Managing Machine Learning Projects

Balance Potential with the Need for Guardrails

V.M. Megler, PhD

February 2019

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS's current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS's products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. AWS's responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Abstract

Today, many companies want to build applications that use Machine Learning (ML). This paper outlines some best practices for managing machine learning projects and offers methods for understanding, managing, and mitigating the risks some organizations might face in the delivery of these complex systems.

The intended audience for this paper includes business stakeholders, managers, data scientists, and software development engineers.

Introduction

Today, many organizations are looking to build applications that use Machine Learning (ML). 86% of data science decision makers across the Global 2000 believe machine learning impacts their industries today. However, many enterprises are concerned that only a fraction of their ML projects will have business impact.[1] In some cases, investments made in ML projects are questioned and projects abandoned when the implementation does not match the vision. [2]

As you create the plan for your machine learning project, it's important to consider some emerging best practices. These best practices put ML projects in an economic context, so you can identify, quantify and manage the business impact of your project. They also provide methods to help you understand, manage, and mitigate the risks your organization might face as you deliver your complex ML system, which could make it more difficult to achieve your business goals. These best practices, referred to as *ML project guardrails*, try to help you improve the communication between your ML practitioners and your business stakeholders, who are financially responsible for the costs and benefits of your system by using familiar management methods.

ML project guardrails help you to maximize the odds of your project succeeding by:

- Placing the ML project in an economic context, identifying its expected benefits and balancing that with its costs and risks
- Focusing on issues frequently encountered in ML projects
- Supporting early identification of risks, for subsequent management and mitigation
- Providing transparency of project controls
- Enabling executive review of project risk/reward trade-offs

While these methods are most applicable to larger, established organizations, or to large ML projects with potential for broad impact, they can be helpful to every organization considering an ML project. The formality of the practice implementation can vary depending on the project, the company, and the needs of the stakeholders involved.

AI/ML – A (Very) Brief Summary of the Current State

The potential of Machine Learning (ML) and Artificial Intelligence (AI) has been widely discussed. Recent successes described in the popular press include these topics from the "It Was a Big Year for A.I." article:[3]

- A.I. Spotted an Eight-Planet Solar System
- Beat the World Champion Go Player
- Bested Poker Pros at No-Limit Texas Hold'Em
- Two A.I. agents, Bob and Alice, started out speaking in English but then...developed their own language to speak in
- … And AI Taught Itself to Program

However, the ML industry is beginning to understand the need for more *engineering discipline* around ML. As a National Science and Technology Council (NTSC) report recently noted, "AI currently relies on a set of separate methods or approaches, each useful for different types of applications."[4] In particular, the report defined the public understanding of AI and ML, as follows:

- *Narrow AI (or ML*) – Specific application areas such as playing strategic games, language translation, self-driving vehicles, and image recognition. Here, "remarkable progress has been made," including specific, technical, grand challenges, such as winning Jeopardy. However, transferring that knowledge to other areas has been identified as a challenge.
- *General AI* – Exhibits apparently intelligent behavior, at least as advanced as a person, across a full range of cognitive tasks. This capability is still far in the future.

While some problems in specific application areas are a good fit for ML, others are less so. While this fit changes over time, ML is most successful when there is a good match between a particular problem, a particular ML technology or approach, and the existence of a particular data set that aligns with the problem and approach.

There is a growing belief that the characteristics that make a particular problem a good fit for ML techniques is not yet well-defined or understood. This sentiment has been expressed by leaders of the AI field, for example:

- "At present, the practice of AI, especially in fast-moving areas of machine learning, can be as much art as science. Certain aspects of practice are not backed by a well-developed theory but instead rely on intuitive judgment and experimentation by practitioners. This is not unusual in newly emerging areas of technology, but it does limit the application of the technology in practice."[5] – NSTC
- "It turns out AI technology is complicated, more than I thought."[6] – Andrew Ng, Co-Founder Coursera, former Chief Scientist at Baidu
- "Thus, just as humans built buildings and bridges before there was civil engineering, humans are proceeding with the building of societal-scale, inference-and-decision-making systems that involve machines, humans and the environment. Just as early buildings and bridges sometimes fell to the ground—in unforeseen ways and with tragic consequences—many of our early societal-scale inference-and-decision-making systems are already exposing serious conceptual flaws… What we're missing is an engineering discipline with its principles of analysis and design."[7] – Prof. Michael Jordan, UC Berkeley

Adapting and applying emerging best practices to ML projects can help to narrow the gap between ML projects and engineering disciplines. Better project processes, including those described in this whitepaper, can help you better identify and manage the risks of your ML projects.

ML Projects – Not Your Traditional Software Project

Machine learning projects have some aspects that frequently make them different from traditional software engineering projects. Some of those aspects include:

- Whether the ML project is currently in a research or development stage
- What the economic value of the ML project is expected to be
- Verifying that the ML project assumptions are valid

Research vs. Development

For machine learning projects, the effectiveness of the project is deeply dependent on the nature, quality, and content of the data, and how directly it applies to the problem at hand. Frequently, for ML projects the initial question to be answered is: *Is it possible to use this data to improve results in [specific business process]?*

Answering this initial question requires a series of explorations: What data is available? Is this data directly relevant to the problem? What ML method(s) should we try? What metric should be used to measure "success"?

These are research-oriented questions, so the work during this part of the project is exploratory and iterative. Initial results might seem positive, but can be followed by a complete lack of progress, despite the effort you invested.[8] Results of initial exploration can cause you to look for additional data sources that you can integrate into the project. You might discover that your business problem cannot be solved with the data available. You might also find that the solution is to modify your systems so that the data does become available, or becomes available in the timeframe the model requires (for example, clickstream data might be needed in real-time). Or you might find that the solution is to modify the business problem. Alternatively, you might discover that you can formulate a solution to the business problem using a heuristic or rule-of-thumb more cheaply or appropriately than with an ML solution. While ML is a powerful tool, it is not the solution to every problem.

These research-oriented questions directly contrast with the engineering-oriented questions that occur after you answer the initial question.

One strategy that can help you to clarify the progress of your project and find appropriate approaches, is to clearly identify which stage of the Research-Development spectrum the ML project is currently in:

- **Research stage** — Exploratory in nature, with an unknown answer. Questions are framed as, "is it possible to …?", or "can we use this data to solve the following problem?", or "surely we must be able to …" The answer is unknown, as is the appropriate method to apply to achieve results. The outcome of this effort is either "yes, it is possible, and here's one way to do it that supports that assertion", or "no, we do not believe it is possible; here are all the things we've tried that did not work." This kind of exploration is most often performed by data scientists.

- **Development stage** – The method for solving the problem is now known. Then the questions shift to: how do we implement this method at scale? How do we pipe the data into the model in a timely fashion? How do we collect, store and transform data so models can be retrained consistently and accurate predictions can be calculated within a required SLA? How do we build an A/B testing environment, in order to test future model iterations? This work fits the skill set of data and software engineers.

Many ML projects in large corporations today follow this model: "Research, followed by Development, if positive results are found." Because these two, distinct project stages are different in nature, they require different project approaches:

- **Research stage** – Because this stage is iterative and exploratory in nature, a Kanban project approach is often a better fit for tracking and exposing project progress. Data exploration, initial model building, and analysis typically require large blocks of uninterrupted focus time. Even short meetings can break concentration and lead to loss of momentum and context, which are key to complex analysis, thus slowing progress. Kanban approaches support the difficult-to-predict duration of much data exploration and experimentation during this phase, allowing for longer, more productive time blocks, while still reflecting progress, dependencies, and blockers.
- **Development stage** – In this stage, because the implementation requirements are becoming clear, Agile methods like Scrum and XP are more appropriate project approaches.

Evaluating the current state of the project on the Research-Development spectrum is important in helping you to effectively create work management and staffing plans, and project timelines. Research projects with uninformed timelines create friction between teams and introduce quality tradeoffs for the sake of perceived progress. Instead, include time for data exploration and iteration early in the project, with a major checkpoint marking transition into the Development stage. Revise your project timelines frequently, as your confidence estimates change.

Assessing Economic Value

For most ML projects, stakeholders assume that there is an incremental economic value to the project. Because the value is incremental, you can estimate and assess it, and compare it to the project costs and risks. One way to accomplish this is within an economic framework.

This economic framework is based on a return on investment (ROI) model, which trades costs for expected benefits. Given the speed and the volume of data over which many ML models operate, it is important to consider issues such as the quality of data and the worst-case economic impacts of ML inferences. Therefore, we add some additional factors for ML projects, such as inspecting risks and their potential costs and using tools such as scorecards, as described in the *Using Scorecards to Manage and Mitigate Risk* section.

At a high level, an end-to-end view of ML projects includes the data collection and pipeline, the model itself, and the inferences, which result in the business value (see Figure 1). The data pipeline consists generally of (potentially multiple instances) of several processing steps—filter, merge, transform—and data storage. The economic value and risk analysis should include the end-to-end process.

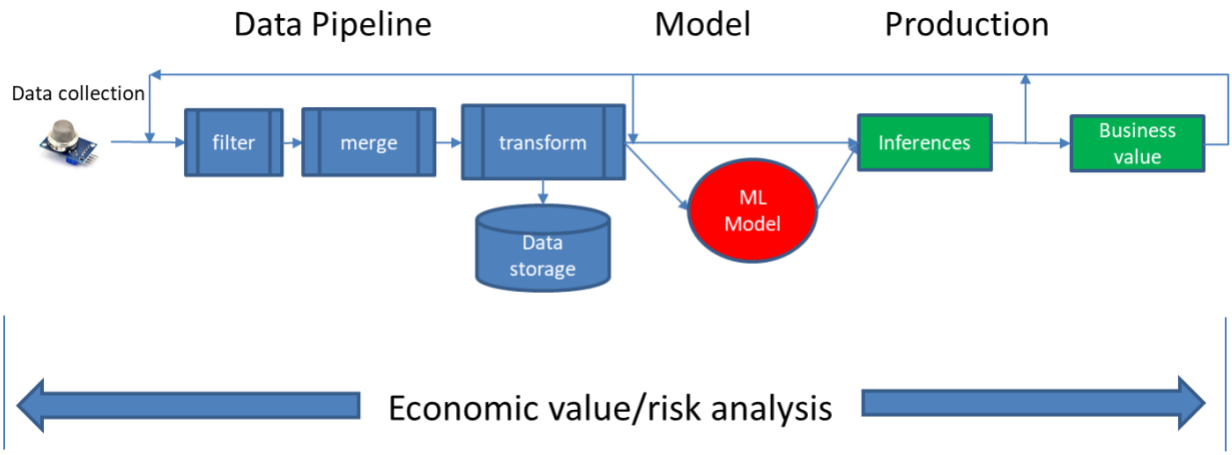

Figure 1– An economic view of ML projects

The economic view has several considerations:

- First, specify the current stage of the project—Research or Development— and then evaluate, assess, and fund accordingly. Some ML projects are intended for explicit production improvements, while others are more speculative, with *big bet* possibilities. The economic model for the project should be consistent with the project type.

  Make sure that both business management and data scientists agree on whether this project is speculative and research-oriented, or uses a well-understood technique in a well-understood environment.

- Evaluate and assess the data pipeline, the ML model, and the expected quality of production inferences. For more information, see the *Data Quality* and

- *From Research to* Production sections.
- Maintain a cost/benefit model, and reassess as changes occur. For example, changes in the external business environment, or addition of an expensive data source, can modify the initial cost/benefit model.
- Understand, evaluate and monitor project risks. Summarize & report regularly to stakeholders

The ROI model also feeds key information into the ML modeling activity. What level of inference quality is required to support the expected value? What is the cost of different kinds of errors (false positives, false negatives)? These values can act as constraints when you train and evaluate candidate models.

To make sure the project remains on track, you must apply the assessment of the risks and economic value on a regular basis, and that deviations or major changes in direction, approach, or risk are understood by all concerned.

Verify Assumptions

ML projects operate under a certain set of assumptions, many of which are unstated and unverified. For most successful projects, this set of assumptions is valid. The following examples are amongst those frequently assumed to be true:

- Variables relevant to the problem are captured and available in the data.

  For example, the major influence of traffic times to a store might be the local school schedule, which might not be captured in the dataset of historic traffic counts. The data used to train an ML model was generally captured for a different business problem…with different assumptions.[9]

- Training, validation, and test sets adequately represent *reality*.

  Are the sets large, representative samples of the populations that the model needs to make predictions about? Does the target variable that the model predicts represent the actual outcome that the ML model is trying to predict, or is it a proxy for that outcome? Are the extraction methods used to generate these datasets the same as for the production data? How similar is the *sample data* to the real data, and is the belief about the similarity supported by testable facts? Are the error sources or treatments the same?

- Training, validation, and test data captured at a point in time remains valid.

  As upstream systems change, the data they produce might have different characteristics, which changes the validity of the model.[10] Temporal changes in the nature or influences of the system being modeled can also affect the validity of the training, validation, or test data.

- The ML model is valid.

  For example, assumptions that the appropriate model was chosen, that *rare cases* were sufficiently well represented, and that correct statistical analysis was performed.[11]

- Transfer learning assumes that the source model is appropriate and the learning is indeed transferable.[12]

- The correlations found are relevant. For example, the divorce rate in Maine is highly correlated with per capita consumption of margarine. Is this relevant to the problem?[13] Correlation is not causation.

- An ML model can be built and deployed in production within the time constraints imposed by management or the business case.

  If not, other avenues should be considered, or the time constraint should be relaxed.

- Concept drift or model drift happen relatively slowly, or can be adequately corrected with model retraining while still supporting the economic model.[14]

- Operational ownership is economically justified.

  The overall economic upside of the system justifies the cost of hiring the ML skills, labeling data or researching errors, maintaining the model, identifying model drift when it occurs, and retraining as necessary.

A best practice is to actively consider the assumptions your project is making, and test them for validity. It's appropriate to be aware of these assumptions and explicitly question whether they apply to your situation.

If they are appropriate, it is worth documenting that assessment and the reasons for it at the beginning of the project, then revisiting periodically to make sure they're still true. *Scope creep*, better understanding of the problem space and available data, or business problem, or data changes over time can change that evaluation.

If the assumptions are not appropriate, then discussing the situation might change the approach or highlight actions that can be taken to correct the incorrect assumptions. For example, a task or project might acquire additional data sources, resources, or connections to other teams.

Data Quality

The quality of the ML model directly depends on the quality of the data used to build it. A significant portion of the *research* component of ML projects is to assess the data quality and whether it's appropriate for the problem. While assessing data quality is still more of an art than a science, some helpful methods include:

- Documenting the data catalog and pipeline
- Estimating the impact of data quality

Documenting the Data Catalog and Pipeline

Frequently, the sources of data and the process used to extract them while building and testing the initial ML model are very different from those used in production to obtain inferences. For example, initial research ("is it possible?") is frequently performed on cleaned and possibly enriched data extracted from a data lake, for convenience and speed of access. The implicit assumption is that the data operated on in production will be the same, and can be provided quickly enough to act on. This assumption should be tested to ensure the ML model will work as expected.

A simple method of identifying potential challenges is to clearly diagram the data pipeline used to build the model, showing where all data is from and how it is transformed. The following is an example of how to annotate the pipeline diagram with this information:

- Major data cleaning operations performed
- Records dropped, either as an actual count or as a percentage
- Major issues found with the data, for example, "duplicate records found and dropped"
- Assumptions made at the time, for example, "data was extracted for US only, other countries are assumed to be similar"

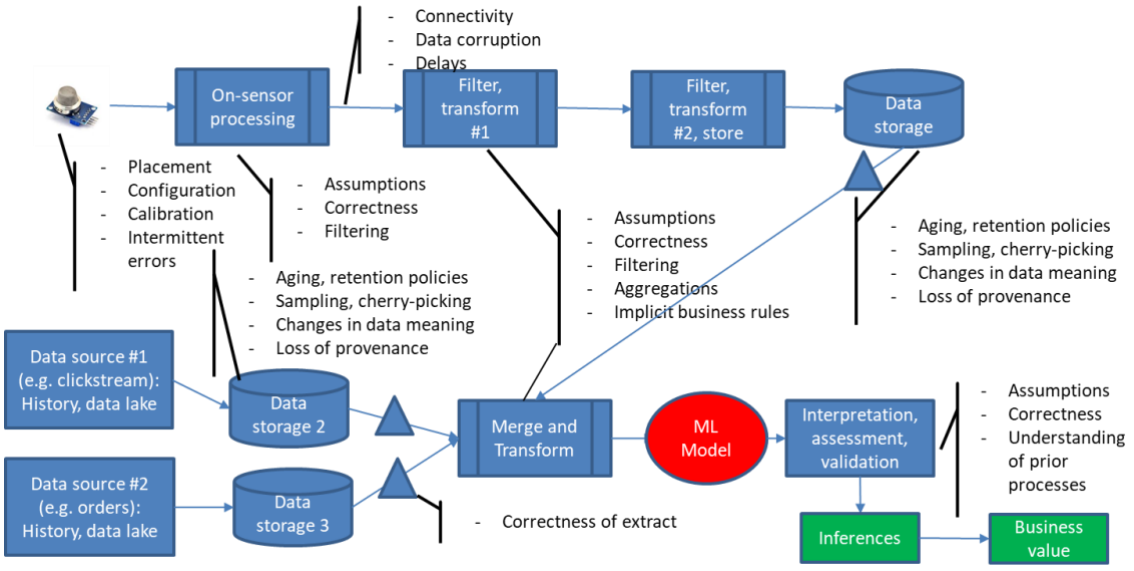

Figure 2 – Plotting the data pipeline and potential sources of error

As shown in Figure 2, characteristics of the source data can also be captured in a *Data Catalog* table, such as Table 1. This catalog documents the current understanding of the data source, communicates to stakeholders the sources to be used and some basic facts about them, and helps identify potential mismatches, concerns, or clarify misunderstandings. For example, in Table 1, the change in storage format of data source #2 potentially adds a conversion or regularization task to the task list.

Table 1 – Sample data catalog during research stage

Source	Contents	Duration	Quantity	Comments
Data source #1: data lake	Clickstream data	Jan 2018–Jan 2019	1.6M	User IP address only; user name not known
Data source #2: data lake	Order history	June 1 2016–Oct 3 2018	55k orders	Format stored in changed on Jan 1 2018 Final order only (not change history) Orders with errors are deleted
Sensor data	Readings from factory sensors. Streaming data is batched and stored	90 days history retention only	50/sec; 5k/sec expected	Data cleaning unknown; is perceived outlier data being dropped?

You should also create diagrams of the pipeline and data catalog to be used in production. Make note of the similarities and differences between the two pipelines.

- If they are the same, then they are likely subject to the same errors. Are these errors important?
- If they are not the same, then different sources, different processing has been applied. Do any of these differences impact the model? How do you know?

Share the diagrams and summaries with SMEs and project sponsors. Discuss the differences and get agreement that they appear reasonable to all stakeholders. If the gap between the two is found to be too large (a very subjective assessment), consider other approaches to sourcing data that is more representative of the data the ML inference endpoint will receive under production circumstances.

The more data sources that are involved, the more disparate the data sources that are to be merged, and the more data transformation steps that are involved, the more complex the data quality challenge becomes.

Estimating Impact of Data Quality

During initial ML training, it is usual to begin with cleaned data (for example, from a data lake), or to clean data prior to running the training. Examples are:

- When merging data, data might be dropped if no direct key match is found.
- Records with null or extreme values might be dropped.

Frequently, there are many individual cleaning and transformation steps performed before the data is used for ML training.

However, when the model is used in production, the data it is sent to perform inferences on is generally coming from a different source, and comes to the model's inference endpoint through a different path, as described in *Documenting the Data Catalog and Pipeline*. The model was built to work well for *cleaned* data inputs. How do you ensure that the model's performance in production will be similar? Here are two approaches that work together: comparing statistics, and validating the model against unclean data inputs.

Compare Statistics

To make sure that the model performs well in production, add a formal checkpoint that takes a step back from the ML model, and compares the source input data to the data the ML model actually used to train on. Make sure to evaluate the data from both a quantitative and qualitative perspective.

- Quantitative evaluation – Review counts, data durations, and the precision of the data inputs.
    - Comparing counts lets you identify, track, and highlight data loss, and test against what seems reasonable. What percentage of source data was used to actually build and test the model? Is there any potential bias as a result of unintentionally dropped data as a result of a merge? Is the data storage subsystem filtering, averaging, or aggregating results after some time period (for example, for log messages)?

  - Reviewing data duration lets you determine what time period each dataset is for. Are all the potentially relevant business cycles included (for example, Black Friday, Singles Day for retail)?
  - Quantify precision, by comparing the mean, median and standard deviation of the data source and the data used to train the model. Calculate the number or percentage of outliers. For lower dimensional data or key variables, boxplots can provide a quick visual assessment of reasonableness.

- Qualitative evaluation – Accuracy is equally important as precision, but likely can only be assessed qualitatively. For example, based on experience, and perhaps some sample exploration, how confident are you that the data is accurate? Are there sufficient anecdotes of errors? For example, do operators report this sensor is always running high?

The actions to take based on this evaluation vary widely. A frequent result is to segment the data based on some factor discovered during the analysis, and take a different action on each segment. For example, in Improving Data Quality in Intelligent Transportation Systems[15], a set of sensors were identified as misconfigured and routed to be repaired, while another subset was used in traffic analysis.

Validate Model Against Unclean Data Inputs

A simple but powerful technique to validate your data model is to take a subset of data that was eliminated during every cleaning or transformation step from the raw data, and compare it to the data eventually used to train the model, and send those items to the ML inference endpoint. Then, assess the resulting inferences. Does the endpoint provide reasonable responses in all cases? Use the results to identify where checks and error handling should be added. Should error handling be added to the inference endpoint? Or, should the applications that are calling the inference endpoints be required to identify and remove problematic inputs, or handle problematic outputs?

A similar practice is to take examples that the ML model incorrectly classified or predicted, and feed those into an end-to-end test of the process within which the ML inferences are made.

Approaches to Apply

Here are some of the methods and best practices that manage and mitigate these risks, and make the best use of the opportunities. The categories described are:

- Staffing the project appropriately
- Assessing economic value
- Using scorecards as a risk management technique
- Using incremental investment approaches

Staffing the Project

There is some confusion about the kinds of people that work in ML. The NTSC[16] noted that the AI workforce includes several largely-distinct types of people:

- A number of AI *researchers* who drive fundamental advances in AI. For researchers, AI training is inherently interdisciplinary, often requiring a strong background in computer science, statistics, mathematical logic, and information theory.
- A larger number of *specialists* who refine AI methods for specific applications. For specialists, training typically requires a background in software engineering and in the application area.
- A much larger number of *users* who operate those applications in specific settings. For users, familiarity with AI technologies is needed to apply AI technologies reliably.

Most ML projects are staffed by people considered *specialists* in this definition. These specialists can be further refined into data scientists and data engineers:

- Data scientists frequently have backgrounds in applied math, statistics background, and perform advanced analytics
- Data engineers frequently have backgrounds in programming and analysis, and specialize in big data technologies[17]

Because up to 80% of effort is in data access and engineering[18], the project team should not consist only of data scientists, though that is the specialty most people think of first. While the exact mix will depend on the size of the project, its potential impact

and visibility, and the risks involved, the following roles should be represented on the team.

- **Steering committee** – The steering committee must include the business stakeholders and the financial *owner* of benefits and risks. The committee makes decisions about the direction of the project and its fit to the business opportunities, including potentially canceling a project if the economic model no longer shows positive returns. It can also bring in external specialists—such as legal, Human Resources, and Public Relations—as needed to manage or mitigate risks. The steering committee can be as formal as needed, depending on the organization structure and financial commitments involved.
- **Subject Matter Experts (SMEs)** – Project success depends on knowledge of how the data is collected, the systems involved, the pre-processing that's been performed, sometimes the history of why data is that way, or the embedded assumptions (for example, data values below x are arbitrarily stripped out before they get to this system). This skill set and knowledge is tightly linked to the business processes involved, and the existing systems and methods used to generate the data. SMEs identify potential data input sources, understand and interpret the source data, validating the *source of truth* if there are conflicts, evaluate data quality, and advise on implementation.
- **Data scientists** – These specialists provide statistical and ML specialty knowledge. They build more complex ML models, and perform statistical evaluation of model performance. Rigorous experimental design is critical, particularly for companies with large user bases or in highly regulated industries. Frequently, only a small number of data scientists is needed compared to other roles.
- **Data, application development, and infrastructure engineers** – These specialists perform data acquisition, ETL (extract, transform, load), and build the data pipeline. This group includes application developers who own integrating the model into an application, and using the inferences in the context of a business process. Typically, a larger number of data engineers is needed than of data scientists. Often, data engineering work is focused at the beginning of the project, acquiring the data for the data scientists and building an initial data pipeline. Later, data engineers or application development engineers take an initial ML model, productionize it, and build out the data pipelines required to support the inferences in production.

Assessing Economic Value

A key first step in your ML project is to build an economic model of the expected value from the project. This model will provide context to inform project decisions, moving the focus from the ML technology to its impact on the business.

The complexity of the economic model, and its components, can vary as appropriate. The goal is to capture the desired cost/benefit equation behind the problem that the ML project will address, at whatever level of detail is required to support decision making. Building an economic model also allows the ML project staff to elicit key business drivers or constraints that the model must meet (such as, “must be at least as accurate as the current process,” or “must provide transparency into how decisions are being made”). These constraints become requirements for the ML system, risks to be managed, or decision criteria on whether the model is sufficiently good to proceed. Whether a model is sufficient to support the business case might be a higher bar than whether the model is a good model.

An additional aspect of the ML project is assessing the cost of errors. Implicit with the speed and volume that many ML models address, is that human intervention and oversight that might exist today are removed. What is the cost of errors? If there is a cost for each error, how much tolerance is there, before the economic model ceases to be positive? If *model drift* occurs, the number of errors might increase. How serious a problem is that? As shown in Figure 3, calculating error costs can show that an otherwise well-performing model (based on otherwise good metrics) might not be economically feasible.

N=16,500	Predicted: No	Predicted: Yes	Count
Actual: No	(True negative) 5,000	(False positive) 1,000	6,000
Actual: Yes	(False negative) 500	(True positive) 10,000	10,500
Count	5,500	11,000	

Classification Rate, Accuracy	0.90
Recall	0.95
Precision	0.91
F-measure	0.92

$$ VALUE	Cost Per	Number	Totals
True Negative	0	5,000	0
False Negative	-1,000	500	-500,000
False Positive	-11,000	1,000	-11,000,000
True Positive	1,000	11,000	11,000,000
TOTALS		**16,500**	**-500,000**

Figure 3 – Assessing the impact of error costs

It might be appropriate to use aspects of the economic model to modify the ML model itself. For example, if the costs of different kinds of errors-such as false negatives or false positives—are widely different, that information can be used to train a model with more desirable outcomes. An example of such an approach is shown in Training models with unequal economic error costs using Amazon SageMaker.[19] Here, the differential costs of errors changes the ML model used to predict breast cancer, producing fewer false negatives (undesirable and expensive) at the *cost* of more false positives, while still producing a cheaper model overall.

Recent experiences with building economic models at the beginning of projects show they can dramatically change the direction and focus of the project. In one case, the cost of required processing was found to be so small that a larger, more holistic ML model replaced the initially-planned, targeted system. In another case, the cost of data acquisition, and its expected fragility due to rapid changes in the data sources, quickly made it obvious that the project was not economically viable. In a third case, gathering data to build an early version of the economic model showed that the problem the ML project was supposed to address could be addressed by a much simpler, more direct method. In each case, presenting the economic model allowed responsible managers and executives to make rapid decisions.

Using Scorecards to Manage and Mitigate Risk

A major component of project management is risk management. Risk management depends on identifying the risks to be managed in the project at hand, then mitigating and monitoring those risks. A core method that assists in this process is scorecards.

Scorecards are a deceptively simple mechanism that can be used to identify, manage, and report on risks. They can be used to make sure that project leaders have discussed potential risk items and captured potential mitigations. The identified risks can be summarized and reported on. The scorecards provide a method to communicate technical issues to responsible executives, ensuring identified risks are communicated to appropriate levels. Sample scorecards are included in the next section.

At Project Inception

At the start of the project, discuss each line item in the scorecards with the project sponsors and identify whether or not this issue applies to the project.

- If not, then document the reasoning.

  This action captures the fact that you discussed this risk and why it didn't apply. If you discover later that the risk does apply, you can revisit the reasoning, identify errors, and modify the scorecard process. For example, if you identify a faulty assumption, you can devise a test of that assumption and include it for next time. You might choose to move these items to a *does not apply* list, which can make it easier to focus on the remaining items of concern.

- If yes, then establish potential impact (perhaps only as an order of magnitude), and identify what mitigations or actions you should apply.

  Is this a relatively minor risk that can be handled within the team? Or, is this a more severe risk, that should involve other teams such as Human Resources or Legal? Are there specific processes that should be applied, such as additional validation or testing steps? These mitigations might be handled within the ML process, or they might be outside the scope of the data science team.

If you or your project team believe the risks are significant—particularly if you've identified potential moral or ethical concerns—it is appropriate to escalate them to the appropriate management level or team within your organization. For example, if you identify a concern that could impact personnel, it might be appropriate to contact Human Resources and ask for their input. They might be able to suggest mitigations, such as offering staff retraining, or assisting with new placement for displaced personnel. It's

appropriate that issues be handled by groups that have the associated responsibility, and that these groups are involved early so they can provide input, plan their responses, and if necessary, provide guidance to the ML team. As is common with other forms of automation, ML can impact many issues that are outside the scope of authority of the data scientists who build the ML models.

The identified risks can then be managed and monitored as the project proceeds. Many of those risks should become tasks on the project plan, to be managed by the usual project management approaches. A number of these tasks might be outside of the current project team.

Project Operations

When preparing for each review, or for each scheduled project report:

- For each *applies* risk, update the current status. Decide whether current mitigations are adequate or additional actions are needed. Revise mitigation plans if appropriate.
- Quickly review the *does not apply* items to verify whether the reasoning is still accurate. If any item should be reassessed, move it to the *active* list.

As the project progresses, the nature of the risks and mitigations should change. While at the beginning of the project, many of the items might have many unknowns and uncertain mitigation plans, shown by a status of *Yellow* or *Red*, as the project proceeds, more of these risks should be mitigated and the status for more items should become *Green*. If the project proceeds and items either become *Red* or do not change, the project should be reviewed in detail, the underlying issues identified and addressed, or the project should either be refocused or potentially cancelled.

Sample Scorecards

The sample scorecards are separated into five topics:

- **Project context** – Addresses the social, business, and regulatory environment of the project
- **Financial** – Identifies the costs and benefits of the problem you are trying to solve with ML, and of the ML system you are developing
- **Data quality** – Highlights areas that are frequently problematic in ML projects, and that can easily mislead the project—missing a signal that exists in the data or believing a signal exists where there is none—if not identified and addressed

- **Project processes** – Addresses processes in the ML project that are easily overlooked in the excitement of developing and testing the algorithm and identifying promising results
- **Summary** – Captures the key risk areas to bring to executive attention

Scorecard topics can be customized to an industry, organization, or team. These samples are not exhaustive, nor are they specific to a country, industry, or company. They do provide a useful starting point, and you should modify them when the issues common to ML projects shift. You can further adapt them by adding, removing, or modifying items to better apply to the context in which they're being used.

Each category is, by itself, a complex topic, often with a plethora of associated research. We give an example in each case of projects where that issue could arise. To spur discussion, we also give some examples of popular articles or research papers that discuss the topic. If a category is judged as affecting the project, you might consider consulting with an expert in that field.

Blank versions of the scorecards are included in the *Appendix – Sample Scorecards*

section.

Project Context

The *Project Context* scorecard addresses the social, business, and regulatory environment of the project.

Table 2 – Sample Scorecard: Project Context

Category	Example	Issue for project? (Y/N)	Status (Red/ Yellow/ Green)	Comments: Applicability, Status, Mitigations
Ethics	Weapons targeting systems Predictive policing[20] AI imitating humans[21]			
Model makes *consequential decisions*	Denying people entry to country, or loans Criminal risk assessments for arrests, bail, sentencing			
Privacy	HIPAA / GDPR applies			
Fairness, Bias[22]	Race identified as loan risk factor Displaced jobs disproportionately held by minorities			
Risk of bad press	Photo labeling app labels African-American as *gorilla* Self-driving car kills pedestrian[23]			
Need for transparency & auditability	Recommendations must be independently verifiable[24]			
Applicability/ success of ML for this application	Natural language assistant chat bots vs free-form conversational understanding			
Closed-world development/ testing vs open-world deployment	Robot in lab vs open house environment (children, pets, stairs)			
Impact of ML model inferences	Self-driving car *sees* pedestrian but ignores it as false-positive			

Financial

The *Financial* scorecard identifies the costs and benefits of the problem you are trying to solve with ML, and of the ML system you are developing.

Table 3 – Sample Scorecard: Financial

Category	Example	Issue for project? (Y/N)	Status (Red/ Yellow/ Green)	Comments: Applicability, Status, Mitigations
Financial model built	Model with anticipated ROI available for review			
Potential upside return	Increased customer retention of 5% Decreased cost per transaction of 5%			
Potential downside risk	Decreased customer retention (10%) Increased cost per transaction (5%)			
Worst-case downside	Automated trading algorithm causes Great Financial Crash			
Liability	Self-driving car kills pedestrian			
Cost of building model	6 months, team of 2[25]			
Cost of maintaining model	Ongoing, 10 hours/month			
Quality of model predictions vs expectations	Economic model assumes 100% correct predictions, but results 85% correct			
Uncertainty in model predictions	Prediction might be accurate +/– 10% Extreme data points cause bad predictions			

Data Quality

The success of an ML project depends on the signal inherent in and extracted from the data. The *Data Quality* scorecard highlights areas that are frequently problematic in ML projects, and that can easily mislead the project—missing a signal that exists in the data or believing a signal exists where there is none—if not identified and addressed.

Table 4 – Sample Scorecard: Data Quality

Category	Example	Issue for project? (Y/N)	Status (Red/ Yellow/ Green)	Comments: Applicability, Status, Mitigations
Input data precision[26]	Test & production data have same characteristics; outliers discarded for both model & production			
Input data accuracy	Sensor values estimated to be +- 5% of actual			
Data volumes & duration	Model data only available for 3 months (but business cycle is 1 year) 50% of source #3 data discarded			
Data sources & pre-processing validated	Data source #1 now undergoing additional quality checks Data extract #2 discovered to be flawed; re-training required			
Production vs model data pipeline	Prod inferences will use separate data source than model trained on			
Data change over time: processes considered	Upstream system changes logic & meaning of its input to model[27]			

Project Processes

The *Project Processes* scorecard addresses processes in the ML project that are easily overlooked in the excitement of developing and testing the algorithm and identifying promising results.

Table 5 – Sample Scorecard: Project Processes

Category	Example	Issue for project? (Y/N)	Status (Red/ Yellow/ Green)	Comments: Applicability, Status, Mitigations
Research or Development	Research, leading to development if successful			
Project Team Skills & Availability	Team is missing production application development skills			
Project Timelines	Committed timelines assume data is available, appropriate, and sufficient for model			
Tests for bias applied	Recidivism rates the same across all populations			
Long tail analysis performed	Minority groups disadvantaged because system is trained on majority			
Statistical analysis validated[28]	Incorrect statistical analysis shows correlation where none exists, leading to incorrect inferences			
Economic analysis of model metrics	Results of model are still in range of project economic value estimates			
Team *temperature check* [29]	Team *gut check*: team willing to be a customer of the system			
Verification and validation procedures completed	Testing completed: privacy assurances, A/B testing Security assessment completed			

Category	Example	Issue for project? (Y/N)	Status (Red/ Yellow/ Green)	Comments: Applicability, Status, Mitigations
Integration with existing processes & systems	New interfaces or APIs required User process changes required			
Instrumentation & monitoring in place	Model endpoint is instrumented to feed data back into training process Monitoring process that identifies model drift is defined			
Production model revisions	Plan to retrain, relaunch models in place & funded			

Summary

The *Summary* scorecard captures the key risk areas to bring to executive attention. It can be prepared for presentation to the steering committee and other interested executives. This simple scorecard can present a compelling summary of a project team's concerns, which enables senior management to focus on the key issues. This summary scorecard is supported by the more detailed documentation provided by the individual scorecards. The following Summary scorecard example shows a single line for each detailed scorecard. Other examples of line items you could include are categories of major risk, such as legal or bias.

Table 6 – Summary Scorecard Example

Scorecard	Status	Summary
1. Project context	Yellow	Implementation risk not mitigated
2. Financial	Green	Upside potential sufficient, downside risk low
3. Project processes	Red	Insufficient analysis performed
4. Data quality assessment	Yellow	Data errors reduced due to new data source

Investing Incrementally

We suggest you take an incremental investment approach to ML projects, and recommend regular reviews with the steering committee or responsible management. The gap between reviews should be long enough for progress to have been made, but short compared to the overall project timeline; we recommend no less than every two months for large projects. At a minimum, a review should occur at every milestone or change of project phase.

The inputs to the review include:

- Document the current assessment, status, and mitigation plans
- Present the summary scorecard and detailed scorecards, if appropriate
- Present the latest data pipelines used for model development and for production (for more information, see *Data Quality*)
- Reassess expected benefits and risk factors when you have new knowledge or updates
- Provide a recommendation on the future course of the project, based on prior items

During this review, the Summary scorecard can be used to focus discussion on the aspects of the project that need the most focus; detailed scorecards can provide supporting documentation. A diagram such as *Figure 4* can be used to track the *arc* of the project over time, as the risk and rewards shift.

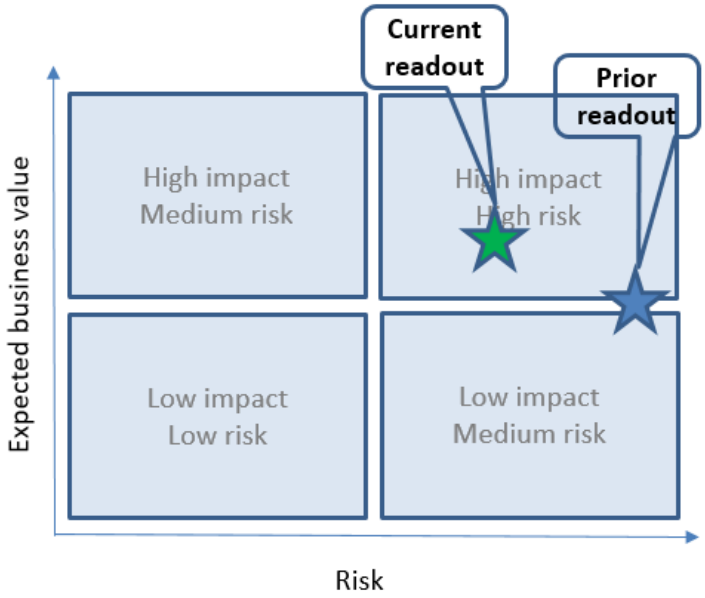

Figure 4 – Plotting the risk-reward arc *of a project over time*

The outcome of the review should be a decision. For example, to continue to invest, to redirect or refocus, or to terminate the project. The outcome of each review should include:

- A formal decision to continue to invest, to pivot in a different direction based on initial findings, or to cancel the project
- Direction on other teams to involve (such as Public Relations, Legal, Finance, or Human Resources)
- Direction on mitigations of specific risks
- Sign-off on risks that have been accepted

From Research to Production

After you have developed an ML model that seems to produce good results, the goal is to put it into production and begin generating predictions. This is the point at which many companies struggle. The following practices can begin to smooth that transition.

Moving from Research to Development

There are fundamental differences in method between research and development:

- **Research** – During this stage, many approaches and analyses are tried quickly and discarded. Some ideas are partially explored and then abandoned. Frequently, a Jupyter or Zeppelin notebook is used as a history of approaches you tried, instead of the physical notebooks that researchers used to use to document their research. Time is not spent writing error recovery code or writing subroutines, because it's not known whether the code will ever be used.
- **Development** – During this stage, there are now *requirements*, produced as a result of the research stage. There is a known method to solve the problem. However, the code in the researcher's notebook is generally not production quality. Reengineering the researcher's code is frequently required to make this code a good fit for a production environment.

While ML models are built to identify frequent statistical patterns across a population (a *portfolio view*), the inference endpoint in production is assessing and acting on an individual case. This is a shift in perspective between research and production.

Unfortunately, the method to communicate the requirements to the development team is frequently by giving them the researcher's Jupyter or Zeppelin notebook, or a set of Python or R scripts. If the development team redevelops and optimizes the code for production while the research team continues from their base notebook, you have the problem of versioning the code and identifying changes.

While understanding of this challenge is still preliminary, these methods have shown promise:

- Encourage your data scientists to understand the deployment phase and your engineers to understand the research phase. Developing an end-to-end understanding is invaluable.

- Encourage your data scientists to consider the production requirements and technical complexity of deployment when considering model designs in the research phase.[30] For example, if the requirement is for real-time inferences, is the model lightweight enough to support that?[31]
- We've found that embedding the data engineers with the data scientists can be successful. These data engineers work to extract the data needed by the data scientists and prepare the data for them. This allows the data scientists to focus on their highest value, while ensuring that the data engineer is aware of the compromises being made during the model building process, as compared to production data.
- Job rotations are another method to encourage this understanding. For example, a data scientist could become a data engineer for approximately 3 months.
- Functional tests must be designed by the data scientists to enforce that the same outcomes—or statistically defined similar outcomes—are produced by production code on the same data. A representative data set is selected by the data scientist for testing. For testing deterministic functions such as feature engineering, the production results must equal the functions written by the data scientists. If a training loop is included in the test, then a statistical tolerance must be agreed upon based on risk management criteria. The production team will then implement the tests in the production environment and include the tests in any production QA.

ML Model as Part of the Software Ecosystem

The ML model you develop is one component in a larger software ecosystem. As described in But What Is This 'Machine Learning Engineer' Actually Doing?[32], all usual software engineering and management practices must still be applied, including security, logging and monitoring, task management, API versioning, and so on.

This ecosystem must be managed using cloud and software engineering practices. For example:

- End-to-end and A/B testing
- API versioning, if multiple versions of the model are used
- Reliability and failover
- Ongoing maintenance
- Cloud infrastructure best practices, such as Continuous Integration/Continuous Deployment (CI/CD)

Extra attention to the following points can be warranted when an ML model is delivered into a production environment:

- Apply software engineering disciplines.

  Add error recovery code and make sure that tests for unexpected data inputs exist. Perform the same kind of unit testing, Quality Assurance, and User Acceptance Testing that is performed for other systems. If the ML system has moved from the research stage to development, some of these expected software engineering practices might have been inconsistently applied.

- Track, identify, and account for changes in data sources.

  The data might change over time. Changes in software that produces a data source can have flow-on effects.

- Perform ongoing monitoring and evaluation of results.

  Evaluate the expectations versus the results of the ML system. Build methods to check the error rate and the classes of errors being made against project expectations. If the overall error rate is the same, are the same proportions of the different classes of errors still the same? Is model drift occurring?

- Create methods to collect data from production inferences that can be used to improve future models.

Conclusion

This whitepaper provides some proven practices for maximizing the success of your ML projects. It clarifies how ML projects can fit within a well-understood economic framework. Identifying the classes of risks that apply to the specific project allows them to be managed using risk-management techniques. Using an incremental project investment approach, which includes regular project reviews against the economic model defined for the project and the identified risks, provides executive management with insight into the risks they are accepting and the possible project outcomes. These practices can help you to make sure that your ML project reliably delivers on the expected outcomes.

In the words of Andy Jassy:

> It's not just the right ML models and services that allow you to do Machine Learning at scale the way you want to; it's being able to place them in the right secure, operationally performant, fully featured, cost-effective system, with the right access controls, that allows you to gain the business results you desire.

Document Revisions

Date	Description
February 2019	First publication

Appendix – Sample Scorecards

This section includes a set of blank scorecards that you can use in your projects.

Project Context

The *Project Context* scorecard addresses the social, business, and regulatory environment of the project.

Table A.1 – Sample Scorecard: Project Context

Category	Requirement or Risk	Issue for project? (Y/N)	Status (Red/ Yellow/ Green)	Comments: Applicability, Status, Mitigations
Ethics				
Model makes *consequential decisions*				
Privacy				
Fairness, Bias				
Risk of bad press				
Need for transparency & auditability				
Applicability/success of ML for this application				
Closed-world development/testing vs open-world deployment				
Impact of ML model inferences				

Financial

The *Financial* scorecard identifies the costs and benefits of the problem you are trying to solve with ML, and of the ML system you are developing.

Table A.2 – Sample Scorecard: Financial

Category	Requirement or Risk	Issue for project? (Y/N)	Status (Red/ Yellow/ Green)	Comments: Applicability, Status, Mitigations
Financial model built				
Potential upside return				
Potential downside risk				
Worst-case downside				
Liability				
Cost of building model				
Cost of maintaining model				
Quality of model predictions vs expectations				
Uncertainty in model predictions				

Project Processes

The *Project Processes* scorecard addresses processes in the ML project that are easily overlooked in the excitement of developing and testing the algorithm and identifying promising results.

Table A.3 – Sample Scorecard: Project Processes

Category	Requirement or Risk	Issue for project? (Y/N)	Status (Red/ Yellow/ Green)	Comments: Applicability, Status, Mitigations
Research or development				
Project team skills & availability				
Project timelines				
Tests for bias applied				
Long tail analysis performed				
Statistical analysis validated				
Economic analysis of model metrics				
Team *temperature check*				
Verification and validation procedures completed				
Integration with existing processes & systems				
Instrumentation & monitoring in place				
Production model revisions				

Data Quality

The *Data Quality* scorecard highlights areas that are frequently problematic in ML projects, and that can easily mislead the project if not identified and addressed.

Table A.4 – Sample Scorecard: Data Quality

Category	Requirement or Risk	Issue for project? (Y/N)	Status (Red/ Yellow/ Green)	Comments: Applicability, Status, Mitigations
Input data precision				
Input data accuracy				
Data volumes & duration				
Data sources & pre-processing validated				
Production vs model data pipeline				
Data change over time: processes considered				

Summary

Table A.5 – Sample Scorecard: Summary

Scorecard	Status	Summary
1. Project context		
2. Financial		
3. Project processes		
4. Data quality assessment		

Notes

[1] Fleming, Reetika, and Phil Ferscht. “How to Avoid Your Looming Machine Learning Crisis.” HFS Research, July 2018. https://1pcll3wzgyqw5kf62erficsw-wpengine.netdna-ssl.com/wp-content/uploads/2018/07/RS_1807_HfS-POV-Machine-Learning-Crisis.pdf.

[2] Polonski, PhD, Vyacheslav. “AI Solutionism.” Towards Data Science, June 21, 2018. https://towardsdatascience.com/risks-of-ai-solutionism-dangers-of-machine-learning-and-artificial-intelligence-in-politics-and-government-728b7577a243.

[3] Bonnington, Christina, and Rachel Withers. “It Was a Big Year for A.I.” *Slate*, December 28, 2017. http://www.slate.com/technology/2018/06/why-doctors-offices-still-use-fax-machines.html

[4] “Preparing for the Future of Artificial Intelligence.” National Science and Technology Council, October 2016

[5] “Preparing for the Future of Artificial Intelligence.” National Science and Technology Council, October 2016

[6] “AI Expert Andrew Ng Wants to Improve Manufacturing.” Fortune, April 24, 2018. http://fortune.com/2018/04/24/data-sheet-andrew-ng-ai-manufacturing/.

[7] Jordan, Michael I. “Artificial Intelligence: The Revolution Hasn’t Happened Yet,” April 18. https://medium.com/@mijordan3/artificial-intelligence-the-revolution-hasnt-happened-yet-5e1d5812e1e7.

[8] Biewald, Lukas. “Why Are Machine Learning Projects so Hard to Manage?” *Medium* (blog), January 28, 2019. https://medium.com/@l2k/why-are-machine-learning-projects-so-hard-to-manage-8e9b9cf49641.

[9] For example: Hollingsworth, Eric. “Unintentional Data,” October 12, 2017. http://www.unofficialgoogledatascience.com/2017/10/unintentional-data.html.

[10] Jordan, Michael I. “Artificial Intelligence — The Revolution Hasn’t Happened Yet,” April 18. https://medium.com/@mijordan3/artificial-intelligence-the-revolution-hasnt-happened-yet-5e1d5812e1e7.

[11] For example, the multiple testing problem, as is humorously described by XKCD: https://xkcd.com/882/

[12] For example: Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson. "How Transferable Are Features in Deep Neural Networks?" *ArXiv:1411.1792 [Cs]*, November 6, 2014. http://arxiv.org/abs/1411.1792. Karpathy, Andrej. "CS231n Convolutional Neural Networks for Visual Recognition." Accessed August 3, 2018. http://cs231n.github.io/transfer-learning/ .

[13] For example: "15 Insane Things That Correlate With Each Other." Accessed January 4, 2019. http://tylervigen.com/spurious-correlations.

[14] https://en.wikipedia.org/wiki/Concept_drift

[15] Megler, V. M., Kristin Tufte, and David Maier. "Improving Data Quality in Intelligent Transportation Systems." *ArXiv:1602.03100 [Cs]*, February 9, 2016. http://arxiv.org/abs/1602.03100.

[16] "Preparing for the Future of Artificial Intelligence." National Science and Technology Council, October 2016.

[17] Anderson, Jesse. "Data Engineers vs. Data Scientists." O'Reilly Media, April 11, 2018. https://www.oreilly.com/ideas/data-engineers-vs-data-scientists

[18] Anderson, Jesse. "Data Engineers vs. Data Scientists." O'Reilly Media, April 11, 2018. https://www.oreilly.com/ideas/data-engineers-vs-data-scientists

[19] Megler, Veronika, and Scott Gregoire. "Training Models with Unequal Economic Error Costs Using Amazon SageMaker." *Amazon Web Services* (blog), September 18, 2018. https://aws.amazon.com/blogs/machine-learning/training-models-with-unequal-economic-error-costs-using-amazon-sagemaker/.

[20] Julia Angwin, Jeff Larson. "Machine Bias Risk Assessments in Criminal Sentencing." Text/html. ProPublica, May 23, 2016. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[21] Vincent, James. "Google's AI Sounds like a Human on the Phone — Should We Be Worried?" The Verge, May 9, 2018. https://www.theverge.com/2018/5/9/17334658/google-ai-phone-call-assistant-duplex-ethical-social-implications. Statt, Nick. "Google Now Says Controversial AI Voice Calling System Will Identify Itself to Humans." The Verge, May 10, 2018. https://www.theverge.com/2018/5/10/17342414/google-duplex-ai-assistant-voice-calling-identify-itself-update

[22] For example: Corbett-Davies, Sam, and Sharad Goel. "The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning." *ArXiv:1808.00023 [Cs]*, July 31, 2018. http://arxiv.org/abs/1808.00023. Bolukbasi, Tolga, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. "Man Is to Computer Programmer as Woman Is to Homemaker? Debiasing Word Embeddings." In *Advances in Neural Information Processing Systems*, 4349–4357, 2016. http://papers.nips.cc/paper/6228-man-is-to-computer-programmer-as-woman-is-to-homemaker-debiasing-word-embeddings.pdf; Snyder, Kieran. "Language in Your Job Post Predicts the Gender of Your Hire." Textio Word Nerd, June 21, 2016. https://textio.ai/gendered-language-in-your-job-post-predicts-the-gender-of-the-person-youll-hire-cd150452407d.

[23] "Software In Fatal Self-Driving Uber Crash Reportedly Recognized Woman, Then Ignored Her." HuffPost South Africa, May 8, 2018. https://www.huffingtonpost.co.za/2018/05/08/software-in-fatal-self-driving-uber-crash-reportedly-recognized-woman-then-ignored-her_a_23429459/ [Note that there are subsequent articles that refine this understanding of the accident: in fact it did see her, but there was no mechanism to inform the driver of the action that the system assumed the driver would take. The example stands, though.]

[24] For example: In 2018, the European Union will begin enforcing a law requiring that any decision made by a machine be readily explainable, on penalty of fines.

[25] Amadeo, Kimberly. "How a 1998 Bailout Led to the 2008 Financial Crisis." The Balance. Accessed January 29, 2019. https://www.thebalance.com/long-term-capital-crisis-3306240.

[26] https://labwrite.ncsu.edu/Experimental%20Design/accuracyprecision.htm

[27] For example, Jordan, Michael. "Artificial Intelligence — The Revolution Hasn't Happened Yet." *Medium* (blog), April 19, 2018. https://medium.com/@mijordan3/artificial-intelligence-the-revolution-hasnt-happened-yet-5e1d5812e1e7

[28] Young, S. Stanley, and Alan Karr. "Deming, Data and Observational Studies: A Process out of Control and Needing Fixing." *Significance* 8, no. 3 (September 2011): 116–20. https://doi.org/10.1111/j.1740-9713.2011.00506.x

[29] Jordan, Michael I. "Artificial Intelligence — The Revolution Hasn't Happened Yet," April 18. https://medium.com/@mijordan3/artificial-intelligence-the-revolution-hasnt-happened-yet-5e1d5812e1e7.

[30] For example, “We evaluated some of the new methods offline but the additional accuracy gains that we measured did not seem to justify the engineering effort needed to bring them into a production environment.” Blog, Netflix Technology. “Netflix Recommendations: Beyond the 5 Stars (Part 1).” *Medium* (blog), April 6, 2012. https://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429.

[31] For example, combining models to create offline, nearline and real time recommendations is described in: Blog, Netflix Technology. “System Architectures for Personalization and Recommendation.” *Medium* (blog), March 27, 2013. https://medium.com/netflix-techblog/system-architectures-for-personalization-and-recommendation-e081aa94b5d8.

[32] See diagram from: Dudek, Tomasz. “But What Is This ‘Machine Learning Engineer’ Actually Doing?” *Medium* (blog), May 27, 2018. https://medium.com/@tomaszdudek/but-what-is-this-machine-learning-engineer-actually-doing-18464d5c699.