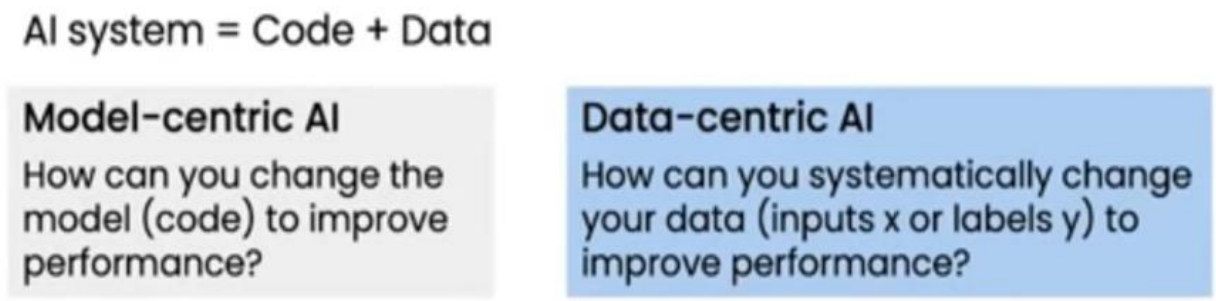
Moving from Model-centric to Data-centric approach

Google researchers found that “data cascades — compounding events causing negative, downstream effects from data issues — triggered by conventional AI/ML practices that undervalue data quality… are pervasive (92% prevalence), invisible, delayed, but often avoidable.”

Lets discuss on the trend being followed widely for most or all of the AI use cases across the organizations. Just to make myself clear the term AI over here in being referred as an umbrella encompassing our DataScience/Machine Learning and Deep Learning Use cases.

The Two basic components of all AI systems are Data and Model, both go hand in hand in producing desired results. We do realize that the AI community has been biased towards putting more effort in the model building One plausible reason is that AI industry closely follows academic research in AI. Owing to open source culture in AI , most cutting edge advances in the field are readily available to almost everyone with who can use github but working on data is sometimes thought of as a low skill task and many engineers prefer to work on models instead but the equation suggests in order to improve a solution we can either improve our code or, improve our data or, of course, do both.



Model-Centric Approach

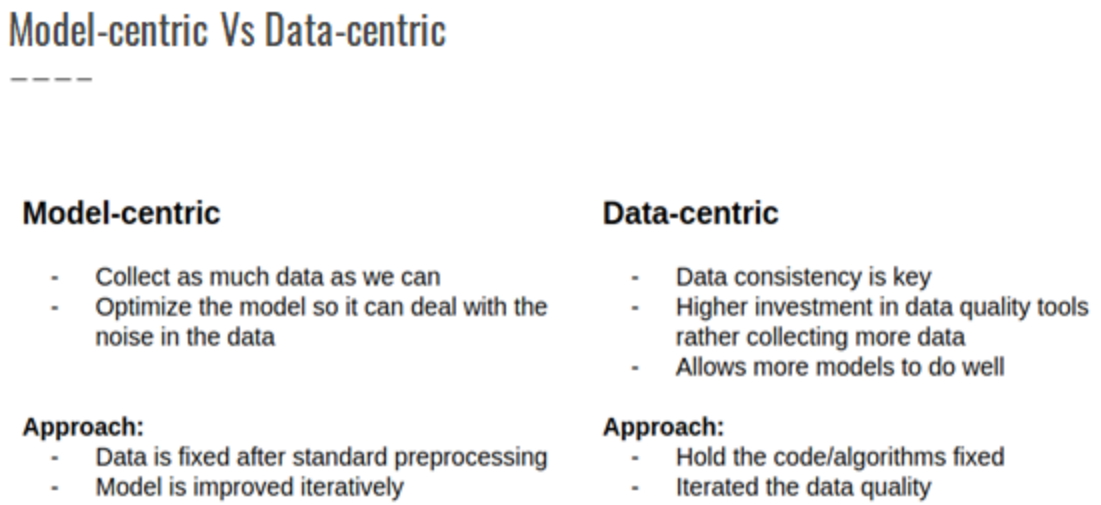
ML is an iterative process this involves designing empirical tests around the model to improve the performance. This consists of finding the right model architecture and training procedure among a huge space of possibilities to arrive to a better solution.

In the dominant model-centric approach to AI, according to Ng, you collect all the data you can collect and develop a model good enough to deal with the noise in the data. The established process calls for holding the data fixed and iteratively improving the model until the desired results are achieved.

Data-centric approach

This consists of systematically changing/enhancing the datasets to improve the accuracy of your AI system. This is usually overlooked and data collection is treated as a one off task.

In the nascent data-centric approach to AI, “consistency of data is paramount,” says Ng. To get to the right results, you hold the model or code fixed and iteratively improve the quality of the data.



Who are talking about it?

Research scientist Martin Zinkevich emphasizes implementing reliable data pipelines and infrastructure for all business metrics and telemetry *before* training your first model. He also advocates testing pipelines on a simple model or heuristic to ensure that data is flowing as expected prior to any production deployment.

The Tensorflow Extended (TFX) team at Google has cited Zinkevich and echoes that building real world ML applications “necessitates some mental model shifts (or perhaps augmentations).”

Recently, however, more attention has been paid to the role of low-quality data is playing in what Ng has identified as the proof-of-concept to

production gap, or the inability of AI projects and machine learning models to succeed when they are deployed in the real world

The message from both of these leaders is that deploying successful ML applications requires a shift in focus. Instead of asking, *What data do I need to train a useful model?*, the question should be: *What data do I need to measure and maintain the success of my ML application?*

Data and AI Divide

Popular machine learning frameworks such as TensorFlow, PyTorch, and SciKit-Learn don't do data processing. Because these data systems don't "do AI" and these AI technologies don't "do data", it's extremely hard for enterprises to succeed with AI, which after all requires both ingredients to be successful. Data science tools that emerged from a model-centric approach offer advanced model management features in software that is separated from critical data pipelines and production environments. This disjointed architecture relies on other services to handle the most critical component of the infrastructure — *data*.

As a result, access control, testing and documentation for the entire flow of data are spread across multiple platforms.

Need for Data Centric ML Platform

At this point before proceeding further I would reiterate More data is always not equivalent to better data. A data-centric ML platform brings models and features alongside data for business metrics, monitoring and compliance. It unifies them, and in doing so, is fundamentally simpler.

a. Data is often siloed in various business applications and is hard and/or slow to access. Likewise, organizations can no longer afford to wait for data to be loaded into data stores like a data warehouse with predefined schema.On the one hand, data in aggregate becomes more valuable over time — as you collect more of it. The aggregate data provides the ability to look back in time and see the complete history of an aspect of your business and to discover trends. Real-time data is most valuable the moment it is captured. In contrast, a newly created or arriving data event gives you the opportunity to make decisions — in the moment — that can positively affect your ability to reduce risk, better service your customers, or lower your operating costs

b. From the infrastructure invested to have data collected, to the number of human resources dedicated to it and how rare can be to have it collected in the ideal situations, makes data one of the most expensive assets nowadays. The industry trend is to move away from

large capital expenditures (capex) to pay for network and server capacity in advance — and toward a “just-in-time” and “pay-for-what-you-use” operating expense (opex) approach

c. Improving how your entire organization interacts with data. Data must be easily discoverable with default access to users based on their role(s), prioritize use cases that make use of similar or adjacent data. If your engineering teams need to perform work to make data available for one use case, then look for opportunities to have the engineers do incremental work in order to surface data for adjacent use cases

d. MLOps (machine learning operations) is *the active management of a productionized model and its task, including its stability and effectiveness*. In other words, MLOps is primarily concerned with maintaining the function of the ML application through better data, model and developer operations. Simply put, MLOps = ModelOps + DataOps + DevOps.

e.Unified Analytics brings the disparate worlds of data science and engineering together with a common platform — making it easier for data engineers to build data pipelines across siloed systems and prepare labelled datasets for model building while enabling data scientists to explore and visualize data and build models collaboratively. Unified Analytics provides one engine to prepare high quality data at massive scale and iteratively train machine learning models on the same data. Unified Analytics also provides collaboration

capabilities for data scientists and data engineers to work effectively across the entire AI lifecycle.

So now that we have defined the problems/ distinguished between the two approaches why need for a Data centric platform. Let's look into the capabilities required to support any organisations transition to a Data Centric Approach. I am not here to advocate usage of a particular product or tool but rather hand hold on general capabilities to look out for before you decide on a build vs buy decision and also this is no hard written fact everybody may have their own trajectory quite different from another.

1.Data processing and management

Since the bulk of innovation in ML happens in open source, support for structured *and* unstructured data types with open formats and APIs is a prerequisite. The system must also process and manage pipelines for KPIs, model training/inference, **target drift**, testing and logging. Not all pipelines process data in the same way or with the same SLA. Depending on the use case, a training pipeline may require GPUs, a monitoring pipeline may require streaming and an inference pipeline may require low latency online serving.

2.Secure Collaboration-

Real world ML engineering is a cross-functional effort — thorough project management and ongoing collaboration between the data team

and business stakeholders are critical to success. Access controls play a large role here, allowing the right groups to work together in the same place on data, code and models while limiting the risk of human error or misconduct.

3.Testing

Ideally automated, tests reduce the likelihood of human error and aid in compliance. . Data should be tested for the presence of sensitive PII or HIPAA data and training/serving skew, as well as validation thresholds for feature and target drift. Models should be tested for baseline accuracy across demographic and geographic segments, feature importance, bias, input schema conflicts and computational efficiency

4.Monitoring

Regular surveillance over the system helps identify and respond to events that pose a risk to its stability and effectiveness. How soon can it be discovered when a key pipeline fails, a model becomes stale or a new release causes a memory leak in production? When was the last time all input feature tables were refreshed or someone tried to access restricted data?

5.Reproducibility

We know AI modes are non deterministic so it is important to validate the output of a model by recreating its definition (code), inputs (data) and system environment (dependencies). If a new model shows unexpectedly poor performance or contains bias towards a segment of the population, organizations need to be able to audit the code and data used for feature engineering and training, reproduce an alternate version, and re-deploy.

6.Documentation-

Documenting a ML application scales operational knowledge, lowers the risk of technical debt and acts as a bulwark against compliance violations Documentation is an important features that bring human judgement and feedback to an AI system.