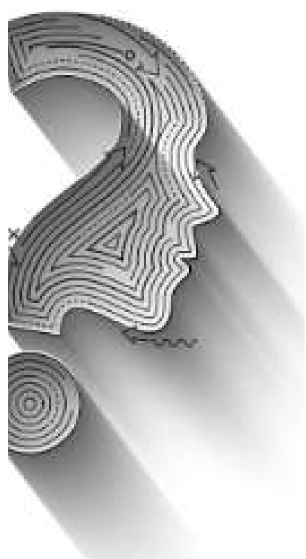
EXAFLUENCE
Data Driven Influence

Are We Just Replicating History or Bettering It Through ML Models?

Malaya Rout, Karthikeyan Sankaran
October 2023

Why is Machine Learning so different from classical programming paradigm?



In rule-based programming, we give inputs and rules to produce an output. In contrast, machine learning programming takes inputs and outputs in various instances and returns the best relationship between the inputs and outputs. That's why machine learning programming is called learning by examples. For the same reason we also say that machine learning doesn't require explicit programming. ML modelling approach is quite in line with inductive reasoning which moves from specific observations to broad generalizations. The inputs and output (let me consider there is a single output) that are provided to the ML algorithm are events that have taken place or decisions that were made in the past. Therefore, we call them historical data. The factors impacting the decision are called independent variables while the decision itself is called the dependent variable.

Can Machine Learning improve the way decisions are made in future?

In our pursuit of training a model to behave as close to historical events as possible, we often forget that matching history should not be our objective. Performing better than history should. The way classical ML model building is set, only an ideal history might produce an ideal future through prediction. History was never ideal. Moreover, repeating history is too boring an objective to pursue. Won't you agree?

What can we do to solve this? What can we do to enable us make decisions in future better than those of the past?

We can do two things to handle this situation.

1. Instead of relying on decision made (output) to be the dependent variable, we should find out the effect of the decision made (outcome) and make it the dependent variable. Only that dataset will give us what we will believe as a good dataset for the model to be trained on. If I say, originally the captured and stored historical data had only 60% happiness in it, I should strive to make the historical data represent, let's say, 90% happiness by using the outcome and not the output. Only then use it to train the model. Only then use the model to predict my future. Only then I can expect a close to 90% happiness in future. In other words, a 60% happy past is converted to an 85% happy future through my ML model (of course based on accuracy of the model).

   However, challenges are plenty. Output is easily stored and available to us. How do we determine the outcome? Outcome could be good or bad assuming a binary scenario. Good or bad from whose point of view? An outcome that is good for an organisation might be bad for the society or the country. What is defined as good and what is bad? How long after the event we should wait to measure the outcome? I don't have answers to any of these questions.

2. The second way is to start with the baseline model of trying to replicate history and manage to perform at less than historical levels. Keep on gathering feedback on whether our ML model is performing as per KPIs (Key Performance Indicators) set. Keep on updating our knowledge base. Use the knowledge base to improve our model. Repeat this cycle till we achieve 85% happiness. The biggest challenge here is, how long does it take to arrive at the set goals? How much is an acceptable unit change? Do data scientists or sponsors have the patience?

Is it important to set our expectations right from standard ML models?

1. Supervised ML, defined as creating a mathematical model that maps inputs to output, helps to generate “predictions” and not “decisions”. The predictions can potentially help with decisions but they cannot be equated. Decision Theory or Decision Intelligence is a different area altogether and deals with topics such as

   a. One Time Decisions (Ex: Secretary problem)

   b. Repeated Decisions (Ex: Inventory ordering)

   c. Adaptable decisions (Ex: Chess moves)

2. Decision Intelligence is complex for reasons such as

   a. Full Observability vs. Partial Observability (Ex: Decisions in a chess game vs. decisions in Poker)

   b. Discrete vs. Continuous States (Ex: Bottling plant vs. Fertilizer Manufacturing plant)

   c. Deterministic vs. Stochastic Environments (Ex: Good ATM Machine vs. Weighing Scale)

   d. Benign vs. Adversarial Situations (Ex: Writing an exam vs. Playing a tennis game)

3

Reinforcement Learning is a branch in ML that tries to bridge the "Prediction" to "Decision" gap. RL takes historical data on State, Action pairs and constructs a model that can predict the "best" action given any state. There are many algorithms in this space but the one that has given the best results so far has been 'Deep Reinforcement Learning' (thanks to fantastic work by DeepMind). While Temporal Difference, Policy Optimization, Q-Learning etc. have yielded good results, Deep Q-Learning has worked best in scenarios that have continuous states, stochastic environments & adversarial situations (Ex: AlphaStar, AlpaZero and AlphaGo)

Do you see ML as a solution to all business problems?

Bottomline, ML by itself does not claim to improve decision making. It does not even try to understand the real world (as statistics attempts to do). ML is just a fast & quick way of formulating a mathematical function to map inputs to outputs. The function can be very simple & parametric as in a Linear Regression model or can be complex & non-linear as in Neural Networks but at the end of the day, it is just fitting a curve to a set of observed, historical data points. The fact that even such a simple prediction machine is useful in many business contexts is fascinating to all of us as analytics practitioners but we should not fall into the trap of using ML as a hammer to hit all nails (read business problems).