Machine Learning

Multiple Linear Regression

ALBERTSCHOOL

MINES PARIS | PSL

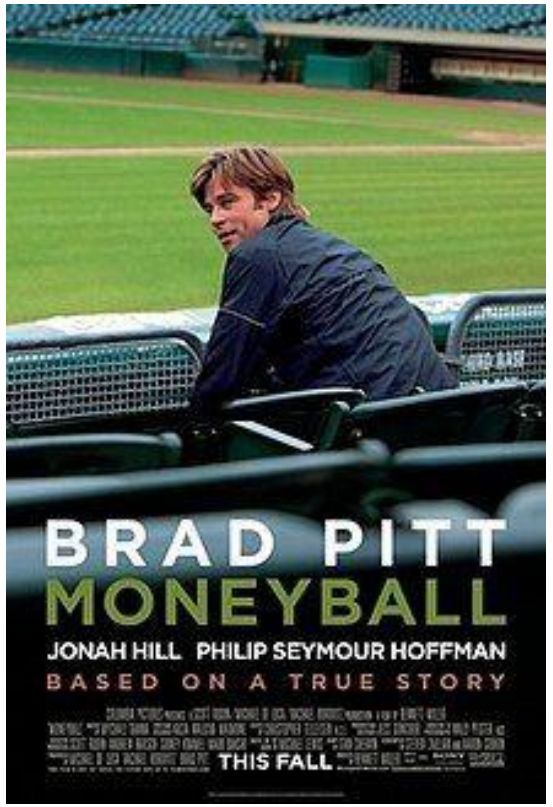
Thomas Milcent - 2025

Machine Learning - Multiple Linear Regression

An example - Money Ball Oakland A's
In order to find the best possible team in their budget, they needed to "underdogs" (undervalues players).

Using statistics ("sabermetrics"), they managed to gather those undervalues talents.

Example:
"Predictions *[like Expected Salary/Next season performance]* can be made using a logistic regression model with explanatory variables including: opponents' runs scored, runs scored, shutouts time at bat, winning rate, and pitcher whip."
Using several features, your model is better, and you can win!



Reference: https://en.wikipedia.org/wiki/Moneyball_(film)

Machine Learning - Multiple Linear Regression

What is Multiple Linear Regression?
A statistical method that allow us to summarize and study relationship between two continuous (quantitative) variables:

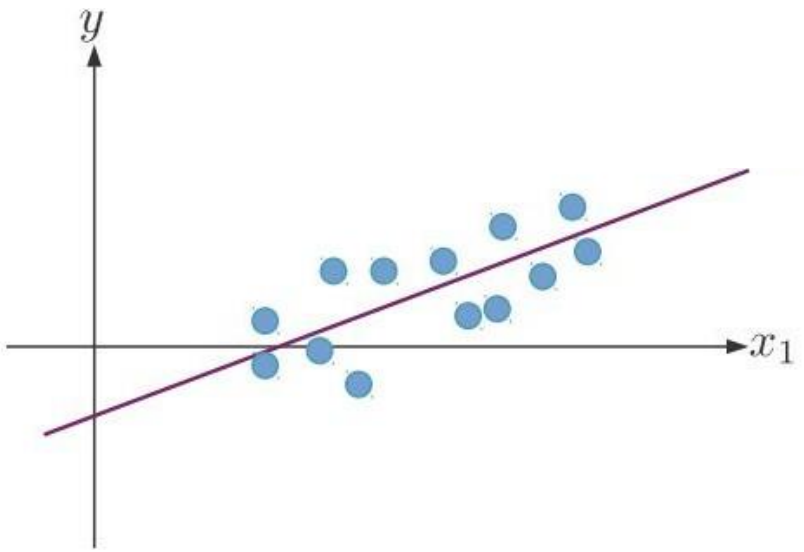
- **x**, is regarded as the **predictor**, **explanatory**, or **independent** variable
- **y** is regarded as the **response**, **outcome** or **dependent** variable

Simple because there is only one predictor (number of features p equal to 1)
In contrast, we study **Multiple** linear regression (two or more predictor variables) to build a better model taking into account more variables!

Machine Learning - Multiple Linear Regression

Definition

$$\boldsymbol{x} \in \mathbb{R}^p, y \in \mathbb{R}$$



$$f(\mathbf{x}) = \sum_{j=1}^{p} \beta_j x_j + \beta_0$$

$$D = \{x^i, y^i\}_{i=1,\ldots,n}$$

$$e_i = y_i - \hat{y}_i$$

Machine Learning - Multiple Linear Regression

Correlated variables

If the variables are **uncorrelated**

- Each coefficient can be estimated separately
- **Interpretation** is easy
  "A change of 1 in xi is associated with a change of β_j in y, while everything else stays the same."

Correlations between variables cause problems:

- The variance of all coefficients tend to increase.
- Interpretation is much harder
  When xj changes, so does everything else
- (Extreme Example) $x_1 = 2x_2$ then $y = 10x_1 + 2 \Leftrightarrow y = -3x_1 + 26x_2 + 2$
  What equation do you prefer?

Machine Learning - Multiple Linear Regression

Correlated variables

If the variables are **uncorrelated**

- Each coefficient can be estimated separately
- **Interpretation** is easy
  "A change of 1 in xi is associated with a change of βj in y, while everything else stays the same."

Correlations between variables cause problems:

- The variance of all coefficients tend to increase.
- Interpretation is much harder
  When xj changes, so does everything else
- (Extreme Example) $x_1 = 2x_2$ then $y = 10x_1 + 2 \Leftrightarrow y = -3x_1 + 26x_2 + 2$
  What equation do you prefer?
  - The first one: less features, better generalization! Simpler!

- Ccl: multicolinearity can have no negative impact on the predictions, but it will make interpretation harder and coefficient estimates more unstable

Machine Learning - Multiple Linear Regression

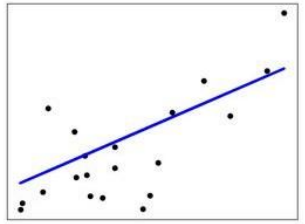
Implementation
You can directly use **LinearRegression** on Scikit Learn.
It is already minimizing MSE (Ordinary Least Squares solution)

[1.1. Linear Models — scikit-learn 1.3.2 documentation]

1.1.1. Ordinary Least Squares

`LinearRegression` fits a linear model with coefficients $w = (w_1, \ldots, w_p)$ to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. Mathematically it solves a problem of the form:

$$\min_w ||Xw - y||_2^2$$



`LinearRegression` will take in its `fit` method arrays `X`, `y` and will store the coefficients w of the linear model in its `coef_` member:

```
>>> from sklearn import linear_model
>>> reg = linear_model.LinearRegression()
>>> reg.fit([[0, 0], [1, 1], [2, 2]], [0, 1, 2])
LinearRegression()
>>> reg.coef_
array([0.5, 0.5])
```

The coefficient estimates for Ordinary Least Squares rely on the independence of the features. When features are correlated and the columns of the design matrix X have an approximately linear dependence, the design matrix becomes close to singular and as a result, the least-squares estimate becomes highly sensitive to random errors in the observed target, producing a large variance. This situation of *multicollinearity* can arise, for example, when data are collected without an experimental design.

Examples:

- Linear Regression Example

Machine Learning - Multiple Linear Regression

Common Metrics - ANOVA table

Source	Degree of freedom	Squared Sum	Mean Squares	F value
Regression	p-1	SSR	MSR = SSR / (p-1)	MSR / SME
Error	n-p	SSE	MSE = SSE / (n-p)	
Total	n-1	SSTO		

As in simple linear regression **R^2** = SSR/SSTO = 1 - SSE/SSTO represents the proportion of variation of y "explained" by the multiple regression model with predictors x1, x2…

If we increase the number of predictor variable xi, R^2 will increase (or stay the same). Thus, by itself, R^2 cannot be used to identify which predictors should be included in a model or excluded.

$$Adjusted R^2 = 1 - \left(\frac{n-1}{n-p}\right)(1-R^2)$$

When comparing two models used to predict the same response variable, we prefer the model with the higher value of Adjusted R^2