Incorporate Explainability and Fairness Within the AI Platform

Published 16 June 2022 - ID G00738714 - 40 min read

By Analyst(s): Sumit Agarwal

Initiatives: Analytics and Artificial Intelligence for Technical Professionals; Build Trust and Mature D&A Culture

Organizations looking to adopt machine-learning-driven AI solutions for key business functions are exposed to algorithm risk, regulatory scrutiny, bias and lack of transparency. Data and analytics technical professionals must mitigate these pitfalls by implementing AI explainability and fairness.

Overview

Key Findings

- The adoption of artificial intelligence (AI) solutions and dependency on algorithms for decision making with impact on core business processes has surfaced certain risks and regulatory implications, requiring organizations to establish a framework to govern machine learning (ML)-based solutions.

- Increasing dependency on AI models has stimulated substantial academic research, regulations and industry demand. This is evidenced by recurring workshops at leading academic conferences such as the Conference on Neural Information Processing Systems (NeurIPS), the International Conference on Machine Learning (ICML), and the Computer Vision and Pattern Recognition Conference (CVPR); new regulation proposals across the EU, U.S., China and elsewhere; and framing of risk management and audit requirements by the National Institute of Standards and Technology (NIST) and other nongovernment organizations.

- There are several open-source packages and toolkits for explainability analysis and bias detection and mitigation. While these packages do not provide a perfect explanation and bring new implementation challenges, they provide easy access to an environment to experiment with to understand the capabilities and limitations.

- Process and role enhancements are requirements to establish the accountability of an ML model — its usage, decisions and organizational impact.

Recommendations

Data and analytics technical professionals responsible for ML model development and implementation must:

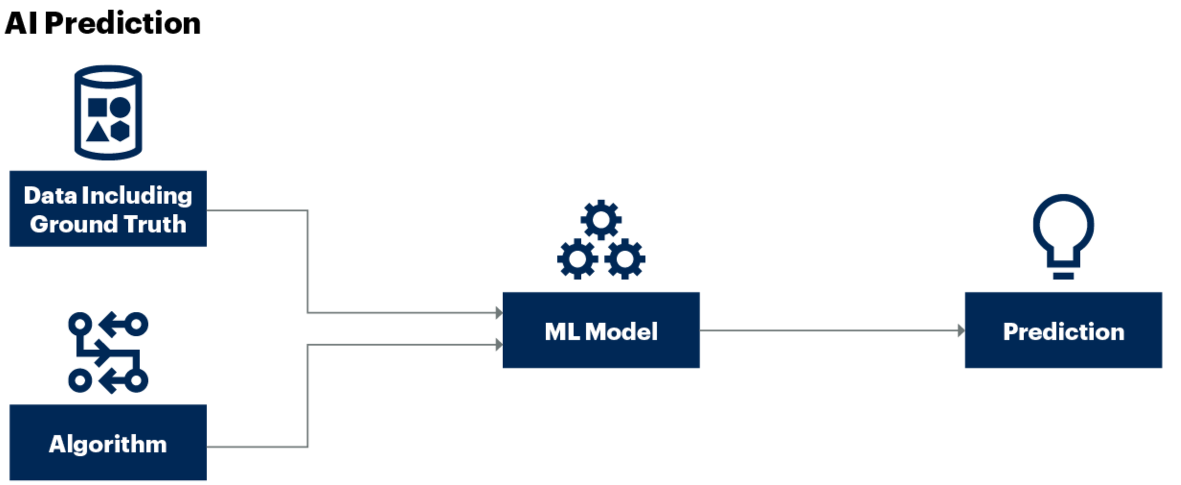
- Evaluate ML explainability toolkits and product features and align them with the use case (regulatory, end-user or customer-centric, or troubleshooting). Instead of focusing on one algorithm or toolkit, embrace the business problem and select a combination of tools for effectively explaining outcomes.
- Embed fairness and explainability functions within the entire model development life cycle. Include change management processes and train technical and nontechnical teams on the tools and methods.
- Establish accountability by defining the role of model owner and engaging the risk, quality assurance, functional and technical teams within the model requirements, development, testing and implementation process.
- Establish fairness definitions and other nonfunctional requirements early, providing balanced guidelines for inference and explanation accuracy, interpretability and execution performance. At the same time, instead of striving for the perfect toolkit, focus on incremental implementation.

Analysis

AI is a transformational technology. The ability to identify and unlock new patterns from a variety of structured and unstructured data creates new business opportunities for organizations, providing innovative services and solutions to customers. Despite this, there are concerns about ethics, fairness, security, privacy, accountability and transparency due to the black-box nature of the prediction process (see Figure 1). These concerns are inhibiting the successful adoption of AI. Without trust and transparency, few organizations are willing to take the leap to apply AI to high-value use cases. Technical professionals who wish to revolutionize mortgage approvals, investment decisions, claim approvals, underwriting decisions, supply chain optimization and grid operations with AI must incorporate AI fairness and explainability.

Download All Graphics in This Material

Figure 1: AI Prediction



Source: Gartner
738714_C

AI is pervasive. People are exposed to services built upon AI algorithmic decision making every day. The recommendation engines inside online movie or shopping portals, and the optimization of traffic signals and weather forecasting applications are just a few of the algorithmic decisions that influence our daily life. These AI-powered services sometimes get it wrong. Email spam filters sometimes tag important emails as spam. Snowstorm predictions of 10 inches of snow are proven wrong by a light dusting of snowfall.

Some mistakes have more serious repercussions, such as racial bias in prediction of repeat offenders, gender bias in selection of candidates for jobs, failure of self-driving cars to identify the object, or an untimely early discharge of a patient from a hospital. These erroneous predictions and decisions have real-world negative impacts. When coupled with the black-box nature of these models, it is understandable why organizations are cautious in implementing these capabilities.

The inherent inability to interpret the behavior of a black-box model increases risk and reduces trust in the usage of ML models. Organizations striving to leverage AI algorithms to improve business efficiency would need to recognize these risks and take actions to mitigate them.

Actions to mitigate the risks must include a focus on model quality and implementation of several best practices across the entire process. Regulatory and government bodies have recognized the implications of bad models. Several countries have defined regulations recently. These regulations provide specific guidance to model development and management:

- **U.S. Federal Reserve SR 11-7:** Guidance on Model Risk Management recognizes "… model risk can lead to financial loss, poor business and strategic decision making, or damage to a banking organization's reputation." It puts the onus on senior management to ensure appropriate model risk mitigation steps and to develop and maintain a strong governance framework.
- **U.S. Equal Credit Opportunity Act**: "…prohibits creditors from discriminating against credit applicants on the basis of race, color, religion, national origin, sex, marital status, age, because an applicant receives income from a public assistance program, or because an applicant has in good faith exercised any right under the Consumer Credit Protection Act."
- **EU General Data Protection Regulation (GDPR):** stipulates that "…the controller shall provide the data subject… meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject."

A combination of regulatory requirements, business needs, customer expectations and the implicit expectations of the data science team's ability to develop good quality models has taken us to a point where the ability to explain the predictions and interpret the models is not an option but a necessity. While most of the model development focus is on predicting the "What," the "Why" behind the prediction is equally important to engender trust in the "What." The decision makers using the technology should have enough information to answer the "Why."

Technical professionals and AI leaders have raised several questions as they work to achieve the above goals:

- How do you understand the functioning of the AI system so you can explain the model to the business experts?
- How do you develop a fair model? How do you measure fairness?
- What are the various tools and techniques to implement model explainability and mitigate bias?

This research integrates the various solution approaches in the following sections:

- Model explainability and fairness tools and frameworks
- Integration within the ML life cycle
- Next steps

Evaluate Machine Learning Explainability and Fairness Toolkits

As AI systems usage increases for enterprise decision making, the requirements to manage related risk becomes critical. Transparency requirements and resolution of bias challenges need increased prioritization and focus. This is because organizations and executives will not implement models they don't trust, or that come with business or regulatory risks. However, model transparency and resolution of bias are nontrivial to implement. In general, building explainability into models has become more complex due to increased use of nonlinear models (such as gradient boosting) and deep learning models in the quest for more accurate predictions. Additionally, large datasets have made it harder to create fair models.

Explainability and fairness may seem like two different concepts. However, explainability helps identify fairness bias in model predictions during the model development process. Because of this interdependency, it is reasonable to analyze the two concepts together. This section provides an overview of the approaches, along with open-source tools and products for model explainability and for identifying and mitigating model bias. This also includes solutions for explainability and fairness provided by various product leaders.

Explainable AI is not just about understanding the prediction. It includes understanding the model as well. This makes it extremely important to understand the available model explainability frameworks along with their core function and scope. Most of the model explainability frameworks can be divided into two broad technical categories:

- Model-agnostic explainability
- Model-specific explainability

Model-Agnostic Explainability

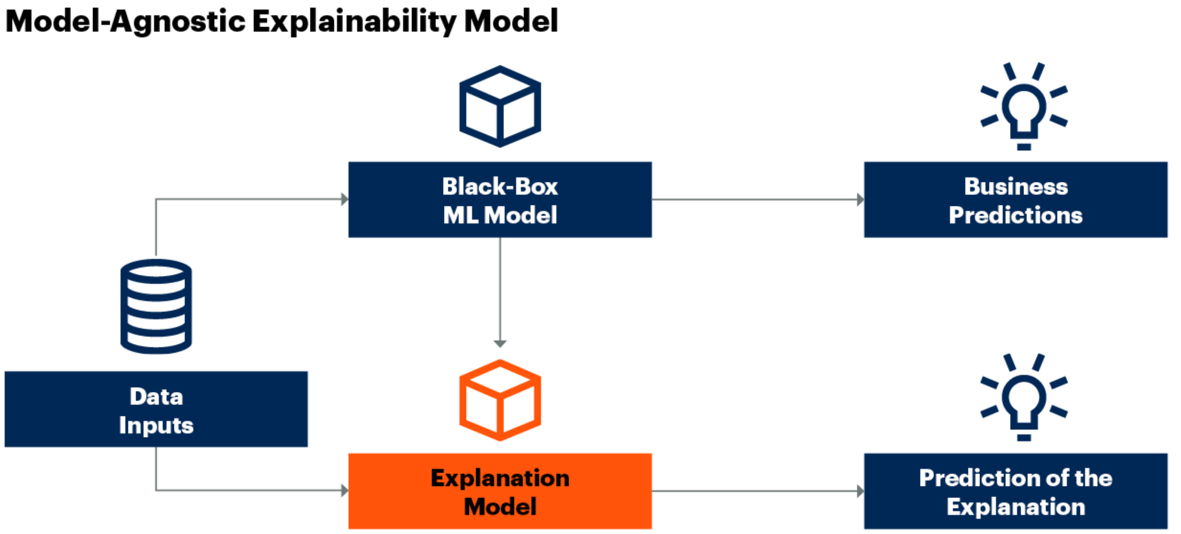
This is also known as post hoc explainability. It provides an easy approach to explainability for models that have already been built. This approach involves development of an explainability model alongside an already trained prediction model. The explainability model uses the same training data as the prediction model. Output from the prediction model is also used to train the explainability model. This is depicted in Figure 2, where the black-box ML model is used for making business predictions, and the explainability model uses the same input data to predict the explanation.

Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin as part of their published research, "Model-Agnostic Interpretability of Machine Learning," [1] identified several benefits of model-agnostic explainability:

- **Model Flexibility:** Models with a very large number of features, or complex neural networks with large numbers of layers (for example, ImageNet and ResNet), or even ensembles of convolutional neural networks (CNNs) and recurrent neural networks (RNNs), may help resolve complex problems with higher accuracy. However, interpretability requirements may constrain data scientists into using simpler methods or structures. Model-agnostic interpretability separates interpretability from the model. This allows the data scientist to focus on model accuracy and not have to worry about the complexity. This also provides explanation flexibility (presenting only the explanations relevant to the use case) and representation flexibility (presenting explanations different from the original dataset; for example, using words to explain image data).

- **Lower Cost to Switch:** The application of explainability to existing models, or changing the algorithm used to explain predictions, would not require any changes to the model itself. This may still require changes to the end-user presentation layer. However, this certainly lowers the cost and system impact to switch from one explanation algorithm to another.

Figure 2: Model-Agnostic Explainability Model



Source: Gartner
738714_C

The key model-agnostic frameworks are:

- Local Interpretable Model-Agnostic Explanations (LIME)
- SHapley Additive exPlanations (SHAP)

Local Interpretable Model-Agnostic Explanations (LIME)

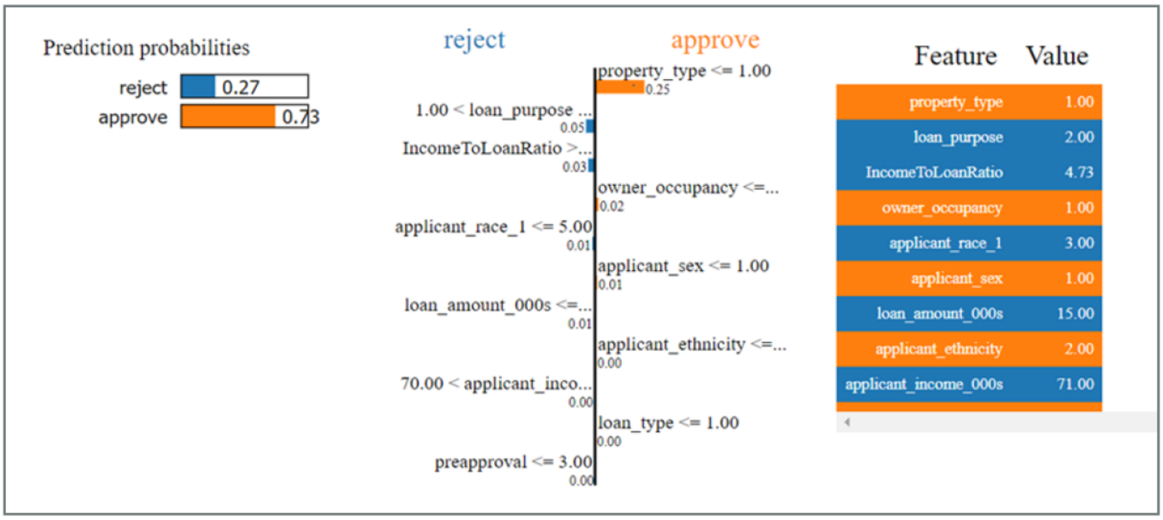
LIME provided a breakthrough by establishing a model-agnostic framework for local interpretation of specific predictions. [2] In 2016, Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin published their research “Why Should I Trust You? Explaining the Predictions of Any Classifier” proposing LIME as “a novel explanation technique that explains the predictions of any classifier in an interpretable and faithful manner by learning an interpretable model locally around the prediction.” [3]

LIME creates a linear, interpretable model locally around the predictions. It samples perturbed instances around a prediction. It then weighs these perturbed data points by their proximity to the original example. These help to train an interpretable model that approximates the model in the vicinity of the prediction. This approach provides a local explanation of the specific prediction.

The screenshot in Figure 3 provides a prediction explanation. This is based on an open dataset that includes mortgage submissions by financial institutions as part of the U.S. Home Mortgage Disclosure Act (HMDA).

Figure 3: Screenshot of LIME Explanation

Screenshot of LIME Explanation



Feature	Value
property_type	1.00
loan_purpose	2.00
IncomeToLoanRatio	4.73
owner_occupancy	1.00
applicant_race_1	3.00
applicant_sex	1.00
loan_amount_000s	15.00
applicant_ethnicity	2.00
applicant_income_000s	71.00

Source: Gartner
738714_C

Gartner

The screenshot in Figure 3 includes prediction probabilities for mortgage approval (left), the contribution of each feature to the prediction (middle), and the list of features and their values (right). The feature explanation shows “property type” with a value of 1 (One-to-four family dwellings other than manufactured housing) [4] as having the biggest positive contribution, while “loan purpose” with a value of 2 (home improvement) has the biggest negative contribution. This can also lead to interpretations, such as that a loan purpose of 1 (home purchase) would increase the approval probability. This would provide actionable insights behind the decision and even possible actions to remedy the decision.

LIME provides easy-to-understand explanations for the prediction. However, LIME provides an approximation. It’s a useful and easy-to-implement Python library for model debugging and even for explaining the results to a stakeholder within the organization. However, the approximation may result in inaccurate results impacting the usage for compliance or business decisions.

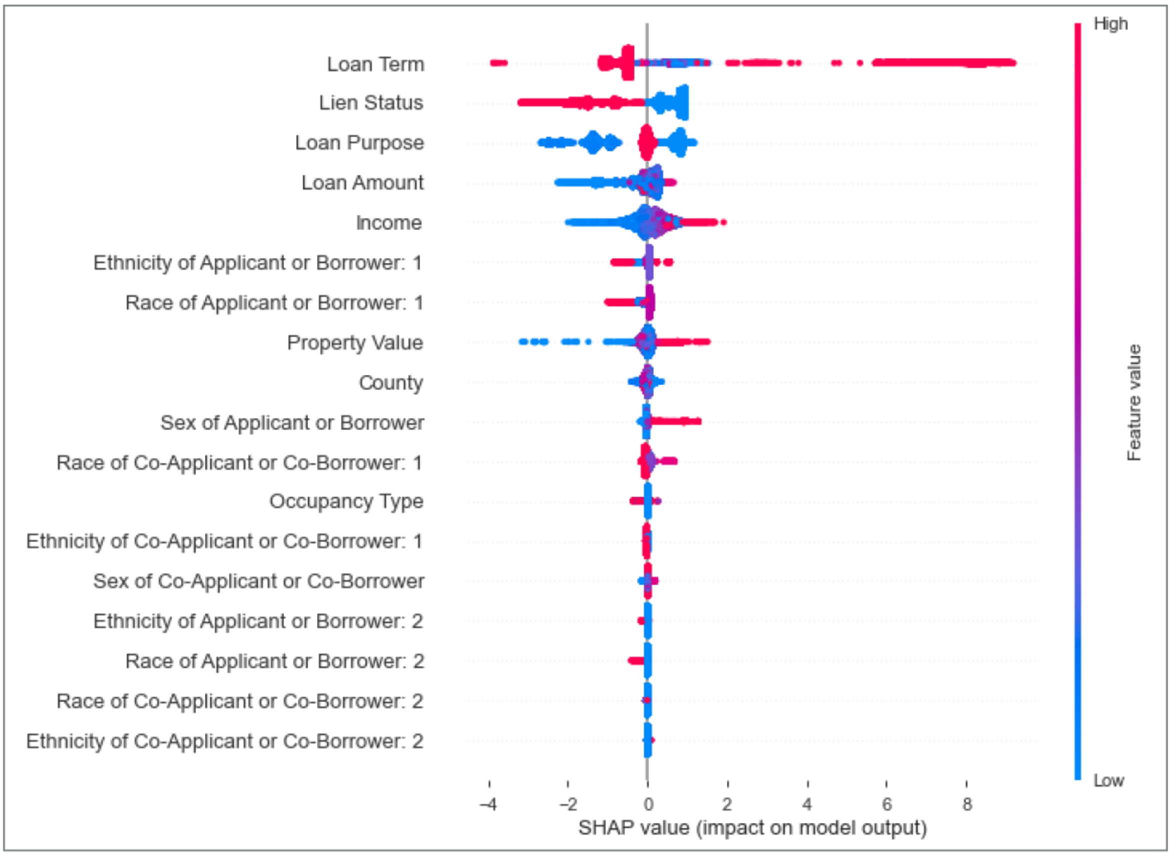
SHapley Additive exPlanations (SHAP)

SHAP is based on Shapley values, a technique used in game theory to determine how much each player in a collaborative game has contributed to its success. [5,6] The Shapley value is the average of all marginal contributions to all possible coalitions. It essentially connects game theory with local explanations.

SHAP provides both global and local interpretability visualizations. At the global level, SHAP provides the significance of the features to the prediction value. The visualizations include a bar chart that displays the features in order of significance, and a dot chart that displays the positive and negative significance of the features. The dot chart (Figure 4) displays visualization for each record in the dataset, providing local interpretability. The colors in the dot chart depict the features that are high or low for each row of the dataset, with red as high and blue as low. In addition, the horizontal location shows a positive or negative contribution of a feature.

Figure 4: Screenshot of SHAP Explanation

Screenshot of SHAP Explanation



Source: Gartner
738714_C

In addition, SHAP decomposes the final prediction into the contribution of each feature variable to provide an explanation for a specific prediction. The feature variables in red account for positive contribution and increase the prediction value, while the blue provide the negative contribution. The SHAP framework provides much more accurate prediction explanations as compared to LIME. However, SHAP is compute intensive and takes a long time to execute for most algorithms. Various commercial products have enhanced the open-source SHAP to increase its execution performance. SHAP may be scaled by applying distributed compute frameworks such as Spark. [7]

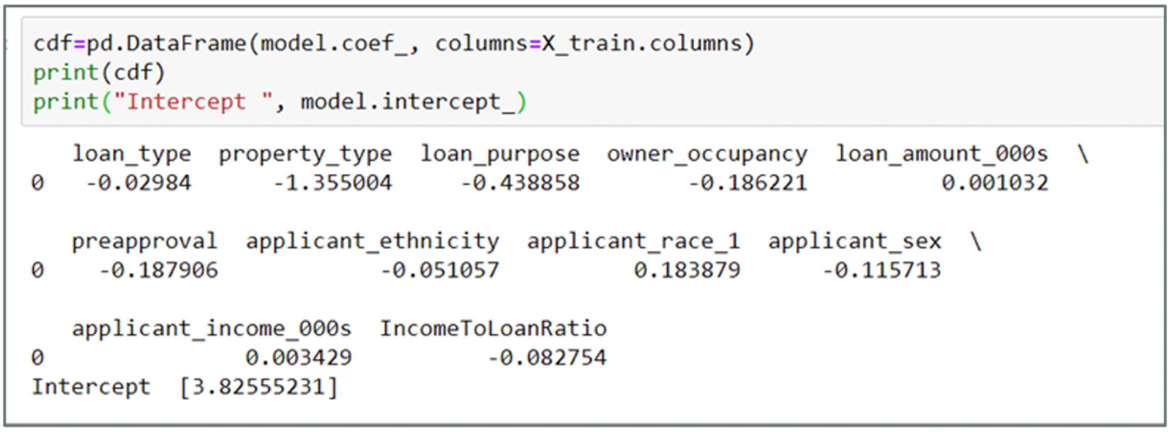
Model-Specific Explainability

Many organizations may prefer using regression or decision trees to avoid the problem of an inability to explain a model. Their use cases have a higher requirement of transparency versus accuracy. The fact remains that regression (linear and logistic) and decision tree models, because of their discrete structure, have higher interpretability.

Regression models: The availability of feature coefficients and the intercept used in the trained model makes it easy to reverse-engineer the prediction based on the input values. The code snippet in Figure 5 shows a sample of feature coefficients and intercept.

Figure 5: Coefficients and Intercept Values of a Regression Model

Coefficients and Intercept Values of a Regression Model



```
cdf=pd.DataFrame(model.coef_, columns=X_train.columns)
print(cdf)
print("Intercept ", model.intercept_)
```

```
   loan_type  property_type  loan_purpose  owner_occupancy  loan_amount_000s  \
0   -0.02984      -1.355004     -0.438858        -0.186221          0.001032

   preapproval  applicant_ethnicity  applicant_race_1  applicant_sex  \
0    -0.187906            -0.051057          0.183879      -0.115713

   applicant_income_000s  IncomeToLoanRatio
0               0.003429          -0.082754
Intercept  [3.82555231]
```

Source: Gartner
738714_C

Decision tree models: Treeinterpreter, a Python open-source package, provides the ability to interpret scikit-learn's decision tree and random forest predictions. [8] It allows decomposing each prediction into bias and feature contribution components.

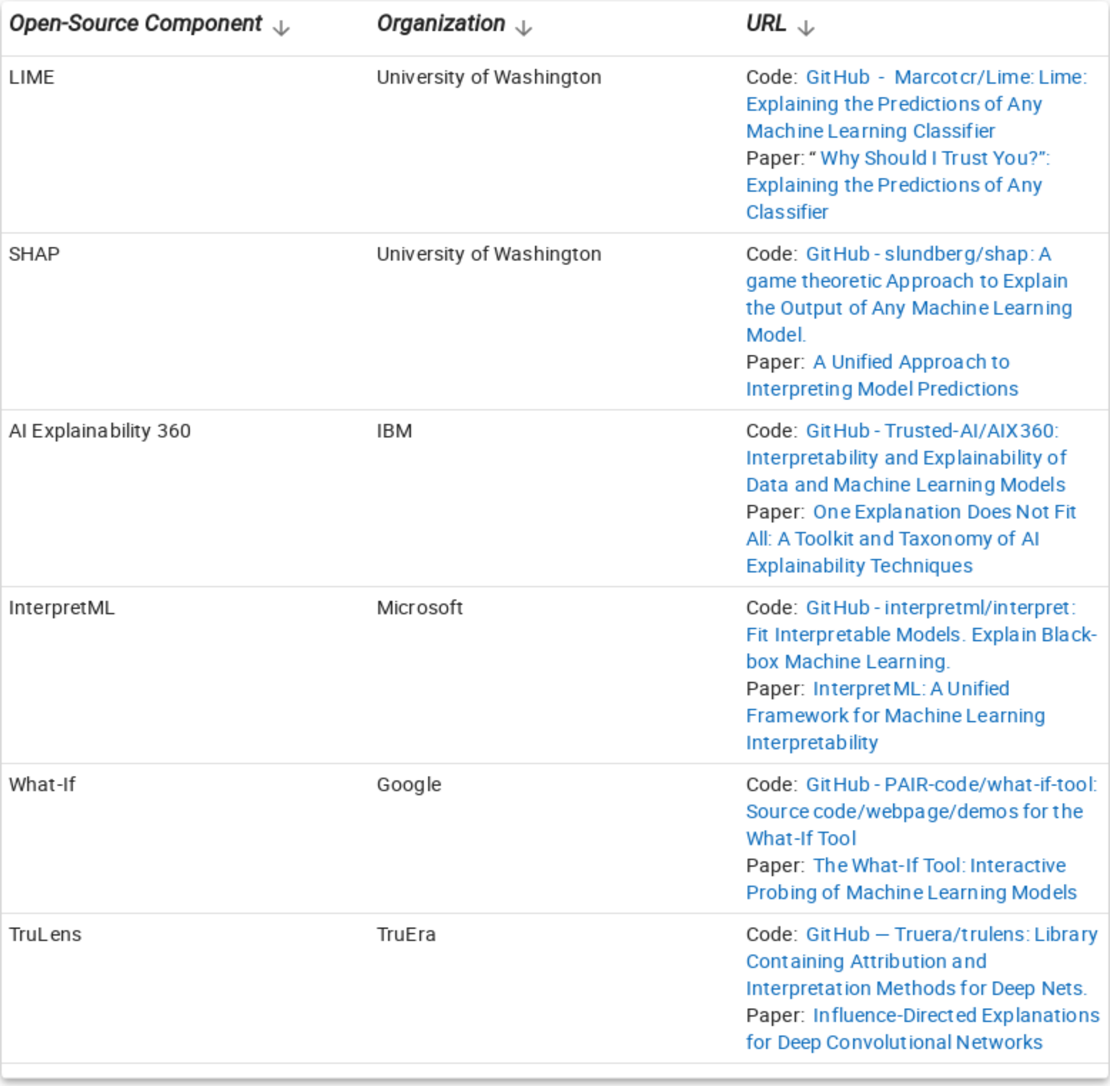
Explainability Products and Open-Source Toolkits

SHAP and LIME provided the early innovations and accessibility of tools to data scientists to incorporate explainability within their model development process. Both packages are still very relevant and provide foundations to a big majority of commercial products. At the same time, the challenges associated with explainability and fairness have driven new product and open-source innovation.

Table 1 provides a list of open-source components:

Table 1: Open-Source Explainability Components

(Enlarged table in Appendix)



Open-Source Component ↓	*Organization* ↓	*URL* ↓
LIME	University of Washington	Code: GitHub - Marcotcr/Lime: Lime: Explaining the Predictions of Any Machine Learning Classifier Paper: " Why Should I Trust You?": Explaining the Predictions of Any Classifier
SHAP	University of Washington	Code: GitHub - slundberg/shap: A game theoretic Approach to Explain the Output of Any Machine Learning Model. Paper: A Unified Approach to Interpreting Model Predictions
AI Explainability 360	IBM	Code: GitHub - Trusted-AI/AIX360: Interpretability and Explainability of Data and Machine Learning Models Paper: One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques
InterpretML	Microsoft	Code: GitHub - interpretml/interpret: Fit Interpretable Models. Explain Black-box Machine Learning. Paper: InterpretML: A Unified Framework for Machine Learning Interpretability
What-If	Google	Code: GitHub - PAIR-code/what-if-tool: Source code/webpage/demos for the What-If Tool Paper: The What-If Tool: Interactive Probing of Machine Learning Models
TruLens	TruEra	Code: GitHub — Truera/trulens: Library Containing Attribution and Interpretation Methods for Deep Nets. Paper: Influence-Directed Explanations for Deep Convolutional Networks

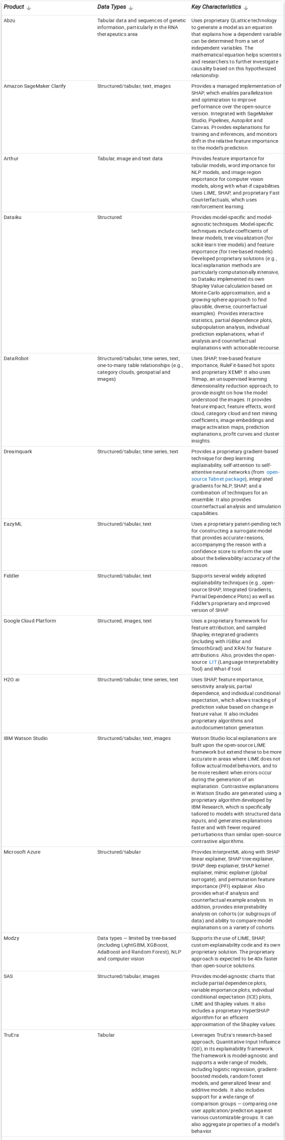
Source: Gartner (June 2022)

Many of these components are also available through data science and machine learning platforms. For example, Azureml-interpret python package, available as part of Azure Machine Learning SDK for Python, is based on the open-source InterpretML (see Table 2).

In addition to open-source explainability frameworks, several commercial solutions provide the added advantage of ease of integration, usability, and possibly, better execution performance.

Table 2: Explainability Capabilities Within Commercial Products

(Enlarged table in Appendix)



Product	Data Types	Key Characteristics
Abzu	Tabular data and sequences of genetic information, particularly in the RNA therapeutics area	Uses proprietary QLattice technology to generate a model as an equation that explains how a dependent variable can be determined from a set of independent variables. The mathematical equation helps scientists and researchers to further investigate causality based on this hypothesized relationship.
Amazon SageMaker Clarify	Structured/tabular, text, images	Provides a managed implementation of SHAP, which enables parallelization and optimization to improve performance over the open-source version. Integrated with SageMaker Studio, Pipelines, Autopilot and Canvas. Provides explanations for training and inferences, and monitors drift in the relative feature importance to the model's prediction.
Arthur	Tabular, image and text data	Provides feature importance for tabular models, word importance for NLP models, and image region importance for computer vision models, along with what-if capabilities. Uses LIME, SHAP, and proprietary Fast Counterfactuals, which uses reinforcement learning.
Dataiku	Structured	Provides model-specific and model-agnostic techniques. Model-specific techniques include coefficients of linear models, tree visualization (for scikit-learn tree models) and feature importance (for tree-based models). Developed proprietary solutions (e.g., local explanation methods are particularly computationally intensive, so Dataiku implemented its own Shapley Value calculation based on Monte-Carlo approximation, and a growing-sphere approach to find plausible, diverse, counterfactual examples). Provides interactive statistics, partial dependence plots, subpopulation analysis, individual prediction explanations, what-if analysis and counterfactual explanations with actionable recourse.
DataRobot	Structured/tabular, time series, text, one-to-many table relationships (e.g., category clouds, geospatial and images)	Uses SHAP, tree-based feature importance, RuleFit-based hot spots and proprietary XEMP. It also uses Trimap, an unsupervised learning dimensionality reduction approach, to provide insight on how the model understood the images. It provides feature impact, feature effects, word cloud, category cloud and text mining coefficients, image embeddings and image activation maps, prediction explanations, profit curves and cluster insights.
Dreamquark	Structured/tabular, time series, text	Provides a proprietary gradient-based technique for deep learning explainability, self-attention to self-attentive neural networks (from open-source Tabnet package), integrated gradients for NLP, SHAP, and a combination of techniques for an ensemble. It also provides counterfactual analysis and simulation capabilities.
EazyML	Structured/tabular, text	Uses a proprietary patent-pending tech for constructing a surrogate model that provides accurate reasons, accompanying the reason with a confidence score to inform the user about the believability/accuracy of the reason.
Fiddler	Structured/tabular, text	Supports several widely adopted explainability techniques (e.g., open-source SHAP, Integrated Gradients, Partial Dependence Plots) as well as Fiddler's proprietary and improved version of SHAP.
Google Cloud Platform	Structured, images, text	Uses a proprietary framework for feature attribution; and sampled Shapley, integrated gradients (including with IGBlur and SmoothGrad) and XRAI for feature attributions. Also, provides the open-source LIT (Language Interpretability Tool) and What-if tool.
H2O.ai	Structured/tabular, time series, text	Uses SHAP, feature importance, sensitivity analysis, partial dependence, and individual conditional expectation, which allows tracking of prediction value based on change in feature value. It also includes proprietary algorithms and autodocumentation generation.
IBM Watson Studio	Structured/tabular, text, images	Watson Studio local explanations are built upon the open-source LIME framework but extend these to be more accurate in areas where LIME does not follow actual model behaviors, and to be more resilient when errors occur during the generation of an explanation. Contrastive explanations in Watson Studio are generated using a proprietary algorithm developed by IBM Research, which is specifically tailored to models with structured data inputs, and generates explanations faster and with fewer required perturbations than similar open-source contrastive algorithms.
Microsoft Azure	Structured/tabular	Provides InterpretML along with SHAP linear explainer, SHAP tree explainer, SHAP deep explainer, SHAP kernel explainer, mimic explainer (global surrogate), and permutation feature importance (PFI) explainer. Also provides what-if analysis and counterfactual example analysis. In addition, provides interpretability analysis on cohorts (or subgroups of data) and ability to compare model explanations on a variety of cohorts.
Modzy	Data types — limited by tree-based (including LightGBM, XGBoost, AdaBoost and Random Forest), NLP and computer vision	Supports the use of LIME, SHAP, custom explainability code and its own proprietary solution. The proprietary approach is expected to be 40x faster than open-source solutions.
SAS	Structured/tabular, images	Provides model-agnostic charts that include partial dependence plots, variable importance plots, individual conditional expectation (ICE) plots, LIME and Shapley values. It also includes a proprietary HyperSHAP algorithm for an efficient approximation of the Shapley values.
TruEra	Tabular	Leverages TruEra's research-based approach, Quantitative Input Influence (QII), in its explainability framework. The framework is model-agnostic and supports a wide range of models, including logistic regression, gradient-boosted models, random forest models, and generalized linear and additive models. It also includes support for a wide range of comparison groups — comparing one user application/prediction against various customizable groups. It can also aggregate properties of a model's behavior.

Source: Gartner (June 2022)

Fairness Through Data and Models

AI models are often the manifestation of data. A bias in a dataset can be amplified and create an even more biased model. Fairness is primarily measured through metrics and monitoring. Data profiling using packages such as pandas-profiling, algorithms such as SMOTE (Synthetic Minority Oversampling Technique), and class weight adjustments are commonly used to correct class imbalances. These methods help identify potential bias in the data and implement model improvements. However, these methods have limited impact when the data has high dimensionality. Exploration of the open-source algorithm (see Table 3) and commercial solutions (see Table 4) provide additional solution approaches.

Table 3: Open-Source Fairness Components

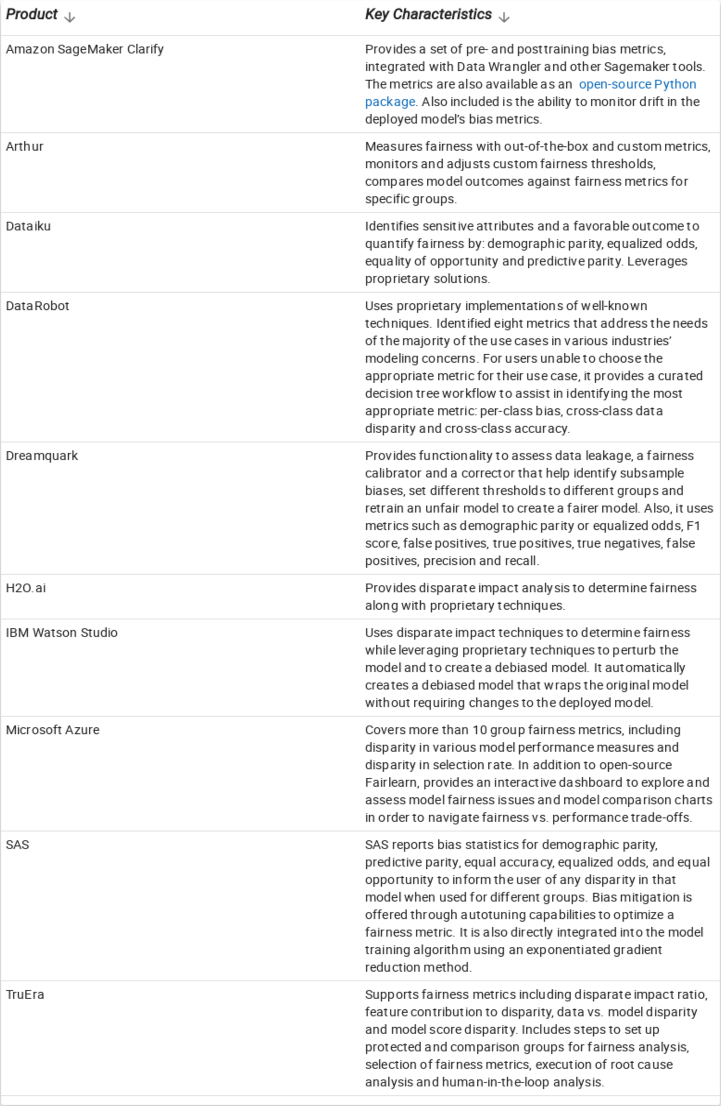
Open-Source Component ↓	*Organization* ↓	*URL* ↓
AI Fairness 360	IBM Research	Code: GitHub - Trusted-AI/AIF360 Paper: AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias
Aequitas	Carnegie Mellon University	Code: GitHub - dssg/aequitas: Bias and Fairness Audit Toolkit Paper: Aequitas: A Bias and Fairness Audit Toolkit
Fairlearn	Fairlearn Organization (Project initiated by Microsoft)	Code: GitHub - fairlearn/fairlearn Paper: A Reductions Approach to Fair Classification
TensorFlow Fairness Indicators	TensorFlow community, initiated by Google	Code: GitHub - tensorflow/fairness-indicators: Tensorflow's Fairness Evaluation and Visualization Toolkit Research blog: Google AI Blog: Setting Fairness Goals with the TensorFlow Constrained Optimization Library

Source: Gartner (June 2022)

In addition to the open-source frameworks in Table 3, the products in Table 4 have embedded model fairness capabilities.

Table 4: Fairness Capabilities Within Commercial Products

(Enlarged table in Appendix)



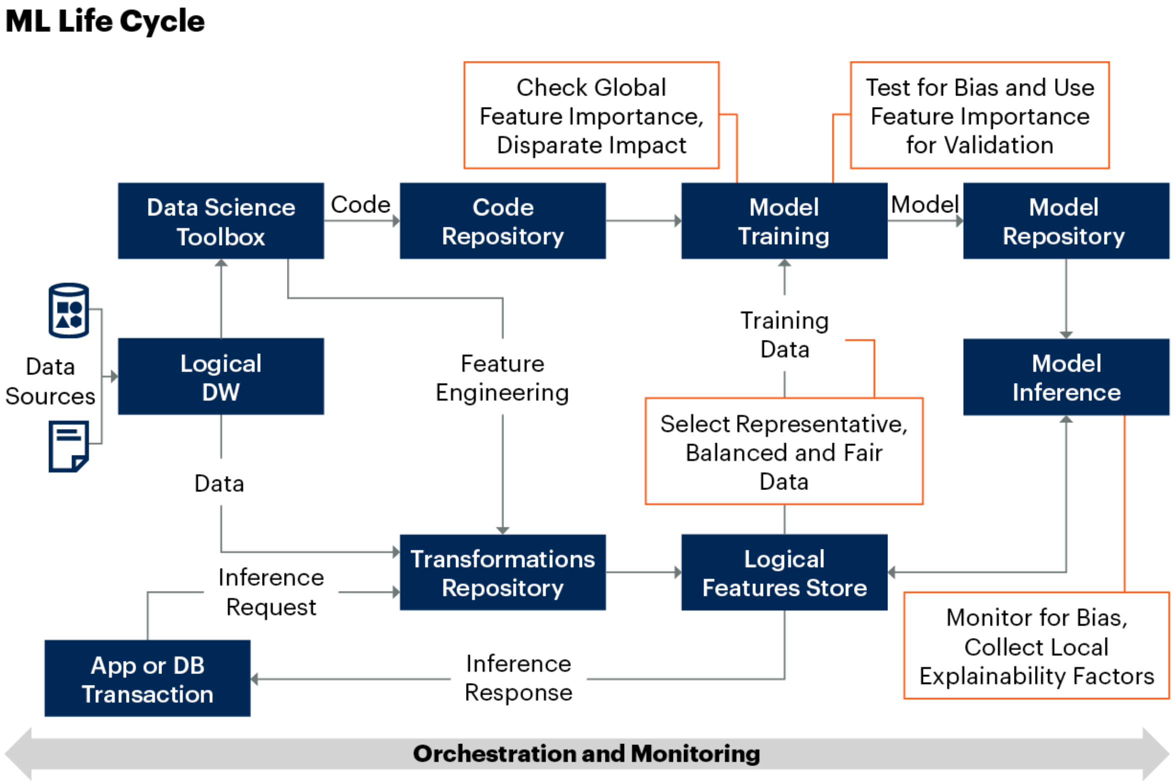
Product ↓	Key Characteristics ↓
Amazon SageMaker Clarify	Provides a set of pre- and posttraining bias metrics, integrated with Data Wrangler and other Sagemaker tools. The metrics are also available as an open-source Python package. Also included is the ability to monitor drift in the deployed model's bias metrics.
Arthur	Measures fairness with out-of-the-box and custom metrics, monitors and adjusts custom fairness thresholds, compares model outcomes against fairness metrics for specific groups.
Dataiku	Identifies sensitive attributes and a favorable outcome to quantify fairness by: demographic parity, equalized odds, equality of opportunity and predictive parity. Leverages proprietary solutions.
DataRobot	Uses proprietary implementations of well-known techniques. Identified eight metrics that address the needs of the majority of the use cases in various industries' modeling concerns. For users unable to choose the appropriate metric for their use case, it provides a curated decision tree workflow to assist in identifying the most appropriate metric: per-class bias, cross-class data disparity and cross-class accuracy.
Dreamquark	Provides functionality to assess data leakage, a fairness calibrator and a corrector that help identify subsample biases, set different thresholds to different groups and retrain an unfair model to create a fairer model. Also, it uses metrics such as demographic parity or equalized odds, F1 score, false positives, true positives, true negatives, false positives, precision and recall.
H2O.ai	Provides disparate impact analysis to determine fairness along with proprietary techniques.
IBM Watson Studio	Uses disparate impact techniques to determine fairness while leveraging proprietary techniques to perturb the model and to create a debiased model. It automatically creates a debiased model that wraps the original model without requiring changes to the deployed model.
Microsoft Azure	Covers more than 10 group fairness metrics, including disparity in various model performance measures and disparity in selection rate. In addition to open-source Fairlearn, provides an interactive dashboard to explore and assess model fairness issues and model comparison charts in order to navigate fairness vs. performance trade-offs.
SAS	SAS reports bias statistics for demographic parity, predictive parity, equal accuracy, equalized odds, and equal opportunity to inform the user of any disparity in that model when used for different groups. Bias mitigation is offered through autotuning capabilities to optimize a fairness metric. It is also directly integrated into the model training algorithm using an exponentiated gradient reduction method.
TruEra	Supports fairness metrics including disparate impact ratio, feature contribution to disparity, data vs. model disparity and model score disparity. Includes steps to set up protected and comparison groups for fairness analysis, selection of fairness metrics, execution of root cause analysis and human-in-the-loop analysis.

Source: Gartner (June 2022)

Embed Fairness and Explainability Functions Within the ML Life Cycle

Development of a fair model and understanding the explanations behind a prediction must not be an afterthought or a stand-alone activity. In order to make tasks repeatable, the actions need to be embedded throughout the entire ML development and deployment process. Figure 6 shows the various steps and actions required at each of the steps.

Figure 6: ML Life Cycle


ML Life Cycle
Check Global Feature Importance, Disparate Impact
Test for Bias and Use Feature Importance for Validation
Data Science Toolbox
Code
Code Repository
Model Training
Model
Model Repository
Data Sources
Logical DW
Feature Engineering
Training Data
Model Inference
Select Representative, Balanced and Fair Data
Data
Inference Request
Transformations Repository
Logical Features Store
Monitor for Bias, Collect Local Explainability Factors
App or DB Transaction
Inference Response
Orchestration and Monitoring

Source: Gartner
738714_C

The steps include:

- Data selection
- Model development and training
- Model testing and verification
- Model inference and monitoring

Apply Fairness Metrics During Data Selection

Data is the foundation of ML models. Biased data has the potential to create a biased foundation. In software development, fixing a bug in production is considered 100 times more expensive than fixing it in development. Similarly, resolving or mitigating bias as part of data selection is a lot more efficient and cost-effective than identifying bias through customer feedback.

All data is biased. It is important to identify, understand and mitigate bias.

Bias mitigation requires several considerations and actions:

- **Define fairness:** Data scientists are often tasked with developing a fair model. However, defining fairness requires an understanding of the business processes and regulator landscape. Business or product owners need to take the lead and collaborate with the compliance or risk expert and the data scientist to define what fairness means for the specific use case. They should also understand the ramifications of biased predictions and determine the guardrails to be implemented when the ML model is implemented in production. A fairness definition may include the following steps:

- Identify impacted stakeholders
- Assess regulations and ethical principles
- Determined favored groups and protected attributes
- Define which attributes must be excluded

The above fairness definition may require exclusion of features that identify a person's racial or gender attributes. As an example, ZIP Code has been found to be a proxy for race in areas where neighborhoods are broadly segregated based on race. A correlation analysis is required to exclude even the strong proxies that would bring in model bias. However, this would need to be balanced against business requirements. For example, property insurance risk analysis is aggregated at ZIP-Code-level. In such cases, removal of ZIP Codes would disrupt the business models. Similarly, income level may be another proxy for racial or ethnic groupings. However, income- or debt-related features are essential elements to determine creditworthiness for a home loan underwriter model. A more careful analysis is required to identify any proxies.

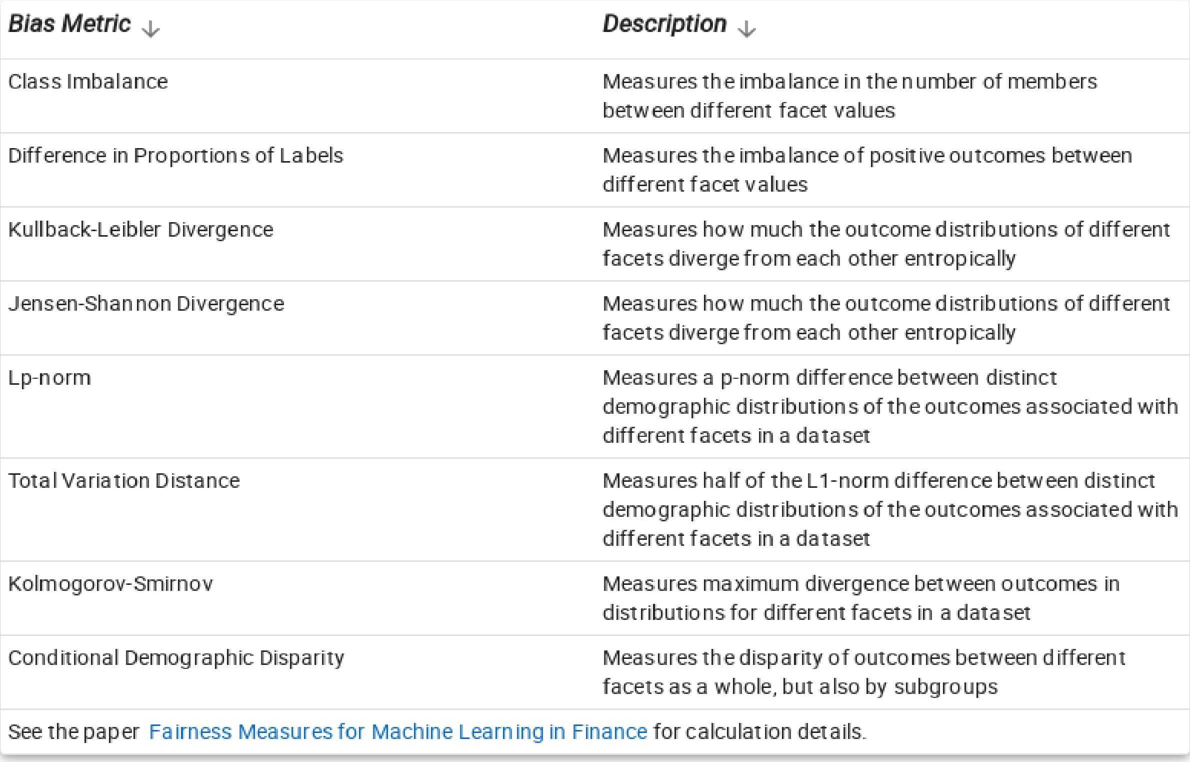
- **Select representative and balanced dataset:** Enterprises have large volumes of data, making it essential to ensure that selected data is representative of the business scenarios and customer segments within the context of the use case. For example, the U.S. Food and Drug Administration (FDA) recognized the lack of diversity within clinical trials. The FDA highlighted that "Participants in clinical trials should represent the patients that will use the medical products." [9] Nonrepresentative participation creates a real risk of medicinal products developed based on efficacy for a very specific population group.

Data scientists must use various metrics to measure the data bias and tools to balance datasets.

Table 5 provides a list of metrics:

Table 5: Metrics to Measure Data Bias

(Enlarged table in Appendix)



Bias Metric ↓	***Description*** ↓
Class Imbalance	Measures the imbalance in the number of members between different facet values
Difference in Proportions of Labels	Measures the imbalance of positive outcomes between different facet values
Kullback-Leibler Divergence	Measures how much the outcome distributions of different facets diverge from each other entropically
Jensen-Shannon Divergence	Measures how much the outcome distributions of different facets diverge from each other entropically
Lp-norm	Measures a p-norm difference between distinct demographic distributions of the outcomes associated with different facets in a dataset
Total Variation Distance	Measures half of the L1-norm difference between distinct demographic distributions of the outcomes associated with different facets in a dataset
Kolmogorov-Smirnov	Measures maximum divergence between outcomes in distributions for different facets in a dataset
Conditional Demographic Disparity	Measures the disparity of outcomes between different facets as a whole, but also by subgroups

See the paper Fairness Measures for Machine Learning in Finance for calculation details.

Source: Amazon SageMaker

There are several open-source toolkits available to mitigate bias in the datasets. Refer to the earlier section on Fairness Tools for more details.

Use Explainability to Improve Training

Good, robust model training is fundamental to the development of an ML model. Data scientists work through data cleansing, feature engineering and transformations to define the best features. They validate model performance based on various metrics including accuracy, F1 score, AUC and RMSE. They analyze specificity versus sensitivity to create a balance between false positives and false negatives. These are important steps.

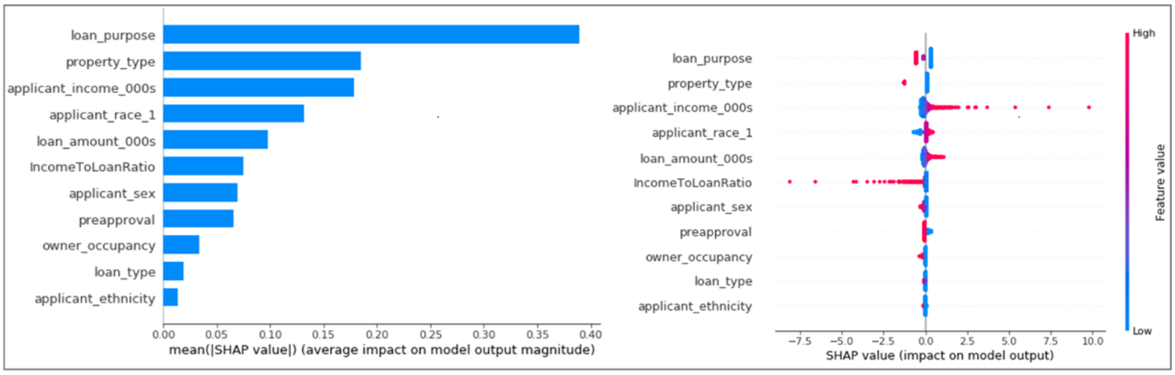
They still need to validate whether the model aligns with the business significance of the various features. Regression algorithms and decision trees, to an extent, provide access to the coefficients for this analysis and validation. However, decision tree ensembles (such as Random Forest and XGBoost) and deep learning models need additional tools to provide this level of transparency. Programming languages provide access to step-by-step execution and other techniques to analyze and debug code. However, ML models are a manifestation of data, and require different tools and techniques to debug. Explainability tools such as SHAP (refer to the section on Explainability Tools for more details) provide an approach to understand feature importance and contrast it with the business function of each of the features.

At the global level, SHAP provides the significance of the features to the prediction value. The visualizations include a bar chart, which displays the features in order of significance, and a dot chart displaying the positive and negative significance of the features. The dot chart displays visualization for each record in the dataset, providing local interpretability. The colors in the dot chart depict the features that are high or low for each row of the dataset, with red as high and blue as low. In addition, the horizontal location shows a positive or negative contribution of a feature.

As an example, the visualization in Figure 7 includes a bar chart and a dot chart. The bar chart shows “loan purpose” as the most important feature. The dot chart provides additional insights. For example, the blue dots concentrated near the center line for applicant income imply that low applicant income did not impact the prediction by much. At the same time, the red dots on the right show that higher applicant income provided a substantial positive contribution to the prediction.

Figure 7: SHAP Summary Plots

SHAP Summary Plots



Source: Gartner

738714_C

In addition, SHAP decomposes the final prediction into the contribution of each feature variable to provide an explanation for a specific prediction. The feature variables in red account for positive contribution and increase the prediction value, while the blue provide the negative contribution. The visualization in Figure 7 shows the loan purpose as being the biggest positive contributor, while property type was the most negative. The global explanations provide an average of all the predictions.

Leverage Feature Importance for Validation

Model testing and validation is probably the hardest problem to solve within the model life cycle. It involves several aspects of testing to ensure that the developed model meets the defined requirements and the business or customer needs.

The Federal Reserve's SR 11-7 has described the model validation requirement as “all model components, including input, processing, and reporting, should be subject to validation; this applies equally to models developed in-house and to those purchased from or developed by vendors or consultants.” [10] SR 11-7 provides a holistic view of testing requirements. However, most of the model testing is focused on validation of technical metrics such as RMSE, accuracy, confusion matrix, F1 score and AUC. Specificity and sensitivity analysis is used to validate use cases where class imbalance exists. As the impact and pervasiveness of AI increases, this testing is not sufficient. It needs additional testing approaches.

Software testing includes approaches like white-box, black-box and A/B testing. Lack of transparency and understanding of the model predictions create challenges for the testers, and they have to rely on black-box testing. Testers struggle to define the expected outcomes or validate if the prediction is a correct one. Explainable AI provides increased transparency into the features used within the model and the relationships at a global and local scope. Access to feature importance would help technical and business testers to relate the feature importance to the business significance of the features for the specific test script or inputs. Unlike the limitation of structured data for fairness tests, explainable AI can be leveraged for structured and unstructured data.

The open-source model validation toolkit from the Financial Industry Regulatory Authority (FINRA) provides an example of a toolkit using several approaches for model validation by the testing teams.

Explicitly Test for Model Bias

For use cases that have a potential of bias against a specific set of customers, employees or any other organizational stakeholders, explicit testing for bias is extremely important.

Testing scripts would be required, which, as an example, may first validate a prediction for one gender and then switch the gender to a different value and expect a very similar prediction. A home loan approval model would need to be tested for income ranges; property-value ranges; various combinations of race, sex and ethnicity; loan type; and so on. The included features may need to be aligned with the underwriting or pricing models. For example, ZIP Code is often considered a proxy for income levels or even racial identity. However, insurance models often quantify risk and pricing based on ZIP Codes. For such cases, a ZIP Code should not be considered a bias indicator. Proper analysis is needed to determine features that may be indicators of bias, and these may differ use case by use case.

Several metrics provide measures to validate bias in trained models. However, **disparate impact** is the most important metric and provides a good starting point. Disparate impact is defined as the ratio of rate of outcome for the unprivileged group to that of privileged group. This metric is often associated with the four-fifths rule defined by U.S. government agencies for employee selection procedures. The four-fifths or 80% rule is a rule of thumb where a group based on race, sex or ethnic group that has an 80% lesser selection rate than the group with the highest selection rate is considered adversely affected. There is an ongoing debate on whether the four-fifths measure provides a way to identify and remediate bias for all use cases. However, measuring disparate impact does provide a good indicator to the presence of bias in the model.

In addition, there are several metrics defined as part of various toolkits and algorithms. The following documents provide several metrics examples:

- Amazon AI Fairness and Explainability White Paper
- Fairlearn Metrics Package documentation
- Google Machine Learning Glossary: Fairness
- IBM AI Fairness 360 toolkit API documentation of Fairness Metrics

Model testing and validation should include steps to generate and assess such metrics based on a representative dataset. The above metrics apply to structured data, for the most part. Identification of bias within unstructured data, specially images, requires more rigorous testing and may include manual verification.

Monitor the Predictions

The previous steps ensure the development of a good model. However, models may not always perform the same way in production. The production data distributions may change over a period of time, or as business conditions change. For example, the image-based solution for detection of diabetic retinopathy developed by Google research had better than 90% accuracy in the lab. [11] However, in production, inconsistent image quality reduced the quality of predictions substantially. It is essential to monitor the input data and model predictions, raising alerts as they go beyond the tolerance thresholds. In addition to alerts, these additional actions enable further analysis:

- Log and store the input, output and explainability factors.
- Provide the explainability factors, including feature attributions, to the business user and data scientists in case of anomalous behavior.
- Debias the model output by applying a decision framework embedding business rules or postprocessing algorithms from the AI Fairness 360 toolkit.

Next Steps

Explore Counterfactual Explanations for End Users

Counterfactual explanations are an active area of research. The explanations help identify the changes in the inputs that would change the outcome. This "what-if" approach may be used to provide an actionable explanation. However, within current applications, this may not necessarily represent a causal relationship. Presenting these explanations may require a thorough business review and validation. For example, for rejection of a customer loan application, a suggestion to pay bills on time for two months is an actionable recommendation. However, an action to increase age by 20 years is not actionable. Consequently, caution is required before implementing counterfactual explanations presented directly to an end user. Microsoft's open-source DiCE project provides a good option to experiment with.

Define Role of Model Owner for Accountability

Data scientists and data and ML engineers bring technical skills and are responsible for development and deployment of ML models and data pipelines. Business owners define the use case and business requirements. While the model is a technology component, it supports business decision making and needs to be considered an asset. SR 11-7 defines the role of a model owner as "... ultimate accountability for model use and performance within the framework set by bank policies and procedures. Model owners should be responsible for ensuring that models are properly developed, implemented, and used."

Model owners have the ownership of model quality, approvals and continued model usage. As an example, a bank loan approval model may be trained for a specific geographical region or income range. However, without knowing the limitations of the model, the model may be used within an application framework for different customer incomes. A model's performance may also be impacted by data or concept drift. It may be biased against a specific group of customers. The model owner is accountable for usage and retraining approvals or decommissioning decisions for the deviations.

Strengths

Understanding of machine learning models empowers both data scientists (model developers) and business users (model consumers) to improve and act upon the model recommendations. Explainable machine learning provides a key component for this understanding. The key strengths of explainability and fairness toolkits and frameworks are:

- **Availability of model insights reduces algorithm risk:** Deep learning neural networks (DNNs), such as image recognition and natural language processing, are said to have achieved human parity. Despite some of the controversies surrounding the conclusion of human parity, the fact remains that DNNs have achieved tremendous growth with overall AI adoption by organizations. At the same time, the black-box nature of DNNs creates challenges for use in key decision-making systems. Doctors would not be able to rely on an ML-engine prescription recommendation unless they were able to access the patient diagnosis scores and alternatives considered. Explainability solutions illuminate some of the black box and provide solutions to reduce algorithm risk to the business operations. The combination of model-agnostic algorithms along with local and global interpretability provides the ability to explain decisions to customers and regulators.

- **Increasing academic research and industry investments:** The Defense Advanced Research Projects Agency (DARPA) has been an innovation guide for both academia and industry. DARPA has recognized the importance of machine learning for the success of AI applications and created the Explainable AI (XAI) program. In addition, increasing submissions at key academic research conferences and recent availability of explainability functions within the major data science and machine learning platforms is a good indication of continued evolution of such capabilities. Open-source toolkits, such as InterpretML and AI Explainability 360, have shared innovative algorithms with ML engineers and data scientists to experiment and integrate without dependency on commercial solutions.

- **Enhanced model quality:** A confusion matrix is the most commonly used measure of model accuracy and quality. However, there is a lot of dependency on the quality and diversity of training data. Fairness (or lack thereof) and bias in decision making are issues that still plague the application of machine learning. Explainability toolkits provide an ability to validate the key features at a global and local prediction level. The functional domain experts can review these findings and determine if additional data or algorithm improvements are required. The ability to troubleshoot the model outcomes adds an important testing and validation layer that aids in development of better-quality models.

- **Fairness metrics combined with explainability help identify and mitigate bias:** Development of a fair model requires analysis and measurement of data bias metrics. In addition, insights provided by explainability tools help determine if protected features are influencing the model prediction. These, combined with various fairness algorithms, provide a good approach to identify and mitigate bias in models.

Weaknesses

Explainable machine learning might seem like the perfect solution to resolve challenges related to the black-box nature of ML models. It does, in fact, provide various options to solve this puzzle. However, the implementation requires some caution and effort to overcome the weaknesses:

- **High computation resource requirements:** Most of the explainability packages like LIME and SHAP execute a large number of model predictions to understand the model's behavior. Model training would require additional time and compute resources to build the explainability model, partial dependence, feature importance and other perturbation-based models. Similarly, local explanations for a specific record or prediction would require additional time for the service call when making real-time predictions.

- **One method does not solve the entire problem:** Even though SHAP provides both global and local explainability, it still requires a combination of tools to validate the explanations. Data and analytics technical professionals would need to use a combination of LIME, SHAP, partial dependence and feature importance, among others, to validate the explanation matrix and use-case alignment. This requires the model analysts to understand results presented by each toolset.

- **Explanation models are approximations:** Model-agnostic explanation algorithms execute the ML model for the training dataset and use feature perturbation to create an explanation model. This model tries to approximate the behavior of the ML model. However, this is based on a static training dataset. The dataset used for inference may be entirely different. Consequently, the prediction explanations may not always be correct. Data scientists and business owners need to understand the limitations of the frameworks and validate using multiple tools to lower the possibility of error. A blend based on explanations from multiple algorithms and methods must be used.

- **Increased risk of model inversion attacks and membership inference attacks:** The explainability surrogate model can be used to reverse engineer the prediction model. The surrogate model, depending on the accuracy, can be used as a sandbox to estimate the training data. This method, also known as inversion, can be used to determine confidential or private information used to train the model. Another method, called membership inference attack, uses similar techniques to inversion, and recovers information as to whether a particular individual was in the training set. This use of surrogate models creates security risks. The original prediction model along with the explainability surrogate model need appropriate access controls and monitoring of the model usage to prevent such attacks. Repeated scoring of similar input datasets should require additional investigation and a potential throttling of the model's usage for real-time prediction models.

Guidance

AI adoption within organizations is witnessing rapid growth. Many organizations are expecting AI to provide the strategic differentiation from their competitors. This reliance and dependency on AI, and specifically on ML, requires organizations to take a step back and evaluate the controls, the rigor, and most importantly, a greater understanding of the prediction process. Explainable AI promises to resolve some of these challenges. However, organizations, specifically data and analytics technical professionals, should evaluate the strengths, weaknesses and an alignment of the frameworks with their requirements.

Dynamic Landscape Needs Proper Assessment

The landscape for explainability frameworks has been evolving rapidly. A combination of academic research, industry product extensions and open-source community contributions has provided several tools to understand machine learning predictions. Academic research often works as a leading indicator to show upcoming product capabilities. Organizations should monitor academic research, conference presentations and publications from thought leaders to understand the forthcoming solutions and capabilities. This dynamic landscape also means that organizations evaluating the various explainability frameworks and products do not have to strive for the perfect product or framework. Most leading product vendors recognize the significance of machine learning explainability and are investing substantially in product development. This continuing product development supported by research should result in improved and more user-friendly capabilities.

In addition, there is a substantial ongoing debate on definitions of terms surrounding explainable AI, such as ethics, privacy, bias, interpretability and overall responsible AI. While this document is focused on the technical aspects of machine learning explainability, organizations may need to assess and evaluate each of these factors as they apply to their customers, employees, suppliers, vendors and shareholders.

Align Explainability Tools and Frameworks With Use-Case and Impacted Personas

Several open-source and proprietary tools and frameworks provide both model-agnostic and model-specific views into the prediction model. However, there is no single solution to the problem. There are several decision points in the process: algorithm, accuracy, interpretability, transparency, compute requirements, execution runtime and ease of implementation, to list a few.

An insurance actuary or a credit analyst would prefer accuracy and transparency over execution runtime and interpretability. They would look to use multiple tools — such as SHAP, partial dependence plots and feature importance — and analyze both local and global explanations to identify any model discrepancies. In contrast, a loan officer or a claims processor would prioritize interpretability and faster availability of results as they look to explain the decision to their customer. LIME, combined with feature importance for a specific prediction, would be more relevant for such use cases.

Often many of these requirements, especially accuracy and performance constraints, are defined as nonfunctional requirements. Data and analytics technical professionals should review both functional and nonfunctional requirements, along with journey maps for different personas for a more holistic solution design.

Not All Business Problems Require Explainability

Organizations should not look to use ML explainability for all use cases. Explainability frameworks add additional complexity and overheads. For applications that create minimal or no adverse impact, the overhead associated with explainability may not be justified. ML models within games, emails, postal code sorting, medical devices with precertified models that cannot be modified and movie recommendation systems are some examples. These examples belong to two categories:

- There are no significant consequences for unacceptable results.
- The problem and the model went through rigorous testing, validation and certification in actual scenarios and market testing. In this case, there is enough trust in the model's decision.

Business subject matter experts should collaborate with data scientists and ML architects to understand the associated overheads with explainability and have an in-depth discussion about the use case and requirements for explainability.

Evidence

[1] Model-Agnostic Interpretability of Machine Learning, arXiv.

[2] Lime, GitHub.

[3] Why Should I Trust You? Explaining the Predictions of Any Classifier, arXiv.

[4] Loan Application Register Code Sheet, Federal Financial Institutions Examination Council.

[5] A Unified Approach to Interpreting Model Predictions, arXiv.

[6] SHAP, GitHub.

[7] Scaling SHAP Calculations With PySpark and Pandas UDF, Databricks.

[8] Treeinterpreter, GitHub.

[9] Clinical Trial Diversity, U.S. Food and Drug Administration.

[10] Supervisory Guidance on Model Risk Management, Board of Governors of the Federal Reserve System.

[11] A Human-Centered Evaluation of a Deep Learning System Deployed in Clinics for the Detection of Diabetic Retinopathy, Association for Computing Machinery.

Document Revision History

Improve the Machine Learning Trust Equation by Using Explainable AI Frameworks - 30 December 2019

Recommended by the Author

Some documents may not be available as part of your current Gartner subscription.

Video: Why Is Responsible AI Important for Data and Analytics Professionals?

Expert Insight Video: What Is Responsible AI and Why You Should Care About It?

Innovation Insight for Bias Detection/Mitigation, Explainable AI and Interpretable AI

Machine Learning Playbook for Data and Analytics Professionals

Analytics and Artificial Intelligence for Technical Professionals Primer for 2022

Table 1: Open-Source Explainability Components

Open-Source Component ↓	*Organization* ↓	*URL* ↓
LIME	University of Washington	Code: GitHub - Marcotcr/Lime: Lime: Explaining the Predictions of Any Machine Learning Classifier Paper: " Why Should I Trust You?": Explaining the Predictions of Any Classifier
SHAP	University of Washington	Code: GitHub - slundberg/shap: A game theoretic Approach to Explain the Output of Any Machine Learning Model. Paper: A Unified Approach to Interpreting Model Predictions
AI Explainability 360	IBM	Code: GitHub - Trusted-AI/AIX360: Interpretability and Explainability of Data and Machine Learning Models Paper: One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques
InterpretML	Microsoft	Code: GitHub - interpretml/interpret: Fit Interpretable Models. Explain Black-box Machine Learning. Paper: InterpretML: A Unified Framework for Machine Learning Interpretability

Open-Source Component ↓	*Organization* ↓	*URL* ↓
What-If	Google	Code: GitHub - PAIR-code/what-if-tool: Source code/webpage/demos for the What-If Tool Paper: The What-If Tool: Interactive Probing of Machine Learning Models
TruLens	TruEra	Code: GitHub — Truera/trulens: Library Containing Attribution and Interpretation Methods for Deep Nets. Paper: Influence-Directed Explanations for Deep Convolutional Networks

Source: Gartner (June 2022)

Table 2: Explainability Capabilities Within Commercial Products

Product ↓	*Data Types* ↓	*Key Characteristics* ↓
Abzu	Tabular data and sequences of genetic information, particularly in the RNA therapeutics area	Uses proprietary QLattice technology to generate a model as an equation that explains how a dependent variable can be determined from a set of independent variables. The mathematical equation helps scientists and researchers to further investigate causality based on this hypothesized relationship.
Amazon SageMaker Clarify	Structured/tabular, text, images	Provides a managed implementation of SHAP, which enables parallelization and optimization to improve performance over the open-source version. Integrated with SageMaker Studio, Pipelines, Autopilot and Canvas. Provides explanations for training and inferences, and monitors drift in the relative feature importance to the model's prediction.
Arthur	Tabular, image and text data	Provides feature importance for tabular models, word importance for NLP models, and image region importance for computer vision models, along with what-if capabilities. Uses LIME, SHAP, and proprietary Fast Counterfactuals, which uses reinforcement learning.

Product ↓	*Data Types* ↓	*Key Characteristics* ↓
Dataiku	Structured	Provides model-specific and model-agnostic techniques. Model-specific techniques include coefficients of linear models, tree visualization (for scikit-learn tree models) and feature importance (for tree-based models). Developed proprietary solutions (e.g., local explanation methods are particularly computationally intensive, so Dataiku implemented its own Shapley Value calculation based on Monte-Carlo approximation, and a growing-sphere approach to find plausible, diverse, counterfactual examples). Provides interactive statistics, partial dependence plots, subpopulation analysis, individual prediction explanations, what-if analysis and counterfactual explanations with actionable recourse.
DataRobot	Structured/tabular, time series, text, one-to-many table relationships (e.g., category clouds, geospatial and images)	Uses SHAP, tree-based feature importance, RuleFit-based hot spots and proprietary XEMP. It also uses Trimap, an unsupervised learning dimensionality reduction approach, to provide insight on how the model understood the images. It provides feature impact, feature effects, word cloud, category cloud and text mining coefficients, image embeddings and image activation maps, prediction explanations, profit curves and cluster insights.

Product ↓	*Data Types* ↓	*Key Characteristics* ↓
Dreamquark	Structured/tabular, time series, text	Provides a proprietary gradient-based technique for deep learning explainability, self-attention to self-attentive neural networks (from open-source Tabnet package), integrated gradients for NLP, SHAP, and a combination of techniques for an ensemble. It also provides counterfactual analysis and simulation capabilities.
EazyML	Structured/tabular, text	Uses a proprietary patent-pending tech for constructing a surrogate model that provides accurate reasons, accompanying the reason with a confidence score to inform the user about the believability/accuracy of the reason.
Fiddler	Structured/tabular, text	Supports several widely adopted explainability techniques (e.g., open-source SHAP, Integrated Gradients, Partial Dependence Plots) as well as Fiddler's proprietary and improved version of SHAP.
Google Cloud Platform	Structured, images, text	Uses a proprietary framework for feature attribution; and sampled Shapley, integrated gradients (including with IGBlur and SmoothGrad) and XRAI for feature attributions. Also, provides the open-source LIT (Language Interpretability Tool) and What-if tool.

Product ↓	*Data Types* ↓	*Key Characteristics* ↓
H2O.ai	Structured/tabular, time series, text	Uses SHAP, feature importance, sensitivity analysis, partial dependence, and individual conditional expectation, which allows tracking of prediction value based on change in feature value. It also includes proprietary algorithms and autodocumentation generation.
IBM Watson Studio	Structured/tabular, text, images	Watson Studio local explanations are built upon the open-source LIME framework but extend these to be more accurate in areas where LIME does not follow actual model behaviors, and to be more resilient when errors occur during the generation of an explanation. Contrastive explanations in Watson Studio are generated using a proprietary algorithm developed by IBM Research, which is specifically tailored to models with structured data inputs, and generates explanations faster and with fewer required perturbations than similar open-source contrastive algorithms.

Product ↓	Data Types ↓	Key Characteristics ↓
Microsoft Azure	Structured/tabular	Provides InterpretML along with SHAP linear explainer, SHAP tree explainer, SHAP deep explainer, SHAP kernel explainer, mimic explainer (global surrogate), and permutation feature importance (PFI) explainer. Also provides what-if analysis and counterfactual example analysis. In addition, provides interpretability analysis on cohorts (or subgroups of data) and ability to compare model explanations on a variety of cohorts.
Modzy	Data types — limited by tree-based (including LightGBM, XGBoost, AdaBoost and Random Forest), NLP and computer vision	Supports the use of LIME, SHAP, custom explainability code and its own proprietary solution. The proprietary approach is expected to be 40x faster than open-source solutions.
SAS	Structured/tabular, images	Provides model-agnostic charts that include partial dependence plots, variable importance plots, individual conditional expectation (ICE) plots, LIME and Shapley values. It also includes a proprietary HyperSHAP algorithm for an efficient approximation of the Shapley values.

Product ↓	*Data Types* ↓	*Key Characteristics* ↓
TruEra	Tabular	Leverages TruEra's research-based approach, Quantitative Input Influence (QII), in its explainability framework. The framework is model-agnostic and supports a wide range of models, including logistic regression, gradient-boosted models, random forest models, and generalized linear and additive models. It also includes support for a wide range of comparison groups — comparing one user application/prediction against various customizable groups. It can also aggregate properties of a model's behavior.

Source: Gartner (June 2022)

Table 3: Open-Source Fairness Components

Open-Source Component ↓	*Organization* ↓	*URL* ↓
AI Fairness 360	IBM Research	Code: GitHub - Trusted-AI/AIF360 Paper: AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias
Aequitas	Carnegie Mellon University	Code: GitHub - dssg/aequitas: Bias and Fairness Audit Toolkit Paper: Aequitas: A Bias and Fairness Audit Toolkit
Fairlearn	Fairlearn Organization (Project initiated by Microsoft)	Code: GitHub - fairlearn/fairlearn Paper: A Reductions Approach to Fair Classification
TensorFlow Fairness Indicators	TensorFlow community, initiated by Google	Code: GitHub - tensorflow/fairness-indicators: Tensorflow's Fairness Evaluation and Visualization Toolkit Research blog: Google AI Blog: Setting Fairness Goals with the TensorFlow Constrained Optimization Library

Source: Gartner (June 2022)

Table 4: Fairness Capabilities Within Commercial Products

Product ↓	*Key Characteristics* ↓
Amazon SageMaker Clarify	Provides a set of pre- and posttraining bias metrics, integrated with Data Wrangler and other Sagemaker tools. The metrics are also available as an open-source Python package. Also included is the ability to monitor drift in the deployed model's bias metrics.
Arthur	Measures fairness with out-of-the-box and custom metrics, monitors and adjusts custom fairness thresholds, compares model outcomes against fairness metrics for specific groups.
Dataiku	Identifies sensitive attributes and a favorable outcome to quantify fairness by: demographic parity, equalized odds, equality of opportunity and predictive parity. Leverages proprietary solutions.
DataRobot	Uses proprietary implementations of well-known techniques. Identified eight metrics that address the needs of the majority of the use cases in various industries' modeling concerns. For users unable to choose the appropriate metric for their use case, it provides a curated decision tree workflow to assist in identifying the most appropriate metric: per-class bias, cross-class data disparity and cross-class accuracy.

Product ↓	***Key Characteristics*** ↓
Dreamquark	Provides functionality to assess data leakage, a fairness calibrator and a corrector that help identify subsample biases, set different thresholds to different groups and retrain an unfair model to create a fairer model. Also, it uses metrics such as demographic parity or equalized odds, F1 score, false positives, true positives, true negatives, false positives, precision and recall.
H2O.ai	Provides disparate impact analysis to determine fairness along with proprietary techniques.
IBM Watson Studio	Uses disparate impact techniques to determine fairness while leveraging proprietary techniques to perturb the model and to create a debiased model. It automatically creates a debiased model that wraps the original model without requiring changes to the deployed model.
Microsoft Azure	Covers more than 10 group fairness metrics, including disparity in various model performance measures and disparity in selection rate. In addition to open-source Fairlearn, provides an interactive dashboard to explore and assess model fairness issues and model comparison charts in order to navigate fairness vs. performance trade-offs.
SAS	SAS reports bias statistics for demographic parity, predictive parity, equal accuracy, equalized odds, and equal opportunity to inform the user of any disparity in that model when used for different groups. Bias mitigation is offered through autotuning capabilities to optimize a fairness metric. It is also directly integrated into the model training algorithm using an exponentiated gradient reduction method.

Product ↓	***Key Characteristics*** ↓
TruEra	Supports fairness metrics including disparate impact ratio, feature contribution to disparity, data vs. model disparity and model score disparity. Includes steps to set up protected and comparison groups for fairness analysis, selection of fairness metrics, execution of root cause analysis and human-in-the-loop analysis.

Source: Gartner (June 2022)

Table 5: Metrics to Measure Data Bias

Bias Metric ↓	*Description* ↓
Class Imbalance	Measures the imbalance in the number of members between different facet values
Difference in Proportions of Labels	Measures the imbalance of positive outcomes between different facet values
Kullback-Leibler Divergence	Measures how much the outcome distributions of different facets diverge from each other entropically
Jensen-Shannon Divergence	Measures how much the outcome distributions of different facets diverge from each other entropically
Lp-norm	Measures a p-norm difference between distinct demographic distributions of the outcomes associated with different facets in a dataset
Total Variation Distance	Measures half of the L1-norm difference between distinct demographic distributions of the outcomes associated with different facets in a dataset
Kolmogorov-Smirnov	Measures maximum divergence between outcomes in distributions for different facets in a dataset
Conditional Demographic Disparity	Measures the disparity of outcomes between different facets as a whole, but also by subgroups
See the paper Fairness Measures for Machine Learning in Finance for calculation details.	

Source: Amazon SageMaker