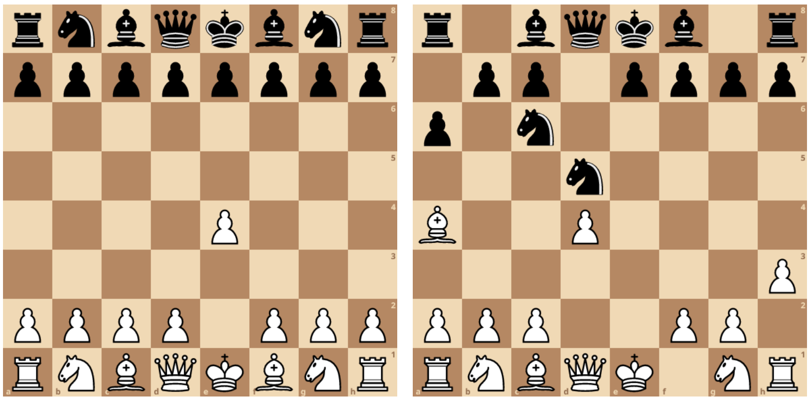
Regression vs Classification of numbers

An example with chess

Daniel Stribrand, July 18, 2023

> This little paper is a retelling of my experiences with developing regression and classification machine learning models for chess. The results are ”fictive”, and not to be taken literally. Treat it rather as a little, silly thought experiment.

Consider the problem that you want to compute the number of moves made, given a position in chess.



(a) One move made (b) Eleven moves made

This is an interesting application, because in theory you want to predict an integer y in the positive unbound range $[0, \infty)$ of integers. Imagine that you develop one regression model r and one classification model c for the task, such that:

$$r(x_i) = r_i \qquad c(x_i) = c_i$$

Assume that both achieve an accuracy α that guarantees **near** certainty, with a loss of $\epsilon_i = (1 - \alpha)y_i$ for position x_i. This will thus give the following result:

x_i	...	x_n	x_m	x_o	...
y_i	...	y_n	y_m	y_o	...
r_i	...	$y_n \pm \epsilon_n$	$y_m \pm \epsilon_m$	$y_o \pm \epsilon_o$	...
c_i	...	y_n	y_m	y_o	...

Despite the incredible accuracy, the results from the regression model are terrible if:

$$abs(y_i - round(y_i \pm \epsilon_i)) \geq 1$$

The difference between having made *e.g.* n moves or $n + 1$ moves is detrimental to the context of the game (for instance in determining which player is to make the next move). This is a good example of a drawback with regression: **the loss** in the computation directly affects the predicted **label**.

On the other hand, the classification model does not suffer from this problem: **the loss** in the computation does not directly affect the predicted label, only the **confidence** of the label.