Predicting Book Sales: Lasso, Ridge, and Elastic Net Regression

Promothesh Chatterjee[1]

Introduction

This analysis aims to predict book sales using both text data from book reviews and numerical features. We will utilize three regularization techniques: Lasso, Ridge, and Elastic Net regression. These models help to handle high-dimensional data and prevent overfitting by adding a penalty to the model's complexity.

Load Libraries

First, we load the necessary libraries for our analysis.

```
library(tidyverse)
library(caret)
library(glmnet)
library(quanteda)
library(knitr)
library(kableExtra)
```

Load and Preprocess Data

Next, we load the dataset and preprocess it. Preprocessing involves converting text reviews into numerical features using TF-IDF (Term Frequency-Inverse Document Frequency).

```
# Load the dataset
data <- read.csv("Combined_Books_Reviews_Data.csv")

# Extract relevant columns
text_data <- data$Review
numerical_features <- data %>% select(Publishing.Year, Book_average_rating,
Book_ratings_count, gross.sales, sale.price, sales.rank)
target <- data$units.sold
```

Text Preprocessing with quanteda

Convert text reviews to TF-IDF features using the quanteda package.

```
# Create a corpus
corpus <- corpus(text_data)

# Create tokens and remove punctuation
tokens <- tokens(corpus, remove_punct = TRUE)

# Create a document-feature matrix
dfm <- dfm(tokens)

# Convert the dfm to TF-IDF
dfm_tfidf <- dfm_tfidf(dfm)

# Convert DFM to a dense matrix
text_features <- convert(dfm_tfidf, to = "matrix")
```

Combine Features

Combine the TF-IDF features with the numerical features to create a comprehensive feature set.

```
# Combine numerical features and text features
X <- cbind(numerical_features, text_features)
y <- target
```

Split Data

Split the data into training and testing sets to evaluate the models' performance on unseen data.

```
set.seed(42)
trainIndex <- createDataPartition(y, p = .8,
                                  list = FALSE,
                                  times = 1)
X_train <- X[ trainIndex,]
X_test <- X[-trainIndex,]
y_train <- y[ trainIndex]
y_test <- y[-trainIndex]
```

Model Training and Evaluation

Train and evaluate Lasso, Ridge, and Elastic Net regression models. These models are part of the generalized linear model family and include regularization techniques to prevent overfitting.

Lasso Regression

Lasso regression adds a penalty equal to the absolute value of the coefficients, which can shrink some coefficients to zero, thus performing feature selection.

```
# Define training control
train_control <- trainControl(method = "cv", number = 5)

# Train Lasso model
lasso_model <- train(X_train, y_train, method = "glmnet",
                     trControl = train_control,
                     tuneGrid = expand.grid(alpha = 1, lambda = 10^seq(2, -2,
by = -0.1)))
```

Ridge Regression

Ridge regression adds a penalty equal to the square of the coefficients, which shrinks coefficients but does not set any to zero.

```
# Train Ridge model
ridge_model <- train(X_train, y_train, method = "glmnet",
                     trControl = train_control,
```

```
                        tuneGrid = expand.grid(alpha = 0, lambda = 10^seq(2, -2,
by = -0.1)))
```

Elastic Net Regression

Elastic Net regression combines both Lasso and Ridge penalties, controlled by a mixing parameter alpha.

```
# Train Elastic Net model
elastic_net_model <- train(X_train, y_train, method = "glmnet",
                           trControl = train_control,
                           tuneGrid = expand.grid(alpha = 0.5, lambda =
10^seq(2, -2, by = -0.1)))
```

Model Evaluation

Evaluate the performance of the models using RMSE (Root Mean Squared Error), which measures the average magnitude of the prediction errors.

```
# Predict and calculate RMSE for Lasso
lasso_pred <- predict(lasso_model, X_test)
lasso_rmse <- sqrt(mean((y_test - lasso_pred)^2))

# Predict and calculate RMSE for Ridge
ridge_pred <- predict(ridge_model, X_test)
ridge_rmse <- sqrt(mean((y_test - ridge_pred)^2))

# Predict and calculate RMSE for Elastic Net
elastic_net_pred <- predict(elastic_net_model, X_test)
elastic_net_rmse <- sqrt(mean((y_test - elastic_net_pred)^2))

# Print RMSE values
lasso_rmse

## [1] 2685.774

ridge_rmse

## [1] 3474.473

elastic_net_rmse

## [1] 3063.544
```

Model Coefficients

Examine the top 5 significant coefficients of each model to understand the impact of different features on the predicted sales.

Lasso Coefficients

Lasso regression performs feature selection by shrinking some coefficients to zero. Let's examine the top 5 non-zero coefficients.

```
lasso_coef <- as.data.frame(as.matrix(coef(lasso_model$finalModel, s =
lasso_model$bestTune$lambda)))
lasso_coef <- lasso_coef[lasso_coef != 0, , drop = FALSE]
colnames(lasso_coef) <- "Coefficient"
lasso_coef <- rownames_to_column(lasso_coef, var = "Feature")
top5_lasso_coef <- lasso_coef %>% top_n(5, abs(Coefficient))
kable(top5_lasso_coef, caption = "Top 5 Lasso Regression Coefficients") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Top 5 Lasso Regression Coefficients

Feature

Coefficient

boy

-29926.98

soldier

28596.03

about

30930.15

aesopica

-35140.12

need

-28783.69

Ridge Coefficients

Ridge regression shrinks coefficients but does not set any to zero. We can examine the top 5 coefficients by absolute value.

```
ridge_coef <- as.data.frame(as.matrix(coef(ridge_model$finalModel, s =
ridge_model$bestTune$lambda)))
colnames(ridge_coef) <- "Coefficient"
ridge_coef <- rownames_to_column(ridge_coef, var = "Feature")
top5_ridge_coef <- ridge_coef %>% top_n(5, abs(Coefficient))
kable(top5_ridge_coef, caption = "Top 5 Ridge Regression Coefficients") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Top 5 Ridge Regression Coefficients

Feature

Coefficient

absolute

16448.85

anne

16418.21

dragonflight

14595.36

finale

14860.18

deception

15370.60

Elastic Net Coefficients

Elastic Net regression combines the properties of both Lasso and Ridge. We can examine the top 5 non-zero coefficients.

```
elastic_net_coef <-
as.data.frame(as.matrix(coef(elastic_net_model$finalModel, s =
elastic_net_model$bestTune$lambda)))
elastic_net_coef <- elastic_net_coef[elastic_net_coef != 0, , drop = FALSE]
colnames(elastic_net_coef) <- "Coefficient"
elastic_net_coef <- rownames_to_column(elastic_net_coef, var = "Feature")
top5_elastic_net_coef <- elastic_net_coef %>% top_n(5, abs(Coefficient))
kable(top5_elastic_net_coef, caption = "Top 5 Elastic Net Regression
Coefficients") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Top 5 Elastic Net Regression Coefficients

Feature

Coefficient

boy

-24883.17

about

24329.81

anne

23991.08

lola

21683.30

horse

22153.93

Results

The RMSE values for the models are as follows:

- **Lasso Regression**: 2685.7739103
- **Ridge Regression**: 3474.4730961
- **Elastic Net Regression**: 3063.5443595

These results indicate that the models have similar performance on this dataset.

Conclusion

In this analysis, we applied Lasso, Ridge, and Elastic Net regression models to predict book sales using both text data and numerical features. Each model showed comparable performance, demonstrating the potential of using text data in predictive modeling. Lasso regression was particularly useful for feature selection, while Ridge regression provided a more stable prediction by shrinking the coefficients. Elastic Net offered a balanced approach by combining the strengths of both Lasso and Ridge.

Examining the top 5 significant coefficients provided insights into the importance of different features. For instance, Lasso regression helped in identifying key features by setting less important ones to zero.