Quality and Equity in Data Science

Anita Taucher

University of Virginia
Charlottesville, VA

ABSTRACT

This paper proposes the introduction of a modified Software Delivery Lifecycle, in order to eliminate operational bias and recognize impact vs intent issues with data models. The observation that minimum standards for data science have not yet emerged, comes from classroom interactions where various field experts were not able to answer students' questions concretely, regarding how to execute fairness and equity while performing in the workforce. The best advice to date, such as "be vigilant" and "check your data sources" will not result in broad or long-term effectiveness. Instead, we need a clear framework to establish principles for accomplishing data science modeling with built-in ethical standards.

Keywords: Ethics, Quality, Data Science

INTRODUCTION

It is possible that data scientists are largely coming from business, statistics, or finance backgrounds, and not coming from engineering backgrounds. That is an area of interest because engineering training encompasses a framework for delivering software or systems, that inherently includes a testing phase, before delivering a final product. But, such training is not necessary in non-engineering fields, and as a result, is being overlooked as a necessary step in the overall process of data science modeling, possibly due to the lack of engineering influence.

As Data Science is an amalgamation of engineering, statistics and business, necessary practices from all fields must be adopted and standardized. This paper describes the need for Data Science to adopt the Software Delivery Lifecycle, incorporating data science specific phases, and emphasizing ethical concerns that are not prominent in legacy software cycles.

SOFTWARE DEVELOPMENT LIFECYCLE

While researching which standards are followed in data science for producing, publishing, and maintaining models, I was not able to find where Data Scientists are following an established Quality Assurance process. But further, there seems not to be any standard lifecycle methodology, at all. For this reason, I start with describing the Software Development Lifecycle.

The Software [or System] Development Lifecycle (SDLC) is a framework that defines activities performed during the software development process, from planning to implementation and release (BlogTeam). Within the SDLC framework, which is a mature process having countless text books and article descriptions, there exist variations for achieving each step, depending on the priorities of the system or software implementation. For example, the waterfall method emphasizes a more-or-less linear path from needs identification to feature delivery, whereas the agile method promotes an emphasis on coding prototypes and failing fast, then adjusting the direction, and trying again. (4)

Any internet search reveals articles, papers, textbooks and certification classes for project managers to learn the methodologies associated with the SDLC.

Yet, I was not able to find any solid references for using the SDLC, or any standardized methodology, to produce Data Science models. I can find only a smattering of one-off articles that suggest that data science projects should follow a standardized development path.

One example of an article promoting the idea of standardization is "Building a Data Science Life Cycle" by Doug Rose (5), which is on the right track, proposing a less rigid version of SDLC. The author makes the point that data science is by nature exploratory, so a rigid standard might not be widely adopted.

But this article fails to identify model deployment and production reassessment, and also falls short of specifically identifying testing and ethical considerations in either the source data or the model results.

Like Mr. Rose, I believe the SDLC is a good general framework to describe the work that needs to be accomplished.

With some variation, the SDLC is generally represented by a circle, which emphasizes the phases for producing a software product, beginning and ending with a set of activities that result from the previous deployment cycle, such as illustrated here (5).

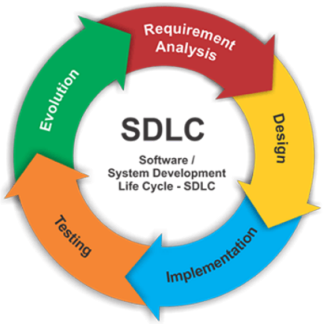

Figure 1. Software Development Lifecycle

THE DATA SCIENCE MODIFICATION OF SDLC

I propose a Data Science SDLC that follows a similar path, but is improved by adding ethical data handling, and specific testing procedures. Its phases are:

- Requirements Analysis
- Data Handling
- Design & Implementation
- Testing & Quality Assurance
- Evolution

Requirements Analysis

The Requirements Analysis phase is identical to the standard SDLC. The work during this phase includes understanding the needs of the business and setting expectations for measuring outcomes. For example, healthcare-related models need an accuracy score of 99% while marketing may be considered successful at 70%.

One departure from SDLC is, this step will be revisited during the Data Handling phase.

Data Handling

I define a novel Data Handling phase, which requires presentation back to the stakeholders before moving forward with the next phase.

The Data Handling effort encompasses identifying data sources, establishing data access requirements, methods for extraction, and decisions on storage formats, as well as an understanding of how the live data will be provided to the production model.

Over and above these bullet points, the legal constraints and ethical expectations need to be discussed and documented. Depending on the industry, there are federal regulations on what types of data can be used to target individuals, such as for Direct Mail campaigns in the Financial industry. And when data points are people, there needs to be clear expectations on how marginalized groups are treated, making sure they are neither exclusively targeted nor excluded by the model, depending on the business requirement.

Further requirements, such as honoring requests from individuals (unsubscribed or opted out or do not call or deceased, etc), need to be identified and removed here, so that data points that will be ultimately excluded from the production runs are also not part of the model training process. The team must ask: to what extent is it an ethical consideration, that people who have actively requested to be removed from a

process, might have their personal information used for training a model that achieves the business goals of profiting from that same process?

Because further requirements are driven by Exploratory Data Analysis (EDA), the EDA is conducted on the training data as a significant step in this phase. The full EDA is then presented to the business, to ensure the requirements have been properly interpreted, and can be addressed by the training data. When the data are people, the EDA must include distributions of any available group-level demographic information, so that the requirement for up-sampling or down-sampling is established before the model is built. In order to treat demographics fairly, the training data must represent all identified groups equally.

Had Amazon stakeholders reviewed summary statistics of the training data before their ill-fated recruitment engine systematically discriminated against female applicants (1), they would have had the opportunity to correct the engine before it was built. A data handling project phase opens the door for stakeholders to gain knowledge of the training data on which model parameters are built.

Such a presentation back to the business intends to give the stakeholders insight into the types of data that are being used to define the model. It opens the door for a business to recognize potential discrimination, that a model developer is not inclined to identify. And, it describes the model to some extent, so that it is harder for stakeholders to shirk responsibility by indicating the science is a black box which no one can interpret (6).

At the end of this stage, a presentation has been made to the stakeholders, and requirements have been updated as a result of the feedback from stakeholders.

Design & Implementation

The design phase mimics the SDLC, but uses data science approaches. This phase includes decisions regarding feature selection, feature reduction (can reduction be sufficiently explained for the project?), model selection, and choosing hyperparameters (i.e. grid search, manual tuning, or lessons learned from previous deployments). If the model is a neural network, this step includes establishing the architecture of the model, itself.

This step is iterative and collaborative among data scientists, as model selection requires model testing and comparison before its effectiveness can be fully understood.

At the end of this stage, the model is fully functional and ready for testing.

Testing & Quality Assurance

It is well established that it is ineffective for developers to test their own code (7). Yet, in Data Science we fully expect the developer to create and test their own models, which are rapidly moved into production. It seems we are treating a data model like the production of a graph – just get some data together to show a measure.

But, in Data Science we are not simply producing a numeric measure or a graphic result for consumption on a slide deck; such a view is held by a developer where everything boils down to the final measure. Instead, we are actually producing a software product to actively process live production data, with the added potential of positively or negatively affecting human experience, and the obligation of realizing stakeholder requirements. The gravity of this reality demands a formal review before release.

To become aware of the discrimination introduced by population generalization, the following activities should be added to any previously identified QA testing steps:

- A report of summary statistics on the data dropped from the training set. This is intended to identify any patterns on data dismissed as ”outliers”. If all outliers are from the same pattern, the model must be adjusted, and the stakeholders informed.
- A report of summary statistics on misclassified data. This is intended to identify patterns in the excluded population. If the misclassification is centered on a single group, such as dark-skinned individuals in image classification, the model must be adjusted and the stakeholders informed.
- A report of summary statistics on properly classified data to identify which classes are actually being targeted by the model. It is just as important to understand whom the model is targeting, and to make a determination if this results in preying on a specific group.
- Reviewing approaches to Feature Engineering
- Reviewing final features for compliance and ethics.

During this stage, the model inputs and outputs are reviewed using summary statistics to ensure fairness and equity in the training and output of the model.

Training in Diversity, Inclusion and Belonging (DIB) warns that the intention of a spoken statement does not matter, it is instead how the statement is received that determines if it is discriminatory (impact vs intent) (Waters). Impact vs. Intent can be inferred from a data model where humans are data points: If the output (likened to a speaker's statement) of a model appears to discriminate against or preys upon a group, then the model is discriminatory, no matter its intended design. Therefore, in order to accept responsibility for what is built, we need to study both the inputs and the outputs of a model, in order to understand model's impact despite its intent with a human population.

At the end of this stage, documentation is delivered in the form of summary statistics on the various identified datasets: dropped, training, misclassification, proper classification, feature selection, and output.

After presenting the model and the results of its QA process to stakeholders, the model is ready for production deployment.

Evolution

The Evolution stage of a Data Science model includes monitoring the measure of accuracy over time. The model will need adjustment, and potentially retraining as a result of natural changes in data and market conditions, over time. When the measure falls below an acceptable threshold, the model should be reassessed, starting again from the requirements analysis phase.

IMPORTANCE OF ETHICAL PRIORITIES

While it may feel like a "black box" to those outside of data science, a model can be somewhat described by the data that is being used during the training phase. With this information, a model is not as incomprehensible, as is being normalized across the industry. If an organization understands the data with which the model is trained, they have some understanding of the up-front cost of the effort, and can make informed decisions on ethical priorities.

For example, people generally do not understand how ham is made, but they can understand that it involves a routine mass slaughter of pigs, and it results in a well preserved, edible meat. For those that have ethical concerns about the mass slaughter of animals, they do not need to understand how ham is made before drawing conclusions about consuming the result. They only need to be presented with the up-front cost, to make responsible decisions about their tolerance of the known part of the process. This can be generalized to data science models.

Separately, there was a time when software development project plans did not include line items for accessibility features that support persons with disabilities. But, "In 1998 the US Congress amended the Rehabilitation Act to require Federal agencies to make their electronic and information technology accessible to people with disabilities." (3). Since the federal government is also a very large consumer of commercial software, accessibility features became the norm for enterprise software companies, so that accessibility features are now fully incorporated into everyday technology.

The first example describes awareness, while the second example describes influence. Both examples show that when ethics are prioritized above the convenience of ignorance, we can normalize our expectations to be inclusive and serve the broader population.

CONCLUSION

It is time to call attention to the missing boundaries and sidesteps we are normalizing in the growing field of Data Science. Building a model is the equivalent of delivering a software product, but with the added risk of marginalizing groups through the necessary statistical step of generalizing a population.

An obvious gap exists from not systematically emphasizing the importance of Quality Assurance in the development of data science models. In the name of efficiency, or possibly ignorance, the data science segment is turning a blind eye to the importance of a testing process, and as a result is exacerbating the problems inherent in highlighting patterns in a generalized population.

The result is that data science is promoting the next generation of oppression of marginalized groups.

In order to stop operationalizing bias, we must first identify and actively remove the bias from our operations. And in order to move the conversation past "the model just said what the data said", we need to quantify intent vs. impact. Implementing a standard framework that provides entry points to open conversations, and helps to preempt potential discrimination, is a first step in addressing the problem broadly.

Following such a framework can go a long way in normalizing the checks and balances needed to ensure the entire population is considered with fairness and equity.

REFERENCES

[1] Acuña, S. T. and Ferré, X. (2001). Software process modelling. *ISAS-SCI (1)*, 1:237–242.

[BlogTeam] BlogTeam. What is sdlc? understanding the phases of the software development life cycle.

[3] contributors, W. (2022). Section 508 amendment to the rehabilitation act of 1973.

[4] Pinheiro, J. (2018). Software development life cycle (sdlc) phases.

[5] Rose, D. (2021). Building a data science life cycle. 16.

[6] Rudin, C. and Radin, J. (2019). Why are we using black box models in ai when we don't need to? a lesson from an explainable ai competition. *Harvard Data Science Review*, 1(2).

[7] Sirikrai, S. (2013). Quality assurance: Choices and changes.

[Waters] Waters, Shonna, P. *Intent vs. Impact: a Formula for Better Communication*.