Microsoft Azure

Responsible AI with Azure Machine Learning

Tools and methods to understand, protect, and control your models

3Cloud

Contents

Organizations across industries have realized the need to invest in artificial intelligence (AI) to remain on pace with market leaders. As innovators are discovering, AI helps us accomplish tasks faster, engage customers, uncover hidden patterns in data to make more informed decisions, and more. It is a powerful lever to help accomplish what's important, from developing new revenue streams to creating measurable efficiencies.

But while many are looking for ways to innovate with AI, most recognize it's a unique technology that must be handled carefully. While AI provides significant benefits, it can also create risks. For example, an ad-targeting engine driven by machine learning (ML) might perpetuate historical inequalities by showing ads for high-income jobs to men more often than women. As another example, an AI model that combs through data to detect financial or tax fraud might wrongly accuse someone if it's not adequately trained and supervised. If AI will become as ubiquitous as cars, we need to subject it to the equivalent of auto safety standards.

"The most critical next step in our pusuit of AI is to agree on an ethical and empathetic framework for its design."

Satya Nadella

CEO, Microsoft

In order for AI to have a positive impact on both businesses and society as a whole, its development and use must be guided by strong ethical frameworks. Each organization has an obligation to ensure they're creating and using AI in a manner that's fair, reliable, secure, and accountable. Doing so opens the door to transformative opportunities while still protecting customers, employees, and organizational values.

At Microsoft, we've been creating and using AI solutions for many years, and we've developed an approach to help us navigate our journey responsibly. This includes establishing guiding principles and developing a system for internal oversight. But while those steps are essential, data scientists and developers need tools to put principles into practice. IT teams are where the rubber meets the road.

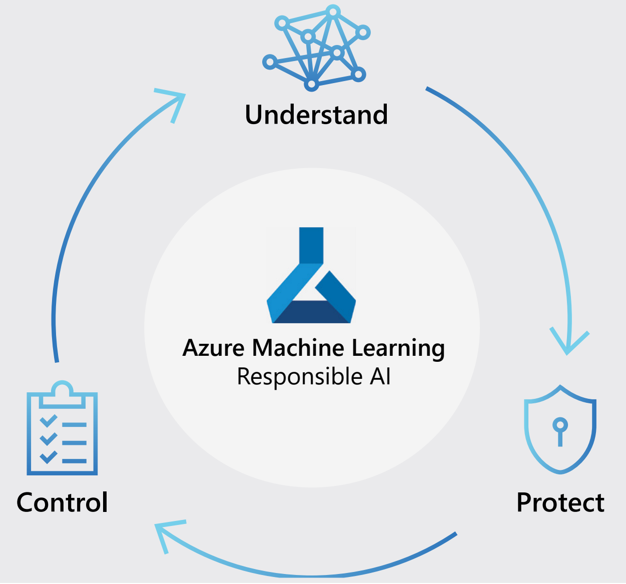
In this paper, we'll share some of the tools and methods that help our IT teams honor our AI principles. The same tools we use internally are available to everyone in Azure Machine Learning, which provides state-of-the-art capabilities to help users understand, protect, and control their data, models, and processes.

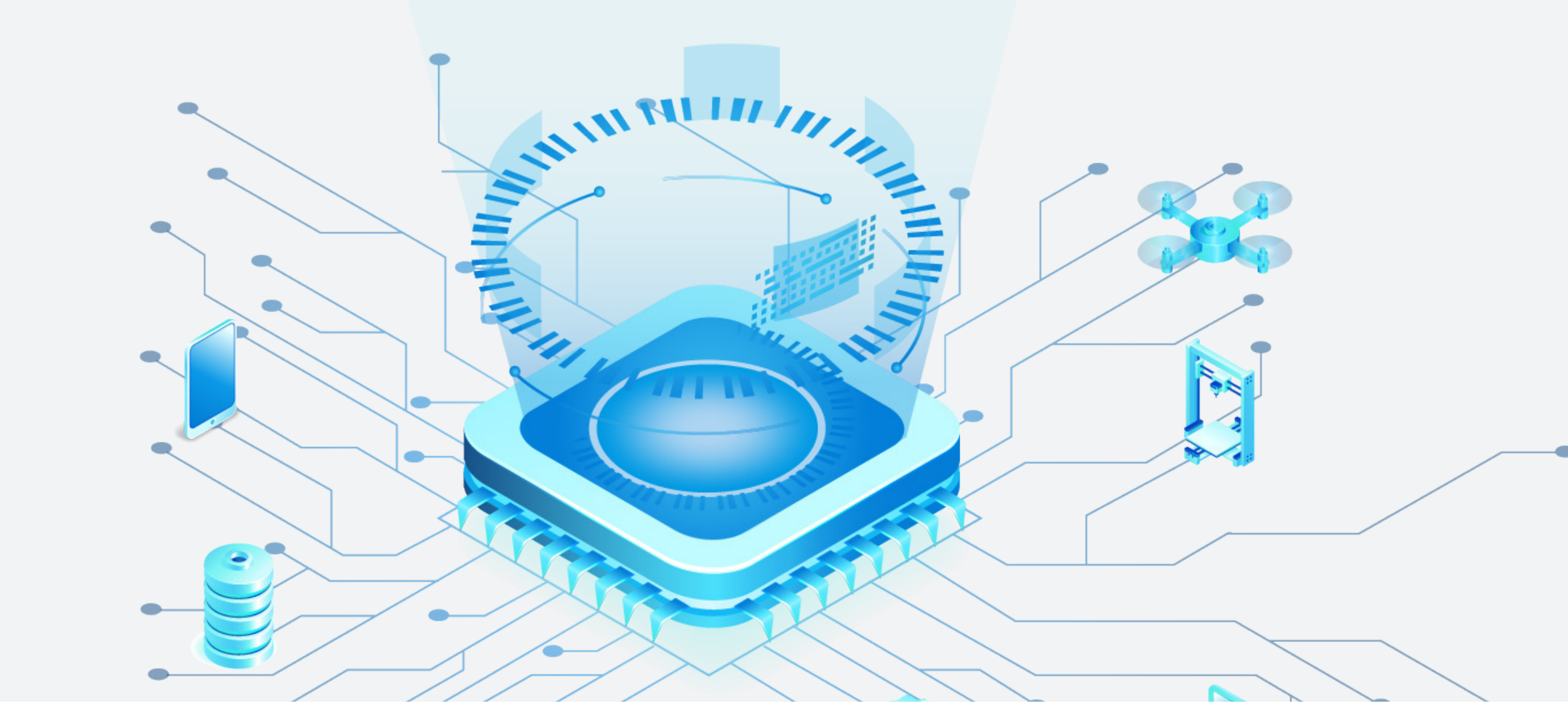
Three Responsible Machine Learning pillars

In Azure Machine Learning (Azure ML), we focus on three categories of responsible ML capabilities, which ladder up to our company-wide AI principles. These categories are:

- **Understand:** Gain visibility into your models, explain model behavior, and detect and mitigate model bias.
- **Protect:** Apply differential privacy techniques to protect sensitive data and prevent leaks. Encrypt data and build models in a secure environment to maintain confidentiality.
- **Control:** Use built-in lineage and audit trail capabilities and document model metadata to meet regulatory requirements.

Azure Machine Learning: Understand, Protect, and Control





Understand

As ML systems become deeply integrated into our daily business processes, we need to be able to trust that they are accurate and fair. Transparency is critical to establishing that trust. Organizations need processes and tools to ensure that data scientists can tune models to meet ethical standards and that end users can understand why certain results are generated. Simply put, when AI systems are used to help make decisions that impact people's lives, people must understand how those decisions were made.

To help users fully understand their ML models, Azure ML provides three capabilities compliment and support each other: interpretability, fairness, and error analysis. Below, we dive into each one and then show how they work together.

Model interpretability

ML systems need to be understandable and explainable so that people can ensure they're working as intended. In fact, model interpretability is a vital first step to achieving other goals like fairness, security, reliability, and accountability.

Imagine a bank uses an ML model to recommend whether or not to give loans to prospective home buyers. They'll need to understand the factors that influence the model's recommendation to grant or refuse a loan, such as the applicant's credit score, age, income, or previous mortgages.

Interpretability is valuable for every role. For data scientists, it's important to understand how a model works and why it produces certain output in order to identify performance issues, resolve them, and validate that model behavior matches objectives. For legal auditors and business leaders, interpretability is necessary to validate that models comply with industry-specific regulations.

In the past, interpretability has been challenging because some ML models (often the most advanced) produce output in a way that's less inherently clear. For these "black box" models, simply publishing the underlying algorithms doesn't provide meaningful transparency. While there will likely continue to be trade-offs between accuracy and interpretability, the field is advancing to make even highly advanced models more explainable.

InterpretML

InterpretML provides several powerful methods to increase model interpretability. The core package provides two key capabilities:

1. Training interpretable glass box models: InterpretML makes it easier to train several types of inherently intelligible models, like decision trees, linear models or Explainable Boosting Machines (EBM). EBMs are as accurate as state-of-the-art techniques like "random forests" and "gradient boosted trees," but unlike those models, they produce lossless explanations and are editable by domain experts.

2. Explaining black box systems: In some scenarios, users might have some pretrained models (often more complex than a glass box model) that they want to understand deeper. For these scenarios, InterpretML's built-in black box explainers provide approximate explanations as to how a model behaves or its predictions. These are useful in cases of pipelines where not all components are directly interpretable. Such techniques are accompanied with data visualization dashboards that help users understand overall model behavior, individual predictions, and compare models to each other. InterpretML supports several types of black box explainers, including "SHapley Additive exPlanations" (SHAP) and "Local Interpretable Model-agnostics Explanations" (LIME).

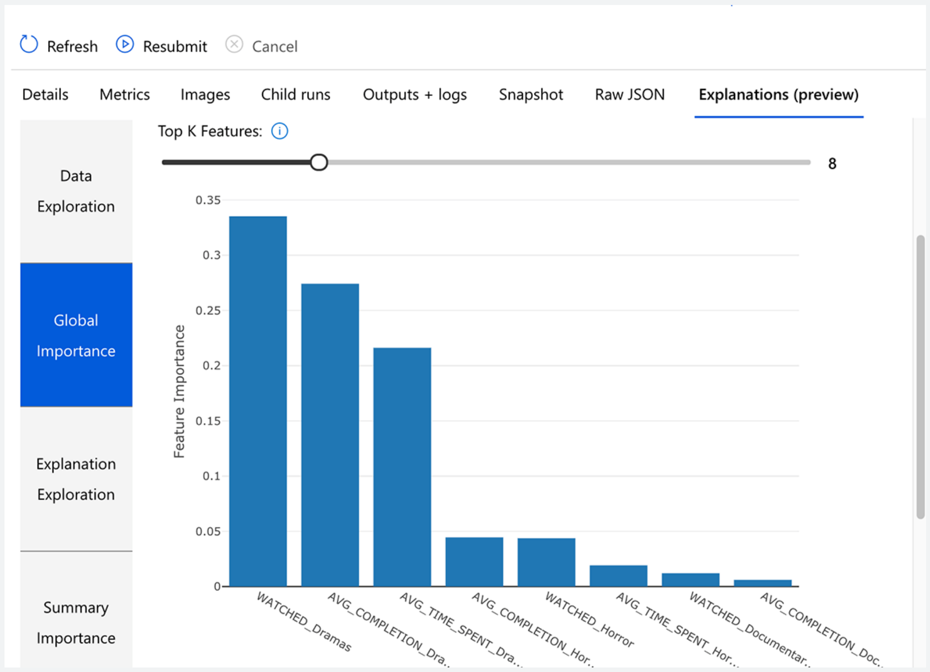
InterpretML can be used within Azure Machine Learning SDK and studio to interpret and explain ML models, including automated ML models.

InterpretML extension: Diverse Counterfactual Explanations (DiCE) for ML classifiers

InterpretML also has another important debugging capability: **counterfactual example analysis,** which computes the most similar data instances that have received different predictions/outcomes. Counterfactual analysis generates explanations for individual outputs or predictions by identifying the smallest change to the input features that would cause the model or system to produce a desired output or prediction.

This is invaluable for data scientists and developers to understand a model and whether it's working the way they want it to. It's also helpful in certain use cases where a customer might need to know what they can do to change a model's decisions next time.

As an example, consider a person who applied for a loan and was rejected by a financial company's loan distribution algorithm. Using DiCE, feature-perturbed versions of the same loan application may show that the applicant would have received the loan, if, for example, the applicant's



InterpretML Diverse Counterfactual Explanations (DiCE) for ML classifiers help data scientists see how small input changes impact model output in 'what-if' scenarios.

income was $10,000 higher. This capability provides what-if explanations for model output and can be a useful complement to other explanation methods, both for end-users and model developers.

InterpretML extension: Interpret-text

The toolkit is another extension of InterpretML specifically for natural language processing (NLP) models. It gathers multiple state-of-the-art explainers in one place, so that data scientists can run experiments across them and efficiently perform comparative analysis for text classification. It also provides an interactive visualization dashboard that helps users uncover insights about which words in their document contribute toward a certain classification prediction.

Model fairness

AI systems should treat everyone fairly and avoid affecting similarly situated groups of people in different ways. It is a common misconception that, because AI is built upon rigorous mathematical and statistical paradigms, it is neutral and, in that sense, "fair." However, AI is the product of human processes and decisions used to create it, the data used to train it, and the environment used to test it. The AI system can exhibit different, sometimes harmful, behaviors as a result of this process. For example, the training data often comes from society and real world, and thus it reflects society's biases and discrimination toward minorities and disadvantaged groups. By manifesting these biases, AI can become a tool in perpetuating historical injustices. This is especially harmful if AI influences the way people access consequential services like employment, housing, insurance, healthcare, or education.

Remember the example of a financial lending institution using an ML-based risk scoring system for loan approvals. If the training data reflects the fact that loan officers have historically favored male borrowers, most approved loans will be for men. Without auditing tools, such fairness issues would persist in the system.

We define whether an AI system is behaving unfairly in terms of its impacts on people—i.e., in terms of fairness-related harms—and not in terms of specific causes, such as societal biases. AI systems can result in a variety of fairness-related harms, including harms involving people's individual experiences with AI systems or the ways that AI systems represent the groups to which they belong. Two very common types of potential harm are:

1. Allocation harms: An AI system extends or withholds opportunities, resources, or information for specific groups. Examples include hiring, school admissions, and lending, where a model might be much better at picking good candidates among a specific group of people than among other groups.
2. Quality-of-service harms: An AI system does not work as well for one group of people as it does for another. For example, a voice recognition system might not work as well for women as it does for men.

To prevent these harms, it's imperative for organizations to be careful about what training data they use and how it is structured. Data scientists should use training datasets that reflect the diversity of society and use tools like Fairlearn to evaluate model fairness.

Fairlearn

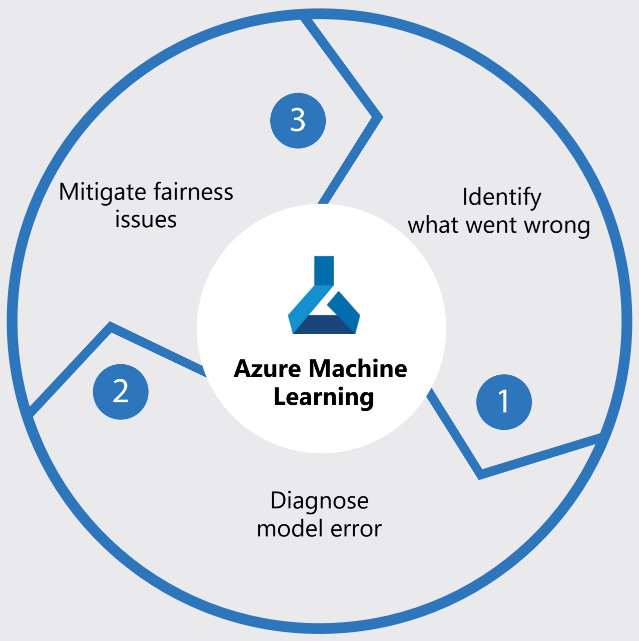
Fairlearn empowers data scientists and developers to assess and improve the fairness of their AI systems. Fairlearn's fairness metrics and interactive dashboard can help with assessing which groups of people might be negatively impacted by a model, while Fairlearn's unfairness mitigation algorithms can help with mitigating unfairness in classification and regression models.

InterpretML

How does my model choose whoes loan application to accept or reject

Fairlearn

Does my model negatively Impact certain groups of people to accept



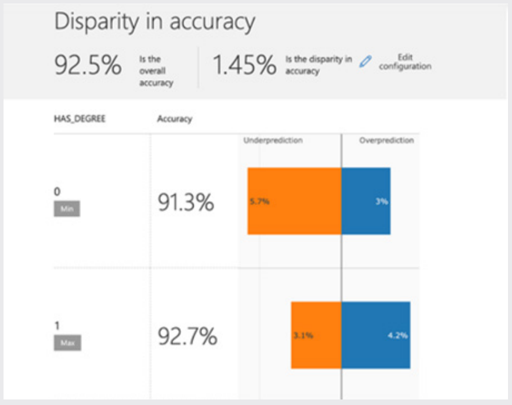
In the Fairlearn open-source package, fairness is conceptualized though an approach known as group fairness, which asks: Which groups of individuals are at risk for experiencing harms? The relevant groups, also known as subpopulations, are those defined in terms of sensitive attributes such as race, sex, age, or disability status.

The open-source Python package consists of the following components:

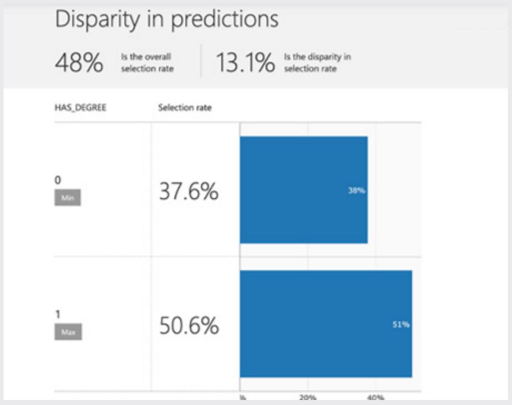
1. The fairness metrics API will enable users to assess the model and measure its behavior when it comes to fairness. There is an interactive dashboard (built within Responsible-AI-Widgets) enabling users to visually assess how a model's predictions impact different groups. They can select different features (e.g., age, gender, ethnicity, disability status, etc.) to assess according to different metrics (e.g. accuracy, precision, selection rate, etc.). Users can also compare multiple models via a chart with axes of fairness and accuracy, allowing them to navigate trade-offs and choose a mitigation strategy appropriate for their use case.
2. Then, Fairlearn provides state-of-the-art mitigation algorithms to mitigate the observed model fairness issues and support classification and regression scenarios. Using the model comparison capability of the assessment dashboard, users can compare the mitigated models and select the one that best fits their needs.

The Azure Machine Learning [Fairness SDK] integrates Fairlearn within Azure Machine Learning. To learn more, check out these [sample notebooks].

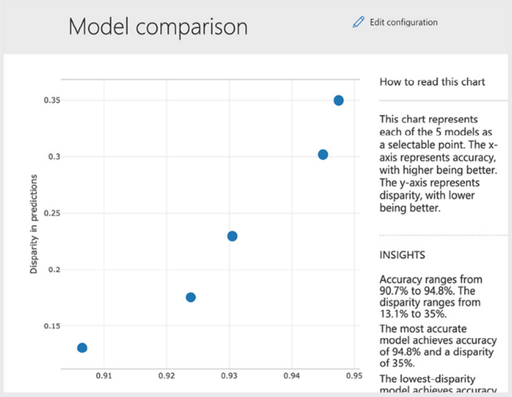
Analyzing model accuracy and fairness with Fairlearn



Investigating HAS_DEGREE as a sensitive feature, we see that accuracy is similar for those with and without degrees.



However, the model is far more likely to recommend interviewing a degree holder.



How to read this chart

This chart represents each of the 5 models as a selectable point. The x-axis represents accuracy, with higher being better. The y-axis represents disparity, with lower being better.

INSIGHTS

Accuracy ranges from 90.7% to 94.8%. The disparity ranges from 13.1% to 35%.

The most accurate model achieves accuracy of 94.8% and a disparity of 35%.

The lowest-disparity model achieves accuracy

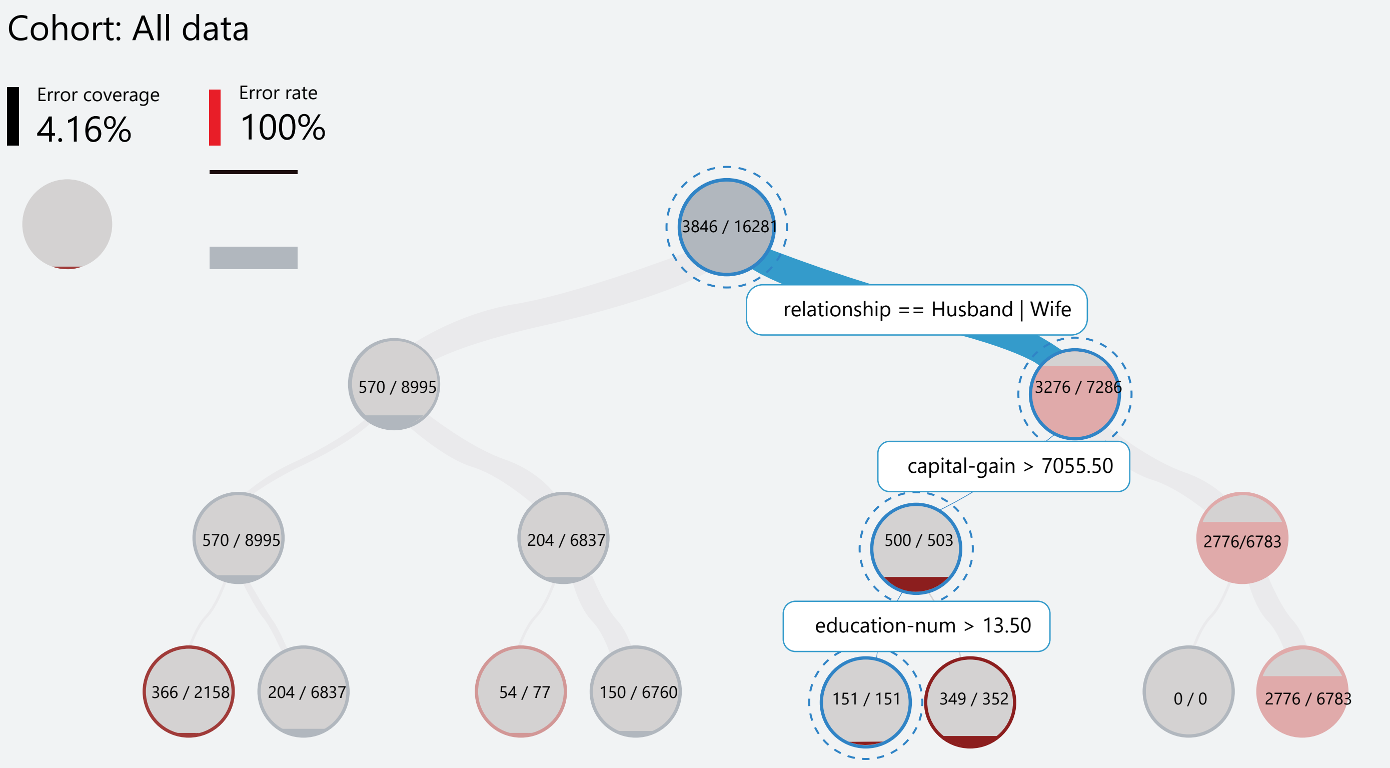
Compare multiple models in order to navigate trade-offs between accuracy and fairness.

Note that while Fairlearn identifies quantitative metrics to assess fairness, developers must also perform their own qualitative analysis to evaluate model fairness. Ultimately, it's up to the humans building ML models to make trade-offs appropriate to their scenario.

Error Analysis

ML teams who deploy models in the real world often face the challenges of conducting rigorous performance evaluation and testing for models. How often do we read claims such as "Model X is 90% on a given benchmark." and wonder what this claim means for practical usage of the model? In practice, teams are well aware that model accuracy may not be uniform across subgroups of data and that there might be input conditions for which the model fails more often. Often, such failures may cause direct consequences related to lack of reliability and safety, unfairness, or more broadly, lack of trust in machine learning altogether. For instance, when a traffic sign detector does not operate well in certain daylight conditions or for unexpected inputs, even though the overall accuracy of the model may be high, it is still important for the development team to know ahead of time about the fact that the model may not be as reliable in such situations.

Visualizing high error rate cohorts with the Error Analysis toolkit



The Error Analysis toolkit makes troubleshooting much easier with two key capabilities:

1. Identify cohorts for which the model is less accurate: To identify cohorts with a higher error rate than the overall benchmark, the toolkit provides visualizations that show how the error rate is distributed across different dimensions (e.g. age, gender, education level, daylight conditions, precipitation, etc.).
2. Diagnose root causes of errors: Then, the toolkit provides interpretability capabilities to diagnose the root causes of the errors so they can be rectified. Built-in capabilities and integration with InterpretML help users understand the most impactful factors responsible for erroneous predictions.

Ultimately, Error Analysis helps data scientists and developers create models that are both more reliable and more fair. It's not only valuable when creating new models, but also when updating and testing existing models.

Combining interpretability, fairness, and error analysis

InterpretML, Fairlearn, and Error Analysis are a family of tools that are meant to work together. As an example, consider how someone could use all three in the scenario where an ML model recommends which loan applications to accept or reject.

Let's say they start in Fairlearn. They can choose to evaluate any sensitive groups by different metrics. In this case, they start by comparing how the model works for men, women, and non-binary groups on the metric of accuracy, and discover that the model is more accurate for women than the other two groups. But then they choose to evaluate the model on the "demographic parity" or "selection rate" metric, discovering that the model recommends loans for women far less often than it does for the other groups.

To better understand why this is happening, they go to InterpretML. There, they can see which data features have the most impact on the model's recommendations. They

SAS

Reducing fraud with Azure Machine Learning

"Azure Machine Learning helps us build AI responsibly and build trust with our customers. Using the interpretability capabilities in the fraud detection efforts for our loyalty program, we can understand models better, identify genuine cases of fraud, and reduce the possibility of erroneous results."

Daniel Engberg

Head of Data Analytics and Artificial Intelligence, Scandinavian Airlines discover that sex is a significant feature in impacting predictions, which potentially is the cause of the demographic parity disparity observed in the fairness assessment stage. They choose an individual data point to do a “what-if analysis”: they flip a data point from female to male, and the outcome flips from rejected to approved. At this stage, they could also use Error Analysis to see if there are any data cohorts for which the model is less accurate and use InterpretML to find out why.

Next, they go back into Fairlearn to use its mitigation algorithms to re-train their model in a way that mitigates the fairness issue. They compare the new model to the old one in the fairness dashboard and see that the selection rate for the new model is much more equitable.

Protect

Privacy and security are key pillars of trust. They require especially close attention in machine learning projects because data is the lifeblood of ML models, and data may contain personally identifiable information (PII).

For certain use cases, models are trained using large quantities of data about people. Then, while they're operating, they make inferences about people. For example, a healthcare organization might develop a model to predict population-wide health trends based on many individual patients' data. Or a retailer might develop a model that generates targeted ads and discounts based on an individual's past purchases.

Organizations need to consider how to protect confidential information while still creating models that are accurate and valuable. They also need to comply with data protection and privacy laws such as Europe's GDPR or the U.S.'s HIPAA.

There are several techniques that help maintain privacy and security in ML systems.

Differential Privacy and SmartNoise

Differential privacy techniques, developed by Microsoft researchers in collaboration with Harvard's OpenDP initiative, help data scientists train highly accurate models while safeguarding individual privacy.

Differential privacy injects a small amount of statistical "noise" into a dataset to ensure that an ML model's output doesn't noticeably change when one individual's data changes. This prevents bad actors from running infinite queries to eventually infer someone's private information. The noise is significant enough to protect privacy, but small enough that it won't materially impact model accuracy.

Differential privacy also protects data using a "privacy budget." The amount of information revealed from each query is deducted from an overall "privacy budget," and queries are halted when they begin to reveal too much information. This is like a built-in shutoff switch that prevents the system from showing data when it may begin compromising privacy.

Microsoft's open-source platform for differential privacy is called SmartNoise. It gives developers the opportunity to leverage expert differential privacy implementations, review and stress test them, join the community, and make contributions.

Confidential computing: Homomorphic encryption and Microsoft SEAL

Homomorphic encryption enables ML models to securely work with encrypted data in the cloud, whether that's during initial training or post-deployment inferencing.

Traditionally, computations can't be performed on encrypted data. If an organization is storing data on the cloud, they must grant the cloud provider permission to decrypt the data before processing it and then re-encrypt it afterward. Alternatively, they have to download their data from the cloud, decrypt and process it themselves, and then re-upload it.

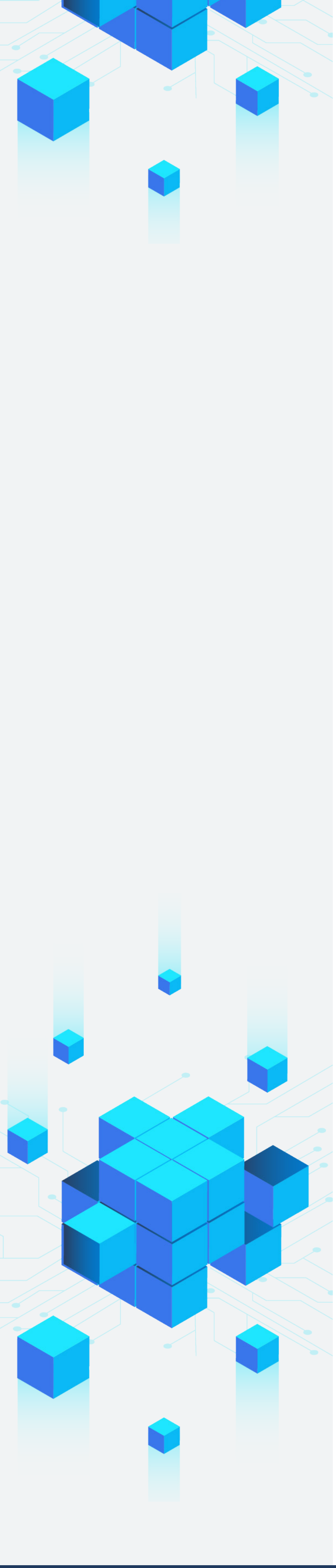
Homomorphic encryption is changing that. It's a technique that makes it possible to perform some types of computation directly on encrypted data. The results of the computations are encrypted and can be revealed only by the owner of the decryption key. Using homomorphic encryption, cloud operators will never have unencrypted access to the data they're storing and computing on. This makes it more secure to train machine learning models in the cloud using encrypted data, or for machine learning models to make inferences related to encrypted data stored on the cloud.

Microsoft SEAL is an open-source homomorphic encryption library that can be applied within Azure Machine Learning using encrypted-inference Python bindings.

Control

Organizations must ensure their ML solutions operate reliably and consistently. This obligation is especially clear in high-stakes use cases, like a manufacturing company predicting when machinery needs maintenance, a healthcare provider using ML to detect abnormalities in X-rays, or an ML model influencing an organization's financial stability. But even in low-stakes scenarios, reliability is key to generating maximum value.

Control and accountability are essential for upholding reliability and accuracy. Control starts in the training phase, where ML systems should be exposed to unusual circumstances and rigorously tested to prevent performance failures down the line. After deployment,



it's equally important for organizations to properly maintain ML systems throughout their lifespan. Maintenance includes continual monitoring to spot issues and periodic retraining to ensure models remain relevant and accurate. Throughout the entire lifecycle, thorough documentation and audit trails keep people accountable for how their systems operate and help organizations meet regulatory requirements.

Azure ML provides many tools to increase control and accountability in ML projects.

MLOps

MLOps is the application of "classical" DevOps principles to data science projects. It involves collaboration between data scientists, ML engineers, software developers, and other IT teams to manage and document the end-to-end ML lifecycle.

MLOps features in Azure Machine Learning helps teams increase control and accountability in two ways:

1. Increase transparency and improve governance through auditable documentation:
    - For each model, Azure ML saves and tracks elements like the ones below in a centralized, shared repository, forming a complete audit trail of the model's history:
        - Code and datasets used in each training and testing iteration
        - Deployment environments
        - Data and model explanations
        - Previous versions of the model
2. Monitor model performance and manage re-training cycles:
    - MLOps tools track model metrics and provide a central log for tracking issues like data drift.
    - Additionally, MLOps provides automation, repeatable workflows, and reusable assets that make it easier for organizations to stay on top of retraining cycles and keep model performance high.

For more information on using MLOps in Azure ML, check out the MLOps Python Git repo and read the MLOps paper.

Additional information about audit trails in Azure ML

Azure Machine Learning provides capabilities to automatically track model lineage and maintain an audit trail of ML assets. Details about each model are captured in a central registry.

- Azure ML Run history stores a snapshot of the code, data, and computes used to train a model.
- Azure ML Model Registry captures all the metadata associated with a model (which experiment trained it, where it is being deployed, if its deployments are healthy).
- Git integration: Azure ML integrates with Git to track the origin of code used for model training.
- Azure ML datasets: Creating a dataset in Azure ML creates a reference to the data's source location along with a copy of its metadata. This helps users track, version, and share data with other users.
- Event Grid: Allows users to act on events in the ML lifecycle such as the completion or failure of training runs, the registration and deployment of models, and the detection of data drift. The events can trigger reactions like sending an email, creating a new ML pipeline, and more.

Datasheets and ABOUT ML

Azure ML can be used to record important details about each ML model, like how and why it was created, using datasheets.

The concept of "datasheets" in ML was inspired by the electronics industry, where products typically come with a sheet of information like operating characteristics, recommended usage, and more.

In machine learning, using datasheets helps data scientists increase transparency and accountability, mitigate unwanted biases, reproduce results more easily, and select more appropriate datasets for their chosen tasks.

Subject of the datasheet	Type of information included
Model	• Intended use • Model architecture • Training data used • Evaluation data used • Performance metrics • Fairness information
Training dataset	• How and why a dataset was created • How the dataset should and shouldn't be used • Potential ethical and legal considerations

Datasheets include the following information:

The "custom tags" capability in Azure ML can be used to implement datasheets today, and over time we will release additional features.

To learn more about documenting machine learning systems, see ABOUT ML, an ongoing multistakeholder initiative led by the Partnership on AI (PAI). The goal of the initiative is to increase ML transparency and accountability by developing, testing, and promulgating best practices for ML documentation.

Azure Machine Learning
Responsible AI

Microsoft's holistic approach to responsible AI

1. **Principles:** We've established guiding principles for how AI should be developed, used, and maintained.
2. **Practices:** We've also established a system for internal oversight that provides guardrails for first and third-party AI solutions. Our AI governance teams are tasked with developing policies and guidance, documenting best practices, helping address issues that arise, and educating employees about responsible AI.
3. **Tools:** Our data scientists and developers use tools and resources that make it easier spot and mitigate potential issues when training and maintaining ML models.

Ready to get started?

From explaining model behavior and mitigating bias to preserving data privacy and tracking model lineage, Azure Machine Learning's state-of-the-art technologies help users develop and maintain ML responsibly. Our offerings will continue to expand and evolve as the field advances.

It is worth noting that tools for data scientists and developers are not be-all-end-all solutions. For long-term success, organizations must use technical tools within a more holistic approach to AI. That approach will be unique for each organization, but there are a few common threads. For instance, in addition to using tools like the ones in this paper, technical teams should establish guidelines,

checklists, and processes that foster ethical practices, rigorous testing, and continual auditing. Organization should have robust systems for governing AI and maintaining accountability. IT teams should be trained in ML-related skills, and all employees should be trained to understand ML solutions and use them appropriately. IT teams should collaborate with their business colleagues to determine the best data and training techniques to use for achieving the desired outcome for each model.

To learn more about developing a holistic approach to responsible AI, explore Microsoft's Responsible AI resources and research.

To start leveraging responsible ML tools today, explore the responsible ML capabilities in Azure Machine Learning.

Additional resources

eBook

Mastering Azure Machine Learning

Explore this free eBook from Packt for hands-on guidance, real examples, and executable code on Azure ML.

eBook

Principles of Data Science

Get a comprehensive beginner's guide to statistical techniques and theory.

Paper

Four Real-Life Machine Learning Use Cases

Dive into four practical end-to-end machine-learning use cases on Azure Databricks.

Demo

Responsible ML Demo

Watch this interactive demo that showcases model interpretability, fairness, and differential privacy in action.

Sample Code

Recommenders Recipe

Get examples and best practices for building recommendation systems.

Sample Code

Computer Vision Recipe

See examples and best practices for building computer vision systems.

Microsoft

3Cloud

Microsoft
Partner

Microsoft

3Cloud is the largest pure-play Azure partner in the ecosystem with unequaled expertise in Azure. We have hundreds of Azure experts, including MVPs and Fast Track Solution Architects, with thousands of hours implementing cutting edge solutions with Microsoft's most advanced technology.

sales@3cloudsolutions.com // 888-88-AZURE // 3cloudsolutions.com