero and a standard deviation of one to allow ease of data proecssing and storage.

2.14 Data Splitting

Training and Testing sets of the data have been re-separated after the data pre-processing stage. Since our Test set does not possess any labels (SalesPrice) we further divide our Training data into a training and validation set to test our models and compare their effectiveness. This is done with random sampling and a validation size of 20% of the training data.

2.15 PCA

We carried out a Principal Component Analysis to figure out the smallest set of variables that still contain the most amount of information. This is a tradeoff between accuracy and simplicity.

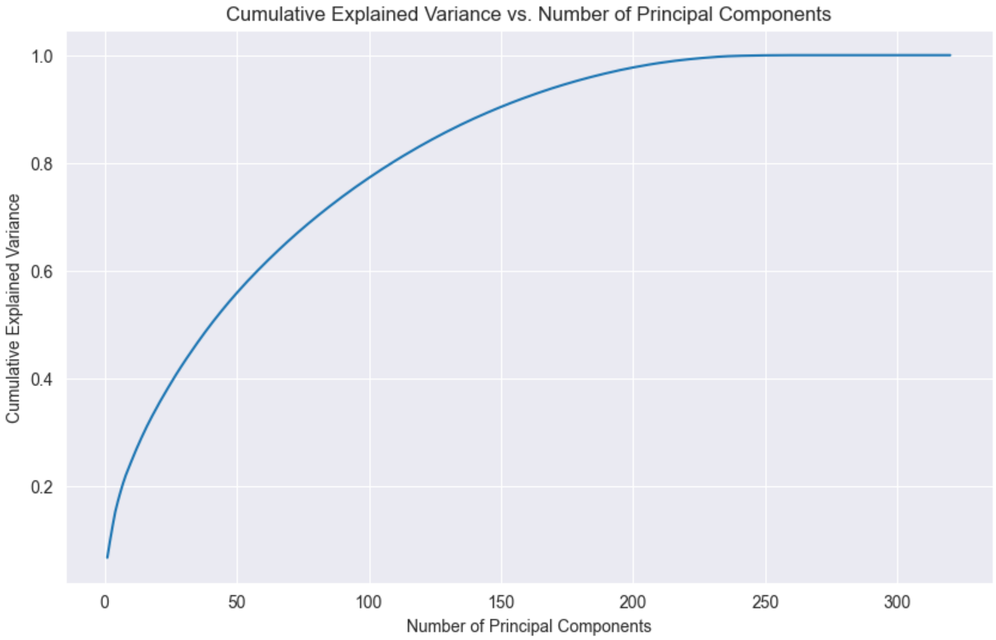


Figure 4: PCA Proportion of Variance Explained

3 Goals

3.1 Main goal of the project

The main goal of this project is to develop accurate predictive models for estimating the final sale price of residential homes in Ames, Iowa, using a dataset comprising various property features and characteristics.

3.2 Specific goals

1. Feature Engineering: To preprocess and engineer the dataset, including handling missing data, encoding categorical variables, and creating relevant features that capture essential information about the properties.

2. Model Building: To build and train predictive regression models using the preprocessed dataset. This includes selecting appropriate algorithms, feature selection, hyperparameter tuning, and model evaluation.

3. Model Evaluation: To assess the performance of the developed models using appropriate evaluation metrics such as Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE). The goal is to achieve the lowest possible error on the test dataset.

4. Model Deployment: If the project permits time, we would also like to create a functional system where users can input property features, and the model will predict the sale price of the property.

3.3 Success metric

The success metric for this project is the R^2 and Root Mean Squared Error (RMSE) on the test dataset.

R^2 is the coefficiet of determination which refers to how well the model fits the given data. The higher the R^2 the better and it ranges between (0,1). We will rank our models based on how high their R^2 values are. We are aiming for values as close to `'1'` as possible.

RMSE measures the average prediction error (in dollars) of the model and provides an indication of how accurately the model estimates the sale prices of homes. The lower the RMSE, the better the model's performance. Achieving a low RMSE indicates that the model can accurately predict sale prices, which aligns with the project's goal. We are aiming for values below `'0.3'`.

3.4 Relation to the problem statement

The problem statement is about predicting the sale prices of residential homes in Ames, Iowa, using a dataset of property features. The main goal of the project is to address this problem statement by developing predictive models that can effectively estimate these sale prices. The specific goals outlined above contribute to achieving this overarching objective.

4 Methods

4.1 Machine Learning Models being Employed

4.1.1 Linear Regression Modeling

A simple Linear Regression model is trained using the training data and validated on the validation data set. This is a good benchmark starting point for the project.

4.1.2 Ridge Regression

A Ridge Regression model is a method of tuning the model with the addition of a shrinkage parameter to deal with the problem of multicollinearity in the data set. It makes use of the L2 norm to shrink coefficient estimates and reduce variance. The tuning parameters considered in this project for Ridge Regression include the fit intercept (Determines whether to calculate the intercept for this model) and the solver(Specifies the algorithm to use for optimization).

4.1.3 Lasso Regression

Least absolute shrinkage and selection operator (LASSO) is a regression method which performs both variable selection using the L1 norm function and regularization (shrinkage) to provide us with regression estimates. The tuning parameters considered in this project for Lasso Regression include the fit intercept (Determines whether to calculate the intercept for this model) and the regularization strength, which controls the amount of shrinkage applied to the coefficients.

4.1.4 Decision Tree Regression

A Decision Tree Regression model is an algorithm which makes use of true or false answers to certain decided question to split up the data in various branches which terminate into nodes. The resulting structure can be visually represented in the form of a tree. The tuning parameters considered in this project for Decision Tree Regression include the criterion based on which the splits are made, the splitter type explains the type of strategy to choose the split at each node, the minimum number of samples required to split an internal node and the maximum number of features to consider when looking for the best split.

4.1.5 Random Forest Regression

Random Forest Regression is an Ensemble learning method that operates via the construction of a multi level decision tree during the training phase. The nodes provide the average value of the output of all the data points in that node. Multiple trees run simultaneously with no relationship between them forming the basis of this ensemble method. The tuning parameters considered in this project for Random Forest Regression include the number of trees in the forest, the criterion based on which the splits are made, the minimum number of samples required to split an internal node and the maximum number of features to consider when looking for the best split.

4.1.6 Gradient Boosting Regression

Gradient Boosting is another ensemble technique that builds an additive model of decision trees. It sequentially adds trees, with each new tree correcting the errors made by the previous ones. Gradient Boosting Regression is known for its high predictive accuracy and can handle both linear and nonlinear relationships. The tuning parameters considered in this project for Gradient Boosting Regression include the number of boosting stages (trees) to be run, The maximum depth of the individual regression estimators (trees) in the ensemble, the minimum number of samples required to split an internal node, the learning rate and the maximum number of features to consider when looking for the best split.

The performance of each individual model is evaluated by calculating the Mean Squared Error

(MSE), Root Mean Squared Error (RMSE), and R-squared (R2). Visualizations are created to compare Residuals vs Fitted Values.

4.2 Explanation of why the methods are relevant to solve the problem

1. **Linear Regression:** Linear regression is relevant because it provides a simple baseline model for predicting home sale prices. It allows for the interpretation of the impact of each feature on the sale price and can serve as a benchmark for more complex models.

2. **Ridge Regression:** Ridge Regression is relevant because it helps in shrinking the coefficients to reduce the variance thus improving accuracy of the model. It also helps rectify issues with the multicollinearity in the data set.

3. **Lasso Regression:** This method will aid in variable selection and understanding feature importance. It performs well when the number of significant parameters are low.

4. **Decision Tree Regression:** Decision Tree Regression is an introduction model into the Tree based decision tree algorithms and the ensemble methods used further along this project. It breaks down the data set into smaller subsets which can be easily interpreted and incrementally developed based on how we choose the cuts.

5. **Random Forest Regression:** Random Forest is relevant because it excels at capturing complex relationships and interactions among features. In real estate, property prices can be influenced by a multitude of factors, and Random Forest can handle these nonlinear relationships effectively.

6. **Gradient Boosting Regression:** Gradient Boosting is relevant because it can improve predictive accuracy by iteratively refining the model and reducing prediction errors. It is well-suited for capturing subtle patterns in the data, which can be crucial in estimating home sale prices accurately.

5 Results

The report summarizes the entire data analysis and modeling process, including data pre-processing, feature engineering, visualization, and regression modeling. The Linear Regression model serves as a starting point for predicting house prices and the more complex models such as Ridge Regression and Boosting aim to improve the overall fit of the model.

5.1 Linear Regression

- **RMSE:** 0.1474
- **R^2:** 0.8635

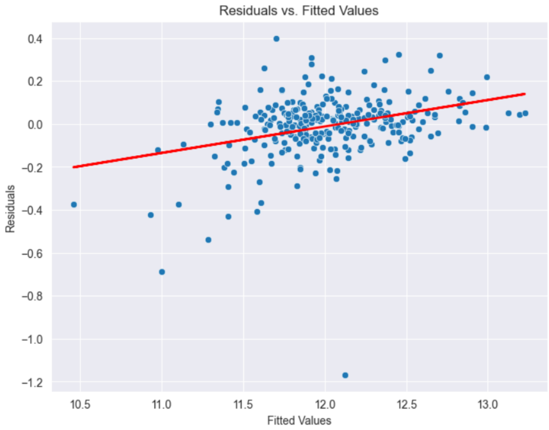


Figure 5: Residuals vs Fitted

5.2 Ridge Regression

- **RMSE:** 0.1475
- **R^2:** 0.8632

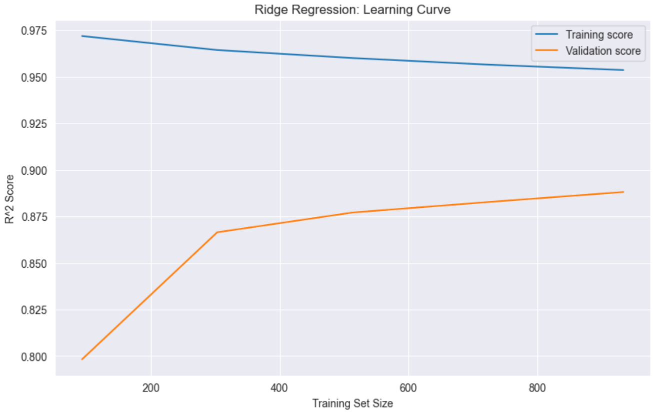


Figure 6: Learning curve

5.3 Lasso Regression

- **RMSE:** 0.149
- **R^2:** 0.859

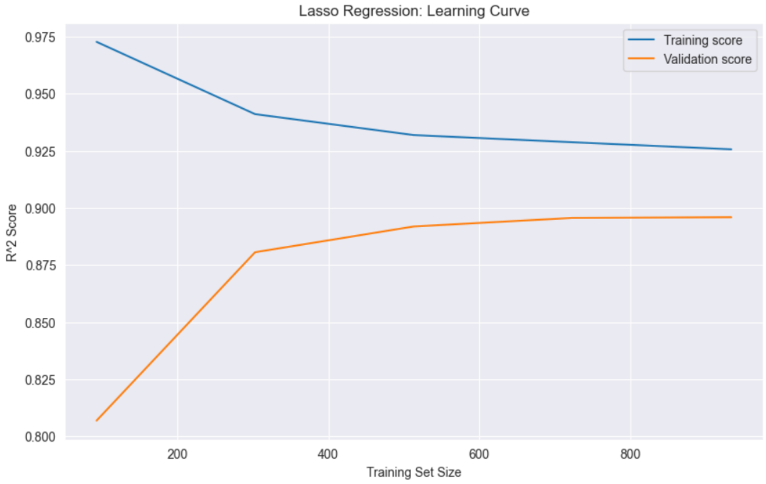


Figure 7: Learning curve

5.4 Decision Tree Regression

- **RMSE:** 0.1737
- **R^2:** 0.7869

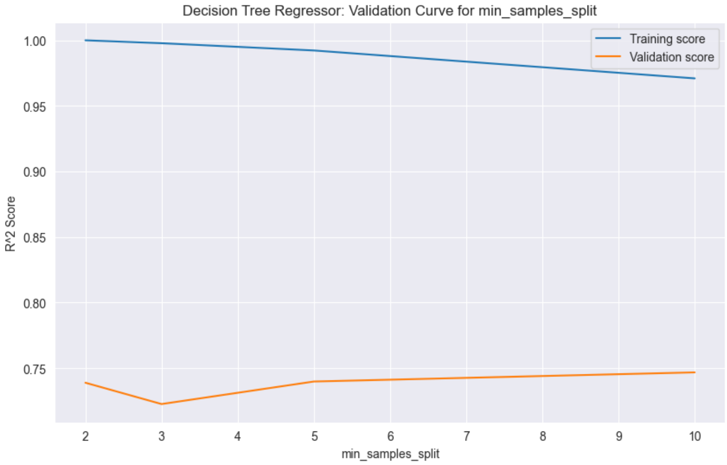


Figure 8: Validation Curve

5.5 Random Forest Regression

- **RMSE:** 0.1036
- **R^2:** 0.9241

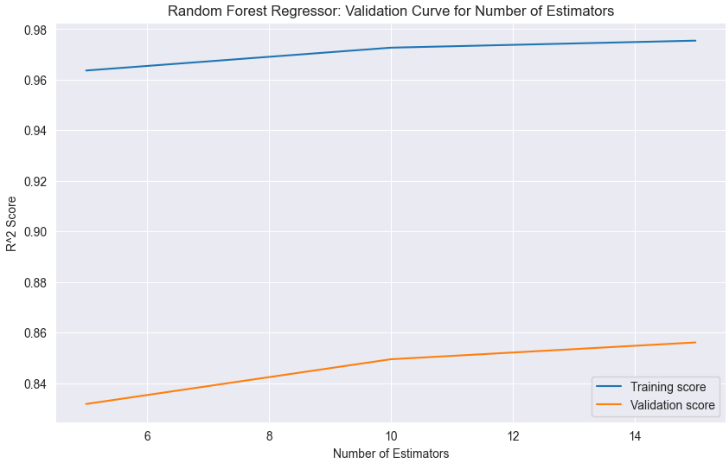


Figure 9: Validation Curve

5.6 Gradient Boosting Regression

- **RMSE:** 0.0852
- **R^2:** 0.9487

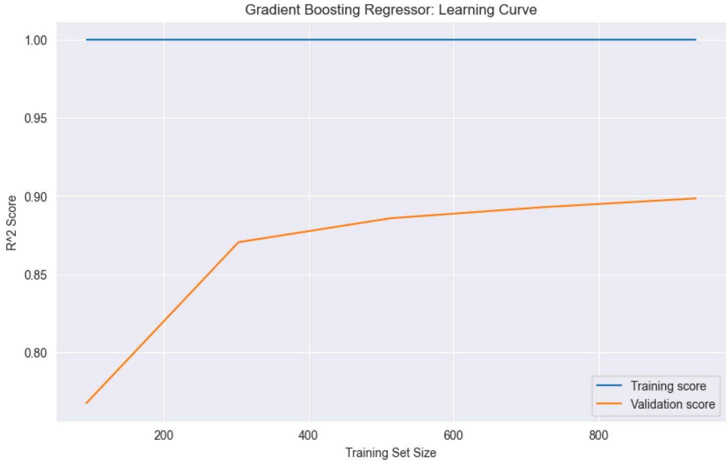
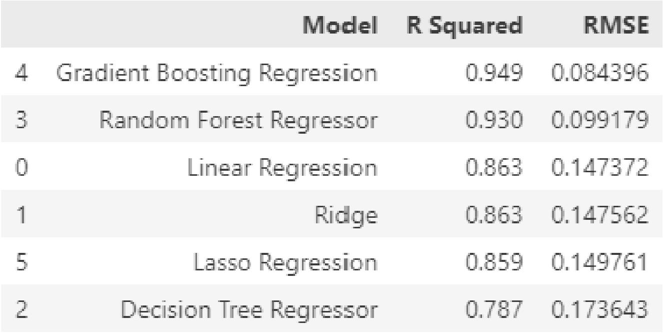


Figure 10: Learning Curve

5.7 Comparison of all the Models



	Model	R Squared	RMSE
4	Gradient Boosting Regression	0.949	0.084396
3	Random Forest Regressor	0.930	0.099179
0	Linear Regression	0.863	0.147372
1	Ridge	0.863	0.147562
5	Lasso Regression	0.859	0.149761
2	Decision Tree Regressor	0.787	0.173643

Figure 11: Models sorted in decreasing order of R^2

As we expected the more complex ensemble methods have risen to the top with better R^2 scores and lower RMSE's. However it is peculiar to note that Linear Regression has outperformed the Ridge Regression model.

5.8 Feature Importance

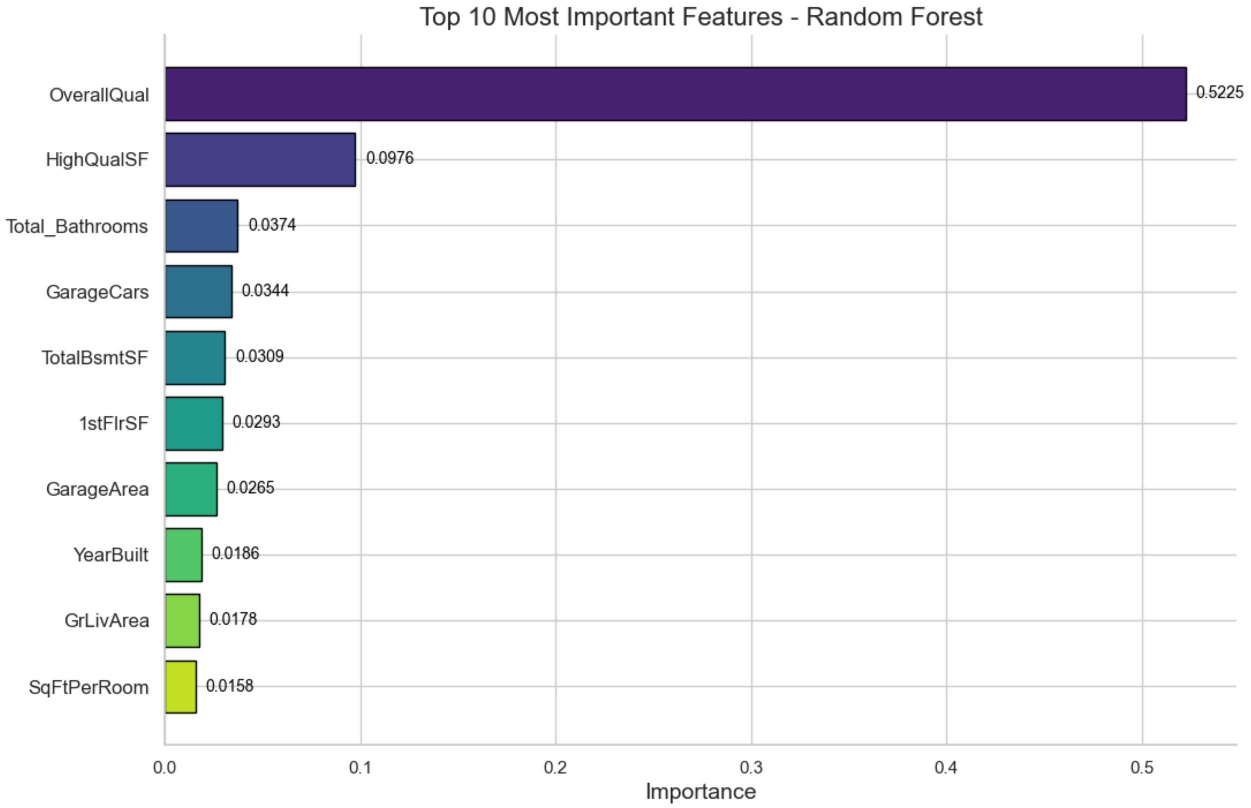


Figure 12: Feature Importance

5.9 Model Performance with PCA

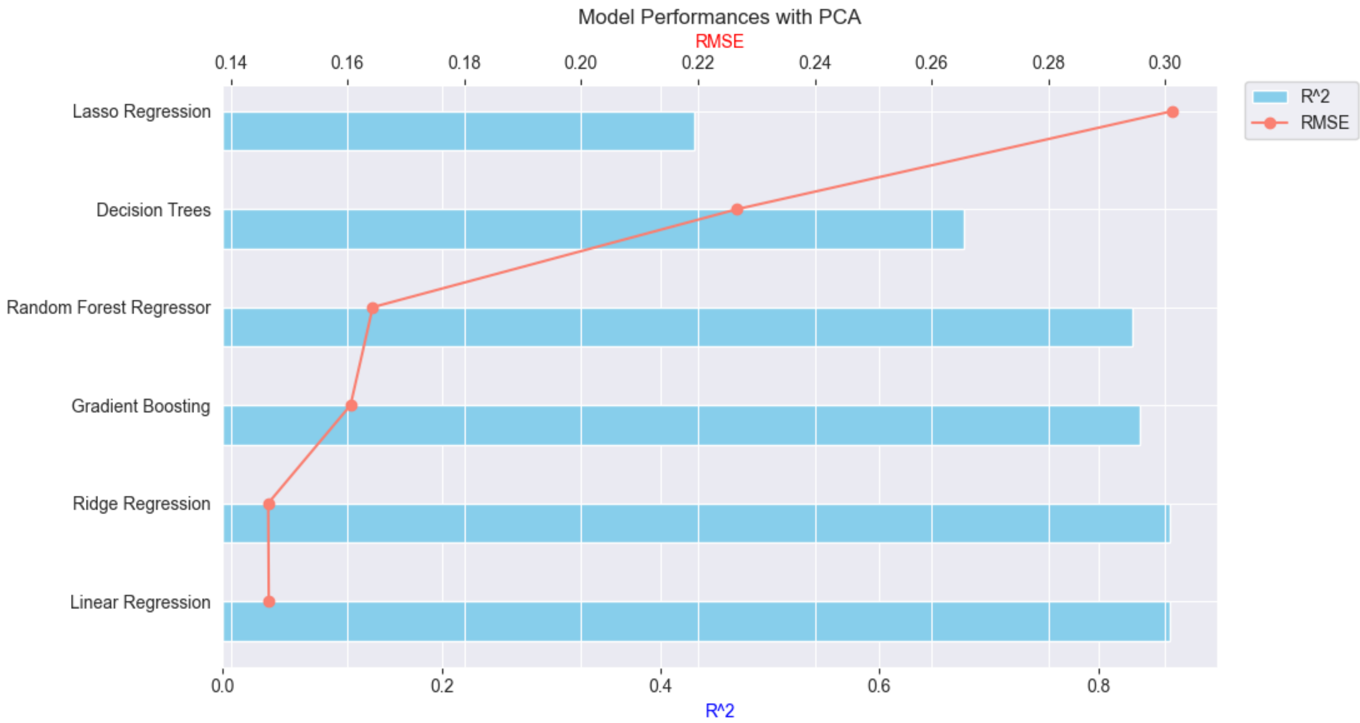
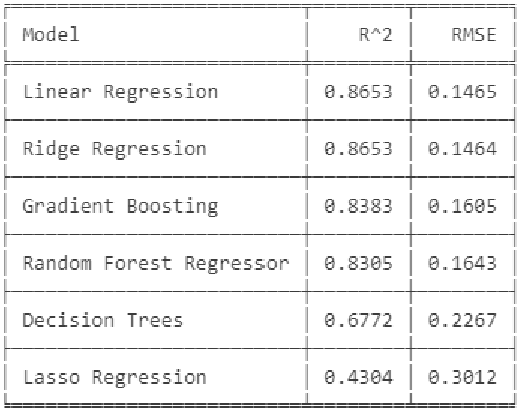


Figure 13: Model Performance with PCA



Model	R^2	RMSE
Linear Regression	0.8653	0.1465
Ridge Regression	0.8653	0.1464
Gradient Boosting	0.8383	0.1605
Random Forest Regressor	0.8305	0.1643
Decision Trees	0.6772	0.2267
Lasso Regression	0.4304	0.3012

Figure 14: PCA Model performance Table

6 Challenges/Lessons Learned

When we look at the current iteration of the linear regression model, we observe that certain observations had to be dropped since they were negatively affecting the performance of the model. Further investigation into why this is occurring could provide some useful insight into the data.

Another key takeaway we had was that we should have considered feature selection and PCA earlier on in the project lifecycle to better correlate our findings. This would perhaps open up an opportunity to find trade offs between model simplicity and accuracy.

A greater focus on the presentation of our results to the classroom should have also been higher up on our priority list.

7 Team Contribution

- **Group Member 1:** Harsh Agarwal - 33.33%
- **Group Member 2:** Rohith Rayson - 33.33%
- **Group Member 3:** Aman Sanjeev Kumar - 33.33%
- **Group Member 4:** Krutik Sunil Panchal - 0.00%