Are your industrial data science models field-ready?

Part 1

How Low is Too Low for R^2

Viswanath Avasarala, PhD.

Why This Article?

Over the last few months, I have been keenly following the rigor with which the healthcare community is conducting clinical studies during the current pandemic. I was contrasting this with the more free-style approach that we in the industrial data science community use in proving out our models and deploying them. While the stakes are arguably lower, the impact of even a simple mistake can be similar- the performance of your AI model in the lab will not be reproducible in the field. This article series is an overview of common mistakes that I have noticed practitioners in the industrial sector commit while building machine learning models and how to avoid them. I hope you will find this article useful if you are:

- A subject matter expert in the industrial sector and are looking to move towards building a career in data science
- An executive charged with generating ROI from digital programs
- An industrial data science expert looking to validate your own experience.

As a first step to delivering value at scale from AI, you need a robust digital capability including:

- Management of machine learning models throughout their lifecycle
- Building end-to-end data integration pipelines from raw data ingestion to analytic output delivery
- Optimization of your models without impacting their generalizability
- Rinse and repeat design patterns that scale to all of your data across the varying modalities, volumes, and velocities.

If you are burdened with a plethora of complex tools and platforms to achieve the above, I recommend that you look at DeepIQ software, which is a simple containerized app that delivers all of the above in a scalable manner. Beyond these, industrial data science has its own set of idiosyncrasies and this article

focuses on these issues. I will start with some core issues and progressively bring in more complex items. I will focus on cross functional collaboration, end to end pipelines and model management needs of industrial workflows. I attempt to create simplistic, contrived examples to illustrate the ideas lucidly and provide real examples from my experience as a way of making them relevant to problems you are looking to solve.

I have divided the article into three different parts, as follows:

a) How do you and your business leaders decide if your model is worth taking to the field?
b) Will your model live up to its lab performance in the field?
c) How do you combine your model with legacy reasoning tools or human intelligence to provide the best possible outcomes to your business?

Issue One: R^2 - How low is too low?

If you have data science colleagues in the digital world, you would have noticed with envy how easy it is for them to prove the impact that their models are generating. For them, graduating a model, from the lab, to an A/B testing environment and on to production when proven, is a run of the mill operation. For example, if you believe you have a better recommendation engine, you deploy it on a portion of your web traffic and compare its performance to the previous model. If more people are clicking on your recommendations, then your model will be in production within no time! Simple. Unfortunately, our non-digital world is much more complex. Before you get an approval to take your model from lab to the field where it will start impacting the business, you will be asked the question: Is your model field worthy?

Let us focus on this question using a machine learning performance metric R^2. R^2 is the standard metric in machine learning regression problems and is defined as "the proportion of the variance in the dependent variable that is predictable from the independent variable(s)." The higher R^2, the better the model is. Ideally, you would want all your output variability captured by your model with R^2 of 1. However, in industrial data science, you will very often be limited by the quality of datasets. The phenomenon you are trying to model may have too many external dependencies that are not captured by your data. The high-frequency data sources may have been down-sampled significantly before they were written to storage. The current sensing systems on your equipment are not designed for advanced analytic use cases. Often, you will end up with a model that provides only a partial explanation of your outputs. You will end up debating with your business users on whether it improves the status quo at all.

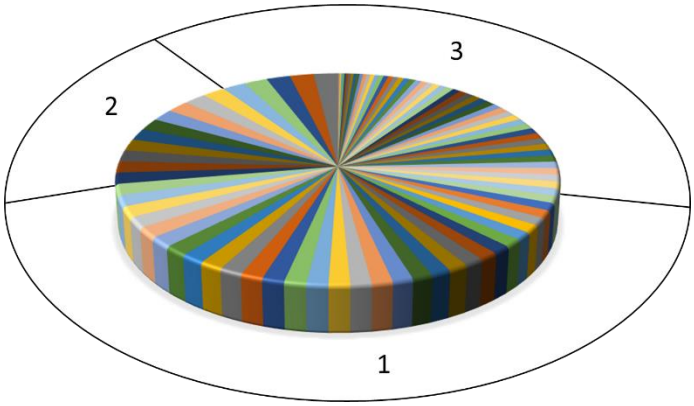
An obvious question arises. What is a good enough R^2 to warrant deploying the model in production? Is there an obvious value of R^2 where the model should be discarded?

I have noticed many data scientists interpret R^2 as if it were a statistics test where a model is rejected when an R^2 threshold is not met. This is possibly inspired by the analogue of statistical testing where you reject the null hypothesis if p-value is less than the significance level.

However, the correct answer is, it depends. Whether the model is useful or not is dependent on the business problem we are solving.

A Small Contrived Example

Let me explain with an example of a contrived game of chance. While the example is R^2 focused, the same ideas apply for other model performance metrics like accuracy, F2 score, adjusted R^2 etc. You are playing a game similar to roulette. You bet on where a ball will land on a spinning wheel that has 100 numbers, partitioned into three categories as follows.



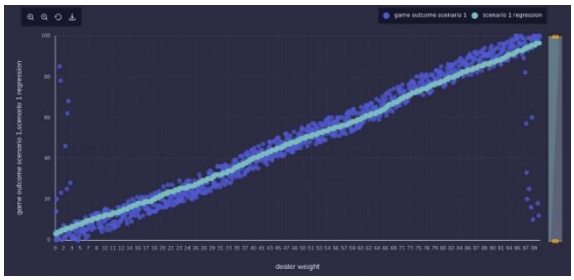
1) 1-45
2) 46-55
3) 56-100

The ball is supposed to fall equally likely on any of the numbers. So, your chances of winning when you bet on category 1 or 3 is 45% and on category 2 is 10%. On each game or ball spin, you can bet $10 dollars. You double your money if the ball falls in your chosen category but lose your money if it does not. So, if you play long enough, you will end up losing all your money because whatever category you bet on, you have a higher likelihood of losing.

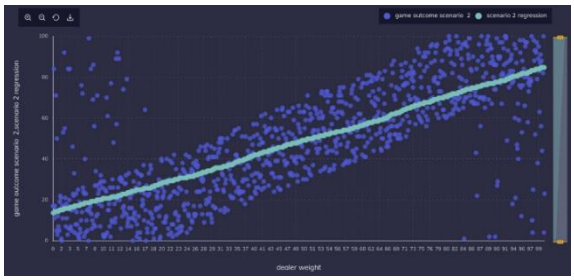
This is where it gets interesting. Because you are a data scientist, you started noticing a pattern. The roulette machine seems to be not functioning correctly, and you have a hunch that the category that the ball is falling in has some dependency on the weight of the person spinning the wheel. You decide to observe 1,000 games and build a model to predict the game outcome based on spinner's weight. Using this model, you plan to play the next 10,000 games in hopes of making a lot of money.

Consider six different scenarios. For each scenario, both the training dataset and the simulated dataset are here https://deepiq.com/docs/training.csv. The only difference between the six scenarios is the strength of dependency between the game outcome and spinner's weight. In the first dataset, your hunch proves accurate. The game result has a strong dependency on the spinner's weight. In the later data sets, I progressively reduced the relationship, and for scenario 6, your hunch is completely wrong. There is no relationship between the weight and game outcome. For each data set, let us fit a simple regression model to predict the game result based on spinner's weight and use this model to play the 10,000 games attached in this dataset https://deepiq.com/docs/simulatedgames.csv. The regression models and the training data are shown in Table 1.

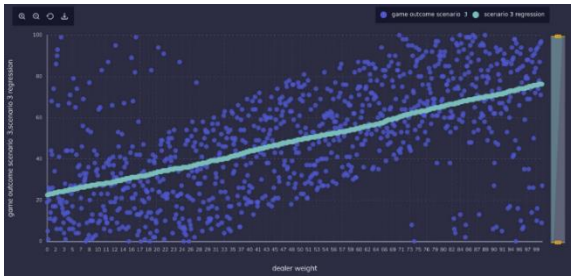
Table 1 Six scenarios with progressively weakening relationship between spinner's weight and game result



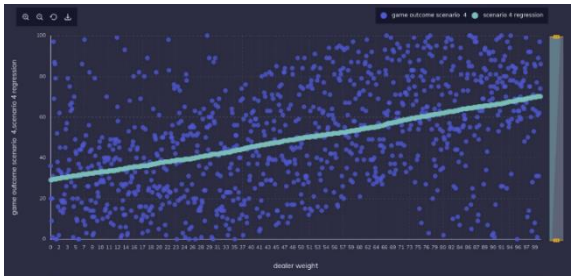
Scenario 1 ($R^2 = 0.9$)



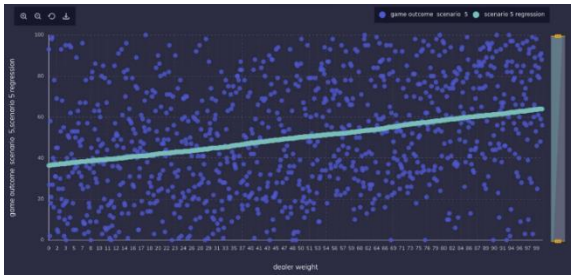
Scenario 2 ($R^2 = 0.58$)



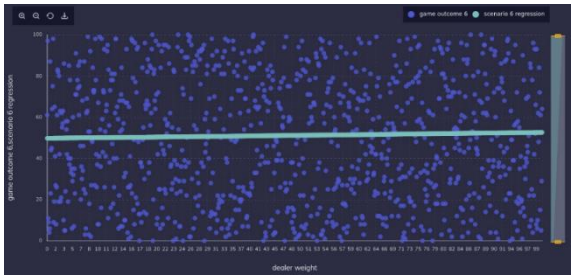
Scenario 3 ($R^2 = 0.35$)



Scenario 4 ($R^2 = 0.2$)



Scenario 5 ($R^2 = 0.1$)



Scenario 6 ($R^2 = 0.0$)

Figure 1 shows your return after playing 10,000 games with your model for each of the six scenarios.

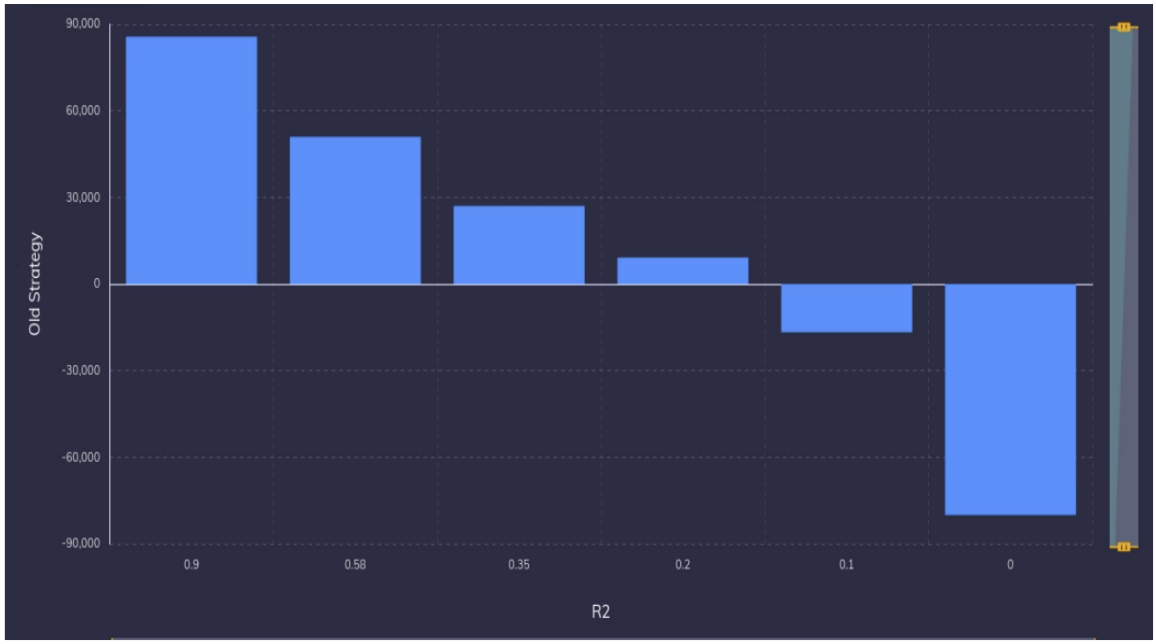


Figure 1: Your return after playing 10,000 games against R^2

In Figure 1, as R^2 decreases, your return from using the model also decreases. However, even with a R^2 of 0.2, you are making a profit of $9,150; much lower than $85,690 you were making with R^2 of 0.9, but much better than playing without a model and incurring an expected loss of $10,000.

However, at R^2 of 0.1, you started incurring losses and you were better off not playing the game with this model. So, if your intention is to make a positive return from this gambling game, all the above models where R^2 is equal to or greater than 0.2 would have been useful.

Long story short, whether a model is good or not is dependent on the problem you are trying to solve and the returns you want from your model. You can still make money or generate a positive ROI using "poor quality" models. Having some model that generalizes well might be significantly better than having no model at all.

In this simulated game, a "poor R^2" model provided you a sufficient advantage over your opponent, the spinner. In real-world business cases, the fact that you have a little less uncertainty in your expected outcomes than your competitors can still provide a differentiating advantage.

Your Problems are not Constant

Businesses typically consider their problems as a given and data science projects as a tool that either succeeds or fails in solving this prescribed, constant problem. However, they leave value on the table by doing this. Optimal use of your analytic models might require you to adapt your business strategy and, in some cases, completely change your business model.

A Small Contrived Example

In the roulette game, we started losing money when our models have an R^2 of 0.1 or less. Let us try to change our strategy a bit. Say you noticed that when the model predicts category 2, it is highly likely to be wrong. So, you change your strategy as follows

- *When the model predicts category 2, you do not bet on the game.*

Figure 2 shows the returns in all the six scenarios between the old and new strategy. Now, you will make money even at a R^2 of 0.1. So, by adapting your strategy, you converted your "bad model" to a useful model. In fact, except for the first model, this strategy will net you more money for all other instances than the first strategy. So, except in the first exceptional case, your business will benefit by changing your strategy of playing the game.

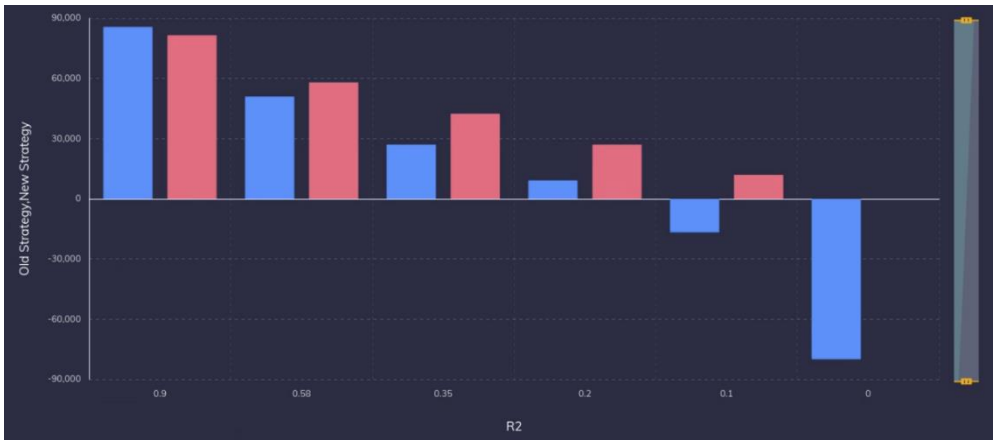


Figure 2: Your ROI with different strategies for all the scenarios

This scenario explains the interplay between business strategy and your model performance. Many times, in industrial data problems, your data is fixed and collecting more data is not an option. It may not be possible to improve the performance of your best models. The only thing you can adapt is your strategy. Once you know the best you can do with your data, the focus should shift to what is the best you can do with this information.

Using the additional information provided by these models, can a business strategy evolve to give the business a possible competitive edge? While decision sciences and risk analysis are mature fields that deal with this issue, I see that many businesses still struggle with this thinking. Consequently, focusing on high R^2 models leads to a rat race. Data scientists feel tremendous pressure to generate higher quality models without a real change in the fundamentals of the problem (better data or more flexible problem definition). When faced with the alternative of cancellation of the entire program, they try multiple strategies to develop "impressive" models including:

1) Complex models - *Maybe you should add more layers to your network model?*
2) Feature engineering - *Maybe you should start normalizing the differential pressure with input pressure and retrain?*
3) More creative data partitioning - *How naïve of you to expect the model capture this failure mode when it never saw this in training data. Maybe, you should swap this portion?*

All these are valid responses to low performing models but need to be done carefully without undermining the validity of your models. More on this in later parts. But, for this problem in question, perhaps the right question to ask may be regarding:

1) Business Strategy: *My model is not good enough to shift completely to a predictive maintenance strategy. But is it good enough to reduce the frequency of our planned* maintenance?

My point is that successful data science is an iterative interaction between technology and business strategy. Both need to adapt to each other's shortcomings for best results. You will need to build business KPIs that measure your model's impact. You will need to run "what if" scenarios to understand the interplay between strategy and model results. You will need an application that will make it easy for your technology teams and business leaders to collaborate and generate the best possible outcomes.

Real World Examples

I will give you two examples of bad models generating value.

The first one is an extreme example. My team was working on a predictive maintenance use case for critical equipment at a large industrial company. We developed a simple interface to pull all the data from the equipment after each job and collected data for multiple months as a first step. Next, we built a model hoping to predict future failures before the tool starts on a new job. Despite our best efforts, we had close to zero predictive value in the model. The onboard data collection procedure down sampled the data significantly, thereby killing any useful signal we could latch onto. We declared the project a disaster and decided to move on. A few months later, the reliability manager reached out to us asking us to repeat the process for other equipment and informed us that they saw an improvement of 11% in reliability because of our project. It turned out that the data collection process we put in forced the field technicians to manually pull the data and then visualize the error logs after every job. The rigor enforced by our model execution environment enabled technicians to see data issues they were previously ignoring and prevented many catastrophic on-job failures.

The second example is from a supply chain inventory management problem. I worked with a pharmaceutical company that was trying to optimize the inventory levels at each of its retail outlets. High inventory levels are an unnecessary expense while having low inventory results in lost business and customer traction. They were using a simple statistical approach where each retail unit would restock each week to meet at least a 99% percentile of expected demand based on historic data. As you can imagine, this resulted in significantly excessive inventory holding costs. We built a simple model that forecasted next week's demand based on the demand in the last few weeks. Because the underlying dynamics of the situation were highly stochastic, the model had low performance. However, when we used the model to calculate the 99% percentile of week inventory, we could significantly reduce the average inventory level at each retail unit while still meeting the business objectives.

How Can DataStudio Help

Impactful data science requires an iterative partnership between business leaders, subject matter experts and data experts. Proving out the value of your analytics and adapting your business strategy to maximize their world in the real world is complicated. Let us say you have a model that provides prognosis of equipment health, so you can avoid catastrophic on field failures. When the model performs well, you will reduce maintenance costs by reducing the frequency of planned maintenance, improve safety and reduce unplanned downtime. When the model generates wrong predictions, your false alarms will increase unnecessary maintenance, distract field personnel, and throw away valuable useful life of the equipment by driving unnecessary maintenance. Your model's missed detections combined with the reduction in planned maintenance can increase catastrophic failures. Your optimal maintenance strategy is a function of the model quality. A perfect model will allow you to shift completely from planned to predictive maintenance. Alternatively, you might have to devise a hybrid strategy to make best use of the new information that the model is providing. To this end, successful industrial data science application requires a partnership between stakeholders. It requires integration of multiple business and operational data sources and allows you to calculate business specific KPIs and run "what if" scenarios.

The self-service capabilities of DeepIQ's DataStudio give your process engineers, geoscientists, digital teams, and business leaders a single place to collaborate and generate value at scale from AI. DataStudio brings strong discipline to your data processes by empowering you to gather all possible data sources including IOT, geospatial and IT sources and tracking their lineage and provenance. You can create integrated datasets combining these different sources and address many of your data issues before the modelling ensues. You can build optimized models and monitor their performance continuously in the production pipeline. You can integrate the model outputs directly into your existing business processes or easily build dashboards to provide insights on a continuous basis. For business KPIs, you can bring in additional business system data like financial data from SAP or maintenance data from IBM Maximo, as needed. Using this data, you can build business specific KPIs that your business leaders can track and understand the impact your work is generating on the business. The rigor and sophistication that DeepIQ brings to your industrial data science processes will be game changing to your data science journey.

In this section, we talked about what kind of models will be useful to your business. In the next section, I will elaborate more on how to build these models in a robust way, so they translate well from the lab to the field.