KEY MACHINE LEARNING CONCEPTS

Explained with Linear Regression

https://econ.lange-analytics.com/aibook/

https://econ.lange-analytics.com/aibook/

WHAT WILL YOU LEARN

- Reviewing the basic idea behind linear regression
- Learning how how to measure predictive quality with Mean Square Error (MSE).
- Understanding the role of parameters in a machine learning model in general and in linear regression in particular
- Calculating optimal regression parameters using OLS
- Finding optimal regression parameters by trial and error
- Distinguish between unfitted and fitted models
- Using the `tidymodels` package to split observations from a dataset randomly into a training and testing dataset.
- Understanding how categorical data such as the sex of a person (female/male) can be transformed into numerical dummy variable.

- Being able to distinguish between dummy encoding and one-hot encoding
- Using `tidymodels` including model design and data pre-processing (recipes) to analyze housing prices.

UNIVARIATE LINEAR REGRESSION - DATA TABLE AND GOAL

The Regression:

$$\hat{y}_i = \beta_1 x_i + \beta_2$$

The Goal

Find values for β_1 and β_2 that minimize the prediction errors $(\hat{y}_i - y_i)^2$

The Data Table

Mockup Training Dataset

	y	x
i	Grade	StudyTime
1	65	2
2	82	3
3	93	7
4	93	8
5	83	4

UNIVARIATE LINEAR REGRESSION - DATA DIAGRAM AND GOAL

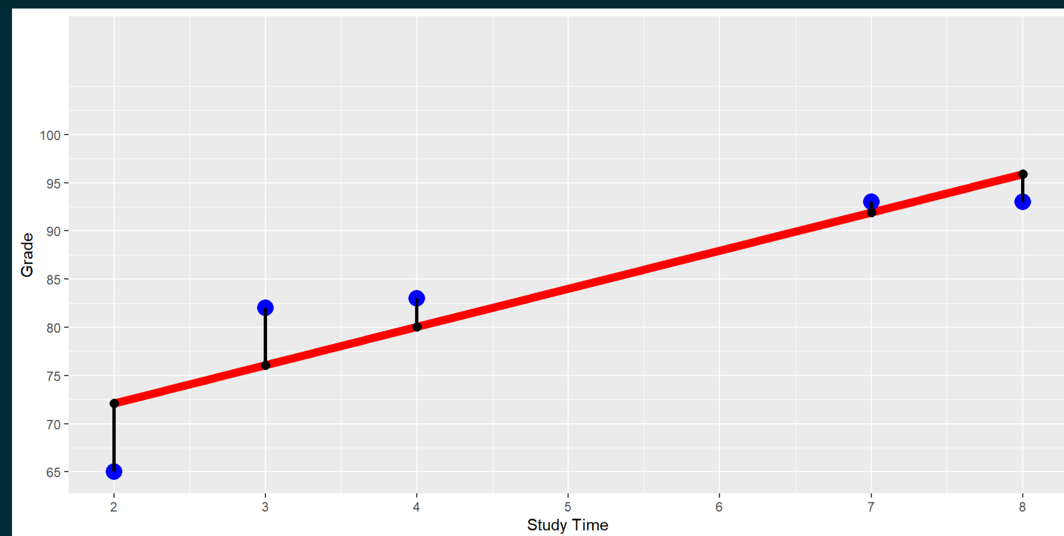
The Regression:

$$\hat{y}_i = \beta_1 x_i + \beta_2$$

The Goal

Find values for β_0 and β_1 that minimize the prediction errors $(\hat{y}_i - y_i)^2$

The Data Diagram



HOW TO MEASURE PREDICTION QUALITY

$$
\begin{aligned}
MSE &= \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \\
&\Longleftrightarrow \\
MSE &= \frac{1}{N}\sum_{i=1}^{N}(\underbrace{\overbrace{\beta_1 x_i + \beta_2}^{\text{Prediction } i} - y_i}_{\text{Error } i})^2
\end{aligned}
$$

https://econ.lange-analytics.com/aibook/

CUSTOM R FUNCTION TO CALCULATE MSE

Function Call:

▶ Code

```
[1] 29.8
```

Function Definition:»

▶ Code

HOW TO FIND OPTIMAL VALUES FOR β_1 AND β_2

Method 1:

Calculate optimal values for the parameters (the βs) based on Ordinary Least Squares (OLS) using two formulas (**Note,** this method works only for linear regression)

Method 2:

We can use a **systematic trial and error process**.

METHOD 1: CALCULATE OPTIMAL PARAMETERS (ONLY FOR OLS!)

$$\beta_{1,opt} = \frac{N\sum_{i=1}^{N} y_i x_i - \sum_{i=1}^{N} y_i \sum_{i=1}^{N} x_i}{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2} = 3.96$$

$$\beta_{2,opt.} = \frac{\sum_{i=1}^{N} y_i - \beta_1 \sum_{i=1}^{N} x_i}{N} = 64.18$$

Mockup Training Dataset

	y	x	y x	x x
i	Grade	StudyTime	GradeXStudyTime	StudyTimeSquared
1	65	2	130	4
2	82	3	246	9
3	93	7	651	49
4	93	8	744	64
5	83	4	332	16

METHOD 2: USE A SYSTEMATIC TRIAL AND ERROR PROCESS 🤓

- **Grid Search (aka Brute Force):**
  1. For a given range of β_1 and β_2 values, build a table with pairs of all combinations of these βs.
  2. Then use our custom `FctMSE()` command to calculate a MSE for each β pair.
  3. Find the β pair with the lowest MSE
- **Optimizer:** Use the R build-in optimizer. Push the start values for β_1 and β_2 together with the data to the optimizer as arguments. The rest is done by the optimizer.
- See the R script in the footnote to see both algorithms in action.»

UNIVARIATE OLS WITH A REAL WORLD DATASET

UNIVARIATE OLS WITH A REAL WORLD DATASET

Data

▶ Code

- King County House Sale dataset (Kaggle 2015). House sales prices from May 2014 to May 2015 for King County in Washington State.
- Several predictor variables. For now we use only $Sqft$
- We will only use 500 randomly chosen observations from the total of 21,613 observations.

Unfitted Model:»

$$\widehat{Price} = \beta_1 Sqft + \beta_2$$

UNIVARIATE OLS WITH A REAL WORLD DATASET

Splitting in Training and Testing Datasets

▶ Code

DataTrain

```
# A tibble: 349 × 2
    Price  Sqft
    <int> <int>
 1 265000  1400
 2 157340   900
 3 280000  1700
 4 207000  1980
 5 226500  1560
 6 295000  1230
 7 260000  2390
 8 175000   670
 9 270000  1010
10 280000  1590
# … with 339 more rows
```

DataTest

```
# A tibble: 151 × 2
    Price  Sqft
```

```
     <int> <int>
 1 735000  2340
 2 475226  2120
 3 242025  1400
 4 510000  2170
 5 325000  1730
 6 560000  2790
 7 733500  2120
 8 785000  1670
 9 580000  2430
10 872500  2040
# … with 141 more rows
```

UNIVARIATE OLS WITH A REAL WORLD DATASET

Run the Analysis»

https://lange-analytics.com/AIBook/Exercises/handler.html?file=05-LinRegrExerc100.Rmd

MULTIVARIATE ANALYSIS

MULTIVARIATE ANALYSIS – THREE PREDICTOR VARIABLES

`Sqft`: Living square footage of the house.

`Grade` Indicates the condition of houses (1 (worst) to 13 (best).

`Waterfront`: Is house located at the waterfront (`yes` or `no`).

▶ Code

Unfitted Model: »

$$Price = \beta_1 Sqft + \beta_2 Grade + \beta_3 Waterfront\_yes + \beta_4$$

MULTIVARIATE REAL WORLD DATASET – SPLITTING

▶ Code

DataTrain

```
# A tibble: 15,128 × 4
    Price  Sqft Grade Waterfront
    <dbl> <int> <int> <chr>
 1 221900  1180     7 no
 2 180000   770     6 no
 3 189000  1200     7 no
 4 230000  1250     7 no
 5 252700  1070     7 no
 6 240000  1220     7 no
 7 228000  1190     7 no
 8 287000  2240     7 no
 9 204000  1000     7 no
10 215000  1610     7 no
# … with 15,118 more rows
```

DataTest

```
# A tibble: 6,485 × 4
     Price  Sqft Grade Waterfront
     <dbl> <int> <int> <chr>
 1 1230000  5420    11 no
```

```
 2  257500  1715      7 no
 3  291850  1060      7 no
 4  229500  1780      7 no
 5  530000  1810      7 no
 6  650000  2950      9 no
 7  285000  2270      8 no
 8  233000  1710      6 no
 9  667000  1400      8 no
10  719000  2570      8 no
# … with 6,475 more rows
```

DUMMY AND ONE-HOT ENCODING

One-Hot Encoding

```
# A tibble: 4 × 2
  Waterfront_yes Waterfront_no
           <dbl>         <dbl>
1              0             1
2              0             1
3              1             0
4              0             1
```

One-hot encoding is easier to interpret but causes problems in OLS (dummy trap) because one variable is redundant. We can calculate one variable from the other (*perfect multicollinearity*):

$$Waterfront_{yes} = 1 - Waterfront_{no}$$

»

DUMMY AND ONE-HOT ENCODING

Dummy Coding

We use one variable less than we have categories. Waterfront has two categories. Therefore, we use one variable (e.g., `Waterfront_yes`):

Dummy Encoding Example

```
# A tibble: 4 × 1
  Waterfront_yes
           <dbl>
1              0
2              0
3              1
4              0
```

Note, dummy encoding can be done with `step_dummy()` in a *tidymodels recipe.*»

MULTIVARIATE ANALYSIS – BUILDING THE RECIPE

▶ Code

Here is how the recipe later on (in the workflow) transforms the data:

```
# A tibble: 15,128 × 4
    Sqft Grade  Price Waterfront_yes
   <int> <int>  <dbl>          <dbl>
 1  1180     7 221900              0
 2   770     6 180000              0
 3  1200     7 189000              0
 4  1250     7 230000              0
 5  1070     7 252700              0
 6  1220     7 240000              0
 7  1190     7 228000              0
 8  2240     7 287000              0
 9  1000     7 204000              0
10  1610     7 215000              0
# … with 15,118 more rows
```

MULTIVARIATE ANALYSIS – BUILDING THE MODEL DESIGN

Unfitted Model:

▶ Code

```
Linear Regression Model Specification (regression)

Computational engine: lm
```

»

MULTIVARIATE ANALYSIS – CREATING WORKFLOW & FITTING TO THE TRAINING DATA

▶ Code

```
# A tibble: 4 × 5
  term            estimate std.error statistic   p.value
  <chr>              <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)    -570056.  15133.        -37.7 6.63e-297
2 Sqft               180.      3.25       55.2 0
3 Grade            95214.   2548.         37.4 1.65e-292
4 Waterfront_yes  868338.  22200.         39.1 7.12e-319
```

▶ Code

```
# A tibble: 1 × 12
  r.squared adj.r.squared   sigma statistic p.value    df   logLik    AIC    BIC
      <dbl>         <dbl>   <dbl>     <dbl>   <dbl> <dbl>    <dbl>  <dbl>  <dbl>
1     0.581         0.581 238574.     7002.       0     3 -208785. 4.18e5 4.18e5
# … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

MULTIVARIATE ANALYSIS – PREDICTING TESTING DATA AND METRICS

▶ Code

```
# A tibble: 3 × 3
  .metric .estimator  .estimate
  <chr>   <chr>           <dbl>
1 rmse    standard   244656.
2 rsq     standard        0.549
3 mae     standard   163358.
```