Supervised Machine Learning vs. Traditional Regression: Exxon-style Oil-Refining vs. Cinderella-style Pea-Picking

STUFF TO KNOW BY HEART - EVEN WHEN DRUNK!

Remember differences between supervised Machine Learning and traditional Regression in terms of:

1. Importance of Theoretical Substance / Interpretability

2. Typical Input Data Size

3. Model Simplicity / Weight Sparsity

4. Direct Optimization vs. Gradual Adaptive Learning

As explained in another note, supervised Machine Learning is similar to traditional Regression in the sense that both try to get Hypothetical Output $\mathbf{H} = h(\mathbf{X}, \mathbf{W})$ to closely mimick actual observed Target Output $\mathbf{Y}$. There are, however, important differences between the two frameworks that we should take note of.

1. IMPORTANCE OF THEORETICAL SUBSTANCE / INTERPRETABILITY

REGRESSION: Good Regression models are constructed according to a theory (e.g. Capital Asset Pricing Model; GDP as function of Labour, Capital and Technology; etc.), or in order to validate/negate a proposed theory. Theory here means a well-analysed, supposedly how-the-world-works, mechanism by which observed output $\mathbf{Y}$ is generated from input variables $\mathbf{X}$. In this setting, the estimated weights $\mathbf{W}$ are meant to explain the the influences of $\mathbf{X}$ in producing $\mathbf{Y}$, and weight magnitudes have important and meaningful theoretical interpretations/implications. If there are inputs x_1 and x_2 that are correlated (hence not independent), giving rise to interpretive difficulties, then people often need to justify why both of them are needed and how to understand the numbers.

MACHINE LEARNING: By contrast, most Machine Learning models practically **DON'T CARE about theory and interpretability**; their ultimate objective is **generalized predictive accuracy**. If we give a trained Machine Learning model new data that it has not seen before and it gives fairly accurate predictions, then the model has been a good student and is now a pretty decent predictive worker. Plus, as elaborated below, because Machine Learning models often have hugely many input variables and weights, people who train Machine Learning models rarely ever spend a second trying to understand what each one single weight means.

2. TYPICAL INPUT DATA SIZE

REGRESSION: Because of their having theoretical bases, Regression models are very selective in terms of what data they admit, that is, only data that are justifiably potentially relevant to the respective theories they support. Regression models are like Cinderellas wanting to get only peas from a pile of ash, knowing in advance that only peas are useful for their jobs.

MACHINE LEARNING: In terms of input consumption, Machine Learning models are nothing like poor Cinderellas, but rather like big Exxon Mobil oil refineries. They admit large amounts of crude inputs (like real oil mixed with sand/dirt, fish bones, and any unfortunate octopi…) **that contain some things, somewhere, that are of value**, but **there is no prior theory or belief to decide which x's in particular are relevant and how exactly they combine**. We hence just freely throw all data that we have into the model and try to adjust the collection of weights $\mathbf{W}$, which easily consists of thousand of weights, to make $\mathbf{H}$ become closer to $\mathbf{Y}$.

3. MODEL SIMPLICITY / WEIGHT SPARSITY

REGRESSION: Because of the need for theoretical interpretability, Regression models often aim to be as simple as possible, and input variables with "statistically insignificant" weights are dropped until only the significant variables remain. Not only do these Cinderellas want peas and not ash, but they also want good peas and not rotten peas. Research papers reporting Regression results have thick appendices detailing meticulous significance tests such as t-tests, F-tests, chi-squared-tests, etc.; variables having weights that are insignificant at a 5% significance level - i.e. the rotten peas - are trashed. (Or, sometimes, if the author is surprised that a variable he/she believes is really important turns out to be insignificant at 5%, then he tries to justify using a 2.5% significance level, or takes the logarithm (ln) of that variable, squares that variable, or in utmost desparation, takes only a subset of data that he likes and pretends not having seen the troublesome cases! Those are Enron-like things people do with Regression models.)

MACHINE LEARNING: Machine Learning models DO also need to be simple, but simplicity is not achieved by carefully/selectively dropping specific input variables or weights. That is not practical because, as mentioned above, Machine Learning models often greedily consume tons of data - thousands of variables - indiscriminately and try to digest them with thousands or millions of weights. Instead of zooming into each input variable / each weight, Machine Learning models achieve simplicity by cleverly **limiting the aggregate magnitudes of the weights**, a technique called "**regularization**". Regularization forces weights to compete with each other for relative relevance (measured by their contributions to the model's predictive accuracy), a process in which valuable weights will grow big and irrelevant ones diminish towards zero.

4. DIRECT OPTIMIZATION vs. GRADUAL ADAPTIVE LEARNING

REGRESSION: Regression models aim to directly optimize their cost functions, achieving absolutely minimal costs in order to "fit" their Hypothetical Output **H** as closely to actual Target Output **Y** as possible. This is partly because regression models often need to have theoretical meanings and hence all relevant variables need to emerge with large, statistically significant weights. Regression models use strong, direct optimization methods to quickly reach their cost functions' minima.

MACHINE LEARNING: When learning through training data sets, Machine Learning models also need to make **H** close to **Y**, but NOT too close. They often **do best by NOT achieving the absolutely minimal cost with the training data**. This is because Machine Learning models' goal is to **generalize well to new cases** beyond the training data, and hence need to learn enough to get the general rules of the game (which are useful for generalization) but not so much as to also memorize the irrelevant, potentially misleading small details that are specific to the training data. Machine Learning models hence use relatively slower, more gradual algorithms in adjusting their weights **W** adaptively, similar to how real-life people learn.