AI in Financial Services: *The Data Science of Explainability*

Frequently Asked Questions

JUNE 2021

FinRegLab periodically releases AI FAQs to provide financial services stakeholders with accessible information about the technological, market, and policy implications of using advanced analytical techniques in the service of a more vibrant and inclusive financial marketplace. This edition introduces key concepts related to the data science of model explainability with a particular emphasis on *post hoc* techniques that may help lenders responsibly use more complex machine learning underwriting models.

One edition of our AI FAQs explored key concepts related to the use of AI in financial services. Another investigated model transparency and explainability and their importance in the context of using machine learning credit underwriting models. To support FinRegLab's forthcoming research on the explainability and fairness of machine learning underwriting models, this edition of our AI FAQs focuses on the data science of model explainability, given that our ability to understand and evaluate algorithmic decisions is at the center of implementing AI safely and responsibly. Lenders will consider transparency as to the bases for a model's predictions, as well as process transparency and other factors that are critical to the trustworthiness of AI systems, in each step of designing, developing, and operating such systems.[1] These FAQs are designed to provide information about different approaches for improving the transparency of machine learning models and implications for the full range of consumer credit stakeholders.

This edition of our AI FAQs answers the following questions:

- **How do current models for credit underwriting work? In what ways are machine learning models different? Are they necessarily more complex?**
- **What kinds of machine learning models are most relevant to credit underwriting?**

- **What factors can affect a model developer's choice of machine learning algorithms?**
- **What are latent features? How do they affect model transparency?**
- **How can model developers improve the transparency of machine learning models?**
- **How can model developers build inherently interpretable machine learning models?**
- **What kinds of *post hoc* explainability techniques can be used to improve model transparency?**
- **What characteristics differentiate *post hoc* explainability techniques?**
- **How can the capabilities and performance of explainability techniques be evaluated?**

In time, our AI FAQs will be presented in digital form on our website to facilitate their use as a reference. Until then, this document refers to, rather than repeats, content previously covered in our AI FAQs. Readers can find the full set of FAQs here or use the links in the cross-referenced sections included with the specific questions below.

How do current models for credit underwriting work? In what ways are machine learning models different? Are they necessarily more complex?

Lenders use underwriting models to evaluate the likelihood that individual applicants for credit will repay the loan. Applicants are assigned to risk tiers based on the relative probability of default that the model (or collection of models) assigns. Lenders then decide which tiers they are willing to approve based on their willingness to take on credit risk in light of market conditions and other factors. Where lenders employ risk-based pricing, they generally impose higher interest rates on applicants that have a greater probability of default as determined by the risk tiering. Finally, lenders will assign the loan amount or credit limit based on the underwriting model's assessment of the applicant's creditworthiness.

Incumbent underwriting models typically use logistic regression or linear classification models to classify a loan applicant's credit characteristics and make a prediction about the likelihood that the requested loan will be repaid.[2] In a classification model, the data are divided into two classes: "good loans" and "bad loans," and the model is trained to predict whether a new application will result in a "good loan" or a "bad loan." These are automated to provide real-time decisioning of applications – to enable point of sale issuance of a credit card or on-demand decisions for digital loan applications, for example. The models typically examine loan applicants based on attributes related to past behavior in using credit and compare them to prior data reflecting experience with borrowers with similar characteristics. For example, many incumbent underwriting models consider applicants' number of delinquencies in their calculations of default probability.[3]

The basic problem that a credit underwriting model is designed to solve – determining whether an applicant is likely to repay a loan – does not change when machine learning is used to develop the model. However, the shift to machine learning modelling methods does have significant implications. Machine learning models can be more accurate than the prior generation of models. This means fewer borrowers are offered loans they are unlikely to be able to repay and that more qualified borrowers are approved for credit. However, this accuracy can result from increased complexity, making it more difficult to explain why a model made a particular assessment of creditworthiness. Our ability to develop such complex models may be ahead of our ability to understand and manage their behavior, especially in use cases like credit underwriting, though much work is being done in academia, industry, and government to understand and address those gaps.

At the same time, not all machine learning underwriting models are less transparent than incumbent models, and some forms of machine learning currently in use for underwriting are not substantially more complex than the models they replaced. Further, even with conventional models, use of complex techniques to construct a model – like LASSO (least absolute shrinkage and selection operator) – does not necessarily result in a model that lacks transparency as to its predictions. In general, model complexity is primarily a function of the type of algorithm being used, the nature of constraints that a model developer designates, and the type and structure of the data to which the learning algorithm is exposed. For example, a deep neural network with many hidden layers in which feature correlations are calculated may not be particularly transparent, whereas a linear regression with hundreds of features identified using machine learning may be roughly as transparent as a conventional regression model.

The complexity of models created by algorithms can also be restricted to improve their transparency – for example, a neural network may be constrained to limit the number of hidden layers, or connections between layers, just as a decision tree may be limited in its depth or number of leaf nodes. However, limitations on model complexity can also decrease predictive accuracy, so practitioners may choose to focus on developing alternative approaches that make complex models more transparent.

Further Reading

Deloitte, Explain Artificial Intelligence for Credit Risk Management, (April 2020), available at: https://www2.deloitte.com/content/dam/Deloitte/fr/Documents/risk/Publications/deloitte_artificial-intelligence-credit-risk.pdf.

Cynthia Rudin & Joanna Radin, Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson from an Explainable AI Competition, Harvard Data Science Review (November 22, 2019), available at https://hdsr.mitpress.mit.edu/pub/f9kuryi8/release/6

Related FAQs

What is the basis for believing that machine learning could improve credit underwriting?

What is model transparency? Why do we need it?

What potential risks are important to consider when lenders replace incumbent credit underwriting models with machine learning?

What kinds of machine learning models are most relevant to credit underwriting?

How can model developers improve the transparency of machine learning models?

How can model developers build inherently interpretable machine learning models?

What kinds of *post hoc* explainability techniques can be used to improve model transparency?

What kinds of machine learning models are most relevant to credit underwriting?

An underwriting model sorts applicants into groups of people based on estimates of their relative likelihood of loan repayment using data about each individual applicant's financial history. Certain kinds of machine learning models were designed for and are particularly well suited to this kind of classification problem,[4] although each can vary in complexity and transparency: decision trees and tree ensembles, support vector machines, and neural networks.[5] Each model type is considered in turn:

- **Decision Trees and Forests:** Decision trees are algorithms for making decisions using a tree-like structure and have been used in various applications for several decades. In machine learning, decision trees are most commonly used for classification tasks.[6] To use a decision tree, we start at its "root node" and then make a series of steps along "branches" based on different attributes of an input data point. Each step is an "if-then" type question; for example, "if the applicant has more than three credit cards, take branch A, otherwise take branch B." After one or more branching steps, we arrive at a "leaf node" which gives us a final prediction (*e.g.*, "80% likely to default" or "10% likely to default"). For example, suppose we have developed a decision tree for predicting an applicant's probability of default on a loan. Our first step states "if the loan is less than $100k, take branch A, otherwise take branch B." The second step along branch A states "if the applicant has less than five credit cards, their default risk is low, otherwise their default risk is high." The second step along branch B might be different: "if the applicant has less than three credit cards, their default risk is low, and otherwise their default risk is high." Larger decision trees can have several branches that use different attributes of each applicant. It is common to develop several different decision trees and average their outputs to create a "decision forest."[7]

- **Support Vector Machines (SVMs):** Support vector machines are another type of algorithm that is frequently used in classification applications and may be useful in constructing underwriting models.[8] SVMs operate by finding "planes" among data points, which are essentially lines on a graph used to separate and classify a dataset into categories. In credit underwriting, for example, an SVM could create planes that categorize applicants as "likely to default" or "not likely to default" based on a variety of characteristics.[9] The goal of finding the best plane is to place a line at the maximum distance between data points, creating a more decisive categorization.[10] A "support vector" is a data point near those planes that ultimately impacts where the line is placed and oriented.[11] For nonlinear trends, planes can encircle or curve around specific data points to create categories. Research has shown that SVMs perform well on smaller datasets, and multiple studies have demonstrated potential for using SVMs for credit underwriting.[12] More recently, however, SVMs have gotten less attention in the context

of credit underwriting due to the emergence of other methods that deliver a better balance of performance gains, interpretability, and operational efficiency.[13]

» **Artificial Neural Networks (ANNs):** ANNs take data inputs and pass them through one or more layers of "neurons" before arriving at an output (prediction).[14] ANNs with multiple layers can represent nonlinear relationships between several variables, allowing them to uncover non-obvious, hidden relationships between variables. However, ANNs with more than a handful of neurons can be difficult to interpret. Limiting the number of layers or neurons in the network can improve an ANN's transparency, but may come with equally significant performance tradeoffs. For example, logistic regression can be represented as an ANN with a single layer of nodes – in other words, certain simple ANNs are similar and sometimes functionally equivalent to logistic regression. Performance-transparency tradeoffs have limited ANNs' use in applications like credit underwriting to date.[15] Additionally, neural networks can require vast amounts of data and are time intensive to train.[16]

What factors can affect a model developer's choice of machine learning algorithms?

A model developer's choice of algorithm or algorithms from available machine learning techniques will depend on a variety of factors, such as the nature and volume of data available for use; the desired goal (such as classification or regression); the level of predictive performance desired; business and regulatory requirements related to the application for which the model is needed; and firm practice and risk tolerance.[17] Further, the number of variables included, types of variables, representation of variables as numbers or percentages, or missing variables could affect the practicality of choosing one type of model or another. In practice, it may not be clear at the outset of model development which machine learning technique will produce the most effective model for a specific use, and developers may test several approaches on a preliminary basis to see what fits the available data and business requirements best.[18]

Apart from data considerations, model developers are also likely to focus on identifying what kinds of explanations are needed for their use case when developing a model, especially in credit underwriting where stakeholders need to be able to understand and explain the operations of the model. Unlike a financial crimes or fraud model that flags suspicious items for further review, underwriting models inform specific decisions that will have significant consequences if the predictions are wrong. Firms and their investors put capital at risk based on an underwriting model's predictions. Some applicants may be denied credit improperly if the model estimates a higher likelihood of default than it should, and others may face a range of long-lasting consequences if they default on a loan for which they were ill suited. Given these stakes, the prudential and consumer protection requirements in law and regulation are more exacting for consumer credit than even in other financial services applications.

Accordingly, the level of scrutiny that underwriting models undergo before being approved and while in use requires higher levels of transparency than fraud models typically do. That scrutiny requires that model developers provide more information about how an underwriting model was

developed and operates than might be required for other use cases. Thus, selecting interpretable models or pairing models with appropriate explainability techniques is an important part of algorithm selection and developing underwriting models.

In this context, two important operational realities for practitioners are worth mentioning. First, many uses of predictive models, including key financial services uses, rely on ensembles of multiple models that are composed of multiple techniques, tools, and technologies. For example, an automated predictive model that relies on traditional statistical techniques may nevertheless apply rules and segments that the model developer first identified through analysis of large data sets using machine learning. Accordingly, firms may use a combination of traditional and machine learning techniques for different components of the model building process.

Second, this is a moment of incredible richness and change in the tools and techniques available for modelling and analytics, including with respect to improving model transparency. The approach taken by any one model builder will in all likelihood reflect the nature and needs of the particular use case at a given time, subject to applicable data, infrastructure, business process, and regulatory limitations. But the toolkit available for predictive modelling and analytics is evolving quickly enough that the combinations of tools and techniques being used in particular contexts are also subject to change as new approaches emerge and are validated in practice.

Further Reading

Majid Bazarbash, FinTech in Financial Inclusion: Machine Learning Applications in Assessing Credit Risk, International Monetary Fund (May 17, 2019), available at https://www.imf.org/en/Publications/WP/Issues/2019/05/17/FinTech-in-Financial-Inclusion-Machine-Learning-Applications-in-Assessing-Credit-Risk-46883.

BLDS, LLC, Discover Financial Services, & H2O.AI, Machine Learning: Considerations for Fairly and Transparently Expanding Access to Credit (July 2020), available at https://www.h2o.ai/resources/white-paper/machine-learning-considerations-for-fairly-and-transparently-expanding-access-to-credit/.

Related FAQs

How are machine learning models developed?

Why is model transparency especially important in the context of AI and machine learning models?

Why is model transparency especially important in the context of machine learning models used for credit underwriting?

What legal and regulatory frameworks apply to the use of machine learning credit underwriting models?

What are latent features? How do they affect model transparency?

Latent features are variables or relationships that inform a model's prediction, but are not part of the training or input data or the prediction itself. They are generated by a machine learning algorithm from attributes in the dataset and serve as interim analyses that help determine the model's prediction. Latent features may result from simple combinations of input variables – not unlike the ratio of average monthly debt obligations to income used in traditional underwriting models – or more complex mathematical processes – such as calculations of the average variance

in monthly disposable income over various periods and economic conditions. Latent features, such as hidden layers in ANNs, are sometimes learned during model training. Developers can also manually calculate inferred features during pre-processing, for example using Principal Component Analysis or Linear Discriminant Analysis.

Latent features calculated by machine learning models and inferred features calculated by developers can be an important source of information in models. The inferred features designed by humans generally involve intuitive computations – such as credit utilization ratios. In other cases, the machine learning algorithm designs latent features that may or may not be intuitive. These kinds of latent features are not unique to complex or "black box" AI or machine learning models – in fact, latent features are commonly used with inherently interpretable models. However, where latent features are generated by an algorithm rather than a human, they can contribute to the challenge of explaining machine learning models. If a machine learning model relies heavily on latent features that are difficult to understand on their own, this further complicates the challenge of understanding the model output. For example, suppose that the three most important features in a machine learning model are latent features. Furthermore, suppose that all of these latent features depend on the applicant's total debt, number of credit cards, and credit score. In this case, it can be very difficult to understand how a single attribute affects the model's prediction.

The need for model transparency and the challenge of producing explanations of model behavior turn on understanding better how these latent features affect a model's predictions, especially in high-stakes areas such as credit underwriting, medical diagnosis and treatment, and criminal sentencing.[19] Given the potential of better predictive performance from machine learning models that rely on latent features, research on explaining such features is increasing in data science fields as varied as machine learning, natural language processing, and computer vision systems, as well as in economics, medicine, and psychology.[20]

Further Reading

Richard A. Berk, Susan B. Sorenson & Geoffrey Barnes, Forecasting Domestic Violence: A Machine Learning Approach to Help Inform Arraignment Decisions, Journal of Empirical Legal Studies (February 8, 2016), available at https://onlinelibrary.wiley.com/doi/abs/10.1111/jels.12098

Filippo Amato et al., Artificial Neural Networks in Medical Diagnosis, Journal of Applied Biomedicine (July 31, 2013), available at http://jab.zsf.jcu.cz/artkey/jab-201302-0001_artificial-neural-networks-in-medical-diagnosis.php

Alexander Amini et al., Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure, 2019 AAAI/ACM Conference, at 289-295 (January 2019)

Scott Zoldi, Adored and Damned: The Ambivalent Relationship to AI (February 22, 2021), from https://www.linkedin.com/pulse/_adored-damned-ambivalent-relationship-ai-scott-zoldi?trk=public_profile_article_view

Related FAQs

How are machine learning models developed?

What is model interpretability?

What is model explainability?

How can model developers improve the transparency of machine learning models?

A model developer will generally seek to enable the minimum degree of model complexity needed to deliver the level of performance required in a model that will work well in the expected conditions in which it will be used. A number of factors may constrain options for any one model, including limitations in law, regulation, and firm policy and resource and business process constraints. Several requirements applicable to consumer credit – such as model risk management, anti-discrimination requirements, and adverse action reporting – point firms in the direction underwriting models that they can describe and explain in a variety of ways to a variety of internal and external stakeholders.

A model developer may opt to build an inherently interpretable model or build a more complex model. An inherently interpretable model is one for which its operations and predictions can be sufficiently understood by inspection and without relying on secondary models or techniques to produce information about how the underlying model operates. For instance, the developer may produce a decision tree or a neural network, but will constrain the learning algorithm prior to training to increase the transparency of its predictions – for example by limiting the model to analyze only linear or monotonic relationships or by limiting certain model characteristics, such as the number of features or layers in a neural network or depth of a decision tree.[21] These constraints can produce machine learning models with transparency akin to traditional methods of statistical prediction, such as the automated logistic regression models commonly used in consumer lending, but may do so at the cost of performance that more complex machine learning models can deliver.

If the developer opts to use a "black box" model – one that is more complex and not inherently interpretable – he or she may use additional techniques to explain the operations and predictions of the underlying model. These *post hoc* explainability techniques are applied after model training and use additional algorithms to generate detailed information about how a machine learning model arrives at a prediction.[22] *Post hoc* methods include techniques like LIME,[23] SHAP,[24] and Integrated Gradients.[25] Explainability techniques work in a variety of ways. For example, techniques like surrogate models (such as simple linear regression or decision tree) approximate the behavior of the model they explain. Others calculate each variable's contribution to a predicted result by iterating the prediction in high volume with small changes to input variables to determine how those changes affect the underlying model's prediction.

Post hoc explainability techniques have the promise of letting firms capture the performance of more complex machine learning models while still delivering sufficient transparency to meet requirements applicable to the model's use case. However, the use of *post hoc* explainability techniques introduces a range of additional questions about when and in what circumstances the explainability technique's outputs can be trusted. These questions may arise because explainability techniques preserve information about key drivers of the model's prediction, but simplify or compress other information about the models that they describe. All explainability methods rely on a reference or basis of comparison, which may be explicitly stated or not. The choice of reference used to create an explanation can be as important as the choice of explainability method itself.

Explainability methods are often used in conjunction with various broadly applicable visualization techniques, such as Individual Conditional Expectation (ICE),[26] Accumulated Local Effects (ALE), or Partial Dependence Plots (PDP) plots,[27] to improve understanding of the information.

In choosing among explainability options, model developers must also give consideration to how the firm will address the particular concerns of policymakers and regulators: model performance for specific sub-groups such as protected classes (as defined in anti-discrimination regulations), sources of performance deterioration in adverse conditions (pursuant to prudential expectations), or explanations for particular kinds of decisions based on the model's predictions (as defined in adverse action notice requirements). Additional research on the capabilities and performance of various explainability techniques for specific applications is needed to understand these questions better.

Further Reading

Diogo Carvalho, Eduardo Pereira, & Jaime Cardoso, Machine Learning Interpretability: A Survey on Methods and Metrics, MDPI (July 26, 2019), available at https://www.mdpi.com/2079-9292/8/8/832

Patrick Hall & Navdeep Gill, An Introduction to Machine Learning Interpretability, O'Reilly Media (April 2018), available at https://pages.dataiku.com/hubfs/ML-interperatability.pdf

Christoph Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable (2019), available at https://christophm.github.io/interpretable-ml-book/

Florian Ostmann & Cosmina Dorobantu, AI in Financial Services, The Alan Turing Institute (2021), available at https://www.turing.ac.uk/research/publications/ai-financial-services

Related FAQs

Why is model transparency especially important in the context of machine learning models used for credit underwriting?

How are *post hoc* explainability techniques being used in practice?

Who needs information about how a credit underwriting model works?

What legal and regulatory frameworks apply to the use of machine learning credit underwriting models?

How can model developers build inherently interpretable machine learning models?

What kinds of *post hoc* explainability techniques can be used to improve model transparency?

How can model developers build inherently interpretable machine learning models?

When building an inherently interpretable model – sometimes referred to as a self-explanatory model – a developer can choose among a variety of approaches for limiting the model's complexity. Measures beyond limiting the scope of the input data include:

Monotonicity Constraints

Monotonicity constraints are imposed on a model to ensure one-directional relationships between the input data and the predictions of a model. When monotonicity constraints are imposed on a model, the relationship between a feature and the outcome always goes in the same direction – an increase in the feature value either always leads to an increase or decrease in the model output (such as the risk of applicant default). For example, on average, it is expected that an individual will have a higher salary with more years of work experience, holding everything else constant. This is an example of a monotonic relationship, and can be expressed by a simple linear regression model. However, adding salt to a savory dish presents an example of a non-monotonic relationship. A small amount of salt will generally make the dish taste better. However, after a certain point, adding salt will not improve the taste of the dish and, in fact, will make the dish taste worse. This is an example of a non-monotonic relationship, as the relationship is positive in some cases and negative in others, which means the relationship is not one-directional.

Monotonicity constraints smooth out the relationship between the feature and outcome. Take an example where a model finds that consumers who are using almost none of their available credit and those who are using almost all of their available credit tend to be higher risk than consumers in the middle. In the context of credit underwriting, without monotonicity constraints, there may be instances where an individual that is not using much of their available credit would be rejected, while an individual that is using more of their available credit would be accepted, even if everything else about these individuals is the same. A lender may find it difficult to explain this decision as the individual with the lower credit utilization would expect to be approved if an individual with a higher credit utilization is approved. If monotonicity constraints are imposed on a model, the constraints ensure that decisions are reached such that individuals with lower credit utilization rates are always approved compared to individuals with higher credit utilization rates. This leads to transparency in a model as the relationship between credit utilization and outcome is one-directional and more intuitive. However, smoothing out the relationship between variables to ensure monotonicity leads to the tradeoff that some information may be lost, especially in the case of non-linear relationships,[28] and as a result the model may be less accurate in order to be more transparent.

Credit underwriting models often use monotonicity constraints to improve model transparency and to enable creation of accurate, consistent information required for adverse action notices.

Sparsity

Regularization and associated techniques create sparse models by, for example, limiting the number of features used as inputs, or by limiting the number of weights in a neural network.[29] Sparsity can be achieved in various ways. For example, feature selection and engineering are ways to limit the model to the features that are the most relevant to predicting the target variable, which can improve predictiveness and stability and lead to more transparent models. In some cases, variables that are highly correlated are dropped to make the feature list smaller, which is another way to achieve sparsity.

Related FAQs

What is model interpretability?

What is model explainability?

How can model developers improve the transparency of machine learning models?

How can the capabilities and performance of explainability techniques be evaluated?

What kinds of *post hoc* explainability techniques can be used to improve model transparency?

In the last decade, data scientists have made considerable strides in developing methods to analyze complex machine learning models so that their operations and predictions can be understood and explained. These *post hoc* explainability techniques are rapidly evolving, especially as more evidence is gathered about their capabilities and performance in the context of specific applications, including credit underwriting. The following represent categories of explainability techniques most relevant to machine learning credit underwriting models:

- **Surrogate Models:** Surrogate models are "simple" models designed to approximate a more complex model's process as closely as possible.[30] Surrogate models require access only to the training data and the underlying model's prediction of the output variable, not the inner workings of the model. A surrogate model charts a simple path from input data to result by using the predictions of the complex model to train an interpretable model on the data. The resulting surrogate model can then be used to interpret the results of the black box model. Surrogate models can be "local" if they reflect model behavior near a specific data point or "global" if they reflect model overall behavior. One advantage of a surrogate model is that any algorithm type can be used to explain an original model. A linear model or decision tree can be a surrogate for a more complex model like a neural network. However, there is no guarantee that the surrogate accurately represents the more complex model or that the surrogate captures all of the important features in the original model that contributed to its predictions.[31]

- **Feature Importance:** Feature importance techniques evaluate how much individual variables contribute to a model's prediction. In these methods, data is usually perturbed or permuted – meaning it is purposefully distorted or altered in a variety of ways. The aggregated effect of those changes on the model's prediction speaks to how much a variable affects the model's predictions.[32] Popular methods for calculating feature importance values include LIME, SHAP, and Integrated Gradients. Feature importance or variable importance scores can be presented in charts with associated predictions, or they can be aggregated together and graphed for comparison.[33]

- **Example-Based Explanations:** Example-based explanations work by selecting individual data instances from the data set, unlike feature-importance techniques which

highlight variables that are likely to impact model output.[34] Example-based explanations are used to explain an individual prediction about a data point X (*e.g.*, a credit card applicant) by showing similar data points and their "ground truth" (*e.g.*, whether the other similar applicants defaulted). Suppose for example that a machine learning model predicts that a borrower is at high risk of default on their loan. An example-based method might find five similar borrowers in the training dataset, along with their "true" default results. If four of the five similar borrowers defaulted on their loan, this is one explanation for why the machine learning model predicted that default is likely.

Further Reading

Hall & Gill (2018)

Molnar (2019)

Related FAQs

What is model explainability?

What techniques can make machine learning models more transparent?

How are *post hoc* explainability techniques being used in practice?

What characteristics differentiate *post hoc* explainability techniques?

How can the capabilities and performance of explainability techniques be evaluated?

What characteristics differentiate *post hoc* explainability techniques?

Currently, there are no uniformly accepted definitions of model transparency or explainability or methodologies for measuring those qualities. However, *post hoc* explainability techniques can be categorized based on the following technical characteristics:[35]

- **Applicable Model Class:** *Post hoc* explainability techniques can be either model-specific or model-agnostic. *Model specific* explainability techniques are based on the unique internal elements of the particular machine learning technique being explained,[36] such as analyses based on the weights used in linear models or the features/thresholds used for splitting a tree, and are applicable only to specific types of machine learning models or classes of models. By using facts about how the model is constructed and other properties of the model, model specific techniques can answer explainability questions by directly analyzing the model instead of generating that information through analysis of a simplified version of the model.[37] Model specific methods are usually intrinsically interpretable methods like decision trees or regressions or they could be very specialized for neural networks and deep learning. In contrast, *model agnostic* techniques can be applied to explain any type of machine learning model and are applied after the model has been trained and makes a prediction.[38] Accordingly, these must make assumptions about the underlying model's structure that may or may not hold true. For example, many such techniques assume the model is linear or that variables are independent,

which may not be robust in the context of credit where a range of dependent variables like number of loans and delinquencies inform a model's prediction. Model agnostic methods usually correspond with *post hoc* explainability methods and include PDPs, ICE plots, ALE plots, LIME, or SHAP.

» **Technique Scope:** Explainability techniques may vary based on the scope of the information they provide about the underlying model's operation. Global explainability techniques will seek to describe a model's operation across an entire set of data and holistically characterize the analytical processes that describe how it functions when making predictions.[39] A local explainability technique will focus on specific areas of the model, features, or input data points to derive the importance of individual inputs to the model's prediction.[40] Some techniques use both kinds of explanations. For example, Shapley-based methods aggregate local explanations into an overall ranking of the importance of individual variables or features to the model's operation.

Further Reading

Diogo Carvalho, Eduardo Pereira, & Jaime Cardoso, Machine Learning Interpretability: A Survey on Methods and Metrics, MDPI (July 26, 2019), available at https://www.mdpi.com/2079-9292/8/8/832

Patrick Hall, On the Art and Science of Explainable Machine Learning: Techniques, Recommendations and Responsibilities (May 31, 2020), available at https://arxiv.org/pdf/1810.02909.pdf

Molnar (2019)

Related FAQs

What is model explainability?

What techniques can make machine learning models more transparent?

How are *post hoc* explainability techniques being used in practice?

What kinds of *post hoc* explainability techniques can be used to improve model transparency?

How can the capabilities and performance of explainability techniques be evaluated?

How can the capabilities and performance of explainability techniques be evaluated?

In the past decade, techniques to increase the transparency of machine learning models have been developed, but there is no established methodology for evaluating the utility of information produced by explainability techniques that use secondary models. Indeed, there may not even be a consensus about the questions we need to ask to assess whether and in what circumstances we can rely on information produced about a model's operation by *post hoc* explainability methods to satisfy applicable legal, regulatory, and firm policy requirements. *Post hoc* explainability methods tend to simplify or compress information about the underlying model's operation or provide general information rather than insights about specific aspects of the model, such as its treatment of protected classes. Even where high degrees of model transparency and explainability are possible, the operational demands of using such models – measured in time, expertise, and infrastructure

costs – may be prohibitive, especially when measured against the anticipated performance gains with respect to incumbent models.

Certain technical considerations provide a starting point for evaluating individual explainability techniques based on use case, priorities, and resources.[41] For example, the accuracy of an explainability technique's outputs can be measured in terms of consistency and stability – that is, whether the explanation produced is similar across similar applicants assessed by the same model or between different models producing similar predictions trained on the same data. How well the explainability technique approximates the underlying model – or its fidelity as measured in metrics like R scores for a surrogate model – can help establish how well the technique works. Finally, the overall complexity and specific data and computational demands of explainability techniques are also relevant – as these factors will increase computation time and cost and may heighten concerns about the trustworthiness of the information being produced.

However, these considerations may not fully speak to whether the information produced by current explainability techniques is sufficiently responsive to applicable legal, regulatory, and firm policy requirements. The fact that most explainability techniques necessarily simplify or compress information about the model they describe naturally raises questions about what information about the underlying model's operation we want the explainability technique to preserve and why. *Post hoc* explainability techniques such as SHAP and LIME have been designed to tell us which features of the model matter most for generating the predictions of the model. We might evaluate the usefulness of different machine learning explanation methods (such as LIME or SHAP) by comparing (a) how consistent their output is over different runs, and (b) whether their output leads to intuitive and actionable insights.

For instance, when we consider how explainability techniques can be used to meet requirements applicable to consumer lending, it is not clear that the questions that legal and regulatory requirements seek to answer can be satisfactorily addressed based only on knowing the 10 variables most important to generating a model's prediction. That information may not address why a model generates differential approval rates across protected classes or how to mitigate this effect. Similarly, knowing the 10 most important variables driving predictions in the training sample might not be informative about why a model suddenly deteriorates with the onset of different economic conditions or a shift in the applicant pool. In these instances, the specific requirements designed to serve distinct policy goals may require more and different information in order to enable responsible oversight of the model by stakeholders of varied backgrounds and expertise.

These questions speak to a critical need to make complex models sufficiently transparent to be comprehensible by a variety of stakeholders, each with their own level and type of expertise and their own need for information. A data scientist or credit risk expert will need different kinds of information about an underwriting model's operation than a financial services executive, a compliance manager, an examiner, an advocate, or a person who applied for credit.

Further Reading

Molnar (2019)

Kacper Sokol & Peter Flach, Explainability Fact Sheets: A Framework for Systematic Assessment of Explainable Approaches (January 2020), available at https://arxiv.org/pdf/1912.05100.pdf

On Evaluating Explainability Algorithms, International Conference on Learning Representations (2020), available at https://openreview.net/pdf?id=B1xBAA4FwH

Related FAQs

What techniques can make machine learning models more transparent?

Who needs information about how a credit underwriting model works?

What legal and regulatory frameworks apply to the use of machine learning credit underwriting models?

How can model developers improve the transparency of machine learning models?

How can model developers build inherently interpretable machine learning models?

What kinds of *post hoc* explainability techniques can be used to improve model transparency?

Endnotes

[1] Transparency as to the process of a model's construction involves technical and governance issues that are beyond the scope of these FAQs and will be examined elsewhere. See generally Paul B. de Laat, Algorithmic Decision-Making Based on Machine Learning from Big Data: Can Transparency Restore Accountability?, Philosophy and Technology (November 12, 2017), available at https://cdn.tc-library.org/Rhizr/Files/4367e301-0301-4e6f-b2d7-f6a54e794a83/03b26612-048b-4b3f-b506-4271a7f34664.pdf; Florian Ostmann and Cosmina Dorobanțu, AI in Financial Services, The Alan Turing Institute at 49 and 56 (2021), available at https://www.turing.ac.uk/research/publications/ai-financial-services (differentiating systems transparency and process transparency in AI systems).

What how do current models for credit underwriting work? In what ways are machine learning models different? Are they necessarily more complex?

[2] Manish Bhoge, Using the Artificial Neural Network for Credit Risk Management, Oracle (January 23, 2019), available at https://blogs.oracle.com/datascience/using-the-artificial-neural-network-for-credit-risk-management.

[3] Peter Carroll and Saba Rehmani, Point of View: Alternative Data and the Unbanked, Oliver Wyman (2017), available at https://www.oliverwyman.com/content/dam/oliver-wyman/v2/publications/2017/may/Alternative_Data_And_The_%20Unbanked.pdf.

What kinds of machine learning models are most relevant to credit underwriting?

[4] A classification algorithm learns how to categorize data into different buckets based on common or contrasting attributes. For example, in healthcare, an image recognition classification algorithm might classify an image as "cancerous" or "not-cancerous." On the other hand, regression methods can be used to predict the value of an outcome based on independent variables/attributes. For example, a linear regression model can be used to predict the price of a house based on relevant variables such as number of bedrooms/bathrooms, square feet, and median house prices in the area. Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar, Introduction to Data Mining (Second Edition), Pearson Education, Ch. 3 (February 2021), available at https://www-users.cs.umn.edu/~kumar001/dmbook/index.php.

[5] Majid Bazarbash, FinTech in Financial Inclusion: Machine Learning Applications in Assessing Credit Risk, International Monetary Fund (May 17, 2019), available at https://www.imf.org/en/Publications/WP/Issues/2019/05/17/FinTech-in-Financial-Inclusion-Machine-Learning-Applications-in-Assessing-Credit-Risk-46883.

[6] Peter Martey Addo, Dominique Guegan, & Bertrand Hassani, Credit Risk Analysis Using Machine and Deep Learning Models, MDPI (February 9, 2018), available at https://www.mdpi.com/2227-9091/6/2/38; Christoph Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable (2019), available at https://christophm.github.io/interpretable-ml-book/; Bazarbash (2019).

[7] Dinesh Bacham & Janet Zhao, Machine Learning: Challenges, Lessons, and Opportunities in Credit Risk Modeling (July 2017), available at https://www.moodysanalytics.com/risk-perspectives-magazine/managing-disruption/spotlight/machine-learning-challenges-lessons-and-opportunities-in-credit-risk-modeling; Leo Breiman, Random Forests, Machine Learning (October 2001), available at https://link.springer.com/article/10.1023/A:1010933404324.

[8] Bazarbash (2019); R. Y. Goh & L. S. Lee, Credit Scoring: A Review on Support Vector Machines and Metaheuristic Approaches (March 13, 2019), available at https://downloads.hindawi.com/journals/aor/2019/1974794.pdf.

[9] Bazarbash (2019).

[10] Bazarbash (2019); Sunil Ray, Understanding Support Vector Machine (SVM) Algorithms from Examples, Analytics Vidhya (September 13, 2017), available at https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/.

[11] Bambrick (2016).

[12] Hafiz A. Alaka et al., Systematic Review of Bankruptcy Prediction Models: Towards a Framework for Tool Selection, Expert Systems with Applications (March 2018), available at https://www.sciencedirect.com/science/article/pii/S0957417417307224; Sheng-Tun Li, Weissor Shiue, & Meng-Huah Huang, The Evaluation of Consumer Loans Using Support Vector Machines (May 2006), available at https://www.sciencedirect.com/science/article/pii/S0957417405001739.

[13] Goh & Lee (2019).

[14] Bacham & Zhao (2017).

[15] Scott Zoldi, Deep Dive: How to Make "Black Box" Neural Networks Explainable, FICO (January 14, 2019), available at https://www.fico.com/blogs/deep-dive-how-make-black-box-neural-networks-explainable; Rudin & Radin (2019); BLDS, LLC, Discover Financial Services, a& H2O.AI, Machine Learning: Considerations for Fairly and Transparently Expanding Access to Credit (July 2020), available at https://www.h2o.ai/resources/white-paper/machine-learning-considerations-for-fairly-and-transparently-expanding-access-to-credit/.

[16] Office of the Comptroller of the Currency, Credit Card Lending Handbook Version 2.0 at 17 (April 2021).

What factors can affect a model developer's choice of machine learning algorithms?

[17] Joseph Breeden, A Survey of Machine Learning in Credit Risk (May 30, 2020), available at: https://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID3616342_code647282.pdf?abstractid=3616342&mirid=1.

[18] Breeden (2020).

What are latent features? How do they affect model transparency?

[19] Alexander Amini et al., Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure, at 289-295 (January 2019).

[20] Daniel Ramage, Christopher Manning, & Susan Dumais, Partially Labeled Topic Models for Interpretable Text Mining (August 2011), available at https://dl.acm.org/doi/10.1145/2020408.2020481; Zhi Chen, Yijie Bei, & Cynthia Rudin, Concept Whitening for Interpretable Image Recognition, Nature Machine Intelligence (December 2020), available at https://arxiv.org/abs/2002.01650.

How can model developers improve the transparency of machine learning models?

[21] Molnar (2019).

[22] Diogo Carvalho, Eduardo Pereira, & Jaime Cardoso, Machine Learning Interpretability: A Survey on Methods and Metrics, MDPI (July 26, 2019), available at https://www.mdpi.com/2079-9292/8/8/832.

[23] Marco Tulio Ribeiro, Sameer Singh, & Carlos Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier (2016), available at https://arxiv.org/abs/1602.04938.

[24] Scott Lundberg & Su-In Lee, A Unified Approach to Interpreting Model Predictions, NIPS 2017 (November 25, 2017), available at https://arxiv.org/abs/1705.07874v2.

[25] Mukund Sundararajan, Ankur Taly, & Qiqi Yan, Axiomatic Attribution for Deep Networks (2017), available at https://arxiv.org/abs/1703.01365.

[26] Alex Goldstein et al., Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation, Journal of Computational and Graphical Statistics (2015), 24:1, 44-65, DOI: 10.1080/10618600.2014.907095, available at https://arxiv.org/pdf/1612.08468.pdf.

How can model developers build inherently interpretable machine learning models?

[27] Daniel W. Apley & Jingyu Zhu, Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models, Journal of the Royal Statistical Society: Series B (2020), 82: 1059-1086, available at https://doi.org/10.1111/rssb.12377.

[28] For example, a higher self-reported income may improve an applicant's credit risk assessment, but at some point too much self-reported income may be indicative of fraud.

What kinds of *post hoc* explainability techniques can be used to improve model transparency?

[29] Robert Tibshirani, Regression Shrinkage and Selection via the Lasso, Journal of the Royal Statistical Society at 267-288 (1996), available at https://www.jstor.org/stable/2346178.

[30] Molnar (2019).

[31] Hall & Gill (2018).

[32] Molnar (2019).

[33] Popular methods for calculating feature importance values include LIME, SHAP, and Integrated Gradients. Other methods for visualizing feature importance include Partial Dependence Plots (PDP) and Accumulated Local Effects (ALE) plots, which quantify how changes in one or two features affects the model output. Individual Conditional Expectation (ICE) plots show how changes to a single feature affects the model output for various data points.

[34] Susanne Dandl & Christoph Molnar, Counterfactual Explanations (2019), available at https://christophm.github.io/interpretable-ml-book/counterfactual.html.

What characteristics differentiate *post hoc* explainability techniques?

[35] Hall & Gill (2018).

[36] Carvalho, Pereira, & Cardoso (2019).

[37] Model agnostic methods can be applied to neural nets, but specific methods may work better for uncovering hidden layers, especially since model agnostic methods work from outside a model. See Molnar (2019).

[38] Carvalho, Pereira, & Cardoso (2019).

[39] Hall & Gill (2018).

[40] Hall & Gill (2018).

How can the capabilities and performance of explainability techniques be evaluated?

[41] Molnar (2019).