ExplainExplore: Visual Exploration of Machine Learning Explanations

Dennis Collaris*
Eindhoven University of Technology

Jarke J. van Wijk†
Eindhoven University of Technology

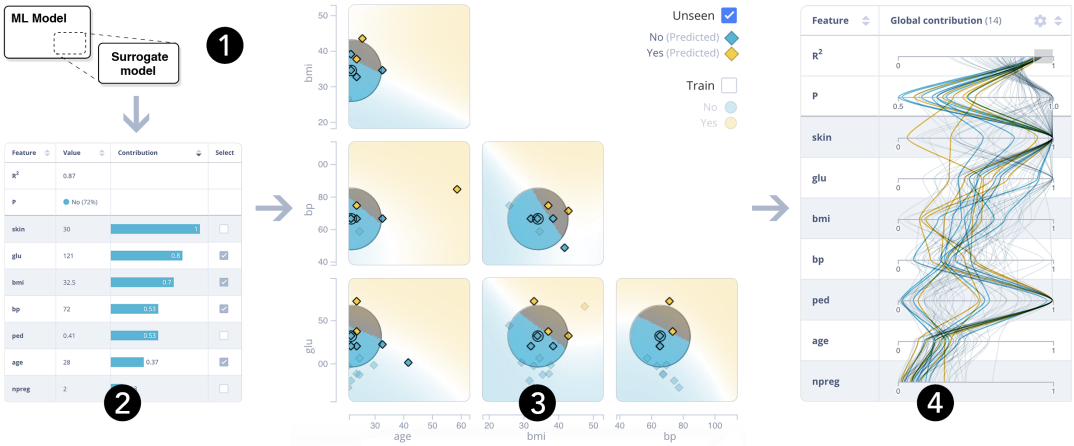


Figure 1: (1) A popular approach to explain a Machine Learning prediction is to extract a simple local approximation or *surrogate* model. (2) This surrogate provides contribution scores for every feature, yielding insight into why predictions are made. However, such an explanation can vary widely based on different parameters. (3) To visually explore different explanations, the local context around the instance is represented using a HyperSlice plot. Data Scientists can leverage their domain knowledge to determine which parameters and perturbations yield an explanation fitting their subjective preference. (4) To verify whether the quality of the explanation generalizes, an overview of all explanations for the dataset are shown. By brushing and linking subsets can be analyzed.

Abstract

Machine learning models often exhibit complex behavior that is difficult to understand. Recent research in explainable AI has produced promising techniques to explain the inner workings of such models using feature contribution vectors. These vectors are helpful in a wide variety of applications. However, there are many parameters involved in this process and determining which settings are best is difficult due to the subjective nature of evaluating interpretability. To this end, we introduce ExplainExplore: an interactive explanation system to explore explanations that fit the subjective preference of data scientists. We leverage the domain knowledge of the data scientist to find optimal parameter settings and instance perturbations, and enable the discussion of the model and its explanation with domain experts. We present a use case on a real-world dataset to demonstrate the effectiveness of our approach for the exploration and tuning of machine learning explanations.

Index Terms: Human-centered computing—Visualization; Computing methodologies—Machine learning.

1 Introduction

With the availability of large amounts of data, machine learning is getting more and more relevant. However, it is often hard to trust and understand the predictions made, as modern machine learning techniques are usually applied in a black-box manner: only the input (data) and output (predictions) are considered; the inner workings of these models are considered too complex to understand.

This lack of transparency can be a major drawback. For instance, the model may not perform adequately: even though it scores well on a test set, it could be based on biases, spurious correlations, and false generalizations [23]. Explanations can enable data scientists to identify such problems during model development.

Understanding the model also plays a crucial role in decision support. In applications such as fraud detection [6, 12], medical diagnosis [10, 29] or bankruptcy prediction [58], models make predictions that have a critical impact on real people. It is not sufficient to base decisions on the prediction score of the model alone [12].

Finally, various stakeholders may have questions about model predictions that require explanation. This got very relevant since the recently introduced General Data Protection Regulation (GDPR) enforces the "right to explanation" [17].

The field of explainable artificial intelligence (XAI) has recently gained a lot of traction as it aims to alleviate these issues. There are two main approaches to provide stakeholders with explanations that can be understood, justified and verified. First, an inherently interpretable model (e.g., a limited set of rules or a linear classifier) can be used that exchanges accuracy for understandability. Second, the reference model can be mimicked with a simpler explanatory (or *surrogate*) model, and explained in terms of this surrogate. We chose the latter approach as it is compatible with preexisting machine learning pipelines and hence widely applicable.

There are many decisions involved in creating explanations using a surrogate model. Parameters include the position, size, and shape of the sampling region, choice of surrogate model, and specific hyperparameters for that model. These choices have a significant impact on the resulting explanation, yet fitting values are rarely discussed. Previous work has shown that techniques may yield incongruent results if parameters are not chosen carefully [12].

By varying these parameters many different explanations can be generated. These may all be considered equally valid and useful [12]. Determining which of these explanations is best remains challenging, as there is currently no consensus on what a good explanation is [14, 21, 33, 57]. What *is* clear is that there is certainly a subjective element to interpretability: different stakeholders may have widely varying definitions of a good explanation [23].

*e-mail: d.a.c.collaris@tue.nl
†e-mail: j.j.v.wijk@tue.nl

2020 IEEE Pacific Visualization Symposium (PacificVis)
978-1-7281-5697-2/20/$31.00 ©2020 IEEE

2020 IEEE Pacific Visualization Symposium (PacificVis) 978-1-7281-5697-2/20/$31.00 ©2020 IEEE 10.1109/PacificVis48177.2020.7090

Due to the subjective nature of interpretability, we argue it is not possible to find the best explanation using purely automated methods. Rather, we propose using visual analytics to leverage domain knowledge to determine the quality of an explanation.

We present EXPLAINEXPLORE: a new approach for analyzing and understanding classification models using state of the art machine learning explanation techniques. The system allows on the fly customization of model and surrogate parameters. Based on that configuration, the data scientist can generate explanations to understand what features are relevant. The system does not encode strict assumptions about the qualities of an explanation, but leverages the domain knowledge of the data scientist to select the optimal explanation. Context for the explanation is provided by showing similar data points, and the effect of perturbations can be interactively explored. Finally, a global overview helps to spot general patterns that could indicate a problem with the model or explanation technique.

Our main contributions are: **1)** an explanation system applicable in many preexisting workflows by supporting a large variety of data sets and models, **2)** in contrast to current literature can be used even when no ground truth is available, and **3)** the system provides both local and global perspectives to tailor for different applications.

We collaborated with a leading insurance company in the Netherlands to obtain valuable insights into the relevance of explanations to data scientists. They provided feedback on our early prototypes and the use case described in this paper.

2 THE NEED FOR EXPLANATION

The value of explanation has been extensively reported in previous work [14, 33, 57]. We outline stakeholders and applications that benefit from explanations.

2.1 Stakeholders

Different stakeholders are involved in the development and application of explanations of machine learning models (see Fig. 2).

Data scientists select local surrogate models and related parameters. Based on this setup, predictions about a *subject* can be accompanied by explanations. *Decision-makers* can use these explanations to judge and understand predictions and communicate these to subjects. Explanations and predictions can also be directly forwarded to subjects, or serve a role in the communication between data scientists and decision-makers.

The work in the machine learning community mostly targets decision-makers, as decision support is a clear use case for explanations [47]. However, as this field revolves around technical and algorithmic advancements, the representation of the explanation is often not thoroughly considered. In contrast, the visualization community usually aims to create systems that expose a level of detail more suitable for data scientists. The scope of most work in this field is limited to the model development stage, as ground truth (which is not available after deployment) is often an integral part of the visualization.

In EXPLAINEXPLORE we target data scientists who work closely with decision-makers. Their familiarity with machine learning is vital for fine-tuning surrogate models, and their domain knowledge helps to assert the quality of an explanation.

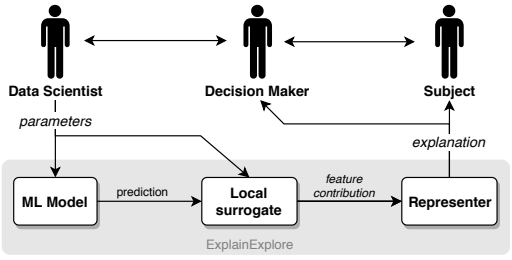


Figure 2: Data flow diagram of stakeholders.

2.2 Applications

Stakeholders can use explanations for a wide variety of applications. We identify four main categories:

Diagnostics

The model may not perform adequately, even though the model scores well on a test set. For instance, it could be based on biases, spurious correlations, and false generalizations [23]. This is demonstrated in the wolf-husky problem from Ribeiro et al. [47], where huskies are classified by detecting snow in the background of an image. Another problem is that traditional models are susceptible to concept drift: the test set generalization may not match with future unseen data. This problem was the reason for the failure of the Google Flu Trends model [8] and is very predominant in adversarial domains (e.g., spam detection, fraud detection).

Refinement

Apart from identifying issues with the model, explanations can also help to improve the model. Analyzing explanations for incorrect predictions can yield insights into how to increase predictive accuracy [4, 52] or remove irrelevant features.

Decision support

In applications such as fraud detection [6, 12], medical diagnosis [10, 29] or bankruptcy prediction [58], models make decisions that have a critical impact on real people. It is not sufficient to base decisions on the prediction score of the model alone [12]. To qualitatively ascertain whether desiderata such as fairness, privacy, and trust are met, explanations are required to verify their behavior [14].

Justification

Various stakeholders may have questions about predictions by the model. For instance, customers subject to predictions may request justification, or authorities may request information to check compliance. The latter got very relevant since the recently introduced GDPR enforces the "right to explanation" in Article 13 and 22 [17].

EXPLAINEXPLORE focuses on data scientists and supports them in all these applications, as depicted in the presented workflow (Fig. 3).

3 BACKGROUND

Various techniques in the category XAI have been proposed to explain machine learning models. The efforts range over multiple fields of research [21, 35, 48]. Here we focus on machine learning and visual analytics.

3.1 Machine learning

There are two main approaches in this field [21, 57]: either a model is used that is *inherently interpretable*, or an explanation is generated by means of a *surrogate* model.

Inherently interpretable models traditionally include linear models, decision trees and rules [16]. However, there are some recent advancements, like linear GA^{2}M models that deal with pairwise interactions [36] and algorithms to induce a concise set of decision rules [3, 15, 32]. For some domains, these types of models can yield predictions with an accuracy close to their complex counterparts, while remaining simple enough to interpret.

This is not always the case though, as simple models will always compromise on expressive power. They also require replacement rather than augmentation of preexisting machine learning pipelines. An alternative approach is to mimic the reference model with a simpler explanatory or *surrogate* model, and explain the reference model in terms of that surrogate. This allows using the full potential of the reference model: rather than compromising its accuracy, the faithfulness of the surrogate is reduced. Surrogate models can be any interpretable model, such as linear models [47] or decision rules [31]. However, as such a simple surrogate cannot perfectly match

the reference model, the explanation yielded from it is only a rough approximation of the real behavior.

In pursuit of better explanations, XAI recently directed its focus from global [7, 49] to local [5, 37, 47, 53] surrogate models. Rather than compromising the faithfulness of the surrogate, the generality of the surrogate is reduced. This means that the scope of the surrogate is limited to part of the reference model, resulting in a simple and *locally* faithful explanation.

3.2 Visual analytics

As interpretability is an inherently subjective concept, many authors from the visualization community have built systems to support machine learning tasks. There is a variety of works ranging different applications as the ones mentioned in Section 2.2.

Approaches like ModelTracker, Squares and work by Alsallakh et al. [1, 2, 46] help *diagnose* the model by highlighting disparity between different predictions. Other approaches compare regression output with ground truth [44] or aim to evaluate fairness [59].

In order to *refine* models, systems such as Manifold, MLCube and RegressionExplorer [13, 26, 61] enable the comparison of different models. Alternatively, approaches such as BaobabView [39, 54] enable interactive construction of models. Post-hoc approaches instead enable intuitive model configuration [34, 41].

Decision support & justification are big topics in visualization. However, most approaches only use data analysis and only use machine learning for recommendations [11, 22, 25]. Decision support using machine learning techniques to provide explanations is a recent development, and as such, the amount of work is scarce [9, 12, 50].

These visual analytics systems often tailor for specific algorithms. Neural networks have received the most attention with systems visualizing or projecting neuron weights [27, 35, 38, 45, 60] or highlighting important regions contributing to a prediction [9, 24, 40]. A few model-agnostic such as Prospector [29] and What-if tool [59] exist, and mainly focus on hypothesis testing.

3.3 ExplainExplore

Compared to traditional visual analytics approaches that only use the prediction of a model, ExplainExplore provides more information by using state-of-the-art machine learning explanation techniques. Rather than considering these explanations as a fixed statistic, we allow interactive tuning of explanation-related parameters to ensure it meets the subjective preference of the stakeholders. Fine-tuning machine learning explanations is, to the best of our knowledge, a novel topic.

The scope of most visual analytics approaches is limited to the model development stage, as ground truth (which is not available after deployment) is often an integral part of the visualization. ExplainExplore does not require ground truth and can thus also be used with machine learning models in production.

Many systems focus only on global [1, 2, 11, 25, 34, 35, 38, 39, 41, 45, 46] or local [9, 12, 29] explanation, but few combine the two [23, 27, 30]. These perspectives are complementary [23] and hence are both supported in our system.

To achieve this, the system uses a technique similar to HyperSlice [55], which has previously been applied to regression models [44]. We extend this method by supporting multiple classes and categorical variables, facilitating machine learning model comparison by exploiting the locality of surrogate models, and offering various options for showing only the data points local to the shown slice.

4 Problem Definition

We interviewed six data science teams at a large insurance firm (dealing with problems ranging from churn prediction, product pricing, recruitment optimization to debtor management) to figure out how they could benefit from explanations. In our study, we found:

- Most teams that were interested in explaining their models used supervised classification for decision making. Contrary to regression, classification models often play a critical role in decision making (e.g., having a significant impact on people) and are much harder to interpret.
- The data scientists used a wide variety of models, created using different technologies, languages, and toolkits.
- They typically use multivariate, tabular input data with a mix of numerical and categorical data.
- The different teams had very mixed preferences for global or local insights.

Our goal is *to assist data scientists in understanding these models*. This understanding will drive many applications as mentioned in Section 2.2. To facilitate these applications, the system should support a wide variety of datasets and models, and enable on the fly customization of model and surrogate parameters. Based on that configuration the data scientists can generate explanations to understand what features are relevant.

The system is aimed at *data scientists* who work closely with decision-makers. Their familiarity with machine learning is vital for fine-tuning surrogate models, and their domain knowledge helps to assert the quality of the model and the explanations given.

4.1 Data

The data for the system consists of a multivariate tabular dataset and a classification model. Ground truth is not required but can be provided to train different types of models within the system. All other data used for explanations and visualizations (e.g., surrogate model and feature contribution vectors) are generated on demand.

4.2 User tasks

We derived a list of user tasks to account for needs in a variety of explanation-driven use cases based on our interviews with six data science teams and previous work in this area [57]:

T1 Adjust the model for performance or better explainability.
T2 Adjust the surrogate for faithfulness and simplicity.
T3 Look up how much a feature contributed to a prediction.
T4 Look up quality metrics for model, prediction and surrogate.
T5 Select instances with noteworthy explanations, such as good or bad faithfulness, or specific feature contribution values.
T6 Query the model sensitivity to feature perturbations.
T7 Compare surrogate and reference model to assert the faithfulness of the explanation.
T8 Explore the effect of input perturbations on prediction and explanation.

To support these tasks we designed two workflows shown in Fig. 3. Arrows depict the common way of interaction, starting from the initial configuration of model and surrogate. Uppercase words summarize the most important actions performed in each view, and user tasks are annotated. In the first workflow (blue) analysis starts with a single prediction and provides more detail with context, whereas the second (orange) starts from a global overview and allows investigating smaller subsets. Workflows correspond to applications of explanations introduced in Section 2.2.

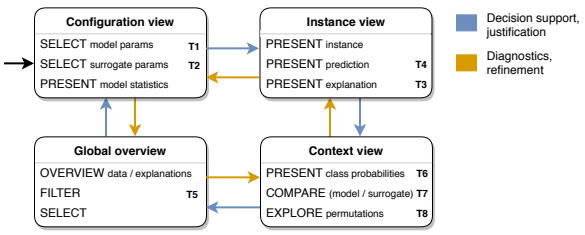


Figure 3: Workflows of ExplainExplore.

Figure 4: Graphical user interface of EXPLAINEXPLORE showing the main components: (A) the configuration view to select and refine model and surrogate parameters; (B) the feature view explaining the prediction by (B1) showing feature-wise contribution values for one case using the chosen configuration, and (B2) providing a global overview to identify whether the selected instance was classified similar to other instances, or whether the model has 'strategies' specific to a subset of instances, and (C) the context view showing neighboring instances, the class probability gradient to assert whether the surrogate model is accurate, and allowing to explore the effect of instance perturbations.

5 EXPLAINEXPLORE

In this section, we first introduce the used explanation technique. Next, we describe how we translated the workflows (Fig. 3) into an interactive visual analytics system. Fig. 4 provides a high-level overview of our approach. We enable users to inspect the model at three different scales: locally, globally and context. One of the data points can be selected as the current focal point for the local view (Fig. 4-B1). The global view provides an overview of (a selection of) the given unseen data points (Fig. 4-B2). Finally, the context view shows an environment of the current focal point (Fig. 4-C).

5.1 Explanation technique

Modern XAI techniques derive model explanations through local surrogates. Our system uses this technique as a basis for understanding machine learning models. A popular local surrogate technique is LIME [47]. The technique generates synthetic data (i.e., *transfer* data) in a local neighborhood around the instance to be explained. Next, the transfer data is labeled by the reference model. By fitting a simpler surrogate regression model to this labeled transfer data, it will mimic the decisions of the reference model in that local region of space (i.e., *sampling region*). The size of this region (i.e., *generality*) and distance kernel used are important parameters whose values should be carefully chosen.

To evaluate the fit of the surrogate, standard goodness-of-fit metrics from machine learning can be used. In this work, we use the coefficient of determination (R^2) as it is a ubiquitous metric familiar to data scientists. The fit of the surrogate is also referred to as *faithfulness* of the surrogate or explanation to the reference model. With an R^2 value of 1, the explanation explains the model perfectly, for any lower value details are lost due to simplification.

LIME uses rejection sampling to generate the transfer dataset, which is inefficient on high dimensional data [28]. We use a modified version of LIME that, rather than rejection sampling, samples transfer data directly from the sampling region. Also, we use a distance kernel with a bounded support instead of a Gaussian kernel.

Our system supports a variety of linear and tree-based surrogate models, in addition to linear regression provided by LIME. Feature contribution vectors are extracted using coefficients for linear models [47] and local increments for tree-based models [42]. To be able to compare feature contributions across different surrogate model types, we use the normalized absolute contribution values.

5.2 Configuration view

The primary goal of the configuration view (Fig. 4A) is to set up the machine learning problem for further analysis. The dataset and classifier can be selected and configured by following the traditional machine learning workflow: data selection, data preparation, modeling, and evaluation. At any time during the analysis, this view can be revisited to adjust the configuration.

First, a dataset can be selected. To allow the system to be applicable for a wide variety of preexisting setups, any tabular dataset with numeric or categorical values can be added (given that train, test, and unseen partitions are separately provided). Basic data preparation is supported by options for feature selection, and data scaling. Data scaling is enabled by default as some classifiers require scaled data.

Next, a classifier model can be selected (Task **T1**). The system supports all classifiers from the Python `scikit-learn` toolkit [43] as well as classifiers from other languages (e.g., R) or applications (e.g., KNIME, SAS Enterprise Miner) using the PMML format [20]. Model hyperparameters are automatically parsed and configurable. The chosen model is fitted to the training partition of the provided dataset on-the-fly, and the performance of the model on the test dataset is displayed (F_1 score, Task **T4**).

Finally, a surrogate model can be selected. This step is an addition to the traditional machine learning workflow and forms the basis for the explanation technique. Options include linear models and shallow tree-based models. Other important parameters affecting the explanation can be configured: model hyperparameters (e.g., regularization constant for linear models, or depth for tree-based models), the size of the sampling region (*generality*) and sampling distance kernel (Task **T2**). Changing these values will immediately update other views, enabling the data scientist to assert the impact of these parameters on the explanation.

5.3 Feature view

The feature view (Fig. 4B) is introduced to explain the prediction by showing feature-wise contribution values obtained using the chosen configuration. The view is formatted as a table with multiple columns in two categories: local, conveying information about the currently selected instance, and global, showing an overview of explanations for all unseen data instances. A local or global oriented workflow can be achieved by reordering columns of the feature view.

Two rows are prepended to the table showing the prediction and R^2 values. These values help to ascertain whether the explanation is sensible, or perhaps misleading. These rows will always appear on top, whereas the rest of the table can be sorted on demand.

Local columns

Local columns in the feature view (Fig. 4-B1) show feature-wise properties for a selected data point.

The value column represents feature values as text, as this is a familiar representation for data scientists, and enables to quickly compare values with other systems they use. Double-clicking brings up an input field to update a feature value manually (Task **T8**).

The contribution column encodes the feature contribution vectors as a vertical bar chart. Values range between 0 and 1, where longer bars mean more contribution to the prediction. The bars are colored according to the predicted class. This view helps the expert to quickly spot which features play a role in the prediction for the selected instance (Task **T3**).

To assert the correctness of these contribution values, information on the prediction certainty (Fig. 4-B1.1), and R^2 value (Fig. 4-B1.2) are shown (Task **T4**). If the prediction is not very certain, the explanation may not be trustworthy; an explanation with a low R^2 score (i.e., a bad surrogate fit) could also be misleading. To alert the expert, low values for these metrics are colored red.

Global column

The global column (Fig. 4-B2) provides a high-level overview of the data. We tried histograms, violin plots and small multiples, but settled on a Parallel Coordinate Plot (PCP) as it was best for conveying clusters in the data. Two types of overviews can be shown: unseen data feature values and the corresponding contribution values.

The global *value* overview encodes the distributions of feature values. This helps to ascertain whether an instance is an outlier, and helps to find interesting clusters in the unseen data that can be selected for further analysis (Task **T5**).

The global *contribution* overview encodes contribution vectors. This helps the data scientist to identify whether the selected instance was classified similar to other instances, and whether the model has 'strategies' (clusters in the contribution vectors and polylines in the PCP) specific to a subset of instances. The expert can use this view to find instances that have diverging explanations, which could indicate a problem with the model or explanation technique.

The global column includes two additional axes for the prediction certainty and R^2 score of the surrogate model. This enables the selection of subsets based on how certain the model was of that prediction, and how faithful the surrogate explanations are to the reference model. Using these axes the data scientist can select subsets or instances for which automated explanation techniques yield misleading or incorrect explanations (Task **T5**).

Line colors correspond to the predicted class of the instance and a thicker black line indicates the selected instance in the PCP. The lines in the PCP are curved by default. This makes it easier to spot the intersections with the axes. Using a smoothly graduating curve also allows experts to discern individual paths better, due to the Gestalt principle of good continuation [19]. When sorting the feature table view by the global column, the rows are sorted by the mean feature value of the unseen data.

Selection of instances is enabled by brushing the axes of the PCP. The selected cases are highlighted in the PCP, as well as linked to the scatter plots in the context view.

5.4 Context view

The context view (Fig. 4C) provides more context for the selected instance and corresponding explanation. Nearby unseen data instances are shown, as well as the class probability of the reference classifier (global) and surrogate model (local). The expert can use this to assert whether the surrogate model is locally faithful to the reference model (Task **T7**), explore neighboring instances and introduce instance perturbations to improve the explanation (Task **T8**).

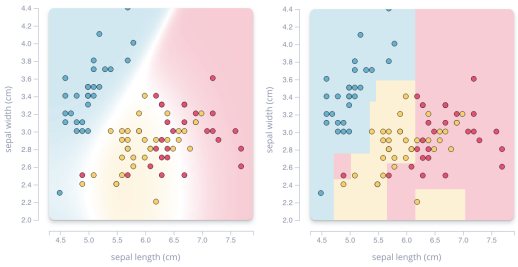
Class probability plot

Class probabilities of machine learning models *in two dimensions* can be visualized as a two-dimensional heatmap. This technique is model-agnostic and can be applied to any model returning a class probability. If this is not supported, the system will substitute a class probability of 1 for the predicted class (as shown in Fig. 5b).

Given a chosen color $\mathbf{c}_k = (r_k, g_k, b_k)$ for class $k \in K$, white color $\mathbf{w} = (1,1,1)$ and predicted class probability $\hat{y}_k$, the color $\mathbf{c}$ for a pixel in the heatmap is computed as

$$\mathbf{c} = \mathbf{w} - \sum_{k \in K} \alpha_k (\mathbf{w} - \mathbf{c}_k), \quad \alpha_k = \max\left(0, \frac{\hat{y}_k - \tau}{1 - \tau}\right), \quad \tau = \frac{1}{|K|} \quad (1)$$

An example is shown in Fig. 5a. White colors in the figure show areas where every predicted class is equally likely. The class probability plot enables the expert to discover which perturbations to a data point would lead to a different prediction (Task **T6**). In the example, a scatter plot of training data points is overlaid. The color of a point corresponds to the true class of that instance. If the color does not match the class probability color in the background the point is incorrectly classified.

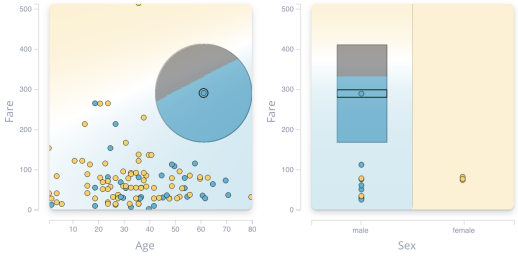


(a) Naive bayes (b) Decision tree

Figure 5: Class probability plot of two models trained on the Iris dataset.

The same technique can be used to visualize the class probabilities of the surrogate model. However, as the local surrogate is trained on a smaller sampling region, we mask the plot to only show that region. The class probability plots are overlaid to enable easy comparison between reference and surrogate model. This helps the expert to ascertain the quality of fit of the surrogate, and hence the quality of the explanation (Task **T7**). An example is shown in Fig. 6a. The surrogate is trained to distinguish one class from all others. Hence the black color represents all other classes in the plot. To increase contrast, colors are discretized by default to show only the color for the predicted class, and black for all others. This can be configured.

For categorical features, the plot is split into regions for each category, as shown in Fig. 6b. The surrogate is overlaid as a rectangle.



(a) Numerical 'Age' (b) Categorical 'Sex'

Figure 6: Overlaid class probability plots of reference and surrogate model, trained on the Titanic dataset. The size of the surrogate overlay (left: circle, right: rectangle) corresponds to the sampling region size.

HyperSlice plot

To deal with higher-dimensional models, we use a similar approach to HyperSlice [55]. Axis-aligned slices intersecting the selected instance (or *focal point*) are displayed as small multiples. An example is shown in Fig. 4C. We chose this encoding to retain meaningful axes for interaction, as opposed to alternatives like multidimensional projections. Unlike the two dimensional example, these slices do not comprise the entire feature space, but they do enable to understand the local neighborhood around the current focal point.

For datasets with a large number of features, not all slices can be shown at once. In this case, the system enables selecting features to be displayed in the table view.

Data

For each two-dimensional slice, a projection of the data points can be shown. Both the unseen and training data can be separately shown and hidden. They are indicated by different glyphs, and colored according to the predicted class, and ground truth respectively.

Showing all data may however be misleading when comparing against the class probability plot, as it is sampled only in the slice rather than all feature space. To account for this, an option is included to filter points based on their distance to the shown slice.

To this end, the Gower similarity coefficient [18] is computed, which is a popular distance metric for mixed data types that combines Manhattan and Dice distance for numerical and categorical features respectively. Given the normalized distance for every point to the slice, the alpha value of each point is obtained by applying a distance kernel h to the distance, given a threshold $\theta \in [0,1]$. Any distance kernel can be used; in the system we use:

$$h_{uniform}(d) = \begin{cases} 1 & d \leq \theta \\ 0 & d > \theta \end{cases}, \quad h_{triangular}(d) \quad = \max\left(0, 1 - \frac{d}{\theta}\right) \tag{2}$$

The latter kernel will linearly fade in points as they get closer to the slice. We considered encoding distance in alternative attributes like size and using focal blur [51]. However, we found the former did not work well with occlusion, and the latter too resource-intensive for large datasets.

Interaction

The context view enables the expert to directly manipulate parameters that affect the explanation provided by the system (Task **T2**).

The focal point can be dragged to introduce perturbations to the selected instance. In this process, the class probability plot serves as guidance to find relevant regions and the expert can observe the effect on the prediction and explanation (Task **T8**). Alternatively, a data point can be clicked on to move the focal point directly to that instance. The class probability plots, and feature table columns update in real-time when the focal point is moved.

Second, the size of the sampling region can be controlled with the mouse wheel or using a slider in the configuration view. This will affect how general the resulting explanation is. Large sampling regions will yield a general explanation (applicable to many instances) but will not be faithful to the reference model. Small regions will be faithful but might overfit to insignificant details of the reference model. The optimal value differs per instance and needs to be determined manually, the system can be used to find a compromise.

Finally, the shape of the distance kernel for the sampling region can be configured with the mouse wheel while holding down the Alt key or using a slider in the configuration view. This affects how the transfer data set is generated (to which the surrogate model is fitted). The effect of the choice of distance kernel on the explanation has gotten little attention so far. Authors of the popular explanation technique LIME [47] mention the choice has no significant impact, but Lundberg and Lee [37] chooses a specific (and different) kernel for LIME to satisfy optimally constraints, and argue that it *is* relevant.

To enable experimenting with distance kernels, the system includes a configurable trapezoid kernel. This is a smooth interpolation between a uniform and triangular kernel, defined as

$$h_{trap}(d) = \begin{cases} 1 & d \leq \sigma \\ \frac{1-d}{1-\sigma} & \sigma \leq d \leq 1 \end{cases} \tag{3}$$

where $\sigma \in [0,1]$ is the *smoothness* parameter. By controlling this variable the probability of generating a transfer data point drops linearly when getting closer to the edge of the area of interest. The advantage of this kernel is that the described region is well specified, as opposed to the Gaussian kernel used in LIME.

6 Use case

We collaborated with a large insurance company to validate our approach in a real-world use case. We found that data scientists were enabled to obtain explanations to identify problems with their model and justify predictions, even when automated techniques fall short.

Debtor management is a crucial part of maintaining a healthy financial administration. The process involves lots of manual labor: staying in contact with various clients, sending reminders and, in extreme cases calling in official debt collectors. Machine learning can help to speed up the process and to prevent resource-intensive debt-collection operations that are unlikely to be effective. However, as the model only provides a prediction, the verification of such a model and justification of decisions is challenging.

The goal of the experiment was to help data scientists from the debtor management department to understand the models they developed. They have extensive domain knowledge and worked closely with the decision-makers at their department. The team created a binary classifier to predict the effectiveness of a debt-collection operation. It is a Random Forest (50 trees) trained on a dataset of 60,000 instances with 16 features (9 numerical and 6 categorical). They provided 250 unseen data points for our experiment.

To validate our approach, two data scientists of the team participated in a user study. The session consisted of two parts: during the first part they were tasked to use the global-oriented workflow to diagnose problems with their model and find possible refinements. The task during the second part was to use the local-oriented workflow to support decisions made by domain experts. The session took four hours, including 30 minutes of introduction. Except for the introduction, only the data scientists used the system. The thinking aloud method was applied throughout the experiment, and all audio and screen activity were captured for further analysis. Figures in this section are taken directly from the screen capture, but are anonymized to protect sensitive information.

Part 1: global-oriented workflow

For this part, we configured a model that was similar to the model they built: it is the same type of model (Random Forest) and has roughly the same F_1 score. We reordered the feature table view columns to show the global columns first. The data scientists were tasked to evaluate if this model behaves as they expected. They had an expectation of the global importance of features based on their own Random Forest.

Diagnostic insight

After having selected a ridge regression surrogate model, they selected an instance to see its feature contributions. They were surprised to find that the two features they expected to be the most important (A and B) were not important at all. Furthermore, the most contributing feature C for this instance was one they deemed redundant and recently removed in a newer version of their dataset. They hypothesized that this instance could just be an outlier. As the class probability for this instance was low (P=0.6), they expected the model might use different features compared to very certain predictions.

To verify their hypothesis they looked at the global contribution column in the feature view. This showed that the different contribution values were persistent across all unseen data. They concluded this model was behaving differently from their own. Next, they argued that the model might infer the values of features A and B from other features. They discussed features that might correlate to A and B in great detail. After this, they used the feature selection option in the configuration view to remove those features from the model. The expectation was that features A and B would have higher contributions. This was not the case; their contributions remained relatively unchanged. They were surprised to find that other features also had predictive power, as they believed only a few features (such as A and B) were important. This insight could help them to refine their current model by leveraging more or different features.

Refinement insight

While they were considering features one by one, the experts realized that a particular feature D (which is only true for a small number of instances) might be an important indicator for the class "effective". They decided to check if the model used this effect, and brushed in the global value PCP to find instances with a specific value. The context view showed that the predicted class for all these points is the same, verifying the effect. However, for these points, the feature had a low contribution. This means that even though the model predicts the cases correctly, the feature was not used for these predictions (Fig. 7). By ensuring that the model uses this feature more effectively, the model could be refined.

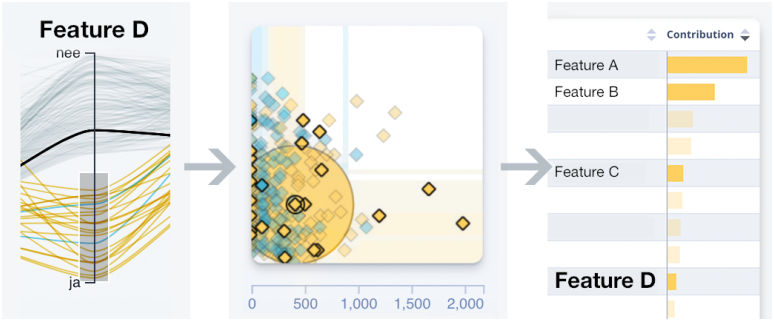


Figure 7: A subset of points is selected, almost all points are predicted as the same class, however feature D does not have high contribution to the predictions.

Part 2: local-oriented workflow

Next, we reordered the feature table view columns to show the local instance columns first. The data scientists were asked to support the decision-making process for the debt-collection operations.

Explanation with high R^2

The data scientists found a point (yellow) amidst a group of points of a different class (blue) that they wanted to investigate further.

They configured a Ridge regression surrogate and considered the feature contributions. There were only four features with a significant contribution. They substantiated that the feature with the highest contribution was important because the value was very high compared to the rest of the data, which increased their trust in the explanation. However, from the context view, they noticed that considering this feature was not enough to explain why the point was classified differently from its neighbors.

The second most contributing feature was a category unique to this point: all neighboring points had a different value. They mentioned "this feature is the deciding factor for the prediction in this neighborhood". Here they used the explanation as guidance to form their hypothesis. They leveraged their domain knowledge to obtain a more logical explanation. They mentioned they would explain their decision-maker in terms of these two features primarily.

Improving explanation with surrogate model choice

The experts note that the explanations seem to be less faithful for their data and model compared to standard simple datasets used in machine learning education. This makes sense, as more complex models are difficult to explain. To improve the explanation they switched from a linear to a decision tree surrogate model. The R^2 axis in the global overview clearly showed that the faithfulness of explanations of unseen data increased and had less variance. This can be explained because tree-based models are better suited to approximate other tree-based models.

Another instance was selected. By considering the class probability plot of the surrogate, it was clear that the surrogate fit improved because it fits non-linear behavior (shown in Fig. 8). The explanation for this instance was faithful and clear.

Figure 8: Switching to a tree-based surrogate improved the R^2 and hence explanation faithfulness, as the decision tree is more suited to fit non-linear boundaries from the Random Forest.

Improving explanation with perturbations

Finally, they selected an instance that was more challenging to explain: it had a class probability of 1 and its neighborhood in the local plot was mostly the same class (Fig. 9). Switching between surrogate models did not improve the fit. To improve the explanation the experts moved the focal point closer to a region with points from a different class. They found that a small change in feature values yielded a significantly better explanation (R^2=0.52 to 0.84). The features that were important also changed. Here the experts used the focal point as a probe to find the nearest faithful explanation for this instance.

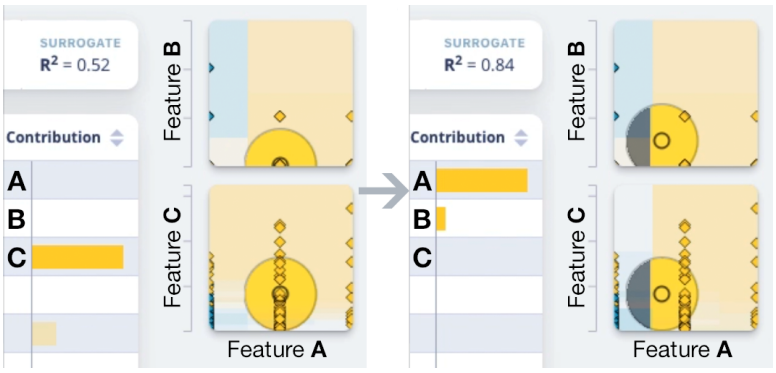


Figure 9: On the left, the R^2 value is low (0.52) and feature C has the highest contribution. A slightly lower value for feature A results in a much higher R^2 value (0.84) and feature A becomes dominant.

Reflection

The data scientists were very positive about the system. They mentioned "Especially for exploration this will really lead to insights" and "Very useful to see if the model is looking at the right aspects; if it behaves logically". The system enabled them to get more insights into their model and data, which was the purpose of this use case.

We got some important insights during the use case. Even though we proposed a global workflow, the data scientists found it more intuitive to start with a single instance and build up from there (our local workflow). As mentioned in Section 4, this preference was mixed for different data science teams.

When selecting instances in the global contribution overview, they were unable to determine the feature values of those selected instances. As soon as they switched from the global contribution overview to the global value overview, their selections were cleared. Even though the values could be inferred from the context plot, this was not straight forward.

Finally, they would have liked some more specialized features to focus on one particular instance. First, a better way to keep track of a selected instance. Once an instance is selected in the current implementation, the focal point moves to that instance. However, this focal point changes during interaction and experts might loose track of the instance they started at.

They also used the system in ways we did not expect. For instance, they used the feature selection creatively: to test the effect of leaving out features on the model. By leaving out features they expected different features to get higher contribution values.

7 Discussion

The basic ideas presented in this paper are all simple in nature: 1) offering broad options for model and surrogate configuration, 2) brushing and linking parallel coordinate plots to analyze subsets of data and explanations, 3) locally representing class probabilities with HyperSlice, 4) overlaid class probability plots to visually ascertain model fit and 5) enabling free navigation of feature space to enable what-if analysis. However, we have shown that combined they form a strong visual encoding that enables data scientists to understand their model by generating explanations.

In contrast to most visual analytics systems that present a single explanation, our system enables the exploration of many explanations. This way domain knowledge of data scientists can be leveraged to discover explanations fitting their subjective preference.

We adopted a hybrid approach, combining global and local representations rather than a single perspective. This helps experts to find global patterns, drill down to a single explanation, but also check whether local explanations are applicable to a larger subset.

Finally, in contrast to visual analytics approaches that are limited to the model development stage (as ground truth is often an integral part of the visualization), our system was built to augment rather than replace preexisting machine learning pipelines. It supports data without any provided ground truth and is compatible with various types of classification models and data.

Scalability

Even though we can visually represent many features in the feature view, there is a practical limitation to the number of features that can be represented in the system. Many features would make it difficult to find and compare axes in the global overview PCP, and tedious to select features to be displayed in the context view. However, for most machine learning problems high dimensional data is ill-advised [56], and the projects of data scientists we interviewed all had acceptable numbers of features. We also support feature selection to reduce the dimensionality ahead of the analysis.

Next, the number of categories the system can represent is limited. Because the class probability plot in the context view is split into regions for categorical variables, it becomes difficult to compare class probabilities and different models if the feature has more than 10 categories. This is not very common for business-related applications. As categories are always assumed to be non-ordinal, a class probability plot for two categorical variables is also not able to guide the expert as well as a slice with a numerical variable can. This is a direction for future work, as the explanation technique also is less effective for purely categorical data.

Finally, the number of unseen data points that can be represented is of course limited. For a large number of instances, the data points in the scatter plots will overlap, and lines in the PCP will start occluding making it challenging to identify local effects in the global overview. The number of unseen data points also affects performance as an explanation needs to be generated for every instance. For this purpose, the number of unseen data points is kept around a thousand. The presented use case shows this is sufficient for understanding a model, and an improvement over automated techniques that can only explain single data points at a time.

Optimization was needed in order to support fluid interaction with the system. Every second, tens of thousands of predictions are made to compute the class probability gradients. Hence, our system relies on the model to be able to generate predictions quickly. Machine learning models are often only computationally expensive during training, but generating classifications is relatively quick. Even so, a hundred layer deep learning model or tree ensembles with thousands of trees will be too slow for the system to be interactive. To alleviate some performance issues the system reduces interactive updates during brushing the PCP and moving the context view focal point as soon as a complex dataset or model is loaded.

After a minute of initial loading time, the system remains interactive using a 30-dimensional dataset (UCI Breast Cancer Wisconsin), a 15x100 layer fully connected multi-layer perceptron, running on a mid-range laptop with an Intel Core i5 (I5-7360U) processor and integrated Intel Iris Plus 650 graphics card.

Limitations & future work

The use of domain knowledge in our system creates the risk of introducing bias. This is not specific to our approach but is inherent when using expert domain knowledge. To counteract this, we show faithfulness metrics, and deliberately left out surrogate feature selection to prevent obvious tampering. However, some risk still remains.

As the system introduces many degrees of freedom for explanations, it may also be overwhelming to new users. We offer suitable defaults for these options, and expect data scientists (our target audience) are sufficiently familiar with parameter optimization.

As we previously mentioned we incorporated metrics in order to assert the quality of explanations, but these metrics are not a perfect proxy for trustworthiness. Such metrics remains elusive, hence finding an 'optimal' explanation with our system does too.

In order to further counteract bias, a direction for future research is to convey the uncertainty of explanations per feature. Even though our system allows to see how explanations vary for input perturbations, directly conveying this uncertainty would be very helpful.

Next, the relevance of different sampling regions for surrogate models is unknown. Our techniques assume circular regions around an instance (as does LIME), but some rule-based techniques [49] consider rectangular regions instead. Our system enables experimentation with different distance kernels, but any in-depth analysis on the relevance would be interesting.

8 Conclusion

In this work, we presented ExplainExplore: an interactive explanation system to assist data scientists in understanding their models. It is built to support a wide variety of different data sets and machine learning models. We demonstrated the value of the system with a use case at a large insurance firm. The participants effectively used explanations to diagnose a model and find problems, identify areas where the model can be improved, and support their everyday decision-making process. For cases where automated techniques fall short, they were able to refine surrogate parameters to improve the explanation and found the closest good explanation that made intuitive sense. We hope that this technique helps to alleviate some of the issues with current explanation techniques, to diagnose problems with the model such as unfairness, and help experts to make informed decisions.

ACKNOWLEDGMENTS

The authors would like to thank Achmea BV for their generous collaboration and feedback. This work is part of the research programme Commit2Data, specifically the RATE Analytics project with project number 628.003.001, which is financed by the Dutch Research Council (NWO).

REFERENCES

[1] B. Alsallakh, A. Hanbury, H. Hauser, S. Miksch, and A. Rauber. Visual methods for analyzing probabilistic classification data. *IEEE transactions on visualization and computer graphics*, 20 (12):1703–1712, 2014.

[2] S. Amershi, M. Chickering, S. M. Drucker, B. Lee, P. Simard, and J. Suh. Modeltracker: Redesigning performance analysis tools for machine learning. In *Proceedings of the 33rd Annual ACM Conf. on Human Factors in Computing Systems*, pages 337–346. ACM, 2015.

[3] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, and C. Rudin. Learning certifiably optimal rule lists. In *Proceedings of the 23rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 35–44. ACM, 2017.

[4] M. Ankerst, M. Ester, and H.-P. Kriegel. Towards an effective cooperation of the user and the computer for classification. In *Proceedings of the sixth ACM SIGKDD int. conf. on Knowledge discovery and data mining*, pages 179–188. ACM, 2000.

[5] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

[6] A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten. Improving credit card fraud detection with calibrated probabilities. In *Proceedings of the 2014 SIAM int. conf. on data mining*, pages 677–685. SIAM, 2014.

[7] O. Bastani, C. Kim, and H. Bastani. Interpreting blackbox models via model extraction. *arXiv:1705.08504*, 2017.

[8] D. Butler. When Google got flu wrong. *Nature News*, 494 (7436):155, 2013.

[9] H. S. G. Caballero, M. A. Westenberg, B. Gebre, and J. J. van Wijk. V-Awake: A visual analytics approach for correcting sleep predictions from deep learning models. In *Eurographics Conf. on Visualization (EuroVis), June 3-7, 2019, Porto, Portugal*. Eurographics Association, 2019.

[10] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare. *Proceedings of the 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining - KDD '15*, pages 1721–1730, 2015. doi: 10.1145/2783258.2788613.

[11] J. Choo, H. Lee, J. Kihm, and H. Park. iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 27–34. IEEE, 2010.

[12] D. Collaris, L. M. Vink, and J. J. van Wijk. Instance-level explanations for fraud detection: A case study. *ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, pages 28–34, 2018.

[13] D. Dingen, M. van't Veer, P. Houthuizen, E. H. Mestrom, E. H. Korsten, A. R. Bouwman, and J. J. Van Wijk. RegressionExplorer: Interactive exploration of logistic regression models with subgroup analysis. *IEEE transactions on visualization and computer graphics*, 25(1):246–255, 2019.

[14] B. Doshi-Velez, Finale; Kim. Towards a rigorous science of interpretable machine learning. In *arXiv:1702.08608*, 2017.

[15] R. Evans and E. Grefenstette. Learning explanatory rules from noisy data. *Journal of AI Research*, 61:1–64, 2018.

[16] A. A. Freitas. Comprehensible classification models – a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1): 1–10, 2013. ISSN 19310145. doi: 10.1145/2594473.2594475.

[17] B. Goodman and S. Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, 2017.

[18] J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871, 1971.

[19] M. Graham and J. Kennedy. Using curves to enhance parallel coordinate visualisations. In *Proceedings on Seventh Int. Conf. on Information Visualization, 2003. IV 2003.*, pages 10–16. IEEE, 2003.

[20] A. Guazzelli, T. Jena, W.-C. Lin, and M. Zeller. The PMML path towards true interoperability in data mining. In *Proceedings of the 2011 workshop on Predictive markup language modeling*, pages 32–38. ACM, 2011.

[21] R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti. A survey of methods for explaining black box models. *arXiv preprint arXiv:1802.01933*, 2018.

[22] D. Guo. Visual analytics of spatial interaction patterns for pandemic decision support. *Int. Journal of Geographical Information Science*, 21(8):859–877, 2007.

[23] F. Hohman, A. Head, R. Caruana, R. DeLine, and S. M. Drucker. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2019.

[24] F. Hohman, H. Park, C. Robinson, and D. H. Chau. Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. *arXiv preprint arXiv:1904.02323*, 2019.

[25] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang. ipca: An interactive system for PCA-based visual analytics. In *Computer Graphics Forum*, volume 28, pages 767–774. Wiley Online Library, 2009.

[26] M. Kahng, D. Fang, and D. H. P. Chau. Visual exploration of machine learning results using data cube analysis. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, page 1. ACM, 2016.

[27] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. P. Chau. A cti v is: Visual exploration of industry-scale deep neural network models. *IEEE transactions on visualization and computer graphics*, 24(1):88–97, 2017.

[28] J. H. Kotecha and P. M. Djuric. Gibbs sampling approach for generation of truncated multivariate gaussian random variables. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 3, pages 1757–1760. IEEE, 1999.

[29] J. Krause, A. Perer, and K. Ng. Interacting with predictions: Visual inspection of black-box machine learning models. *ACM Conf. on Human Factors in Computing Systems*, pages 5686–5697, 2016. doi: 10.1145/2858036.2858529.

[30] J. Krause, A. Perer, and E. Bertini. A user study on the effect of aggregating explanations for interpreting machine learning models. In *ACM KDD Workshop on Interactive Data Exploration and Analytics*, 2018.

[31] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154*, 2017.

[32] B. Letham, C. Rudin, T. H. McCormick, D. Madigan, et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[33] Z. C. Lipton. The mythos of model interpretability. *ICML Workshop on Human Interpretability in Machine Learning*, 2016.

[34] M. Liu, S. Liu, X. Zhu, Q. Liao, F. Wei, and S. Pan. An uncertainty-aware approach for exploratory microblog retrieval. *IEEE transactions on visualization and computer graphics*, 22(1):250–259, 2016.

[35] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE transactions on visualization and computer graphics*, 23(1): 91–100, 2017.

[36] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD int. conf. on Knowledge discovery and data mining*, pages 623–631. ACM, 2013.

[37] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4768–4777, 2017.

[38] Y. Ming, H. Qu, and E. Bertini. RuleMatrix: visualizing and understanding classifiers with rules. *IEEE transactions on visualization and computer graphics*, 25(1):342–352, 2018.

[39] T. Mühlbacher and H. Piringer. A partition-based framework for building and validating regression models. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1962–1971, 2013.

[40] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev. The building blocks of interpretability. *Distill*, 2018. https://distill.pub/2018/building-blocks.

[41] J. G. S. Paiva, W. R. Schwartz, H. Pedrini, and R. Minghim. An approach to supporting incremental visual data classification. *IEEE transactions on visualization and computer graphics*, 21 (1):4–17, 2015.

[42] A. Palczewska, J. Palczewski, R. M. Robinson, and D. Neagu. Interpreting random forest classification models using a feature contribution method. In *Integration of reusable systems*, pages 193–218. Springer, 2014.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[44] H. Piringer, W. Berger, and J. Krasser. Hypermoval: Interactive visual validation of regression models for real-time simulation. In *Computer Graphics Forum*, volume 29, pages 983–992. Wiley Online Library, 2010.

[45] P. E. Rauber, S. G. Fadel, A. X. Falcao, and A. C. Telea. Visualizing the hidden activity of artificial neural networks. *IEEE transactions on visualization and computer graphics*, 23 (1):101–110, 2017.

[46] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams. Squares: Supporting interactive performance analysis for multiclass classifiers. *IEEE transactions on visualization and computer graphics*, 23(1):61–70, 2017.

[47] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[48] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing*, 268:164–175, 2017.

[49] I. Sanchez, T. Rocktaschel, S. Riedel, and S. Singh. Towards extracting faithful and descriptive representations of latent variable models. *AAAI Spring Syposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*, 1:4–1, 2015.

[50] T. Spinner, U. Schlegel, H. Schafer, and M. El-Assady. explAIner: A visual analytics framework for interactive and explainable machine learning. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2019.

[51] J. Staib, S. Grottel, and S. Gumhold. Enhancing scatterplots with multi-dimensional focal blur. In *Computer Graphics Forum*, volume 35, pages 11–20. Wiley Online Library, 2016.

[52] S. Stumpf, V. Rajaram, L. Li, W.-K. Wong, M. Burnett, T. Dietterich, E. Sullivan, and J. Herlocker. Interacting meaningfully with machine learning systems: Three experiments. *Int. Journal of Human-Computer Studies*, 67(8):639–662, 2009.

[53] R. Turner. A model explanation system. In *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th Int. Workshop on*, pages 1–6. IEEE, 2016.

[54] S. Van Den Elzen and J. J. van Wijk. BaobabView: Interactive construction and analysis of decision trees. In *IEEE Conf. on Visual Analytics Science and Technology (VAST)*, pages 151–160. IEEE, 2011.

[55] J. J. van Wijk and R. van Liere. Hyperslice. In *Proceedings Visualization'93*, pages 119–125. IEEE, 1993.

[56] M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. In *Int. Work-Conf. on Artificial Neural Networks*, pages 758–770. Springer, 2005.

[57] D. Weld and G. Bansal. The challenge of crafting intelligible intelligence. *Communications of ACM*, 2018.

[58] D. West, S. Dellana, and J. Qian. Neural network ensemble strategies for financial decision applications. *Computers & operations research*, 32(10):2543–2559, 2005.

[59] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics*, 2019.

[60] T. Zahavy, N. Ben-Zrihem, and S. Mannor. Graying the black box: Understanding DQNs. In *Int. Conf. on Machine Learning*, pages 1899–1908, 2016.

[61] J. Zhang, Y. Wang, P. Molino, L. Li, and D. S. Ebert. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE transactions on visualization and computer graphics*, 25(1):364–373, 2019.