Multivariate Correlation with LASSO

LA

21/8/2025

Background

In this demo, the "Boston" data regarding housing prices in the Boston area will be used to make predictive models on the house price based on other parameters. The LASSO algorithm will be applied, setting it to two feature selection levels:

- Lambda_min (optimal)
- Lambda_1se (fewer features, slightly lower predictive power)

Data Preparation

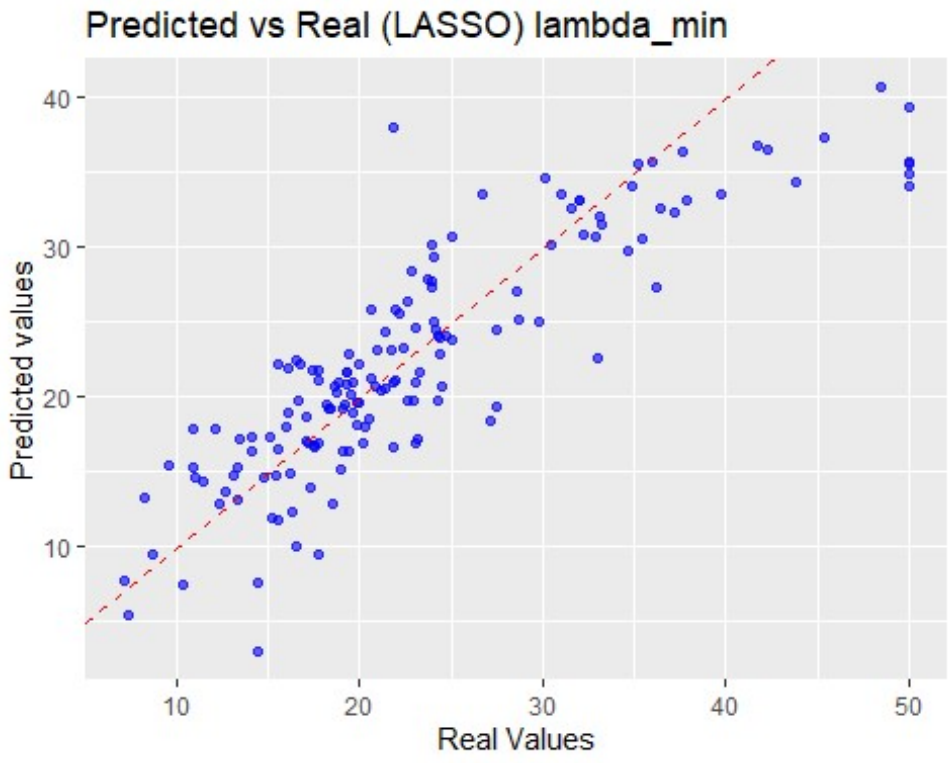
```
data("Boston")
X <- as.matrix(Boston %>% dplyr::select(-medv)) # all but medv
y <- Boston$medv
```

LASSO lambda.min

```
cv_lasso <- cv.glmnet(X_train, y_train, alpha = 1)

# Lambda ottimale
lambda_best <- cv_lasso$lambda.min
lambda_best

## [1] 0.02523631

lasso_model <- glmnet(X_train, y_train, alpha = 1, lambda = lambda_best)
coef_lasso <- coef(lasso_model)
print(coef_lasso)

## 14 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  36.975028468
## crim         -0.101315374
## zn            0.048815584
## indus        -0.058653338
## chas          4.029982442
## nox         -13.480303703
## rm            3.222061794
## age           .
## dis          -1.458836011
## rad           0.259822798
## tax          -0.008609311
## ptratio      -0.845119290
```

```
## black          0.006546153
## lstat         -0.584436311
```

```
y_pred <- predict(lasso_model, newx = X_test)

# Plot predicted vs real
df_plot <- data.frame(
  Reale = y_test,
  Predetto = as.numeric(y_pred)
)

ggplot(df_plot, aes(x = Reale, y = Predetto)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Predicted vs Real (LASSO) lambda_min",
       x = "Real Values",
       y = "Predicted values")
```

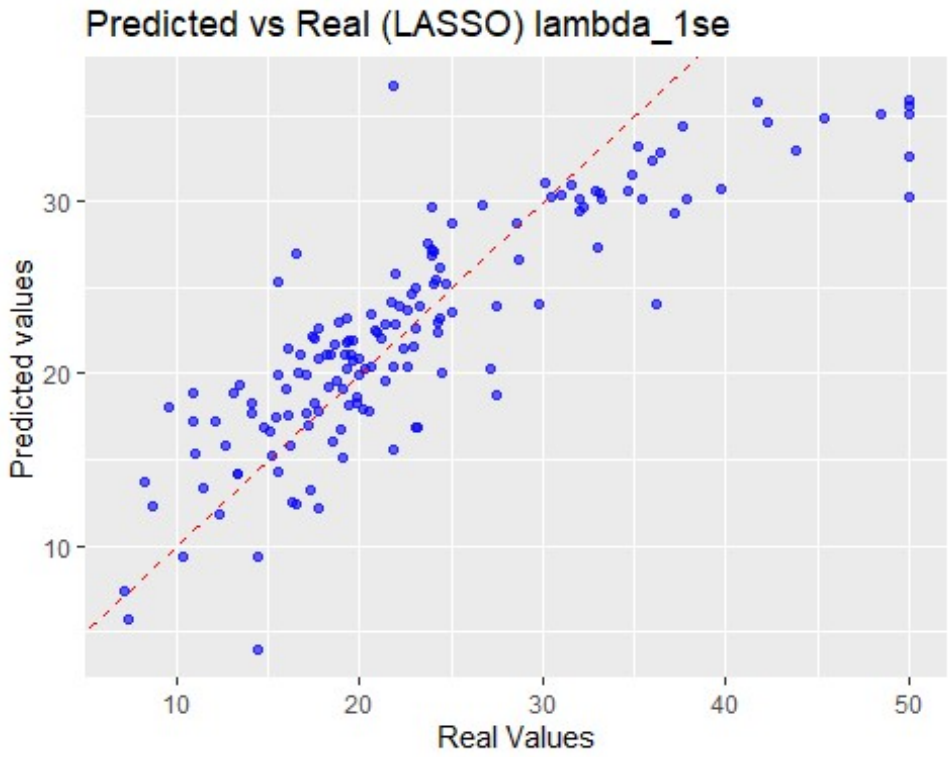



```
## Predction r squared

r_squared_min <- 1 - sum((y_test - y_pred)^2) / sum((y_test - mean(y_test))^2
)
r_squared_min
```

```
## [1] 0.7392826
```

LASSO lambda.1se

```
cv_lasso <- cv.glmnet(X_train, y_train, alpha = 1)

# Lambda
lambda_best <- cv_lasso$lambda.1se
lambda_best

## [1] 0.5967099

lasso_model <- glmnet(X_train, y_train, alpha = 1, lambda = lambda_best)
coef_lasso <- coef(lasso_model)
print(coef_lasso)

## 14 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept) 19.172404397
## crim        -0.007373580
## zn           .
## indus        .
## chas         2.846246144
## nox          .
## rm           3.506760096
## age          .
## dis         -0.040166370
## rad          .
## tax          .
## ptratio     -0.674431147
## black        0.001648574
## lstat       -0.554978330

y_pred <- predict(lasso_model, newx = X_test)

# Plot predizioni vs reali
df_plot <- data.frame(
  Reale = y_test,
  Predetto = as.numeric(y_pred)
)

ggplot(df_plot, aes(x = Reale, y = Predetto)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Predicted vs Real (LASSO) lambda_1se",
       x = "Real Values",
       y = "Predicted values")
```




```
## Predction r squared

r_squared_1se <- 1 - sum((y_test - y_pred)^2) / sum((y_test - mean(y_test))^2
)
r_squared_1se

## [1] 0.6908823
```

Conclusions

```
library(flextable)

## Warning: il pacchetto 'flextable' è stato creato con R versione 4.3.3

##
## Caricamento pacchetto: 'flextable'

## Il seguente oggetto è mascherato da 'package:purrr':
##
##     compose

final_df<-data.frame(rbind(c("r_squared_min", round(r_squared_min,2)),c("r_sq
uared_1s",round(r_squared_1se,2))))
colnames(final_df)<-c("Method","Prediction r_squared")
final_df_table<-flextable(final_df)

final_df_table <- align(final_df_table, align = "center", part = "header")
```

```
final_df_table <- autofit(final_df_table)
final_df_table
```

Method	Prediction r_squared
r_squared_min	0.74
r_squared_1s	0.69

- Lambda.min uses more features and achieves the best predictive performance.
- Lambda.1se uses fewer features with only slightly lower R^2, offering a simpler and more interpretable model.
- Predicted vs real plots highlight that both models struggle with very high-priced houses.
- The trade-off between number of variables and predictive power can guide business decisions on data collection costs.

Please note:

- The report contains code snippets because it is intended for educational purposes; in a non-demo version, they would be hidden.
- The ideal way to find a multivariable correlation would be to iteratively generate a series of training data frames and test data frames and then apply the resulting model to a completely new data frame. This is outside the scope of the demo.