Econ491FinalProject

2024-04-23

Introduction

The primary research question that is driving this project is exploring which variables have the largest influence on changes in house prices. Understanding the most significant factors that drive housing market dynamics is essential for a broad range of stakeholders such as economists, policymakers, real estate professionals, and potential homeowners. By gaining an actional and nuanced insight into the factors that anchor changes in house prices we can expand both our theoretical and practical economic knowledge of the functionalities and intricacies. of the housing market

Therefore a core objective of this research project is to dissect the complexities of the housing market by creating a sophisticated statistical methodology to identify which predictor variables are the most influential in determining house prices. A comprehensive approach will be employed by implementing Machine Learning techniques such as Random Forest, XGBoost, Principal Component Analysis, and Lasso Regularization, for their feature importance metrics. These metrics are pivotal in providing nuanced insights into which specific variables are the most important in affecting house prices from a largely statistical and mathematical perspective.

By leveraging these Machine Learning Techniques, the research project will not only identify key variables but it will also allow for an extremely methodological comparison with existing economic and econometric literature. This dual approach will enable us to see how our insights into what drives changes in house prices compare with economic theories that are posited by economic researchers and housing market experts. Moreover, this comparative analysis provides a deeper understanding of the housing market that surpasses the empirical and theoretical knowledge gained from traditional analytical methods.

Moreover, a secondary yet equally pivotal objective of this project is to ascertain which machine learning predictive model is the most suitable for determining housing prices. This involves a systematic evaluation of a wide range of predictive models to determine the optimal model. The process we will implement will also make use of cross-validation to identify the most optimal hyperparameters for each of the predictive models and then compare them using performance metrics such as Mean-Squared Error, Mean Absolute Error, and the coefficient of determination R^2.

The importance of answering these questions transcends academic curiosity, as gaining a deep understanding of the complexities and nuances of the housing market has real-world applications. Accurately identifying the key predictors in influencing housing market prices can enhance investment decisions, shape governmental policies, and influence urban economic development. We can also further our development of theoretical and practical economic knowledge. We can explore beyond basic economic principles and identify the microeconomic and macroeconomic indicators that drive the economics of real estate and economic development. Moreover, this approach also enables us to further our exploration of the subfield of behavioral economics as we can examine how human behavior is influenced by interactions with the housing market. Moreover, the research project also underscores the importance of machine-learning techniques as by utilizing a variety of supervised and unsupervised learning methods we can make sense of larges swathes of data and can then create a quantitative framework to analyze or critique existing economic theories and generate new ones.

Literature Review

Data Prepocessing and Exploratoroy Data Analysis

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(ggplot2)
```

Decscirption of the Dataset: The dataset is one that is sourced from kaggle which has been further sourced from the Ames Housing Dataset which is a compilation of data describing every aspect of residential homes in Ames Iowa and their prices. The features range from the Sale price of the house to the structural properties of the house such as the lot shape and pool area. The data also includes geographic descriptors of the house such as what neigbourhood it is in and also provides the overall layout of the house by providing features that describe the type of utilities present in the house.

```
library(tidyverse)

df <- read.csv("/Users/adityakakarla/Downloads/archive (9)/AmesHousing.csv")
dim(df)
```

```
## [1] 2930   82
```

```
colSums(is.na(df))
```

```
##           Order             PID     MS.SubClass       MS.Zoning    Lot.Frontage
##               0               0               0               0             490
##        Lot.Area          Street           Alley       Lot.Shape    Land.Contour
##               0               0            2732               0               0
##       Utilities      Lot.Config      Land.Slope    Neighborhood     Condition.1
##               0               0               0               0               0
##     Condition.2       Bldg.Type     House.Style    Overall.Qual    Overall.Cond
```

```
##              0              0              0              0              0
##     Year.Built Year.Remod.Add     Roof.Style      Roof.Matl   Exterior.1st
##              0              0              0              0              0
##   Exterior.2nd   Mas.Vnr.Type   Mas.Vnr.Area     Exter.Qual     Exter.Cond
##              0              0             23              0              0
##     Foundation      Bsmt.Qual      Bsmt.Cond  Bsmt.Exposure BsmtFin.Type.1
##              0             79             79             79             79
##   BsmtFin.SF.1 BsmtFin.Type.2   BsmtFin.SF.2    Bsmt.Unf.SF  Total.Bsmt.SF
##              1             79              1              1              1
##        Heating     Heating.QC    Central.Air     Electrical    X1st.Flr.SF
##              0              0              0              0              0
##    X2nd.Flr.SF Low.Qual.Fin.SF    Gr.Liv.Area Bsmt.Full.Bath Bsmt.Half.Bath
##              0              0              0              2              2
##      Full.Bath      Half.Bath  Bedroom.AbvGr  Kitchen.AbvGr   Kitchen.Qual
##              0              0              0              0              0
##  TotRms.AbvGrd     Functional     Fireplaces   Fireplace.Qu    Garage.Type
##              0              0              0           1422            157
##  Garage.Yr.Blt  Garage.Finish    Garage.Cars    Garage.Area    Garage.Qual
##            159            157              1              1            158
##    Garage.Cond    Paved.Drive   Wood.Deck.SF  Open.Porch.SF Enclosed.Porch
##            158              0              0              0              0
##    X3Ssn.Porch   Screen.Porch      Pool.Area        Pool.QC          Fence
##              0              0              0           2917           2358
##   Misc.Feature       Misc.Val        Mo.Sold        Yr.Sold      Sale.Type
##           2824              0              0              0              0
## Sale.Condition      SalePrice
##              0              0
```

Remove PoolQC, Fence and MisctFeature and impute the rest

```
total_nas_remaining <- sum(is.na(df))
total_nas_remaining
```

```
## [1] 13960
```

To deal with the null values and missing values I use two different methods. First I remove features where maority of the instances are null and missing values. These features include Alley, Misc.Feature, Pool.QC, Fence and Fireplace.Qu. The second technique I use for features with a small or moderate amount of null or missing values. This technique is called imuputation, in which I replaced the null/missing values with the mean of the feature for continius feature variables and the mode of the feature for categorical variables.

```
df_trans <- df %>% select(-c(Alley, Misc.Feature, Pool.QC, Fence, Fireplace.Qu))
```

The code for Imputation Techniques are provided below

```
mode_function <- function(x) {
  ux <- unique(x[!is.na(x)])
  ux[which.max(tabulate(match(x, ux)))]
}
df_trans <- df_trans %>%
  mutate(across(where(is.numeric), ~ifelse(is.na(.), mean(., na.rm = TRUE), .))) %>%  # Mean imputation
  mutate(across(where(is.character), ~ifelse(is.na(.), mode_function(.), .)))  # Mode imputation for ch
```

```
df_trans <- df_trans %>%
  mutate(across(where(is.factor), ~ifelse(is.na(.), as.factor(mode_function(as.character(.))), .)))
```

The code below simulates label encoding in which I convert the categorical variables. This is done through the process of giving each category with a variable a distinct number. In other words it converts categorical features into numerical ones. An example of this is converting a Gender variable with two levels of Male and Female inot 0 and 1.

```
cateogorical_cols <- sapply(df_trans, function(x) is.factor(x) || is.character(x))
cateogorical_cols_names <- names(df_trans)[cateogorical_cols]

for (col in cateogorical_cols_names) {
  df_trans[[col]] <- as.numeric(factor(df_trans[[col]]))
}
```

In the data preprocssing stage the impost important thing that I have checked which columns in the testing and training datasets have null or missing values. First i have dropped the columns that do are majority misisng values and then I hahve used mean imutation for integer based columns and medina imutations for cateogricial colums.

```
dim(df_trans)
```

```
## [1] 2930   77
```

Now I will implement a training and testing split

```
set.seed(123)

train_index <- createDataPartition(df_trans$SalePrice , p = 0.7, list = FALSE)

training_df <- df_trans[train_index,]
testing_df <- df_trans[-train_index,]
```

```
colnames(training_df)
```

```
##  [1] "Order"           "PID"             "MS.SubClass"     "MS.Zoning"
##  [5] "Lot.Frontage"    "Lot.Area"        "Street"          "Lot.Shape"
##  [9] "Land.Contour"    "Utilities"       "Lot.Config"      "Land.Slope"
## [13] "Neighborhood"    "Condition.1"     "Condition.2"     "Bldg.Type"
## [17] "House.Style"     "Overall.Qual"    "Overall.Cond"    "Year.Built"
## [21] "Year.Remod.Add"  "Roof.Style"      "Roof.Matl"       "Exterior.1st"
## [25] "Exterior.2nd"    "Mas.Vnr.Type"    "Mas.Vnr.Area"    "Exter.Qual"
## [29] "Exter.Cond"      "Foundation"      "Bsmt.Qual"       "Bsmt.Cond"
## [33] "Bsmt.Exposure"   "BsmtFin.Type.1"  "BsmtFin.SF.1"    "BsmtFin.Type.2"
## [37] "BsmtFin.SF.2"    "Bsmt.Unf.SF"     "Total.Bsmt.SF"   "Heating"
## [41] "Heating.QC"      "Central.Air"     "Electrical"      "X1st.Flr.SF"
## [45] "X2nd.Flr.SF"     "Low.Qual.Fin.SF" "Gr.Liv.Area"     "Bsmt.Full.Bath"
## [49] "Bsmt.Half.Bath"  "Full.Bath"       "Half.Bath"       "Bedroom.AbvGr"
## [53] "Kitchen.AbvGr"   "Kitchen.Qual"    "TotRms.AbvGrd"   "Functional"
## [57] "Fireplaces"      "Garage.Type"     "Garage.Yr.Blt"   "Garage.Finish"
```

```
## [61] "Garage.Cars"     "Garage.Area"     "Garage.Qual"     "Garage.Cond"
## [65] "Paved.Drive"     "Wood.Deck.SF"    "Open.Porch.SF"   "Enclosed.Porch"
## [69] "X3Ssn.Porch"     "Screen.Porch"    "Pool.Area"       "Misc.Val"
## [73] "Mo.Sold"         "Yr.Sold"         "Sale.Type"       "Sale.Condition"
## [77] "SalePrice"
```

Ensemble Methods

In the first phase of the resarch project we will focus on fitting ensemble models. Ensemble models are supervised machine learning techniques that combine multiple, yet indiviudals model in order to improve the overall performance of the model. it follows the core idea that a groip of weak learners can be combined to create a strong learner. Ensemble models can be categorized into two groups, which are bagging and boostong.

The first machine learning ensemble technique that we will employ a supervised statistical method called Random Forest, which is an example of a bagging ensemble method. Random Forest is a versatile machine learning algorithm that is used for both regression and classification tasks. It is based around the idea of creating multiple decision trees during the training phase and then outputing the mean of all the predictions of individual trees for regression or using majority voting for classifications. A core part of the functionality of Random Forest is something known as bootstrapping aggregating, or in other words bagging, which involves randomly sampling from the original data set with replacement, and then fitting each new bootstrapped sample to each individual decision tree - therefore a different random sample of the original dataset is fit onto each individual tree. Not only does Random Forest rely on bagging but it also utilzities feature randomness when fitting each individual decision tree. This means that at each split of the decision tree it select a random subset of features rather than using all the feeatures in the datset.

Another important aspect of the Random Forest is it has the ability the measure the importance of each predictor variable in making predictions. Feature importance in Random Forest is based on how much each feature decreases the impurity of the nodes in the tree, This often done by measuring the change in the Gini Impurity for classification task and Mean-Squared Eror for regression tasks. Each feature's importance is then average over all if the trees to provide a final measure of importance.

The second machine learning ensemble tehcnique that I will employ is Gradient Boosting Method, which is an example of a boosting ensemble methid. Gradient Boosting is based on the idea combining the predictions of multiple weak learners, for instance decision trees in a sequentiall manner. In other words in sequentially trains multiple individual models, and learns from the weakness of the predeceasing model and each new model assingmns weights according to a certain criteria.

Our goal in this phase is to provide a comprehensive comparative analysis of how the two ensemble models are similar and different in how they use feature importance scores to determine which predictor variables are the most significant in explaining the change in house prices. Moreover, by comparing the error metrics of both models on their predictions on the test data we can want to see which model woudl be more optimal in acting as a predictive model for house prices.

We avoid the use of cross-validation for the ensemble techniques in order to reduce the computational complexity and increase computational speed.

Random Forest Implementation:

For the Random Forest I choose the pick the number of trees as 500 and the number of predictor variables used at each split as 25.

```
library(caret)
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
set.seed(123)

rf_model <- randomForest(SalePrice ~ . ,
                         data = training_df,
                         ntree = 500, mtry = 25,
                         importance = TRUE)

predictions_rf <- predict(rf_model, testing_df)


mse_value_rf <- mean((testing_df$SalePrice - predictions_rf)^2)
mae_value_rf <- mean(abs(testing_df$SalePrice - predictions_rf))
rmse_value_rf <- sqrt(mse_value_rf)
rsquared_value_rf <- cor(testing_df$SalePrice, predictions_rf)^2
```

Adaboost Implementation:

```
set.seed(123)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(ada)
```

```
## Loading required package: rpart
```

```
library(gbm)
```

```
## Loaded gbm 2.1.9

## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.c
```

```
gbm_model <- gbm(SalePrice ~., data = training_df,
                 distribution = "gaussian",
                 n.trees = 5000, interaction.depth = 4,
                 shrinkage = 0.01, cv.folds = 5,
                 n.minobsinnode = 10)


predictions_gbm <- predict(gbm_model, testing_df, n.trees = 5000)


mse_value_gbm <- mean((testing_df$SalePrice - predictions_gbm)^2)
mae_value_gbm <- mean(abs(testing_df$SalePrice - predictions_gbm))
rmse_value_gbm <- sqrt(mse_value_gbm)
rsquared_value_gbm <- cor(testing_df$SalePrice, predictions_gbm)^2
```

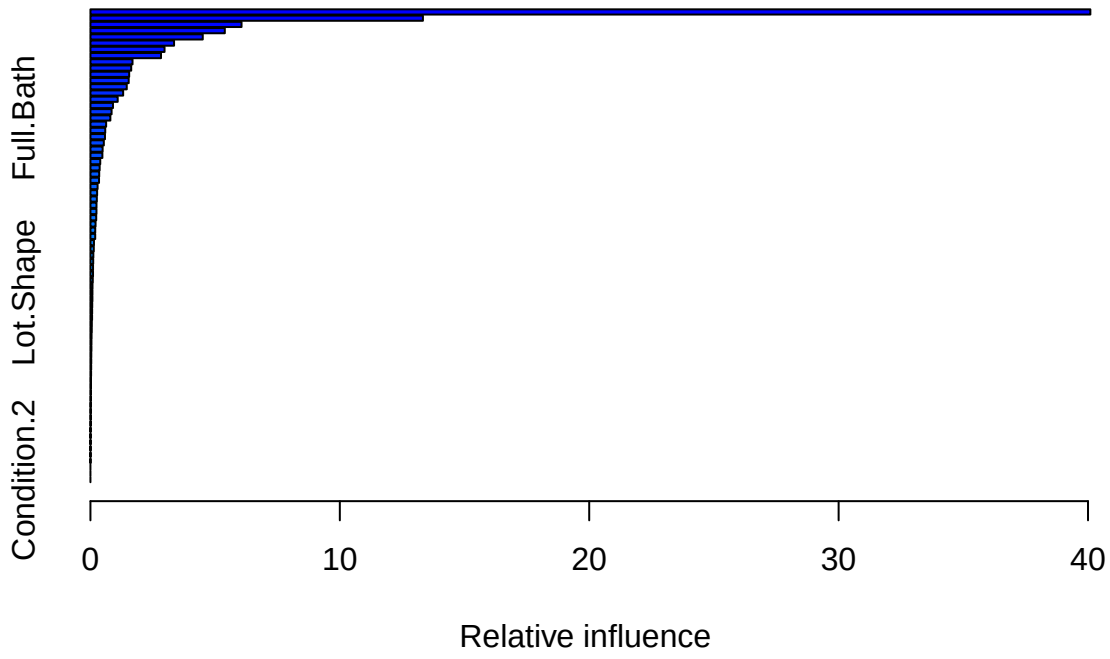
Interpretation of Variable Importance for Ensemble Techniques The table below depicts the the variable importance measures of the top 10 most important predictor varibales in the Random Forest model based on %IncMSE, which is a metric that shows the increase in MSE of the model when a variable is randomly permuted - a higher value indicates that the variable is more important.

```
rf_importance <- as.data.frame(importance(rf_model))
rf_importance <- rf_importance %>% arrange(desc(`%IncMSE`)) %>% head(10)
rf_importance
```

```
##                %IncMSE IncNodePurity
## Gr.Liv.Area   41.84634  1.221080e+12
## Overall.Qual  27.80653  3.154381e+12
## BsmtFin.SF.1  22.51038  2.476918e+11
## X1st.Flr.SF   21.99826  5.955559e+11
## Total.Bsmt.SF 19.88594  6.367878e+11
## X2nd.Flr.SF   19.29479  1.381579e+11
## Year.Built    19.25595  1.267977e+12
## Fireplaces    19.18173  1.796301e+11
## Garage.Area   16.90543  3.814152e+11
## Overall.Cond  16.85627  5.981742e+10
```

Below is the extraction of the variable importance results for Gradient Boosting Model. The table shows the top 10 predictor variables with the highest relative importance in predicting the outcome.

```
importance_gbm <- summary(gbm_model) %>%
  arrange(desc(rel.inf)) %>% head(10)
```




```
importance_gbm
```

```
##                          var   rel.inf
## Overall.Qual   Overall.Qual 40.094239
## Gr.Liv.Area     Gr.Liv.Area 13.332017
## Total.Bsmt.SF Total.Bsmt.SF  6.058584
## Garage.Cars     Garage.Cars  5.388320
## BsmtFin.SF.1   BsmtFin.SF.1  4.504123
## X1st.Flr.SF     X1st.Flr.SF  3.353731
## Year.Built       Year.Built  2.971919
## Lot.Area           Lot.Area  2.829747
## Mas.Vnr.Area   Mas.Vnr.Area  1.686810
## Fireplaces       Fireplaces  1.635007
```

From the importance score generated by both ensemble models we can clearly see that Overall.Qual(Overall Quality of the House) and Gr.Liv.Area(living Area of the house) are the two most important predictors in determining house prices. Additionally variable such as Total.Bsmt.SF(Basement Area), Year.Built, Fireplaces and X1st.Flr.SF(First Floor Area) are also amongst the most important predictors. This suggests that our ensemble techniques shows us that the physical charactersitcs of the house are the most important determinant in the house prices.

This suggests that enseble techniques conclude that variables intrinsic to the model itself are the most important in determine house prices. If we compare this to economic literature we already analyzed we can see that there both similarities and differences in this idea. For instance, if we compare conclusions drawn from the research paper titled "Determinants of housing values and variations in home prices across neighborhoods in Cook County" by by Maude Toussaint-Comeau and Jin Man Lee we can see that there

is a similarity in the emphasis on the physical characteristics of the House being important in determing house prices. Moreover, the paper also states that "that square footage and the lot size are the largest determination features for housing price" - both these variables show up in the ten most important variable for both the Random Forest and the Gradient Boosting model. The article also states that "Other amenities, such as a garage, brick exterior, fireplace, and central air conditioning, all have a positive effect on house price" - this another similarity between the ensemble models and the paper as both random forest and GBM boost contain predictos that depicts characteristics related to garages and fireplaces such as Garage.Area, Garage.Cars and Fireplaces in their ten most important variables. This parallel underscores the shared understanding in waht determines the price of houses.

In contrast to this, the research paper also highlights the importance on the characteristics of the neighborhood that the house is in, for instance the paper states that waterfront properties and are associated with higher prices, while being closer to public train stops can lower price significantly. The interplay between the characteristics of the neighborhood that the houses are located in and the house prices is a crucial aspect that our ensemble models did not recognize.

Another extremely important aspect that our models did not recognize was the importance of macroeconomic indicators in determining house prices. Research papers such "Fundamental Drivers of House Prices in Advanced Economies" by the International Monteray Fund's Nan Geng and "Do the Determinants of House Prices Change over Time? Evidence from 200 Years of Transactions Data" by Amserdam Buisness School's Martijn I. Droes and Alex van de Minnet, suggest that macroeconomic indicaotrs such the GDP per Capita, Unemployment Levels and levels of housing supply and subsets of house income are more important than any other factors in determining the price of houses. Moreover the paper by the IMF also suggests that government policies like rent control and demographic trends such as changes in the wroking age population and changes in the age structure of geographic location can also be extremely influential in determining the price of houses. These elements reflect the broader economic context within the house marekt that influences marekt dynamics and ultimately house prices. Through this we can see that understanding the factors that cause changes in house prices require deeper understanding and larger complexity in analytically methods.

```
data.frame(Type = c("Random Forest", "GBM"),
           MSE = c(mse_value_rf, mse_value_gbm),
           rmse = c(rmse_value_rf, rmse_value_gbm),
           mae = c(mae_value_rf, mae_value_gbm),
           r_squared = c(rsquared_value_rf, rsquared_value_gbm ))
```

Interpretation Error Metrics for Ensemble Techniques

```
##            Type       MSE     rmse      mae r_squared
## 1 Random Forest 727719084 26976.27 15144.73 0.9079913
## 2           GBM 556972597 23600.27 13706.23 0.9238298
```

From the table above that provides a comparative analysis of the MSE, RMSE, MAE and rsquared between the Random Forest and GBM model we can that GBM tends to have a superiro performance in every error metrics as it has a lower MSE, Lower RMSE, lower MAE and higher R-squared value. This mean that not only does GBM lead to less noise and errors when predicting house prices, but it reduces overfitting and explains the variability in house prices better than a Random Forest Model does.

From a statisical perspective the superiority of the GBM model can be explained by the fact that combines the predictions of multiple weak learners, for instance decision trees in a sequentiall manner. This means that it learns from the mistakes of every prvious decision tree, which means it can sequentially correct errors. Through this it has the ability to indentify more complex trends in the underlying data distribution. Mreover, while Random Forests focus on reducing variance, GBMs focus on optimzing the loss functon and therefore have the core objective of reducing loss and error.

Linear Models

The second pahse of the research process will focus on a comparative analysis of linear regression models and how they differ in determining which feature variables have the most influence on the change in house price and what that influence entails, for example is it a strong positive relationship or a strong negative relationship. We implement this process by fitting a traditional linear regression model and then comparing it with a Ridge, Lasso and Elastic Net Regression which are regularization techniques. The comparative analysis will include an analysis of how Lasso and Ridge regression shrink the coefficients of the predictor variables in comparison to linear regression and then we will identifiy the variables with the most significant positive, negative and absolute values. Another core objective of this phase is to verify the conclusions and results we have gained from the initial phase of the research project that focuses on ensemble models

In addition to this, I will also implement cross-validation on the Lasso and Ridge Regression models in order to determine the optimal model for each by identify the optimal regularization parameter called lambda, which is the penalty term for the models. We will then make predictions for each model using the testing data and then compare error metrics such as MSE, MAE, RMSE and R-Squared in order to determine which linear model is the most optimal, in other words leads to noise and error, when determining future house prices.

Before, we implement the first phase of the research process, it is also important to understand what lasso, ridge and elastic net regression. Lasso, Ridge and Elastic Net Regression are essentially regularization techniques that add a penalty to the coefficients of the predictor variables of linear regression model and shrink them in order to prevent overfitting and improve generalization to new unseen data. Lasso Regression adds a penalty to the absolute value of the sum of the coefficients, and can perform variable selection by setting some of the coeffcients to zero. Ridge regressions adds a penalty equal to the square of the magnitude of the coefficients and while it shrinks the coefficients towards zero it does not shrink them to zero like Lasso regression. Elastic Net acts as intermediary between Lasso and Ridge Regression as it combines the properties of both, for instance it can add a penalty equal to the square of the magnitude of the coefficients and add a penalty to the absolute value of the sum of the coefficients. Therefore it can shrink some coeffcients towards zero and shrink others to zero.

Linear Regreesion First I will fit a simple linear model using the training data

```
linear_model <- lm(SalePrice ~. ,data  = training_df)

linear_model_summary <- summary(linear_model)
linear_model_summary
```

```
##
## Call:
## lm(formula = SalePrice ~ ., data = training_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -446426  -14180    -928   11906  196353
##
## Coefficients: (2 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      8.910e+06  8.831e+06   1.009 0.313119
## Order           -5.433e+00  6.925e+00  -0.784 0.432849
## PID             -7.192e-06  7.135e-06  -1.008 0.313590
## MS.SubClass     -8.669e+01  3.273e+01  -2.649 0.008141 **
## MS.Zoning       -7.500e+02  8.600e+02  -0.872 0.383247
## Lot.Frontage    -1.524e+02  4.191e+01  -3.637 0.000283 ***
```

```
## Lot.Area         4.988e-01  8.970e-02   5.561 3.05e-08 ***
## Street           1.921e+04  9.012e+03   2.132 0.033139 *
## Lot.Shape       -7.525e+02  5.001e+02  -1.505 0.132513
## Land.Contour     6.944e+02  1.013e+03   0.685 0.493227
## Utilities       -2.106e+03  9.993e+03  -0.211 0.833071
## Lot.Config       1.267e+02  4.180e+02   0.303 0.761833
## Land.Slope       5.511e+03  3.098e+03   1.779 0.075422 .
## Neighborhood     9.979e+01  1.025e+02   0.974 0.330265
## Condition.1      1.531e+02  7.634e+02   0.201 0.841051
## Condition.2      3.527e+03  3.267e+03   1.080 0.280454
## Bldg.Type       -3.272e+03  1.109e+03  -2.950 0.003215 **
## House.Style     -1.376e+03  4.890e+02  -2.814 0.004945 **
## Overall.Qual     1.216e+04  8.830e+02  13.771  < 2e-16 ***
## Overall.Cond     4.104e+03  7.662e+02   5.356 9.48e-08 ***
## Year.Built       1.853e+02  5.674e+01   3.267 0.001107 **
## Year.Remod.Add   3.272e+01  5.084e+01   0.644 0.519929
## Roof.Style       3.010e+03  8.710e+02   3.455 0.000561 ***
## Roof.Matl       -1.616e+03  1.454e+03  -1.111 0.266575
## Exterior.1st    -1.050e+03  3.732e+02  -2.814 0.004934 **
## Exterior.2nd     6.292e+02  3.447e+02   1.825 0.068080 .
## Mas.Vnr.Type     3.137e+03  6.957e+02   4.510 6.87e-06 ***
## Mas.Vnr.Area     3.847e+01  4.694e+00   8.196 4.41e-16 ***
## Exter.Qual      -9.609e+03  1.474e+03  -6.518 9.03e-11 ***
## Exter.Cond      -2.804e+02  9.268e+02  -0.303 0.762250
## Foundation       1.030e+03  1.224e+03   0.842 0.400164
## Bsmt.Qual       -4.784e+03  7.586e+02  -6.306 3.51e-10 ***
## Bsmt.Cond        2.150e+03  9.600e+02   2.239 0.025253 *
## Bsmt.Exposure   -3.437e+03  6.893e+02  -4.986 6.70e-07 ***
## BsmtFin.Type.1  -7.771e+02  4.831e+02  -1.609 0.107879
## BsmtFin.SF.1     1.172e+01  3.261e+00   3.595 0.000332 ***
## BsmtFin.Type.2  -5.254e+02  9.713e+02  -0.541 0.588633
## BsmtFin.SF.2     2.732e+00  6.503e+00   0.420 0.674416
## Bsmt.Unf.SF      1.783e+00  2.993e+00   0.596 0.551462
## Total.Bsmt.SF           NA         NA      NA       NA
## Heating         -8.690e+02  2.549e+03  -0.341 0.733168
## Heating.QC      -8.540e+02  4.636e+02  -1.842 0.065597 .
## Central.Air     -1.220e+02  3.312e+03  -0.037 0.970630
## Electrical       1.451e+02  6.890e+02   0.211 0.833200
## X1st.Flr.SF      5.378e+01  4.139e+00  12.994  < 2e-16 ***
## X2nd.Flr.SF      4.357e+01  3.831e+00  11.372  < 2e-16 ***
## Low.Qual.Fin.SF  1.700e+01  1.552e+01   1.095 0.273740
## Gr.Liv.Area             NA         NA      NA       NA
## Bsmt.Full.Bath   7.819e+03  1.770e+03   4.416 1.06e-05 ***
## Bsmt.Half.Bath  -1.554e+03  2.800e+03  -0.555 0.579012
## Full.Bath        1.828e+03  1.992e+03   0.918 0.358966
## Half.Bath        1.906e+03  1.932e+03   0.986 0.324026
## Bedroom.AbvGr   -3.331e+03  1.248e+03  -2.669 0.007659 **
## Kitchen.AbvGr   -8.108e+03  3.812e+03  -2.127 0.033552 *
## Kitchen.Qual    -4.304e+03  7.411e+02  -5.808 7.37e-09 ***
## TotRms.AbvGrd    2.209e+03  8.796e+02   2.511 0.012105 *
## Functional       2.654e+03  5.855e+02   4.532 6.19e-06 ***
## Fireplaces       4.563e+03  1.271e+03   3.589 0.000340 ***
## Garage.Type     -8.266e+01  4.712e+02  -0.175 0.860766
## Garage.Yr.Blt    4.906e+01  4.807e+01   1.020 0.307625
```

```
## Garage.Finish   -1.976e+02  1.079e+03  -0.183 0.854728
## Garage.Cars      8.148e+03  2.058e+03   3.959 7.78e-05 ***
## Garage.Area      2.952e-01  7.087e+00   0.042 0.966773
## Garage.Qual     -1.123e+03  1.176e+03  -0.955 0.339596
## Garage.Cond      9.816e+02  1.393e+03   0.705 0.481022
## Paved.Drive      7.457e+02  1.466e+03   0.509 0.610976
## Wood.Deck.SF     1.963e+01  5.833e+00   3.365 0.000779 ***
## Open.Porch.SF   -3.804e+00  1.065e+01  -0.357 0.720941
## Enclosed.Porch   3.092e+01  1.107e+01   2.793 0.005280 **
## X3Ssn.Porch      8.264e+00  2.538e+01   0.326 0.744790
## Screen.Porch     6.418e+01  1.170e+01   5.487 4.62e-08 ***
## Pool.Area       -5.592e+01  1.737e+01  -3.219 0.001309 **
## Misc.Val        -1.515e+01  1.306e+00 -11.601  < 2e-16 ***
## Mo.Sold          6.238e+01  2.402e+02   0.260 0.795089
## Yr.Sold         -4.705e+03  4.392e+03  -1.071 0.284254
## Sale.Type       -1.554e+02  3.618e+02  -0.429 0.667679
## Sale.Condition   3.184e+03  6.206e+02   5.131 3.17e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28490 on 1978 degrees of freedom
## Multiple R-squared:  0.8707, Adjusted R-squared:  0.8659
## F-statistic:   180 on 74 and 1978 DF,  p-value: < 2.2e-16
```

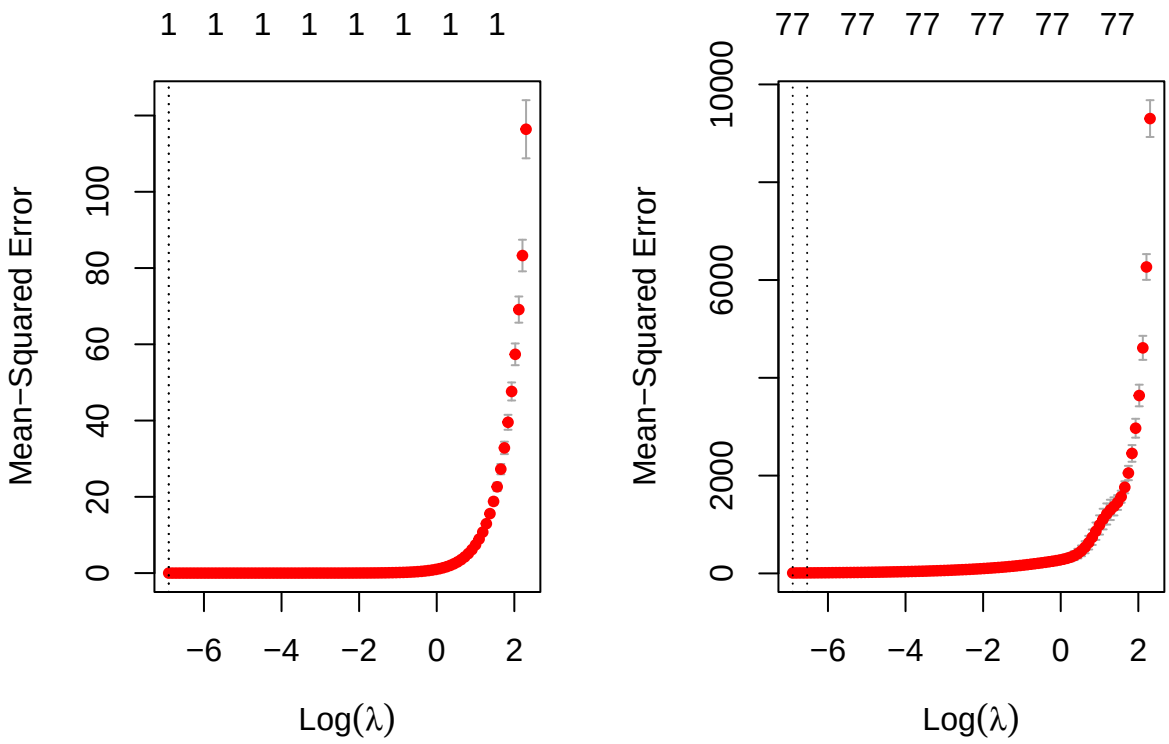
```
set.seed(123)
library(glmnet)
```

Regularization Techniques

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-8
```

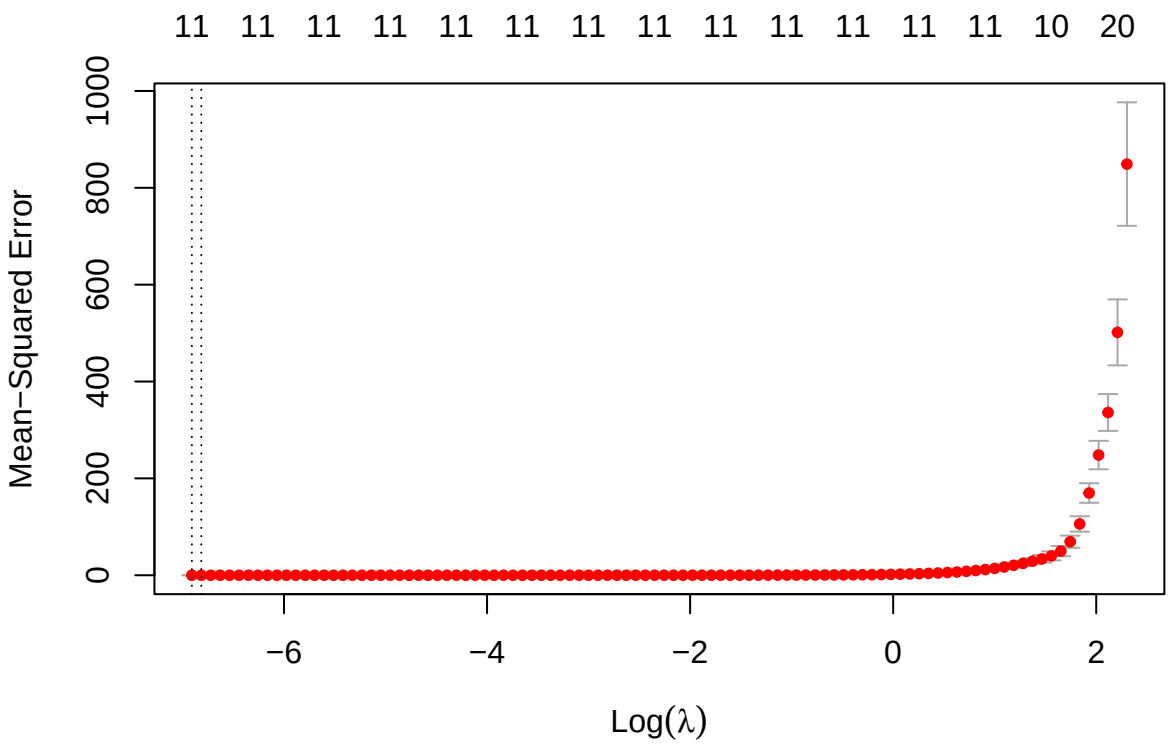
```
x_train <- model.matrix(~ . -1, data = training_df)
y_train <- training_df$SalePrice

cv_lasso <- cv.glmnet(x_train,y_train, alpha = 1,
                      lambda = exp(seq(log(0.001), log(10), length = 100)))
cv_ridge <- cv.glmnet(x_train,y_train, alpha = 0,
                      lambda = exp(seq(log(0.001), log(10), length = 100)))
cv_elastic_net <- cv.glmnet(x_train,y_train, alpha = 0.5,
                            lambda = exp(seq(log(0.001), log(10), length = 100)))
```

```r
par(mfrow = c(1,2))

plot(cv_lasso)
plot(cv_ridge)
```



The First Plot is for Lasso Regression and the Second Plot is for Ridge Regression

```r
par(mfrow = c(1,1))
plot(cv_elastic_net)
```




```r
set.seed(123)
best_lambda_lasso <- cv_lasso$lambda.min
best_lambda_ridge <- cv_ridge$lambda.min
best_lambda_elastic_net <- cv_elastic_net$lambda.min

best_lasso_model <- glmnet(x_train,y_train, alpha = 1, lambda = best_lambda_lasso)
coefficients_lasso <- as.matrix(coefficients(best_lasso_model, s = "lambda.min"))
coefficients_lasso <- as.data.frame(coefficients_lasso, row.names = NULL)
coefficients_lasso_top_10_abs <- coefficients_lasso %>% mutate(Variable = rownames(coefficients_lasso))


best_ridge_model <- glmnet(x_train,y_train, alpha = 0, lambda = best_lambda_ridge)
coefficients_ridge <- as.matrix(coefficients(best_ridge_model, s = "lambda.min"))
coefficients_ridge <- as.data.frame(coefficients_ridge)
coefficients_ridge_top_10_abs <- coefficients_ridge %>% mutate(Variable = rownames(coefficients_ridge))

best_elastic_net_model <- glmnet(x_train,y_train, alpha = 0.5, lambda = best_lambda_elastic_net)
coefficients_elastic_net <- as.matrix(coefficients(best_elastic_net_model, s = "lambda.min"))
coefficients_elastic_net <- as.data.frame(coefficients_elastic_net)

coefficients_elastic_net_top_10_abs <- coefficients_elastic_net %>% mutate(Variable = rownames(coefficie
```

```
coefficients_lm <- as.data.frame(coefficients(linear_model))
coefficients_linear_top_10_abs <- coefficients_lm %>% rename(Coefficients = `coefficients(linear_model)`)

coefficients_linear_top_10_abs$Coefficients
```

Interpratation of Variable Importance

```
##  [1] 8910151.342   19212.675   12159.229   -9608.725    8147.622   -8108.217
##  [7]    7818.917    5511.002   -4784.009   -4704.533
```

```
data.frame(Linear_Model_Variable  = coefficients_linear_top_10_abs$Variable,
           Linear_Model_Coefficients = coefficients_linear_top_10_abs$Coefficients,
           Lasso_Model_Variable  = coefficients_lasso_top_10_abs$Variable,
           Lasso_Model_Coefficients = coefficients_lasso_top_10_abs$Coefficients,
           Ridge_Model_Variable  = coefficients_ridge_top_10_abs$Variable,
           Ridge_Model_Coefficients = coefficients_ridge_top_10_abs$Coefficients,
           Elastic_Model_Variable  = coefficients_elastic_net_top_10_abs$Variable,
           Elastic_Model_Coefficients = coefficients_elastic_net_top_10_abs$Coefficients)
```

```
##    Linear_Model_Variable Linear_Model_Coefficients Lasso_Model_Variable
## 1             (Intercept)               8910151.342          (Intercept)
## 2                  Street                 19212.675              Yr.Sold
## 3            Overall.Qual                 12159.229            Utilities
## 4              Exter.Qual                 -9608.725            Exter.Qual
## 5             Garage.Cars                  8147.622          Paved.Drive
## 6           Kitchen.AbvGr                 -8108.217          Garage.Cars
## 7          Bsmt.Full.Bath                  7818.917          Central.Air
## 8              Land.Slope                  5511.002           Foundation
## 9               Bsmt.Qual                 -4784.009            Bldg.Type
## 10                Yr.Sold                 -4704.533        Bedroom.AbvGr
##    Lasso_Model_Coefficients Ridge_Model_Variable Ridge_Model_Coefficients
## 1                362665.26603          (Intercept)              362800.69657
## 2                  -177.82635              Yr.Sold                -177.89322
## 3                   -70.54782            Utilities                 -70.56570
## 4                    62.47407            Exter.Qual                  62.47817
## 5                    41.03654          Paved.Drive                  41.04841
## 6                    39.39947          Garage.Cars                  39.41883
## 7                   -32.35311          Central.Air                 -32.35348
## 8                   -29.11568           Foundation                 -29.11819
## 9                    27.29529            Bldg.Type                  27.29376
## 10                   26.50043        Bedroom.AbvGr                  26.50308
##    Elastic_Model_Variable Elastic_Model_Coefficients
## 1             (Intercept)               362732.98132
## 2                 Yr.Sold                 -177.85979
## 3               Utilities                  -70.55676
## 4              Exter.Qual                   62.47612
## 5             Paved.Drive                   41.04247
## 6             Garage.Cars                   39.40915
## 7             Central.Air                  -32.35329
## 8              Foundation                  -29.11694
## 9               Bldg.Type                   27.29453
## 10          Bedroom.AbvGr                   26.50175
```

Thr table above output the ten predictor variables that have the highest abosulte coefficients for each of the linear models - through this we can see how each of them differ in regards to determing what are the model important predictor variables in determining house prices.

One importance aspect that we do see across the model is that the three regularization techniques tend to be more similar in terms fo what are the most important variables in comparison to a linear model. This is expected as Lasso, Ridge, and Elastic Net models, being regularization techniques will yield similar parameters rgeading which variables are the most influential since they have common foundation in shrinking cenrtain coeffecients in order to reduce overfitting and deal with multicollienarity amongst predictor variables.

Lasso, Ridge, and Elastic Net models often yield similar assessments regarding which variables are most impactful. This similarity stems from their shared foundation in regularization, which not only helps in reducing overfitting but also in dealing with multicollinearity among predictors

This analysis reveals a strong consensus about the importance of the physical charactersitcs of the house in determining the price of the house. Similarity to the ensemble models we see that characteristics of the house such as characteristics of Garage are the most important determinants. However one notable difference between the ensemble techniques is the lack of importance the linear models give to the predictor variable reating to the overall quality of the house(however it does give importance to the exterior quality of the house. While we the conclsuions gained from implementing the linear models are similar to thoose of the ensemble models we can see that ensemble models did miss out on the temporal nature of the house being an important predictor in infleuncing housing prices.

```
set.seed(123)
x_test <- model.matrix(~ . -1, data = testing_df)
y_test <- testing_df$SalePrice

predictions_lasso <- predict(best_lasso_model, s = "lambda.min", newx = x_test)
actuals <- testing_df$SalePrice
mse_lasso <- mean((predictions_lasso - actuals)^2)
rmse_lasso <- sqrt(mse_lasso)
mae_lasso <- mean(abs(predictions_lasso - actuals))
r_squared_lasso <- 1 - (sum((actuals - predictions_lasso)^2) / sum((actuals - mean(actuals))^2))


predictions_ridge <- predict(best_ridge_model, s = "lambda.min", newx = x_test)
actuals <- testing_df$SalePrice
mse_ridge <- mean((predictions_ridge - actuals)^2)
rmse_ridge <- sqrt(mse_ridge)
mae_ridge <- mean(abs(predictions_ridge - actuals))
r_squared_ridge <- 1 - (sum((actuals - predictions_ridge)^2) / sum((actuals - mean(actuals))^2))


predictions_elastic <- predict(best_elastic_net_model, s = "lambda.min", newx = x_test)
actuals <- testing_df$SalePrice
mse_elastic <- mean((predictions_elastic - actuals)^2)
rmse_elastic  <- sqrt(mse_elastic)
mae_elastic  <- mean(abs(predictions_elastic - actuals))
r_squared_elastic <- 1 - (sum((actuals - predictions_elastic)^2) / sum((actuals - mean(actuals))^2))

predictions_linear <- predict(linear_model , newdata = testing_df)
actuals <- testing_df$SalePrice
mse_linear <- mean((predictions_linear - actuals)^2)
```

```
rmse_linear <- sqrt(mse_linear)
mae_linear <- mean(abs(predictions_linear - actuals))
r_squared_linear <- 1 - (sum((actuals - predictions_linear)^2) / sum((actuals - mean(actuals))^2))


data.frame(Type = c("Linear", "Lasso", "Ridge", "Elastic", "Optimal"), MSE = c(mse_linear, mse_lasso, ms
           rmse = c(rmse_linear, rmse_lasso, rmse_ridge, rmse_elastic, "Lasso"),
           mae = c(mae_linear, mae_lasso, mae_ridge, mae_elastic, "Lasso"),
           r_squared = c(r_squared_linear, r_squared_lasso, r_squared_ridge, r_squared_elastic, "Lasso/
           )
```

Interpratation of Error Metrics for Linear Models

```
##      Type              MSE             rmse              mae
## 1  Linear 1155961895.90542 33999.4396410503 20023.0700191348
## 2   Lasso 10066.4970775567 100.331934485271 72.8052181187232
## 3   Ridge 10068.2920322136 100.340879168032 72.8128969619892
## 4 Elastic 10067.3943968538 100.336406138818 72.8090575403393
## 5 Optimal            Lasso            Lasso            Lasso
##             r_squared
## 1   0.838262522787696
## 2   0.999998591536756
## 3   0.999998591285613
## 4   0.999998591411207
## 5 Lasso/Ridge/Elastic
```

The table above outputs the error metrics for each model for their predictions on the test data. From the table above we can see that Lasso Regression seems to be the most optimal model as it has the lowest MSE, MAE and RMSE. The lasso mdoel is follwoed by elastic net, then ridge and then linear models. However the r-squared value for Ridge, Lasso and elastic net are the same.

This suggests that Lasso regression is the most suitable model for predicting house prices, followed by Elastic net and then Ridge. These conclusions make sense from a statistical perspective due to the fact that since Lasso regression applies the strongest penalty to the coefficients, followed by elastic net and then Ridge regression. However we must not that difference in error metrics between the three regularization models is not large in absolute terms. Therefore eaach of them can be used as a predictive model for house prices.

Unsupervised Learning The last phase of my research methodology includes implementing forwards selection and backward selection techniques to our linear regression model. Forward Selection is where we start with a model with no variables and then add a predictor that signficant improves the model based on a selection criteria. In contrast backwards selection, is we start with the full model(all predictors) and then remove the variable, whose remove leads to the most improvement of the model based on a selection criteria. In our case we will be using AIC as the selection criteria.

```
library(MASS)
```

```
intiial_model <- lm(SalePrice ~ 1 ,data  = training_df)
full_model <- lm(SalePrice ~. ,data  = training_df)

fit_forward <- stepAIC(intiial_model, scope = list(lower = intiial_model, upper =full_model), direction
```

```
coefficients_forward <- as.data.frame(coefficients(fit_forward))
coefficients_forward_top_10_abs <- coefficients_forward %>% rename(Coefficients = `coefficients(fit_forw
coefficients_forward_top_10_abs
```

Forward Subset Selection

```
##                Coefficients       Variable
## (Intercept)     1859687.092    (Intercept)
## Street            17990.580         Street
## Overall.Qual      12382.318   Overall.Qual
## Exter.Qual       -10105.481     Exter.Qual
## Garage.Cars        8418.728    Garage.Cars
## Bsmt.Full.Bath     8308.269 Bsmt.Full.Bath
## Kitchen.AbvGr     -7872.993  Kitchen.AbvGr
## Bsmt.Qual         -4885.464      Bsmt.Qual
## Fireplaces         4502.981     Fireplaces
## Kitchen.Qual      -4393.602   Kitchen.Qual
```

```
intiial_model <- lm(SalePrice ~ 1 ,data  = training_df)
full_model <- lm(SalePrice ~. ,data  = training_df)

fit_backward <- stepAIC(full_model, direction. = "backward", trace = FALSE)
coefficients_backward<- as.data.frame(coefficients(fit_backward))
coefficients_backward_top_10_abs <- coefficients_backward %>% rename(Coefficients = `coefficients(fit_ba
coefficients_backward_top_10_abs
```

Backward Subset Selection

```
##                Coefficients       Variable
## (Intercept)     1859687.092    (Intercept)
## Street            17990.580    MS.SubClass
## Overall.Qual      12382.318   Kitchen.Qual
## Exter.Qual       -10105.481   Mas.Vnr.Type
## Garage.Cars        8418.728         Street
## Bsmt.Full.Bath     8308.269 Enclosed.Porch
## Kitchen.AbvGr     -7872.993      Bldg.Type
## Bsmt.Qual         -4885.464     Fireplaces
## Fireplaces         4502.981     Heating.QC
## Kitchen.Qual      -4393.602    House.Style
```

From the outputs above we can see that the ten predictor variables with the largest absolute coefficients for both forward and backwards selection. An important thing to note is that both the forward and backward selection techniques have the same top ten predictor variables with he largest absolute coefficients.

If we compare this to our linear models and regualzation models, we can see a similarity in the presence of Overall.Qual, Exter.Qual, Fireplaces, and Garage.Cars - this is another indiciation that the physical characterstics of the house are important determinent in their prices and esepically the quality of the house and the characteristics related to the fireplace and garage. However something that differs is the the top 10 variables also include the Street(the street which the house is on).This something that has not shown up in previous importance measures of the linear models or ensemble techniques and is something that does

match up with the idea from the research paper of "Determinants of housing values and variations in home prices across neighborhoods in Cook County" by by Maude Toussaint-Comeau and Jin Man Lee that the neigbourhood characterstics are an important determinent in housing prices.

Conclusion

Through a systemic process the research project has explored the most important determinants of housing prices through the implementation of the various machine learning models and then identifying which of those techniques would be the most optimal in acting as a predictive model in determine the price of houses.

By comparing supervised ensemble techniques such as Random Forest and Gradient Boosting and linear models, such lasso, ridge and elastic net regularization, the research project has created a robust statistical framework that has concluded that the physical characteristics of a house, such as overall quality, living area, garage features, and exterior aesthetics, are the most important in determining the price of a house. These findings alos aligned with establish economic theories and empirical econometric research. However something we foound was that the linear models and the ensemble models did differ in their analysis of how important temporal factors such as the year sold were on the prices of houses. If we were conduct the research project again it would important to fit anther subset of machine learning models to understand how important these temporal factors were.

While the research methodology we implemented did have significant strenghts with its focus on provided on a compehensive and robust statistical analysis with the comparison of errors metrics annd use of cross validation, there were some major weaknesses. Comparative analysis between the machine learning models and tradional economic literature(the three economic papers referenced throughout the paper) we saw that there were important determinents of house prices that the models and the data set we used missed. While both the economic literature and the machine learning models agreed on the importance of physical propery features, the economic literature also outlined the importance on neigbourhood characteristics, proximity to city infrastructure, macroeconomic variables and the country's particular population demographic characteristics in determining and prediciting housing prices. Incorporating such variables in the models and dataset coudl largely enhance the predictive accuaracy of the models and heighted the interpreatations there provide. This is due to the fact that by incorporating these variables we can better reflect and simualte the complexities of housing market dynamics.

Another weakness of the project is the limited scope of our anaylsis. This is not only due to that fact that we did not include importance variables such as neigbourhood characteristics, proximity to city infrastructure, macroeconomic variables and the country's particular population demographic characteristics in determining and prediciting housing prices, but our dataset only provided house prices for the city of Ames in the American state Iowa. Both the city and state are small in terms of population and therefore it would be difficult to apply the conclusions drawn to a wider scale. To improve the project next time I coudl possibly take hosuing dataset of multiple cities and then implement machine learning models to each of them and cary out a comporative analysis.

Another limitation of the model was that by using ensemble models there were computational constraints due to the fact that we could implement cross validation due to the computational complexing. This could have essentially limited the exploration of more complex models or a larger set of hyperparameters during cross-validation.

Another way I would want to improve this project is also to find another way to reduce the number of irrelevant variables, in order to simplify the analysis and possibly implement another subset of machine learning techniques in order to add to the comparative analysis of identifying the most important predictor variables. One way of doing this is by mplementing subset selection methods such as Stepwise selection or Subset Selection(both forward and backward), and using a criteria-based selection using AIC (Akaike Information Criterion), BIC (Bayesian Information Criterion), or Mallows' Cp. These methods can help enhance the analysis by systematically identifiying and retain variables the contribute most to the model's predictive power and then removing irrelvant ones. Another advantage of this is that it can help us identify a range of optimal model for linear regression methods. I would also try to implement a larger range od

regresson models, that significant differ from ensemble learning methods linear models, such as support vector machines, K Nearest Neighbours or Kernel Regression.

References

Jin Man Lee & Maude Toussaint-Comeau, 2018. "Determinants of Housing Values and Variations in Home Prices Across Neighborhoods in Cook County," Profitwise, Federal Reserve Bank of Chicago, issue 1, pages 1-23.

Geng, Nan. (2018). Fundamental Drivers of House Prices in Advanced Economies. IMF Working Papers. 18. 1. 10.5089/9781484367629.001.

Martijn Droes & Alex van de Minne, 2016. "Do the Determinants of House Prices Change over Time? Evidence from 200 Years of Transactions Data," ERES eres2016_227, European Real Estate Society (ERES).

Singh, Aishwarya. "A Comprehensive Guide to Ensemble Learning (with Python Codes)." Analytics Vidhya, 22 Nov. 2023, www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/.