Project 1: Kaggle Challenge

Christina Mourad, Victor Um, Joe De Leon, Martin Ha

2024-11-19

Project Overview

The goal of this project is to build a predictive model that can estimate house prices based on a variety of features. We were given the following files:

- **train.csv** - the training set
- **test.csv** - the test set
- **data_description.txt** - full description of each column, originally prepared by Dean De Cock but lightly edited to match the column names used here
- **sample_submission.csv** - a benchmark submission from a linear regression on year and month of sale, lot square footage, and number of bedrooms

---

Loading Dataset

```
library(dplyr)
library(ggplot2)
library(caret)
library(glmnet)
```

Libraies Utilized

House Prices Dataset train.csv

```
train_dataset <- read.csv("C:\\Users\\btmgc\\Desktop\\MATH444\\Projects\\Project 1\\StatisticalModeling-

head(train_dataset, n=2)
```

```
##   Id MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape LandContour
## 1  1         60       RL          65    8450   Pave  <NA>      Reg         Lvl
## 2  2         20       RL          80    9600   Pave  <NA>      Reg         Lvl
##   Utilities LotConfig LandSlope Neighborhood Condition1 Condition2 BldgType
## 1    AllPub    Inside       Gtl      CollgCr       Norm       Norm     1Fam
## 2    AllPub       FR2       Gtl      Veenker      Feedr       Norm     1Fam
##   HouseStyle OverallQual OverallCond YearBuilt YearRemodAdd RoofStyle RoofMatl
```

```
## 1     2Story           7           5      2003         2003     Gable  CompShg
## 2     1Story           6           8      1976         1976     Gable  CompShg
##   Exterior1st Exterior2nd MasVnrType MasVnrArea ExterQual ExterCond Foundation
## 1     VinylSd     VinylSd    BrkFace        196        Gd        TA      PConc
## 2     MetalSd     MetalSd       None          0        TA        TA     CBlock
##   BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinSF1 BsmtFinType2
## 1       Gd       TA           No          GLQ        706          Unf
## 2       Gd       TA           Gd          ALQ        978          Unf
##   BsmtFinSF2 BsmtUnfSF TotalBsmtSF Heating HeatingQC CentralAir Electrical
## 1          0       150         856    GasA        Ex          Y      SBrkr
## 2          0       284        1262    GasA        Ex          Y      SBrkr
##   X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath
## 1       856       854            0      1710            1            0        2
## 2      1262         0            0      1262            0            1        2
##   HalfBath BedroomAbvGr KitchenAbvGr KitchenQual TotRmsAbvGrd Functional
## 1        1            3            1          Gd            8        Typ
## 2        0            3            1          TA            6        Typ
##   Fireplaces FireplaceQu GarageType GarageYrBlt GarageFinish GarageCars
## 1          0        <NA>     Attchd        2003          RFn          2
## 2          1          TA     Attchd        1976          RFn          2
##   GarageArea GarageQual GarageCond PavedDrive WoodDeckSF OpenPorchSF
## 1        548         TA         TA          Y          0          61
## 2        460         TA         TA          Y        298           0
##   EnclosedPorch X3SsnPorch ScreenPorch PoolArea PoolQC Fence MiscFeature
## 1             0          0           0        0   <NA>  <NA>        <NA>
## 2             0          0           0        0   <NA>  <NA>        <NA>
##   MiscVal MoSold YrSold SaleType SaleCondition SalePrice
## 1       0      2   2008       WD        Normal    208500
## 2       0      5   2007       WD        Normal    181500
```

data_description

```r
data_description <- read.csv("C:\\Users\\btmgc\\Desktop\\MATH444\\Projects\\Project 1\\StatisticalModeli

head(data_description)
```

```
##   MSSubClass..Identifies.the.type.of.dwelling.involved.in.the.sale.
## 1                               20\t1-STORY 1946 & NEWER ALL STYLES
## 2                                          30\t1-STORY 1945 & OLDER
## 3                             40\t1-STORY W/FINISHED ATTIC ALL AGES
## 4                             45\t1-1/2 STORY - UNFINISHED ALL AGES
## 5                                 50\t1-1/2 STORY FINISHED ALL AGES
## 6                                          60\t2-STORY 1946 & NEWER
```

```r
cat("Full train dataset shape is", dim(train_dataset), "\n")
```

Dimensions of Dataset:

```
## Full train dataset shape is 1460 81
```

The House Prices dataset is composed of 81 columns and 1,460 entries.

Methods

To predict house prices, the following methods were applied:

1. **Data Cleaning**: Handling missing values, removing outliers, and converting categorical variables into factors.
2. **Exploratory Data Analysis**: Understanding the relationships between features and the target variable (sale price) through visualizations.
3. **Data Enrichment:** Transforming variables, creating new features, and selecting relevant predictors.
4. **Modeling**: Implementing multiple regression and advanced machine learning techniques such as LASSO, Ridge, and Gradient Boosting.
5. **Evaluation**: Using cross-validation and computing RMSE on the log-transformed sale price.

    1. **Data Cleaning**

Checking for Missing Values:

```r
missing_values <- colSums(is.na(train_dataset))
missing_values <- data.frame(Feature = names(missing_values), Missing = missing_values)
missing_values <- missing_values %>% filter(Missing > 0)

cat("Columns with missing values:\n")
```

```
## Columns with missing values:
```

```r
print(missing_values)
```

```
##                   Feature Missing
## LotFrontage   LotFrontage     259
## Alley               Alley    1369
## MasVnrType     MasVnrType       8
## MasVnrArea     MasVnrArea       8
## BsmtQual         BsmtQual      37
## BsmtCond         BsmtCond      37
## BsmtExposure BsmtExposure      38
## BsmtFinType1 BsmtFinType1      37
## BsmtFinType2 BsmtFinType2      38
## Electrical     Electrical       1
## FireplaceQu   FireplaceQu     690
## GarageType     GarageType      81
## GarageYrBlt   GarageYrBlt      81
## GarageFinish GarageFinish      81
## GarageQual     GarageQual      81
## GarageCond     GarageCond      81
## PoolQC             PoolQC    1453
## Fence               Fence    1179
## MiscFeature   MiscFeature    1406
```

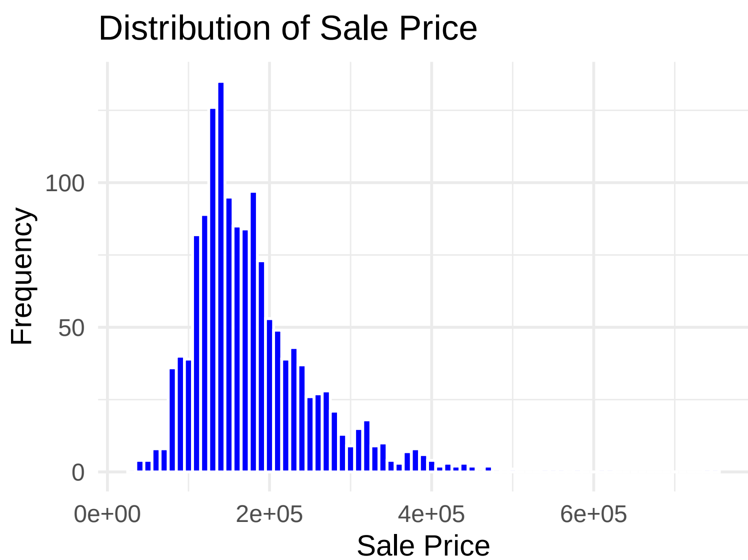
Fill missing values with median or mode based on variable type:

```
train_dataset <- train_dataset %>%
  mutate(across(where(is.numeric), ~ ifelse(is.na(.), median(., na.rm = TRUE), .))) %>%
  mutate(across(where(is.character), ~ ifelse(is.na(.), "None", .)))
```

2. **Exploratory Data Analysis**

Distribution of SalePrice:

```
ggplot(train_dataset, aes(x = SalePrice)) +
  geom_histogram(binwidth = 10000, fill = "blue", color = "white") +
  theme_minimal() +
  labs(title = "Distribution of Sale Price", x = "Sale Price", y = "Frequency")
```



Relationship between GrLivArea and SalePrice:

```
ggplot(train_dataset, aes(x = GrLivArea, y = SalePrice)) +
  geom_point(alpha = 0.5) +
  theme_minimal() +
  labs(title = "Sale Price vs. Ground Living Area", x = "Ground Living Area", y = "Sale Price")
```

3. **Data Enrichment**

Log-transforming SalePrice to normalize it:

```
train_dataset$LogSalePrice <- log(train_dataset$SalePrice)
```

Encoding categorical variables:

```
train_dataset <- train_dataset %>%
  mutate(across(where(is.character), as.factor))
```

Creating new features:

```
train_dataset$TotalSqFt <- train_dataset$GrLivArea + train_dataset$TotalBsmtSF
```

4. **Modeling**

Split data into training and validation sets:

```
set.seed(123)

train_index <- createDataPartition(train_dataset$LogSalePrice, p = 0.8, list = FALSE)
train_data <- train_dataset[train_index, ]
test_data <- train_dataset[-train_index, ]
```

Fit LASSO Regression:

```
x_train <- model.matrix(LogSalePrice ~ ., data = train_data)[, -1]
y_train <- train_data$LogSalePrice

lasso_model <- cv.glmnet(x_train, y_train, alpha = 1)
best_lambda <- lasso_model$lambda.min

cat("Optimal lambda for LASSO:", best_lambda, "\n")
```

```
## Optimal lambda for LASSO: 0.0006191379
```

5. **Evaluation**

Predict on test data:

```r
x_test <- model.matrix(LogSalePrice ~ ., data = test_data)[, -1]
predictions <- predict(lasso_model, s = best_lambda, newx = x_test)
```

Calculate RMSE:

```r
rmse <- sqrt(mean((predictions - test_data$LogSalePrice)^2))
cat("RMSE for LASSO model:", rmse, "\n")
```

```
## RMSE for LASSO model: 0.09044559
```

Analyzing the Results:

- The log-transformation of the sale price improved the model's performance by stabilizing variance.
- LASSO regression was effective in feature selection and regularization, reducing overfitting.
- The RMSE metric was used to evaluate model performance, ensuring a fair comparison with Kaggle benchmarks.

Conclusion

The House Proces dataset presented challenges such as missing values, mixed data types, and a large number of features (81 columns). Tackling this problem required a systematic approach, combining data cleaning, exploratory analysis, and advanced modeling techniques.

The primary objective was to create a model that could accurately estimate house prices while balancing predictive performance with interpretability. By leveraging techniques like LASSO regression, the project demonstrated how regularization can help handle datasets with many predictors by selecting only the most relevant features.

One of the main challenges was managing missing data for key variables such as `LotFrontage` and `GarageType`. Strategies such as imputing medians for numeric data and adding placeholders for categorical data ensured that the dataset was both complete and usable without introducing bias. Additionally, transforming the target variable (`SalePrice`) to its logarithmic scale addressed heteroscedasticity, a common issue in regression problems.

Through visualizations, relationships between house prices and features such as `GrLivArea` and `TotalSqFt` were identified, guiding feature engineering. These insights proved vital in creating a more predictive model. The final model achieved a Root Mean Square Error (RMSE) of **0.0904** on the log-transformed prices, indicating strong performance.