Machine Learning Foundations
Unit 9: Deploying Your Model

Table of Contents

- Module Wrap-up: Deploying Your Model

Lab 8b

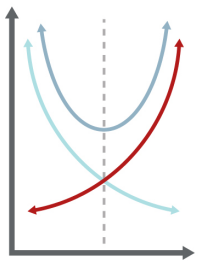
- Assignment: Lab 8b Assignment
- Read: Thank You and Farewell

Unit 9 Overview
Video Transcript

At this point, we have completed our core curriculum and you should have the foundations to build, optimize, and troubleshoot multiple types of machine learning algorithms and apply them to a variety of tasks. In the industry, building your model is just the first step. Models need to be deployed, maintained, and monitored to realize their full value. Model deployment methods vary widely and follow software engineering best practices. We will provide you with resources to begin your career as a machine learning engineer or data scientist. We will also have several experts continuing our discussion on AI fairness as well as real-world applications of AI. Some of the content may overlap with prior weeks' content, but there is a lot of value in both reinforcing important concepts and understanding how those concepts are applied in practice.

What you'll do:

- **Discuss societal failure mode**
- **Understand the sources of discriminatory bias and how to measure and mitigate them**
- **Improve the fairness and accountability of a model**
- **Explore how to deploy, host, and monitor your model**



Unit Description

As you work on more machine learning problems, there are many factors to keep in mind at all stages of model development — before, during, and after! Some of those factors are algorithmic fairness and algorithmic accountability.

In this unit, you will discover why it is important to develop fair and unbiased models. You will also explore how to deploy and host your model to be available for stakeholders to solve the business problems for which the model was developed. A number of experts will share their experiences, explaining best practices

for model development and deployment. By the end of this unit, you will export your project into your portfolio.

Back to Table of Contents

Tool: Unit 9 Glossary

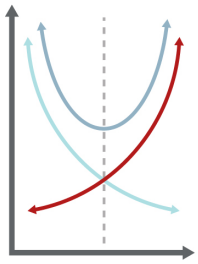
Though most of the new terms you will learn about throughout the unit are defined and described in context, there are a few vital terms that are likely new to you but undefined in the course videos. While you won't need to know these terms until later in the unit, it can be helpful to glance at them briefly now. Click on the link to the right to view and download the new terms for this unit.

Download the Tool

Use this **Unit 9 Glossary** as a reference as you work through the unit and encounter unfamiliar terms.

Back to Table of Contents

Module Introduction: Improving Fairness and Accountability



Other key factors to keep in mind as you work on more machine learning problems are algorithmic fairness and algorithmic accountability; these are topics that Mr. D'Alessandro will introduce and discuss in this module. As model owners and developers, it's important to stay accountable while producing models to be fair and reduce potential societal harm. In this module, you will also encounter many experts describing different types of bias and how to address them.

Back to Table of Contents

Watch: Algorithmic Fairness

Let's continue exploring the failure mode called societal failures. A whole subfield of AI, known as ethical AI, is dedicated to studying model fairness and accountability. Mr. D'Alessandro discusses a few high-profile cases that call into question the fairness and potential societal harms of machine learning algorithms.

Video Transcript

Now I'm going to expand on the discussion of failure mode I've called societal failures. This failure mode is the general focus of ethical AI, which is the subfield of AI that studies model fairness and accountability. I want to begin the discussion by presenting a few high-profile examples. It's important to establish that these problems can exist in everyday products we use, whether we notice it or not. After that, I'll generalize these cases to different types of harms that we need to be mindful of when we consider algorithmic fairness. One of the most high-profile cases of algorithmic bias has been the result of deep investigative reporting by a news organization called ProPublica. They investigated the use of a commercial tool called COMPAS, which predicts whether a criminal defendant might commit future crimes based on their criminal history and other factors. The predictions are used by judges to make sentencing and bail decisions. Public data suggests that this particular model predicts future crime correctly 61 percent of the time. But ProPublica's investigation found that the model's errors were different for defendants of different races. The table here shows the false positive rates in the top row and the false negative rates of the model in the bottom row. A false positive here means being labeled high risk but never actually reoffending. The study found that black defendants had a false positive rate that was twice as high as white defendants. The fairness discussion here usually is centered around the false positives. A positive case here means one is predicted to likely commit a future crime. This prediction could result in key harms to the defendant, such as longer prison sentences or more difficult bail requirements. This was one of the first high-profile cases of algorithmic bias. The makers of the COMPAS tool, a company called Northpointe, responded with a counter-analysis where they claimed that their tool was indeed fair. The challenge for us is that both sides were technically correct. This debate inspired a lot of research

that revealed there are different ways to measure algorithmic fairness, and often these metrics are incompatible with each other. This means that one can achieve fairness with one metric, but violate it with another. In this case, ProPublica was using one fairness metric, while Northpointe was using a different one, which led both to make opposing conclusions that were correct by their own standards of fairness. The next high-profile case I want to highlight focuses on the accuracy of facial recognition systems when comparing people with different skin tones. In 2018, two researchers, Joy Buolamwini and Timnit Gebru, published a study that showed commercial facial recognition systems were generally performing worse on people with darker skin, and particularly, darker-skinned women. When classifying gender based on face, for instance, three commercially available facial recognition systems showed error rates of around 30 percent for darker-skinned women as compared to less than one percent error rate for lighter-skinned men. This paper revealed a major cause for concern, particularly as these commercial facial recognition systems were widely used for a variety of commercial and governmental tasks. This paper and the subsequent advocacy engendered pressured the producers of facial recognition systems to invest more to improve the inaccuracies that were exposed. I love this particular case, both because of the impact it had, but also, it highlights a narrative that a few determined engineers can really make a difference at times. We can generalize these two examples into two types of harms. These are allocative and representational harms. An allocative harm is a discriminatory system that withholds certain opportunities, freedoms, or resources from specific groups. A representational harm is one where a system reinforces negative stereotypes along the lines of identity and protected class. Fairness in machine learning systems is usually centered around whether the system is able to cause specific harms, both by directly disadvantaging people or providing advantages to some groups over another. And a lot of fairness discussions focus on model performance disparities between groups, with those groups usually defined by identification along some protected class. Deciding whether to deploy a machine learning system based on the grounds of ethics and fairness carries a lot of weight. As a junior machine learning engineer, you may not be in a position to override the profit motives of the companies you work for. Fortunately, many of the larger tech companies are embracing ethical machine learning and integrating fairness analysis into the standard machine learning development lifecycle. Certain industries like finance and insurance already have regulations in place that aim to mitigate the risk

that algorithmic decision making in lending and underwriting is impacting different groups at different rates. While this is hard to predict, both European and U.S. regulators are publicly discussing programs for AI and machine learning regulations across all industries. But no matter the reason why your potential future company is or isn't engaged with fair machine learning, a lot of fairness starts with your awareness of the potential harms these systems can cause, as well as your ability to develop models that generalize well. Fair models tend to be good models and vice versa. Reducing societal failures from machine learning applications starts with practicing and improving the craft of building good models.

Back to Table of Contents

Ask the Expert: Mehrnoosh Sameki on AI Ethics

As artificial intelligence becomes more pervasive in society, ethical concerns about how AI systems are developed are growing. For example, there is potential for models to be trained on historically biased data as well as potential for privacy infringement as AI models are fed sensitive data about individuals in order to make predictions.

In this video, Dr. Sameki discusses the role of responsible AI in ensuring that ethical standards are maintained in the field of AI.

Note: The job title listed below was held by our expert at the time of this interview.

Mehrnoosh Sameki
Product Manager, Microsoft

Dr. Mehrnoosh Sameki is a Senior Technical Program Manager at Microsoft, responsible for leading the product efforts on machine learning interpretability and fairness within the Open Source and Azure Machine Learning platforms. Dr. Sameki also co-founded Fairlearn and Responsible-AI-widgets and has been a contributor to the InterpretML offering.

Question

What are AI ethics?

Video Transcript

AI has the potential to drive considerable impact in the way that we do our business. It is literally everywhere. In every single company you can imagine, from the finance to healthcare, to retail, there are many AI systems deployed to impact a lot of decisions that are being made for us as humans. With all this great impact, of course, there is a broad impact on the society as well. As we move forward with AI, it is better to ask this question of what AI should do rather than what AI can do for us. Specifically, AI raises concerns on many fronts due to its potential to be unfair and be trained on top of historically biased data from society. It has the potential to be not transparent and for you to completely lose track of how it has made its prediction. There are also potentials for losing the privacy of individuals while making decisions for them. Over

and over again, we have seen many stories coming up in the press that is talking about some of the negative impacts and influence that AI can have on humans' lives. Now, the great news is there is this area called responsible AI, which is an umbrella term and framework to bring many of these critical considerations around AI transparency, reliability, fairness, privacy, accountability under one roof, and provide companies with best practices, set of tools, guidelines, and truly just a framework to follow in order to make sure that AI is not causing harm on humans. With any responsibility comes accountability. Responsible AI is all about putting humans first rather than always technology first. Responsible AI can guard against the use of historically biased data to train models, ensure that automated decisions are justified and explainable, and ultimately help maintain user trust and individuals' privacy. We hope that responsible AI will provide a clear set of rules and best practices to companies and allow them to innovate and realize the transformative potential of AI while staying 100 percent accountable for the impact it has on humans.

Back to Table of Contents

Read: Metrics and Types of Fairness

Algorithmic fairness is an important topic in machine learning. When improperly implemented, machine learning models can cause societal harm in ways that are detrimental to underprivileged and underrepresented groups. As a responsible machine learning engineer, it is your duty to be aware of such harms and mitigate if not eliminate them. The two most common harms caused by such unfairness in machine learning are allocative harm and representational harm.

Allocative Harm

Allocative harm is concerned with the idea of allocating resources. Machine learning models have the potential to serve all groups of people equally regardless of their gender, race, and religion. When a model favors one group over another, it becomes harmful to society. Some examples of allocative harm are as follows:

- Approving a loan at a higher rate to people of a certain race.
- Assigning more budget to government programs based on management's ethnicity.
- Making positive hiring recommendations only to people of a certain gender.

Imagine an AI model that predicts a candidate's ability to succeed at a given job. It is possible that a model had learned to discriminate against candidates based on their gender or skin color instead of job experiences and educational background.

Representational Harm

Representational harm is concerned with the idea of how reality is presented and hence shaped by a machine learning model. Some examples of representational harm are:

- Recommendation engine showing harmful stereotypes of a certain ethnic group.
- Represent certain groups as less likely to become CEOs.

Summary

While the effect of allocative harm is usually immediate, representational harm may not be realized until years or decades down the road. In either case, machine learning engineers must be cognizant of these harmful pitfalls and reduce if not eliminate them whenever possible.

Back to Table of Contents

Ask the Expert: Brandeis Marshall on Recognizing Algorithmic Harms

In this video, Dr. Marshall explains how we can mitigate algorithmic harms, a process which begins by reflecting on the potential for algorithmic harms before a product is even developed. Considering whether a technology is even needed and who may be affected by it is the first step in producing algorithmic fairness.

Note: The job title listed below was held by our expert at the time of this interview.

Brandeis Marshall
Founder and CEO, DataedX Group

Brandeis Marshall is founder and CEO of DataedX Group, LLC. DataedX provides learning and development training to help educators, scholars and practitioners humanize their data practices. Dr. Marshall is the author of "Data Conscience: Algorithmic Siege on our Humanity" (Wiley, 2022). She speaks, writes, and consults on how to move slower and build better human-centered tech by highlighting the impact of data practices on technology and society.

Dr. Marshall has been a Stanford PACS Practitioner Fellow and Partner Research Fellow at Siegel Family Endowment. She has served as a tenure-track faculty member at Purdue University and Spelman College. Dr. Marshall's research work in data education and data science has been supported by the National Science Foundation and philanthropy organizations. She holds a Ph.D. and Master of Science in Computer Science from Rensselaer Polytechnic Institute as well as a Bachelor of Science in Computer Science from the University of Rochester.

Question

How do you recognize and mitigate algorithmic harms?

Video Transcript

There's a couple of different theories on how do we mitigate algorithmic harms. The first one that I appreciate is talking to communities who are going to be most impacted by the algorithms themselves. And this tends to be black and brown and

indigenous communities. Because we are the first communities that are not considered. We're not part of the main demographic when it comes to the use of these algorithms and systems and platforms. Having conversations, building relationships with these communities in order to make sure that they are testing out these algorithms and processes before they are deployed and scaled. That's number one. The second type of thought process around mitigating harms is to think through whether or not the technology is providing a good to society or is it a technology that is just there to optimize something. Not really having a direction and really being retrospective about whether or not the technologies being birthed in the first place and being deployed needs to happen. That's the second thought of how do we mitigate is sometimes we just don't need the technology. The last one is really a consideration kind of in-between. There are some technologies that we're not sure if they are going to be beneficial to communities that have been historically harmed. We're not sure quite if these technologies are going to work out well. Are they really optimizing or are they just creating more friction? This is where you really need to have a great AB testing. There's a big push in the tech community in order to get as much information and as much of the products out to customers. But I think there needs to be better AB testing in order to understand whether or not a technology is really a viable, quality option versus just a product, just to say that you've released a product. Mitigating the actual harms is asking a bunch of questions. You have to ask questions from the beginning of why. You have to ask questions about how you're doing things. You have to ask questions about what you are doing. All of those questions need to come from different stakeholders, from the software developers to the PR team, to the leadership of an organization. They need to all be asking questions about where's this data coming from? What is being mitigated? How is that impacting and reverberating to other communities and to just other services that the company might have. Hopefully, we'll have less products that are just out and about and more products that are actually solving a problem.

Back to Table of Contents

Watch: Addressing Algorithmic Fairness

Tackling algorithmic fairness concerns is an emerging field that requires deeper study. Currently, the methods used are relatively experimental. Mr. D'Alessandro gives some examples of mitigation techniques for fairness issues you might discover in your machine learning models.

Video Transcript

In this video, I'm going to discuss some mitigation techniques for fairness concerns we might discover in our machine learning models. Mitigating fairness issues is a fairly sophisticated topic that really requires deeper and focused study. This is an emerging field, so a lot of the methods are fairly experimental and there isn't a lot of published work that shows the efficacy of these methods beyond academic studies. On top of that, many of the methods that have been introduced haven't been integrated into standard software packages like scikit-learn This makes mitigation strategies more expensive from a developmental cost perspective. Nonetheless, there are a few straightforward techniques we can integrate into the standard model development process that should go a long way towards creating more fair models. I'm going to walk through different stages of the model development lifecycle and discuss how we might incorporate fairness into the process. Starting with problem formulation, the first thing we want to do is define a fairness goal. Fairness can be defined and interpreted in several ways, and the different fairness definitions often compete with each other. Determining which is best is something you'll have to discuss with your peers. You'll want to be as inclusive as possible, too. This means consulting with technical people, but also product marketing, legal, and others. If possible, it is also good to get points of view from customers. Ultimately, you want to incorporate what customers think is fair into your consideration. People from different communities have different lived experiences and values, so you'll want to cover as diverse a set of customers as possible. As an example, with the COMPAS model that predicts criminal recidivism, the makers of the model anchored fairness from the point of view of their clients, which was the criminal justice system. They noted that because accuracy was similar across groups, that the system was fair from their point of view of their clients. Their clients were more concerned with accidentally releasing a reoffender, which

could be harmful to the public as well as to their political careers. But if we took the defendant's point of view, which ProPublica did, false positive parity is the basis for fairness. In this basis, we care more about withholding freedom from people who are ultimately innocent. This isn't a common example you'll likely encounter in traditional business applications, though. But it does underscore the need to be careful about whose fairness you are solving for and to state your fairness goals upfront. Another part of problem formulation that requires a fairness lens is in creating the label. A lot of problems require us to define labels using subjective judgment. Let's assume we want to build a model to predict whether a job applicant will be a good employee. This is something that might be used to rank resumes for interview prioritization. A natural label would be whether or not someone was a great employee. This makes sense at a high level, but think about how you would measure this. We could use performance ratings or prior promotions as our label here. Performance ratings might be defensible, but what if a company has a history of favoring certain types of demographic groups in its ratings? Also, if you use promotion history, what if the firm has a culture that isn't favorable to some groups, such as working parents? In either case, the label could encode biased decisions or corporate cultural norms that the company regularly practices. Once we encode a bias into a model, we perpetuate it at scale, which is exactly the type of thing we're looking to avoid here. Let's move on to the data preparation phase. One common source of algorithmic bias is when individual features are highly correlated with a protected class attribute. Probably one of the most classic cases of this is a person's zip code. Many US zip codes are a good proxy for an individual's race or ethnicity. So using zip code in a model is almost as effective as using race. Going back to the hiring example, there was a public case where a large e-commerce company had developed an algorithmic approach to screening resumes for the best candidates. They ended up canceling the project because they could not design it in a way that was unbiased towards women. One of the problems is that the input features were derived from resume text, and often such text is indicative of a person's gender. There are obvious ways this can happen, such as someone indicated that they played certain gendered sports or were part of certain gendered professional and social clubs. There are also subtle ways. Researchers have found slight differences in the type of language applicants of different genders use in their writing. When you couple gendered features with labels where men are overrepresented, you get a model that mostly is predicting the gender instead of

what they're trying to do, which is predict a high quality applicant. In industries like financial services and insurance, protected class attributes and proxies of them, like zip code, are explicitly forbidden from use in models. If you are building a model where such regulations aren't in place, it would be up to you as an algorithm designer to avoid using such features. The final point I'm going to cover on this topic is effectively using protected class data if you actually have it. Probably the most important use of such data is to run fairness evaluations of your model. First, starting from your fairness objective, you would evaluate your model against the fairness metrics that are appropriate given your basis for fairness and for the different protected class attributes that are available to you. Many companies make a point to not collect this type of information, either because it is explicitly forbidden or the privacy and legal risks are too large to want to collect it. This is probably the biggest challenge for scaling algorithmic fairness across industries. Provably meeting fairness requirements requires evaluation against these protected class attributes. These attributes are often not collected in order to preserve people's privacy and to protect people from explicitly discriminatory practices. But algorithmic bias often happens unintentionally, and the same efforts taken to prevent explicit discrimination make it challenging to detect implicit or accidental discrimination. There is no easy and fast solution to this problem. Solving it will require new industry, legal, and political standards to be created. The good news, though, is that many companies are working with regulators to work out effective solutions. The bad news is this likely won't resolve quickly. In the meantime, one of the best safeguards is to at least be committed to building robust models that meet their true predictive objectives. As I described earlier, model developers can still examine their labels, features, and data representation for potential discriminatory biases, even if these biases can't be empirically measured.

Back to Table of Contents

Ask the Expert: Brandeis Marshall on Addressing Bias During Model Development

You should be cautious of your model during development to ensure you aren't being biased. In these videos, Dr. Marshall lists some considerations to address bias while building your model as well as how to mitigate it during the early phases of model development.

Note: The job title listed below was held by our expert at the time of this interview.

Brandeis Marshall
Founder and CEO, DataedX Group

Brandeis Marshall is founder and CEO of DataedX Group, LLC. DataedX provides learning and development training to help educators, scholars and practitioners humanize their data practices. Dr. Marshall is the author of "Data Conscience: Algorithmic Siege on our Humanity" (Wiley, 2022). She speaks, writes, and consults on how to move slower and build better human-centered tech by highlighting the impact of data practices on technology and society.

Dr. Marshall has been a Stanford PACS Practitioner Fellow and Partner Research Fellow at Siegel Family Endowment. She has served as a tenure-track faculty member at Purdue University and Spelman College. Dr. Marshall's research work in data education and data science has been supported by the National Science Foundation and philanthropy organizations. She holds a Ph.D. and Master of Science in Computer Science from Rensselaer Polytechnic Institute as well as a Bachelor of Science in Computer Science from the University of Rochester.

Question 1

What are some considerations to address bias in model development?

Video Transcript

When it comes to bias, I think we have to first break down what type of bias we mean. The first type of bias is really about statistical error. That are things that can be fixed. You've added an extra number at the end of something. You've done some things that

just were unexpected by some type of numerical manipulation. That's one type of error. The other type of error is really the societal type of error. That is what is a bit more difficult in order to extract out of a model. In fact, it's so hard that it's not really possible at this point. Because there's so much happening within the data itself. There's people's perceptions and how they view the world that's embedded in every data set. One way in order to mitigate the bias that happens on a societal end is really to do, first round is just descriptive statistics. That is understanding what is the sparseness and the denseness of the data set. Understanding what is the range of values or the range of categories inside of your data set, and understanding how much data is missing, how much data has possibly been interpreted. The first line of defense is understanding your data set so then you can now contextualize it. Understanding its limitations, as well as the power that could potentially have in your model.

Question 2

How should mitigating bias affect your data prep and model selection?

Video Transcript

When it comes to mitigating the bias within a data set, one of the hallmarks that is very important to use is to think through who is represented within that particular data set, especially if it's actually people. What are the demographics of those individuals and where are they placed in your data set? If you were to map them on a 2D or 3D graph, what would happen to those people who happened to be from different demographic groups? If all the people from different demographic groups are somewhere not in the mainstream or the norm of that data set, you then want to make certain adjustments to whether or not you are going to use a particular statistical model, like linear regression, or you're going to decide to use a different model altogether. Or even perhaps you're going to want to actually use different models for different parts of the data set. This helps to ensure that each of the data elements are actually evaluated according to their actual characteristics, rather than trying to generalize and essentially make everyone the same when everything in the data set should not be treated the same inside of your algorithms.

Back to Table of Contents

Algorithmic Accountability

Model owners and developers are accountable for the decisions that their machine learning systems make. In this video, we'll explore methods for model level transparency as well as techniques that are frequently used for example-based model transparency. Mr. D'Alessandro also introduces a tool called the partial dependency plot.

Video Transcript

One topic that goes hand in hand with algorithmic fairness is algorithmic accountability. Algorithmic accountability means model owners and developers are accountable for the decisions that their machine learning systems make. This should cover all impacts of such decision making, from predictive performance, system reliability, to algorithmic discrimination. One way we can achieve algorithmic accountability is to create transparency around how our models are performing. Understanding how a model maps inputs to predictions can help with several concerns. At a high level, it gives us insight into which features are worth investing in and which should be left out. These insights might also shape how we understand the core problem we are trying to model from a cause and effect perspective. At a micro level, algorithmic transparency can help us understand why specific decisions or predictions were made for specific examples. Now I'm going to cover details on methods for model-level transparency, and introduce modern techniques that are frequently used for example-based model transparency. Model transparency at the global level typically involves feature analysis. We can start by looking at feature importances that we derive from tree-based models. This chart shows the Random Forest feature importance on a data set that predicts customer churn from a phone service based on 11 customer behavior features. This chart shows us that the two most important features are called EQP days and revenue. EQP days is defined as the number of days that a customer has had their phone equipment, and revenue is the average monthly amount the customer pays. It is nice to know what features are important, but the feature importance information offers limited insight into how these features actually impact customer churn. To get more insight, I'll introduce a tool called the partial dependency plot. Once we know that a certain feature is

important, we usually like to know the relationship between that feature and the outcome. The simplest way to do this is to plot just x versus y, where x is the feature in question. Qualitatively, this tells us that as a customer has had their equipment longer, they are more likely to churn. Maybe this makes sense when you think about it. A reasonable hypothesis is that customers that have had their equipment longer will be more likely to want to upgrade. But as one decides to upgrade, they may also consider shopping for a new phone service altogether. The issue with this simple plot we introduced is this idea of confounding. Customers with high or low values of this feature may be intrinsically different. The trend that this plot shows they may be representing the intrinsic differences between customers with low and high values, as opposed to the effect of this feature. A partial dependency plot solved this confounding problem. For us, it does this by using a technique called counterfactual analysis. Factual analysis is just measuring x versus y from the observed data, as we just showed. But counterfactual analysis asks, What would churn have looked like if everyone had had a certain value of this particular feature instead of the one they were observed to have? We use this thinking to create the partial dependency plot. We start by defining a range over the feature instead of the points in that range. For each value in that range, we give all examples in our data the same value, make a prediction on our model, and then take the average of those predictions. The partial dependency plot on the EQP days feature would look like this. We can see that the curve is similar up to a point. For higher values of the EQP days, the partial dependency plot shows lower churn than what is actually observed. Again, this is a reflection of how the model captures the relationship between EQP days and churn once all of the other features have been controlled and accounted for. The following is a set of partial dependency plots on the top four features we observed from the feature importances chart. Again, one way to compare the partial dependency plot with a simple x versus y plot is the partial dependency plot tells us how the model we are using relates features to an actual outcome or a prediction. This is what enables us to get some high-level view of how the model is operating. If we use this to understand individual predictions, we can look up the feature values for a given example and see how sensitive the model is around that range of the feature. In this plot, we also note the little black markers along the x axis of each subplot. This is called a rug plot, and it reflects how much actual data the training data had in the range of x. This is helpful for gaining confidence in the partial dependency plot. When

there is a lot of data, the relationship between x and y is more reliable. On the other hand, when there is less data, we should be skeptical that the actual relationship that has been modeled would actually generalize well. From a model prediction performance perspective, if a particular example had feature values that were in the low-density regions, we would expect the model to be more wrong on average. This is an example of model variance at work on feature values that have low support in the training data. I just referenced an ad hoc way to use partial dependency plots to understand why a model might be making a particular prediction. The idea of instance-level model explanations has been growing interest in the research community, and by instance level, I mean an explanation for a particular example. The two most popular methods I typically see and use today to create instance-level model explanations are called LIME and SHAP, spelled S-H-A-P. These are the general name of the methods, but each one has a python package associated with it. Both of these methods aim to measure how much individual features influence the output of the model and the granularity again of a single example. I'll leave it as a follow-up to study the details of each particular method. Instead, I will show a few examples of using SHAP to create explanations for individual examples. The most straightforward way to do this is to use what they call the waterfall plot within the SHAP python package. This chart here shows exactly that. We'll read this from the bottom up. The bottom shows the average churn rate for all examples in our data. The bars then show how each feature for this example is contributing to a change in the prediction where that change is relative to the average churn. Notice how the direction and influence of individual features is consistent with our feature importance and partial dependency plots. This example has a high-churn risk that can be attributed to high values in the revenue and EQP days feature. Here's an explanation for a different example. This example is very different; namely, the churn risk is much lower than in the previous. We can see that the EQP days is still influential, but the value here brings the churn risk down instead of up. We can also see some influence from the feature outcalls, which, again, brings the overall risk down. In this lecture, we showed three different ways to gain explanations and transparency into how a model makes predictions. We showed two global methods, which were looking at feature importance to get relative sense of what features matter, and also the partial dependency plots, which give us more granular detail on how a feature relates to the particular outcome. The third method was the instance-level explanation called SHAP, which provided explanations

for individual examples showing how the values of the features at that example changed the prediction of that example. The use of these methods and tools will enable you to gain valuable insight from your models, but also create a sense of accountability so that you can be more assured that you are protecting your customers and company from any unintentional biases and discrimination.

Back to Table of Contents

Ask the Expert: Miriam Vogel on AI Governance

What does it mean to govern AI? And what does responsible governance look like for AI? Having AI governance means having the ground rules in place for how you'll decide whether or not to use AI responsibly. There are things to consider such as: Is using AI the best option? Who is in charge? What problems may arise? Who will handle those issues?

In these videos, as Miriam Vogel explains the importance of AI governance and why we need it.

Note: The job title listed below was held by our expert at the time of this interview.

Miriam Vogel
President and CEO, EqualAI

Miriam Vogel is the President and CEO of EqualAI, a nonprofit created to reduce unconscious bias in artificial intelligence (AI) and promote responsible AI governance. Ms. Vogel cohosts a podcast, "In AI We Trust," with the World Economic Forum and serves as Chair to the recently launched National AI Advisory Committee (NAIAC), mandated by Congress to advise the President and White House on AI policy.

Ms. Vogel teaches Technology Law and Policy at Georgetown University Law Center, where she serves as chair of the alumni board. She also serves as a senior advisor to the Center for Democracy and Technology (CDT). Previously, Ms. Vogel served in U.S. government leadership, including positions in the three branches of federal government.

Most recently, Ms. Vogel served as Associate Deputy Attorney General, where she advised the Attorney General and the Deputy Attorney General (DAG) on a broad range of legal, policy and operational issues. Under the direction of DAG Sally Yates, Ms. Vogel led the creation and development of the Implicit Bias Training for Federal Law Enforcement. She also spearheaded the department's intellectual property (IP) efforts to identify and dismantle IP theft domestically and internationally, and she worked with the DAG to manage the multibillion-dollar budgets of the department's divisions; resolve high-level challenges; and represent the department in briefings for

White House, congressional, and GAO staff on policy initiatives and oversight matters.

Ms. Vogel served in the White House in two administrations, most recently as the Acting Director of Justice and Regulatory Affairs. She led the President's Equal Pay Task Force to promote equality in the workplace. Ms. Vogel also advised White House leadership on initiatives ranging from women, LGBT, economic, regulatory and food safety policy to criminal justice matters.

Prior to serving in the Obama administration, Ms. Vogel was Associate General Counsel at Dana-Farber Cancer Institute and practiced entertainment/corporate transactional law at Sheppard Mullin in Los Angeles. She began her legal career as a federal clerk in Denver, Colorado, after graduating from Georgetown University Law Center. Ms. Vogel is also a third-generation alumna from the University of Michigan.

Question 1

Why is there a need for AI governance?

Video Transcript

There is a need for AI governance because we are in the midst of a change that I would say is not just a digital revolution but it's of the seismic shift of an Industrial Revolution. We are moving from a technological society where we deploy AI and other technical capacity, other emerging tech, to support our lives in some critical functions to a life where it will support our work in most critical functions in most of our daily lives and routines and our work and our governance. As a result, it is of critical importance that we have ground rules in place; that we as consumers can know what expectations we can have. How can we build trust for the AI that we are so deeply relying on? As an employee, you want to know what the expectations are for you. How will AI be used to support you? What are the expectations for how you will use it? And without a question if you are in a C-suite, if you're a lawyer, if you're on a board or a position of responsibility where tech is being built under your watch or it's being deployed at large scale, whether you're licensing it, acquiring it, etc., you want to make sure that you have level truth with your own employees, with your fellow board members and C-suite, and with the general public about how you will or will not use your AI systems.

Question 2

What is the state of efforts to govern and regulate AI?

Video Transcript

So many people think that AI is just unregulated, and that's not true. First of all, we have many laws on the books in every country, really, that govern current use of systems and processes, regardless of whether AI is used or not. So you don't get off the hook for a discriminatory or risky action because AI supported your action. What do I mean by that? Well, the EEOC -- the Equal Employment Opportunity Commission -- and the Department of Justice had a historic statement about a year ago where they together came to us and said, "If you are using AI in your employment determinations, in your HR system, you have to understand that our traditional civil rights laws will apply." And the example they used was ADA. It's the law that makes sure that if you have a disability, that you are not discriminated based on that disability; you are a protected class and you deserve the full protection of the law. Meanwhile, we know that AI does not operate with that same equality. It doesn't hear female voices as well as males. It does not see all skin colors equally. As a result, if you're using AI in a way that could impact some of these differences and bring about discrimination as a result, you're on the hook. So back to the example of voice recognition, If you're using voice recognition in some function and somebody with a different dialect for which it was not trained is not heard, that's on you to ensure that they are heard equally. And that's challenging when we're talking about AI because we know the way it stands today, it will not hear all people equally. And it's even more problematic if you're talking about someone with a speech impediment, a disability impairing their speech in any way, because then you're violating the ADA. And some of the U.S. institutions that monitor that have told us specifically, "You will be on the hook for the civil rights laws to the full extent even if it's an AI-based decision and not a human-based decision because at the end of the day, the humans are on the hook for the laws that are governing our country and our society." So in addition to those laws, there are many other laws on the books that are particular to financial institutions, healthcare institutions, and so forth, in the U.S. alone, let alone if you look in Europe. GDPR, the privacy law that governs Europe and companies doing business in Europe, has an article, Article 22, that requires you have a human in the loop. So already today, we know there are many laws across the globe that apply to your AI use.

In addition, we talk with lawmakers all the time at EqualAI and know that they are looking seriously at what additional guardrails need to be put into place when they're talking about governing artificial intelligence. So we know that the EU has an AI Act underway, and it looks like a GDPR type of process. And what I mean by that is that it is a huge mechanism that will touch on most companies that are either in Europe or doing business with Europe and have large consequences. In the case of the EU AI Act, currently they divide up their AI system regulations based on level of risk. So in some cases, if it's deemed to be too risky, it's banned outright from the EU. Those types of systems are social monitoring systems, for example, that we've only seen deployed in certain countries such as China. But there are a lot of AI uses in the high-risk, the second-highest category, where we will expect to see significant regulations. So companies or organizations using AI in employment, infrastructure, education, and so on, that will often fit into the high-risk category based on how it's currently framed in the current drafts that were released as of last December. In addition, there are voluntary frameworks that have already been put out and put in broad use. For instance, the Congress in the U.S. mandated that the Department of Commerce, the NIST -- sorry; their division NIST, the National Institute of Standards and Technology, issue a risk management framework on AI. That was released in January and provides a very comprehensive system for AI governance, for managing risks in terms of both potential harms, discrimination, and ensuring that you have an inclusive AI system. And so that's already available. In fact, on our website, we try to make it even more user friendly by creating an AI impact assessment that is available for anyone's use. And it's based on the AI RMF provided by NIST, and it provides a lot of questions you can ask and interrogate your AI system to ensure it's both compliant with the NIST risk management framework as well as larger societal norms about AI safety and use.

Question 3

Whose responsibility is it to ensure responsible AI?

Video Transcript

Whose role is it to ensure that our AI is trustworthy and responsible? The answer is all of us. We all have a critical role to play, and I mean that with absolute sincerity. Because while, on the one hand, a lot of our work in AI is focused on the C-suite, the executives, the lawyers who certainly have a role to play to ensure there is

accountability, clarity, that there's trust -- that if you have a problem, that they will tend to it; that they will be responsible and welcome that feedback as opposed to reprimanding somebody or brushing it under the rug. So certainly, a significant amount of the responsibility is with the senior executives, the board members, who have that fiduciary responsibility. But there's also the policymakers; that's another constituency we work with. They need to understand where are there guardrails that need to be put in place or clarifications as to what the policy implications are; what are the societal norms to which we'll hold our AI? We also have a role as lawyers; that's my bias as a lawyer. We need to make sure that we reduce liability, we reduce harms, while we're increasing the opportunities that we make possible through AI. But everyone has a role quite literally, because we need to make sure that consumers are educated. We need to make sure that they know what they're buying, that they know what the potential risks are, that they know the use cases for which it was built and can ask questions to make sure that companies are held accountable. We need to make sure that employees understand what their rules of the road are; what is the governance system in place for their AI, to what will they be held accountable? Are they expected to test their AI to make sure that it's safe? What are those tests for whom should their AI be built? I would say the right question is to make sure they're always asking, for whom will this fail? And that's a Cathy O'Neil line that we always think of when we're talking about AI. So it's really a question we all can ask: For whom could this fail? Does this see me? Does this AI system hear me? Does it work for me? And if not, what are we going to do about it?

Back to Table of Contents

Quiz: Check Your Knowledge: Improve Fairness and Accountability

You may take this quiz up to three times.

The full contents of this page cannot be rendered in the course transcript. Please complete this activity in the course.

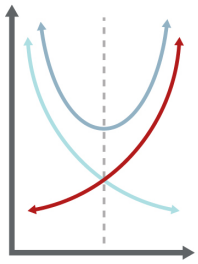
Back to Table of Contents

Module Wrap-up: Improving Fairness and Accountability

Now that you've completed this module, you can see why it's important to be accountable while producing models. Remember that the goal of model owners and developers is to build models that are fair. By doing this, you will reduce potential societal harm. As you continue building future models, keep in mind how you can improve them by being accountable and addressing bias.

Back to Table of Contents

Module Introduction: Deploying Your Model



Once you have trained and tested your model, how do you make it available to stakeholders to solve business problems? You have to deploy and host your model in a production environment. So how do you do that? And what happens after? In this module, you will explore different ways of deploying your model as well as things to monitor once your model is in production.

Back to Table of Contents

Code: Making Your Model Persistent

In this activity, you will see how to make your scikit-learn model persistent so that you can use it in the future to make predictions on new, unseen data without retraining your model every time.

This activity will not be graded.

The full contents of this page cannot be rendered in the course transcript. Log in to the course to view it.

Back to Table of Contents

Ask the Expert: Kathy Xu on Using ML Models in Production

Once a model has been trained and tested, it can be used in a production environment to solve business problems. In this video, Ms. Xu discusses the different ways ML models are used in production.

Note: The job title listed below was held by our expert at the time of this interview.

Kathy Xu
Analytics Extensibility Lead, Pfizer

Kathy (Qingyu) Xu leads a team responsible for extending analytics capabilities across Pfizer. Her team helps create machine learning-powered products (web apps, plugins, and dashboards) for use across the enterprise and provides guidance for fellow data scientists and analysts to optimize their usage of data and analytical technologies that are a part of the enterprise analytics platform.

In particular, Ms. Xu has an interest in MLOPs and industrializing machine learning models. Outside of work, she is an avid art history lover and frequently visits the museums around NYC, such as the Cooper Hewitt and Brooklyn Museum. Ms. Xu received her Master and Bachelor of Science in Statistics from Cornell University.

Question

How to use the model in production?

Video Transcript

I think one of the main challenges that a lot of times we face, especially in the field of data science and data analysis, is once a model is created, how can we actually get that model from the development process into production? There's a lot of different things we can think about. I think one of the main things to think about is understand where's is our data coming from and is there some sort of refresh schedule? Depending on where your data is coming from and the refresh schedule and how often a end user might expect, say, a refresh of the model or new output, you might consider building your model in different ways. What I mean by that is if you have a model that only needs to be retrained once a year, then it might just be enough for a

data scientist or a data analyst, host this model in a particular environment, generate some sort of output, and then send that recommendation to a stakeholder like once a year through like a PowerPoint; I think that might be sufficient. However, if you're looking at a use case where an end user needs to see daily results or a daily refresh or something that's much more frequent than, say, a yearly refresh or a yearly readout, then what I would recommend individuals consider in that creation, in that process from going from development into production, is how do you want to display the end results to you at your end user? A lot of times what our team does is consider, "Hey, maybe we should create some web application or visualization for our end user and then create a model that's hosted in our environment so that it can be automated and refreshed at that" — you know, daily, weekly, or whatever cadence is available. Once that's completed, the actual output of that model gets read into our web application or into our visualization, and then the end user actually sees a live dashboard, for example, with a BI tool, whether that's using Tableau, Spotfire, Power BI, or a visualization that's created through, say, Dash Plotly and other methods, and the end user can actually visualize different types of plots and charts and output directly. So instead of going, say, through like a PowerPoint or the output of your Jupyter Notebook, your model's actually packaged within an environment, automated, and then the end user doesn't even see that part; they only see the end visualization with the cards and the end results that are fed through the visualization. So it's a little bit more — I'll say tangible, especially for users that maybe don't come from a technical background, and they're able to digest that information very readily.

Back to Table of Contents

Read: Deploying, Hosting, and Monitoring Your Model

You just practiced one way to make your trained and tested machine learning model persistent so that you can use it in the future to make predictions on new, unseen data. But in the real world, what do you have to do to make your model available to stakeholders to help them solve business problems? You have to deploy and host your model in a production environment. Deploying a machine learning model can be challenging, but advancements in cloud computing have made the process much easier.

We will outline the considerations for deploying, hosting, and monitoring machine learning models to ensure their ongoing success.

1. Determine the deployment method.

After the model is fully trained, it can be deployed to make predictions on new data through two primary methods: batch inference (or offline inference) and online inference (also known as real time). Batch deployment processes large volumes of data in batches on a recurring schedule and stores the predictions in a database. This information can be provided to stakeholders when needed. Online inference processes data as it arrives in real time.

The choice between the two methods depends on the specific problem requirements, and both come with their own set of advantages and disadvantages.

Deployment Method	Pros	Cons
Batch inference	• Can deploy more complex models with a larger number of inputs • Requires simpler infrastructure requirements and less computational power compared to online inference • Predictions can be analyzed and processed before being seen by stakeholders	• Can result in higher processing latency • Not suitable for applications that require real-time predictions, therefore predictions may not be available for new data
Online inference	• Provides results in real time and on demand • Lower processing latency	• Harder to implement • Cannot handle as complex models as batch inference • More computationally demanding compared to batch inference

Key considerations:

- Your latency requirements
- The complexity of your model
- Specific requirements of your use case

Questions to ask:

- How frequently do you require predictions?
- Do you need results based on individual cases or batches of data?
- What amount of computational power is needed to process the inputs?

2. Choose a hosting environment.

When deploying a machine learning model, one important consideration is where to host it. There are two main options — internal hosting or cloud services — and the decision often depends on the specific needs of the project and organization. Below are some key considerations when choosing where to host your model.

Deployment Method	Pros	Cons
Internal	• Greater control and security for sensitive data • May be more cost effective for larger companies with existing internal infrastructure	• Can be more costly in resources, time, and maintenance • More difficult to scale
Cloud services	• Easily scalable and flexible, with lower maintenance • More cost effective for smaller companies without existing internal infrastructure	• May not be cost effective on large projects • May not meet strict security requirements

Questions to ask:

- **Infrastructure:** Does your organization have the necessary hardware, network, and security protocols to support internal hosting? If not, consider cloud services.
- **Cost:** How much will it cost to host and deploy the model? Internal hosting may require a significant upfront investment, while web services are often charged based on usage.
- **Scalability:** Will your project grow in terms of data volume and user demand? Choose a host that can easily scale up to meet those needs, such as cloud-based web services.
- **Security:** Does your project have specific, strict security requirements? If so, deploying the model on premises can provide better control over the infrastructure and data.

Popular cloud services for deploying and hosting machine learning models

Deploying ML models to the cloud is an increasingly common practice, as it provides easy access to necessary computing power, storage, and network

resources for handling large data volumes and running complex models. Yet there are also potential downsides to consider when weighing your options, including security concerns, high cost, latency, and dependency on the cloud provider.

- **Amazon SageMaker** is a fully managed machine learning service offered by Amazon Web Services (AWS). It provides a complete platform for building, training, and deploying machine learning models.
- **Google Cloud AI Platform** is a suite of tools and services that help you build, train, and deploy machine learning models on Google Cloud.
- **Microsoft Azure Machine Learning** is a cloud-based machine learning platform that enables you to build, deploy, and manage machine learning models.
- **BentoML** is an open-source platform for deploying, managing, and serving machine learning models. It provides a unified interface for packaging and deploying models as production-ready web services
- **Kubeflow** is an open-source platform for deploying and managing machine learning workflows on Kubernetes. It provides a unified interface for managing machine learning pipelines.
- **TensorFlow Serving** is a framework for serving machine learning models using TensorFlow. It provides a flexible architecture for deploying models in production.

3. Deploy your model.

Once you've chosen the deployment method and hosting environment that suits your project, the next step is to package the model along with its dependencies into a deployable format such as a container or a bundle. Containers are popular because they're predictable, reproducible, and easily modifiable, making them ideal for collaboration among engineers.

Industry-standard tools that specialize in preparing for deployment:

- **Docker** is a popular tool for packaging and deploying applications in containers. It can be used to package machine learning models and their

dependencies into a container that can be easily deployed to different environments.
- **Kubernetes** is an open-source container orchestration platform that can be used to manage and scale containerized applications. It can be used to deploy and manage machine learning models packaged in containers.
- **ONNX Runtime** is an open-source runtime engine for deploying models that are compliant with the Open Neural Network Exchange (ONNX) format. It provides high-performance execution of models across different hardware platforms.

Before deploying the packaged model, you should test it to ensure that it performs as expected. This may involve running tests to evaluate the model's accuracy, precision, recall, and other performance metrics. Once you are satisfied with the results of your testing, deploy the packaged model to your target environment that has been determined based on your security, financial, performance, and computational requirements.

Automating deployment

To streamline deployment and testing workflows, some organizations automate the process. Automation ensures the model is tested regularly to maintain its robustness. It can also help scale the model without burdening the team.

4. Monitor for model improvements.

Monitoring a deployed machine learning model is vital to ensure it is performing well and to detect any issues that may arise during production. Monitoring the model's performance, identifying errors, and making necessary adjustments is crucial. Things to monitor for include:

- **Performance deterioration:** Quality degrades over time due to changes in data distribution.
- **Bias or discrimination:** This occurs when the data used to train the model is not representative of the population it is intended to serve.

- **Security risks:** Over time, attackers may be able to manipulate or alter the model's predictions to achieve their goals.
- **Costly revisions:** A model may require costly revisions or even replacement if its performance degrades over time.

Best practices for monitoring model performance

- Constantly evaluate your ML model's performance on real-world data to detect any decrease in accuracy due to changes in the data environment, known as model drift. If model drift occurs, retrain your model with fresh data to improve its accuracy.
- Monitor your deployment pipeline to ensure the model runs smoothly and debug it when necessary.
- Collect feedback from end users to improve the model's performance by identifying areas for improvement and providing valuable insights.

Considering all of the above factors and being methodical when deploying a machine learning model is important to ensure that the model is accurate and reliable and that it delivers the expected value to the business or project for which it is intended.

Back to Table of Contents

Tool: Deploying, Hosting, and Monitoring Your Model

Once you have trained and tested your machine learning model, you have to deploy and host your model in a production environment so as to make your model available to stakeholders to be used to solve business problems.

Download the Tool

Use **Deploying, Hosting, and Monitoring Your Model** tool as a guide to deploy, host, and monitor future models.

This reference tool outlines the considerations for deploying, hosting, and monitoring machine learning models to ensure their ongoing success. Use it as a guide when you are prepared to deploy your machine learning model.

Back to Table of Contents

Ask the Expert: Francesca Lazzeri on Deploying to the Cloud

In this series of videos, Francesca Lazzeri describes what the deployment of a machine learning model entails. She explains the process from algorithm development to application, going through the various deployment methods, expected time frame, and where to anticipate errors. Finally, she describes the different tools, platforms, and architectures most commonly used in the industry.

Note: The job title listed below was held by our expert at the time of this interview.

Francesca Lazzeri
Curriculum Committee Industry Advisor, Microsoft

Dr. Francesca Lazzeri is an experienced scientist and machine learning practitioner with both academic and industry experience as an Adjunct Professor of AI and Machine Learning at Columbia University and Principal Cloud Advocate Manager at Microsoft. She also authored the book "Machine Learning for Time Series Forecasting with Python" (Wiley) and many other publications, including technology journals and conferences. At Microsoft she leads a large international team (across the U.S., Canada, U.K., and Russia) of engineers and cloud AI developer advocates, managing a large portfolio of customers in the research and academic sectors and building intelligent automated solutions on the cloud. Before joining Microsoft, Dr. Lazzeri was a research fellow at Harvard University in the Technology and Operations Management Unit. You can find her on Twitter at @frlazzeri.

Question 1

What does it mean to "deploy your machine learning model"?

Video Transcript

So, machine learning model deployment is simply the process by which a machine learning algorithm is converted into what we call the web service and then into an application, and as a data scientist, as a machine learning predictioner, so we refer to this conversion process as the operationalization, or also deployment of the machine learning algorithm. Operationalizing a machine learning model really means to

transform it into something that is a consumable, so into a consumable service, we said, and as a consequence, then other people can embed it into an existing production environment. So it's really the moment in which you basically transform the model that you build into an AI application. Because thanks to this deployment process, then other people are going to be able to call your model and produce results out of it. So I really like to refer to it as really the moment in which machine learning becomes AI, becomes artificial intelligence.

Question 2

What are the different tools, platforms, and architectures used in the industry?

Video Transcript

In the industry, I would say in general, not just for model deployment, but in general, the main tools that are used for building the machine learning models, training them, and testing them, and then of course, deploying them are like Python is a programming language that's been used a lot, as I said, not just for training, preparing your data, and testing your models, but also for deployment. It is great because, as you know, Python is an open source tool, so it's really supported by the community. There's always this sort of improvement because, as you know, Python has multiple versions, and there are a lot of learnings that we can observe from one version to the other, a lot of learning, and as a consequence, a lot of improvements. So, Python for sure is a tool, and the other, in my opinion, big tool, of course is cloud computing in general. I know that cloud computing can sound like a very broad term. But when I say cloud computing, I mean different typologies of services, such as computer targets that you need in order to attach your experimentation environment to a computer that then you are going to use in order to run your models. But also you have to think in terms of the storage. You need to access different storages in order to store your data, making sure that you have a working end to end data pipeline, meaning that you need to get new and refreshed data at specific internals. Then I think that the other important piece of all of these is also the data visualization. Data visualization is another tool that most of the time, I like to tell to my team of data scientists that they need to think about data visualization as an opportunity to understand the data, so at the beginning of the machine learning development process, but also at the end because data visualization is an extremely important tool that we have in order to

communicate our results. Data visualization, again, we have a lot of different tools that we can think about Power BI, we can think about also Tableau. Also, Python is another great resource for data visualization with their Matplotlib package. I would say always think about not just one single component of the machine learning development process, but think about the end to end process. Then if I have to mention some of the important tools that I have seen the industry use, I would say again, Python, cloud computing, different computer targets, and different tools for data storage, and different tools for data visualizations.

Question 3

How do you deploy a model?

Note: *The functions listed in this video are* `init( )` *and* `run(input_data)`.

Video Transcript

There are many different technologies and tools that you can use. Most of the time we like to say that there are two main functions that you need to use. Most of the time these functions can be written in Python. We refer to these other two functions as the init and run functions. Again, you can write this function in a Python. The init function is more about preparing your data because as you know after the data preparation, the model training, the model testing, validation, you select a model and then you need to deploy it, but the data preparation part has to be there because of course any different typologies of data that you're going to use, any new data that you're going to use, then it needs to be prepared in the same way. This first function that I was referring to is about preparing your data and making sure that basically the input data that you're using is going through the same process. This data, then you need to use them in order to feed your machine learning model. Then there is a second function that is called more like the run function. That is about making sure that your model knows how to read the data, and how to be run in order to produce your results. The combination of these init and run functions actually is really the core part of the model deployment. Most of the time you can use different tools and technologies that helps you to pack these two functions into to a file that is called the pickle file. You have really to think about this file as a folder where you have captured this information that I was referring to earlier; how your data needs to be prepared, and

then how your data is going to be read by the model, and so that the model can produce the predicted results that you want. We refer to this as a pickle file.

Question 4

How long does this process typically take?

Video Transcript

The process, it depends a lot. A lot of the time I use the answer, "it depends", because in machine learning, many things depend on the typology of data that you have and also on the type of problem that you're trying to solve. Generally speaking, I would say that creating just the two functions is just a matter of a few minutes, maximum an hour, because it's just about the data scientist needs to understand how to create these two functions and then you just write them in Python. That part, I would say it's a pretty quick part. However, when then you deploy your model, it depends, of course, on what is the model that you are using. So if it is like I would say, more like a complex model, it can take longer. Again, generally speaking, can be anything from, I would say, 30 minutes up to sometimes also one hour. But again, it depends on how much data you're using and the typology of the model that you are deploying.

Question 5

What are the steps involved in the deployment process?

Video Transcript

There are many different typologies of the steps that you need to follow. I usually like to summarize these three steps in the following way. The first step is about registering your model; the second one is about preparing your models to be deployed, and finally, there is the deployment itself. Registering the model — that is the first step — is really about making sure that there is a sort of a logical container for one or more files that then you are going to use for running your model. For example, if you have a model that is storing multiple files, you can register them as a single model in your machine learning workspace. This is the space that you're going to use as a sort of environment for collaboration and experimentation. After the registration, you can then download or deploy the registered model and receive all the files that were registered there. Then there is the preparation to deploy that — honestly, it's probably the easiest step because it's about specifying the different assets, the usage, and also

the computer target that basically you are going to use. And finally, there is the deployment itself of the model to the computer target. The deployment itself to the computer target is the part that I was mentioning before that is about writing this function, consuming actually is the function that you wrote before so that your model is then deployed into this pickle file. Again, think about this as a folder where you are going to see that there is the model that you created and also all the preprocessing steps that you need to perform on top of your data in order to make sure that your data is ready for your model. These are the three main tools, the three main technologies and steps that you need to follow in order to deploy your model. Again, register your model, prepare to deploy, then deploy the model to your computer target.

Queston 6

How do you detect errors in deployment?

Video Transcript

Most of the time, when you as a data scientist write the init and the run function, you immediately will notice if there is something that is not working well for your scenario. Why? Because the init function, this is the function that loads the model into what we call the global object and this function is run only once. Basically, most of the time when your docker container is starting, so when you need to basically input your data. Immediately there, if there are some problems that part, that function is going to give you an error. Because again, it means basically that your data is not processed in the right way. You will get an error from any different technologies or tools that you're using, because it's a Python function and as a consequence, you will get an error immediately. Another point, another moment where you can get a sense that your deployment is not really working, is at the run function, at the input data moment. This function uses the model to predict a value, because we are in a machine learning scenario, based on the input data. Here, we use also a process that is called the serialization and deserialization. So this means that the inputs and the outputs to the run, typically use what we call a JSON for serialization-deserialization. At this moment also, you will get an error immediately if you didn't get the right data and the model basically is not running in a proper way. Again, this is another moment which you immediately will get an error message or you will see immediately that your model is

not working, and as a consequence, the deployment part didn't work out. This is good, because again, as I mentioned at the beginning that the init and run functions are somehow the first two things that you need to think about when you are ready to deploy your model. As you can understand, you're going to understand immediately, very early staging in the process that your model or the preparation of your data are not working in a good way.

Back to Table of Contents

Ask the Expert: Miriam Vogel on Mitigating Algorithmic Harms After Model Development

There are ways to reduce algorithmic harms when you monitor and maintain the model after deployment. In this video, Ms. Vogel describes what needs to be monitored, maintained, and updated once a model is being used in production.

Note: The job title listed below was held by our expert at the time of this interview.

Miriam Vogel
President and CEO, EqualAI

Miriam Vogel is the President and CEO of EqualAI, a nonprofit created to reduce unconscious bias in artificial intelligence (AI) and promote responsible AI governance. Ms. Vogel cohosts a podcast, "In AI We Trust," with the World Economic Forum and serves as Chair to the recently launched National AI Advisory Committee (NAIAC), mandated by Congress to advise the President and White House on AI policy.

Ms. Vogel teaches Technology Law and Policy at Georgetown University Law Center, where she serves as chair of the alumni board. She also serves as a senior advisor to the Center for Democracy and Technology (CDT). Previously, Ms. Vogel served in U.S. government leadership, including positions in the three branches of federal government.

Most recently, Ms. Vogel served as Associate Deputy Attorney General, where she advised the Attorney General and the Deputy Attorney General (DAG) on a broad range of legal, policy and operational issues. Under the direction of DAG Sally Yates, Ms. Vogel led the creation and development of the Implicit Bias Training for Federal Law Enforcement. She also spearheaded the department's intellectual property (IP) efforts to identify and dismantle IP theft domestically and internationally, and she worked with the DAG to manage the multibillion-dollar budgets of the department's divisions; resolve high-level challenges; and represent the department in briefings for White House, congressional, and GAO staff on policy initiatives and oversight matters.

Ms. Vogel served in the White House in two administrations, most recently as the Acting Director of Justice and Regulatory Affairs. She led the President's Equal Pay Task Force to promote equality in the workplace. Ms. Vogel also advised White House leadership on initiatives ranging from women, LGBT, economic, regulatory and food safety policy to criminal justice matters.

Prior to serving in the Obama administration, Ms. Vogel was Associate General Counsel at Dana-Farber Cancer Institute and practiced entertainment/corporate transactional law at Sheppard Mullin in Los Angeles. She began her legal career as a federal clerk in Denver, Colorado, after graduating from Georgetown University Law Center. Ms. Vogel is also a third-generation alumna from the University of Michigan.

Question

After a model is built, are there mechanisms in place to mitigate algorithmic harms?

Video Transcript

We all know that when you're talking about artificial intelligence, it's going to continue to iterate, it's going to continue to build new patterns and have new answers, and we will have to learn what those are. You have to continually monitor what is happening with your AI system to ensure that it is staying true to your intentions and that it stays true to your values, the company that you're working for, the organization's values, and that it's safe. So are there tools -- this is a growing field. There are algorithmic auditors who are brought in at each stage in the development; it's just really the best way to do it. You want to bring in an expert when you're building the AI system to give you guidance along the way, but you also want to ask that person and continually ask yourself, "Where again will we need another audit? How often?" And that will depend on the AI system. How many patterns is it learning? How quickly is it learning? How pivotal or high stakes are the use for which it's being deployed? And then you want to have a system in place to document, what did I test and when because down the road, other people are likely to be the ones doing that testing, whether it's a different auditing team, whether it's someone who has acquired that AI system and is looking at it with fresh eyes and they'll want to understand what data sets were used, what gaps were there in the population for whom it was tested, what were the use cases for which it was built so we can understand what might be outside of those contexts for which they'll need additional testing. So in addition to

the algorithmic auditors we mentioned -- so one of our senior advisors, for instance, is Cathy O'Neil, who wrote "Weapons of Math Destruction." She has an algorithmic auditing company called Orca and there are several others that are beginning to emerge. There's also tools that people can use. So different companies put out different tools. There's a wide array that are starting to emerge, but one tool that is starting to be more widely deployed is an algorithmic impact assessment. And those are a series of questions that a leadership team or an outside expert helps you come up with to interrogate your AI system and understand where are the risks and the vulnerabilities, where are the opportunities, and where do we need additional testing or oversight. We have one, for example, as I mentioned, on our website. If you look on our home page, you'll see an algorithmic impact assessment that EqualAI offers. Ours is based specifically on the NIST AI risk management framework that was released in January, and it really mostly reflects the questions that they say are best practice to ask. But others have put up their models, too. Microsoft, I think Google; several companies have AIEs, algorithmic impact assessments, that you can make use of as well as other best practices, such as model cards or other ways to routinely systematically document your AI audits and testing.

Back to Table of Contents

Tool: Upload Jupyter Notebooks to GitHub Repository

GitHub is a platform used to host and store code for version control and collaboration. It is free to use as an online directory or storage space for your projects. You may want to store your work from the program. This tool will guide you on how to download assignment files and then upload them to your own private code repository.

Download the Tool

Use this **Upload Jupyter Notebooks to GitHub Repository Tool** to help you with this process.

Back to Table of Contents

Module Wrap-up: Deploying Your Model

In this module, you explored various ways of deploying your model for production as well as things to monitor once your model is available to stakeholders. Remember to use the tool as a reference as you continue building future ML models!

Back to Table of Contents

Lab 8b Overview

In this lab, you will continue implementing the machine learning project plan that you drafted in the written assignment last week.

Using a Jupyter Notebook, you will practice exploratory data analysis techniques that you have learned to investigate your data and plan how to prepare your data to build a modeling data set that is suitable for your predictive problem and model. You will then implement your plan and build a machine learning model for your predictive problem.

When you are done implementing your plan, you will create a portfolio by uploading your project and data set to GitHub.

This three-hour lab session will include:

- **10 minutes** - Icebreaker
- **30 minutes** - Week 8 Overview and Q&A
- **20 minutes** - Breakout Groups: Big-Picture Questions
- **10 minutes -** Class Discussion
- **10 minutes - Break**
- **30 minutes** - Breakout Groups: Lab Assignment Working Session 1
- **15 minutes** - Working Session 1 Debrief
- **30 minutes** - Breakout Groups: Lab Assignment Working Session 2
- **15 minutes** - Working Session 2 Debrief
- **10 minutes -** Concluding Remarks and Survey

By the end of Lab 8, you will:

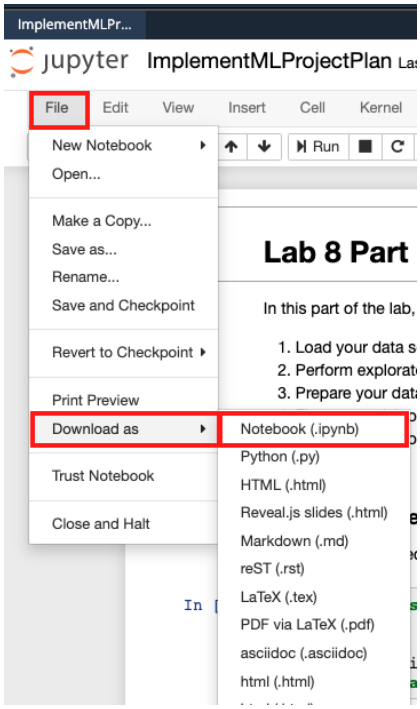
- Load your data set.
- Inspect and analyze the data.
- Prepare your data for your model.
- Fit your model to the training data and evaluate the model's performance.
- Improve the model's performance.
- Create a portfolio to showcase your project.

Back to Table of Contents

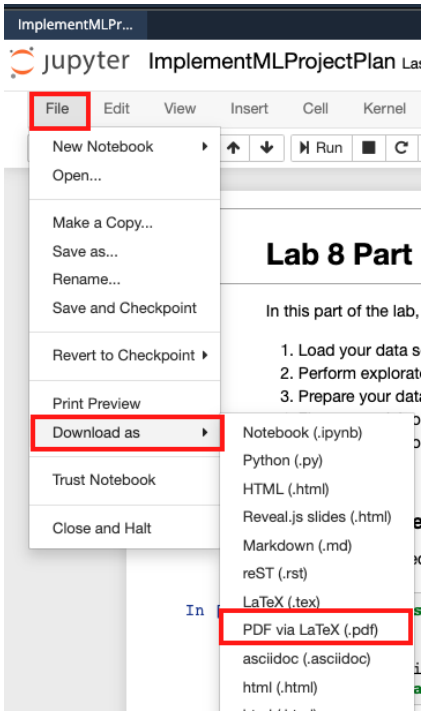
Assignment: Lab 8b Assignment

When you are done implementing your project plan, follow the next few steps to download your project and add it to your portfolio.

1. Download your notebook.
    - Download the `.ipynb` file. Click on `File -> Download As -> Notebook (.ipynb)`.



    - If you would like to save an easy-to-read snapshot of your notebook that contains the output of your code cells, you can download a PDF version of your notebook. After you run all of the code cells in the notebook and produce output, save the notebook and download it as a PDF.



2. Download the data set that you used in your project. Click on the links below to download data sets.
   - **adultData.csv**
   - **airbnbListingsData.csv**
   - **WHR3018Chapter2OnlineData.csv**
   - **bookReviewsData.csv**
3. Follow the instructions and reference the tool **Upload Jupyter Notebooks to GitHub Repository** as you add your files and data set to GitHub.

Back to Table of Contents

Read: Thank You and Farewell

Congratulations on completing your ML Foundations course!

Brian D'Alessandro

Head of Data Science, Social Impact

Instagram

You have just made an important leap forward toward becoming a machine learning practitioner. I hope that you now have the theoretical foundations to understand many key ML algorithms as well as the practical experience to build, optimize, and improve your models. Machine learning is both art and science, and you should now have the foundation to continue your study of the field (the science) while developing your skills as a practitioner (the art). Machine learning is a very important discipline for business and society, and you are on your way to making a meaningful impact on both.

Best of luck in your continued study and practice!

Sincerely,
Brian

P.S. I highly encourage you to download and save any and all course tools, files, and assignments that you have worked on throughout these past weeks. They may prove to be excellent references — as portfolios and concept reminders — as you interview for future jobs.