Chapter 6: Multiple Linear Regression

Machine Learning for Business Analytics in R (2nd ed)

Shmueli, Bruce, Gedeck, Yahav & Patel

We assume a linear relationship between predictors and outcome:

outcome

coefficients

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon,$$

constant

predictors

error (noise)

Topics

- Explanatory vs. predictive modeling with regression
- Example: prices of Toyota Corollas
- Fitting a predictive model
- Assessing predictive accuracy
- Selecting a subset of predictors

Explanatory Modeling

Goal: Explain relationship between predictors (explanatory variables) and target

- Familiar use of regression in data analysis

- Model Goal: Fit the data well and understand the contribution of explanatory variables to the model

- Metrics: "goodness-of-fit" - R^2, residual analysis, p-values

Predictive Modeling

Goal: predict target values in other data where we have predictor values, but not target values

- Classic data mining context
- Model Goal: Optimize predictive accuracy
- Train model on training data
- Assess performance on validation (hold-out) data
- Explaining role of predictors is not primary purpose (but useful)

Example: Prices of Toyota Corolla

ToyotaCorolla.csv

Goal: predict prices of used Toyota Corollas based on their specification

Data: Prices of 1000 used Toyota Corollas, with their specification information

Variables Used

Price in Euros

Age in months as of 8/04

KM (kilometers)

Fuel Type (diesel, petrol, CNG)

HP (horsepower)

Metallic color (1=yes, 0=no)

Automatic transmission (1=yes, 0=no)

CC (cylinder volume)

Doors

Quarterly_Tax (road tax)

Weight (in kg)

Data Sample

(showing only the variables to be used in analysis)

Price	Age	KM	Fuel_Type	HP	Metallic	Automatic	cc	Doors	Quarterly_Tax	Weight
13500	23	46986	Diesel	90	1	0	2000	3	210	1165
13750	23	72937	Diesel	90	1	0	2000	3	210	1165
13950	24	41711	Diesel	90	1	0	2000	3	210	1165
14950	26	48000	Diesel	90	0	0	2000	3	210	1165
13750	30	38500	Diesel	90	0	0	2000	3	210	1170
12950	32	61000	Diesel	90	0	0	2000	3	210	1170
16900	27	94612	Diesel	90	1	0	2000	3	210	1245
18600	30	75889	Diesel	90	1	0	2000	3	210	1245
21500	27	19700	Petrol	192	0	0	1800	3	100	1185
12950	23	71138	Diesel	69	0	0	1900	3	185	1105
20950	25	31461	Petrol	192	0	0	1800	3	100	1185

Preprocessing

Fuel type is categorical (in R - a `factor` variable), must be transformed into binary variables. R's `lm` function does this automatically.

Diesel (1=yes, 0=no)

Petrol (1=yes, 0=no)

None needed* for "CNG" (if diesel and petrol are both 0, the car must be CNG)

*You <u>cannot</u> include all the binary dummies; in regression this will cause a multicollinearity error. Other machine learning methods <u>can</u> use all the dummies.

Fitting a Regression Model to the Toyota Data

```
library(caret)
car.df <- mlba::ToyotaCorolla
# select variables for regression
outcome <- "Price"
predictors <- c("Age_08_04", "KM", "Fuel_Type", "HP", "Met_Color",
     "Automatic", "CC", "Doors", "Quarterly_Tax", "Weight")
# reduce data set to first 1000 rows and selected variables
car.df <- car.df[1:1000, c(outcome, predictors)]

# partition data
set.seed(1) # set seed for reproducing the partition
idx <- createDataPartition(car.df$Price, p=0.6, list=FALSE)
train.df <- car.df[idx, ]
holdout.df <- car.df[-idx, ]

# use lm() to run a linear regression of Price on all 11 predictors in the
# training set.
# use . after ~ to include remaining columns in train.df as predictors.
car.lm <- lm(Price ~ ., data = train.df)
# use options() to ensure numbers are not displayed in scientific notation.
options(scipen = 999)
summary(car.lm)
```

put 60% in training

Output of the Regression Model

```
> summary(car.lm)

Call:
lm(formula = Price ~ ., data = train.df)

Residuals:
Min     1Q    Median 3Q   Max
-9047 -831    -6    832   6057
Coefficients:
            Estimate    Std. Error      t value    Pr(>|t|)
(Intercept)   -3725.59270 1913.92374     -1.95      0.05206 .
Age_08_04      -133.98649    4.92047    -27.23    < 0.0000000000000002 ***
KM               -0.01741    0.00231     -7.53      0.00000000000019238 ***
Fuel_TypeDiesel 1179.18603 724.71141      1.63      0.10425
Fuel_TypePetrol 2173.64897 729.55378      2.98      0.00301 **
HP                36.34253   4.75838      7.64      0.00000000000008997 ***
Met_Color         -7.60255 119.54320     -0.06      0.94931
Automatic        276.55860 267.85985      1.03      0.30227
CC                 0.01517   0.09440      0.16      0.87236
Doors              2.28016  62.30556      0.04      0.97082
Quarterly_Tax      9.64453   2.60048      3.71      0.00023 ***
Weight            15.25566   1.81726      8.39      0.0000000000000035 ***
```

"P-value," a measure of the chances that a random shuffling could produce a coefficient as big as observed (low p-values mean "statistical significance")

Accuracy Metrics for the Regression Model

```
Residual standard error: 1340 on 589 degrees of freedom
Multiple R-squared: 0.869, Adjusted R-squared: 0.867
F-statistic: 356 on 11 and 589 DF,
     p-value: <0.0000000000000002
```

These are traditional metrics, i.e. measured on the training data

Specialized Metrics Used in Regression (lower values are better)

Akaike Information Criterion (AIC)

AIC = $n \ln(SSE/n) + n(1 + \ln(2\pi)) + 2(p + 1)$

Bayesian Information Criterion (BIC)

BIC = $n \ln(SSE/n) + n(1 + \ln(2\pi)) + \ln(n)(p + 1)$

Mallow's Cp

Cp = $SSE/\sigma^2_{full} + 2(p+1) - n$

σ^2_{full} is the estimated MSE for the full model

Mallow's Cp is equivalent to AIC for large samples

Make the Predictions for the Holdout Data (and show some residuals)

```
# use predict() to make predictions on a new set.
pred <- predict(car.lm, holdout.df)

options(scipen=999, digits=0)
data.frame(
   'Predicted' = pred[1:20],
   'Actual' = holdout.df$Price[1:20],
   'Residual' = holdout.df$Price[1:20] - pred[1:20]
)
options(scipen=999, digits = 3)
```

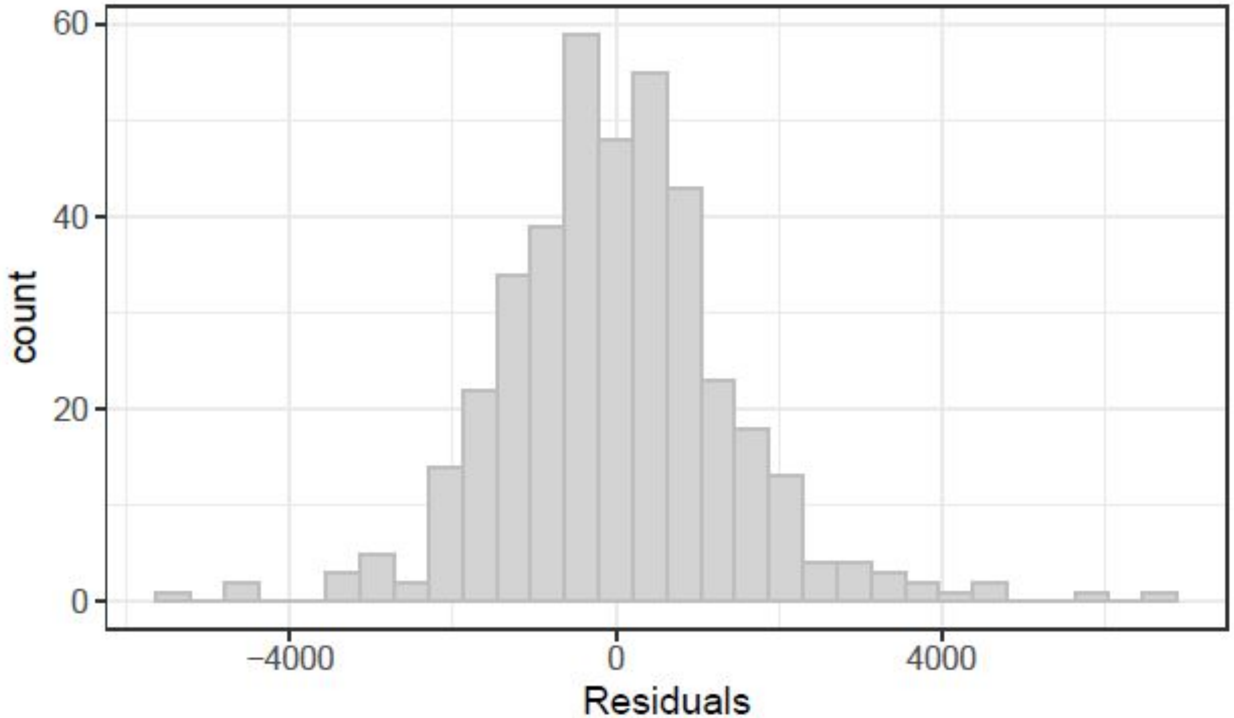
```
     Predicted Actual Residual
1     16652    13500   -3152
14    19941    21500    1559
15    19613    22500    2887
16    20424    22000    1576
18    16553    17950    1397
19    15247    16750    1503
20    15006    16950    1944
21    14949    15950    1001
```

How Well did the Model Do With the Holdout Data?

```
# calculate performance metrics
rbind(
Training=mlba::regressionSummary(pred
ict(car.lm, train.df),
train.df$Price),
Holdout=mlba::regressionSummary(pred,
holdout.df$Price)
)
```

```
          RMSE MAE
Training  1329 1009
Holdout   1423 1054
```

Distribution of Residuals (Holdout Set)



Symmetric distribution

A few outliers

```
library(ggplot2)
pred <- predict(car.lm, holdout.df)
all.residuals <- holdout.df$Price - pred

ggplot() +
     geom_histogram(aes(x=all.residuals), fill="lightgray", color="grey") +
     labs(x="Residuals", Y="Frequency")
```

Feature (Variable, Predictor) Selection

- Why select a subset of attributes to predict the target?
- More predictors/attributes problems:
  - Expensive data collection
  - More missing data
  - Multicollinearity – some predictors behave the same way
  - Uncorrelation with target variable
- The goal
  - Find parsimonious model (simplest model that performs sufficiently well)
  - More robust & higher predictive accuracy
- Variable selection methods
  - Exhaustive search
  - Partial Subset selection: Forward
  - Partial Subset selection: Backward
  - Partial Subset selection: Stepwise

Selecting Subsets of Predictors

Goal: Find parsimonious model (the simplest model that performs sufficiently well)

- More robust
- Higher predictive accuracy

Exhaustive Search

Partial Search Algorithms

- Forward
- Backward
- Stepwise

Exhaustive Search = Best Subset

- All possible subsets of predictors assessed (single, pairs, triplets, etc.)
- Computationally intensive, not feasible for big data
- Judge by "adjusted R^2"

$$R^2_{adj} = 1 - \frac{n-1}{n-p-1}(1-R^2)$$

Penalty for number of predictors

Exhaustive search requires library `leaps` and manual coding into binary dummies

```
# use regsubsets() in package leaps to run an exhaustive search.

library(leaps)
library(fastDummies)

# create dummies for fuel type
leaps.train.df <- dummy_cols(train.df, remove_first_dummy=TRUE,
     remove_selected_columns=TRUE)
search <- regsubsets(Price ~ ., data=leaps.train.df, nbest=1,
     nvmax=ncol(leaps.train.df), method="exhaustive")
sum <- summary(search)

# show models
sum$which

# show metrics
sum$rsq
sum$adjr2
sum$cp
```

Exhaustive output shows best model for each number of predictors

```
sum$which
 (Intercept) Age_08_04 KM    HP Met_Color Auto   CC    Doors Q_Tax Weight  Diesel  Petrol
1    TRUE     TRUE     FALSE FALSE FALSE   FALSE FALSE FALSE FALSE  FALSE  FALSE   FALSE
2    TRUE     TRUE     FALSE TRUE  FALSE   FALSE FALSE FALSE FALSE  FALSE  FALSE   FALSE
3    TRUE     TRUE     FALSE TRUE  FALSE   FALSE FALSE FALSE FALSE  TRUE   FALSE   FALSE
4    TRUE     TRUE     TRUE  TRUE  FALSE   FALSE FALSE FALSE FALSE  TRUE   FALSE   FALSE
5    TRUE     TRUE     TRUE  TRUE  FALSE   FALSE FALSE FALSE TRUE   TRUE   FALSE   FALSE
6    TRUE     TRUE     TRUE  TRUE  FALSE   FALSE FALSE FALSE TRUE   TRUE   FALSE   TRUE
7    TRUE     TRUE     TRUE  TRUE  FALSE   FALSE FALSE FALSE TRUE   TRUE   TRUE    TRUE
8    TRUE     TRUE     TRUE  TRUE  FALSE   TRUE  FALSE FALSE TRUE   TRUE   TRUE    TRUE
9    TRUE     TRUE     TRUE  TRUE  FALSE   TRUE  TRUE  FALSE TRUE   TRUE   TRUE    TRUE
10   TRUE     TRUE     TRUE  TRUE  TRUE    TRUE  TRUE  FALSE TRUE   TRUE   TRUE    TRUE
11   TRUE     TRUE     TRUE  TRUE  TRUE    TRUE  TRUE  TRUE  TRUE   TRUE   TRUE    TRUE
```

Each row is the best model for a given # of predictors,
“TRUE” and “FALSE” show whether the variable is included

Adjusted R^2 and CP for the models with 1 predictor, 2 predictors, 3 predictors, etc. (exhaustive search method)

```
> sum$adjr2
[1] 0.773 0.815 0.847 0.864 0.865 0.867 0.867 0.867 0.867
  0.867 0.867
> sum$cp
[1] 422.90 234.33 92.94 17.09 14.05 5.73 5.20 6.03 8.01
  10.00 12.00
```

Metrics improve until you hit 6-7 predictors, then stabilize, so choose model with 6-7 predictors

Exhaustive search may be computationally infeasible - some alternatives:

FORWARD SELECTION

- Start with no predictors
- Add them one by one (add the one with largest contribution)
- Stop when the addition is not statistically significant

BACKWARD ELIMINATION

- Start with all predictors
- Successively eliminate least useful predictors one by one
- Stop when all remaining predictors have statistically significant contribution

STEPWISE

- Like Forward Selection
- Except at each step, also consider dropping non-significant predictors

Regularization (shrinkage)

- Alternative to subset selection
- Rather than binary decisions on including variables, penalize coefficient magnitudes
- This has the effect of “shrinking” coefficients, and also reducing variance
- Predictors with coefficients that shrink to zero are effectively dropped
- Variance reduction improves prediction performance

Shrinkage - Ridge Regression

- OLR minimizes sum of squared errors (residuals) - SSE
- Ridge regression minimizes SSE subject to penalty being below specified threshold
- Penalty, called **L2, is *sum of squared coefficients***
- λ parameter controls degree of regularization
  (Use cross-validation to set)
- Predictors are typically standardized

Goal - minimize: $$SSE + \lambda \sum_{j=1}^{p} \beta_j^2$$

Shrinkage - Lasso

- OLR minimizes sum of squared errors (residuals) - SSE
- Ridge regression minimizes SSE + penalty
- Penalty, called **L1, is *sum of absolute values for coefficients***
- λ parameter controls degree of regularization
  (Use cross-validation to set)
- Predictors are typically standardized

Goal - minimize: $$SSE + \lambda \sum_{j=1}^{p} |\beta_j|$$

Ridge Regression Using Caret

```
library(caret)
trControl <- caret::trainControl(method='cv', number=5,
allowParallel=TRUE)
tuneGrid <- expand.grid(lambda=10^seq(5, 2, by=-0.1), alpha=0)
model <- caret::train(Price ~ ., data=train.df,
     method='glmnet',
     family='gaussian', # set the family for linear regression
     trControl=trControl,
     tuneGrid=tuneGrid)
model$bestTune
coef(model$finalModel, s=model$bestTune$lambda)
```

Lasso Regression Using Caret

```
tuneGrid <- expand.grid(lambda=10^seq(4, 0, by=-0.1), alpha=1)
model <- caret::train(Price ~ ., data=train.df,
     method='glmnet',
     family='gaussian', # set the family for linear regression
     trControl=trControl,
     tuneGrid=tuneGrid)
model$bestTune
coef(model$finalModel, s=model$bestTune$lambda)
```

When you run both the Ridge and Lasso models, you will see that the coefficients for key predictors are smaller than the equivalent ones in the basic model that was developed initially.

Summary

- Linear regression models are very popular tools, not only for explanatory modeling, but also for prediction
- A good predictive model has high predictive accuracy (to a useful practical level)
- Predictive models are fit to training data, and predictive accuracy is evaluated on a separate validation data set
- Removing redundant predictors is key to achieving predictive accuracy and robustness
- Subset selection and regularization (shrinkage) methods help find "good" candidate models.