Using Visual Analytics to Interpret Predictive Machine Learning Models

Josua Krause JOSUA.KRAUSE@NYU.EDU
New York University Tandon School of Engineering, Brooklyn, NY 11201 USA

Adam Perer ADAM.PERER@US.IBM.COM
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 USA

Enrico Bertini ENRICO.BERTINI@NYU.EDU
New York University Tandon School of Engineering, Brooklyn, NY 11201 USA

Abstract

It is commonly believed that increasing the interpretability of a machine learning model may decrease its predictive power. However, inspecting input-output relationships of those models using visual analytics, while treating them as black-box, can help to understand the reasoning behind outcomes without sacrificing predictive quality. We identify a space of possible solutions and provide two examples of where such techniques have been successfully used in practice.

1. Introduction

There is growing demand among data scientists to generate and deploy predictive models in a variety of domains so that the patterns unearthed from massive amounts of data can be leveraged and converted into actionable insights. Predictive modeling is defined as the process of developing a mathematical tool or model that generates an accurate prediction (Kuhn & Johnson, 2013). Data scientists often turn to machine learning, where the goal is to create predictive models based on information automatically learned from data with ground truth. However, these machine learning techniques are often black-boxes and may be selected based only on performance metrics such as high accuracy scores, and not necessarily based on the interpretability and actionable insights of the model.

Recently, we have worked on a variety of techniques to make predictive models more interpretable by bringing humans-in-the-loop through visual analytics. In this paper, we provide initial reflections on interpretability and the role visual analytics plays in this space based on our experience building such solutions. We first reflect on the role and meaning of interpretation and model transparency, then we reflect on the role of visual analytics in this space, then we describe two visual analytics systems we developed for model interpretation, and finally we conclude by proposing interesting questions for further research.

2. Why and when is interpretation needed?

It is important to start this discussion by clarifying that interpretation may not always be necessary in machine learning. There are plenty of situations in which building a model, testing and refining it, and finally putting it in production is absolutely appropriate (*e.g.*, chess playing or face recognition). It is also important to point out that interpretation is necessarily a human activity and, as such, it may be costly and error prone. When, then, do we need human interpretation? Why do we need to involve humans in the machine learning process? Without loss of generality, we have identified three main opportunities/needs in which interpretability is a highly desirable feature:

1. **Data understanding and discovery.** Machine learning is typically used as a tool to make predictions but it does not have to be used exclusively for this purpose. Machine learning models can also be used as a way to help understand and observe complex realities in providing abstractions that can be used by humans to enable interpretation and discovery.

2. **Trust building and accountability.** Some mission-critical situations, e.g., when models make important decisions about human beings, it is important to have a better understanding of what a model does and why. Increasing model interpretation and transparency can play a role in increasing trust and accountability.

3. **Model comparison and diagnostics.** Model developers often need to understand where models fail

2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 2016), New York, NY, USA.

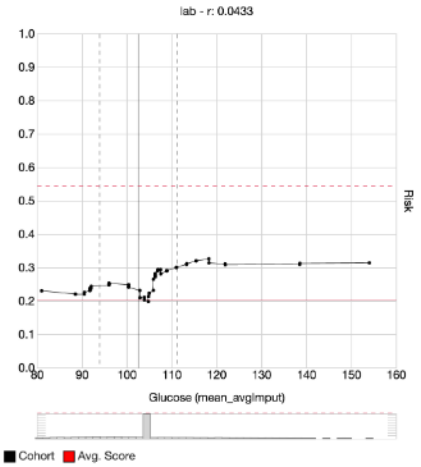

Figure 1. Debugging model performance using partial dependence. Instead of a direct relationship between higher Glucose lab values (x-axis) and higher risk scores (y-axis) the model predicts a low risk for average Glucose lab values. The histogram below the plot, showing the distribution of the values found in the input data, indicates that most patients have the average value. Since missing values are imputed using the average Glucose value the valley in the plot can be explained by the outcome independence of this value due to the high number of missing values.

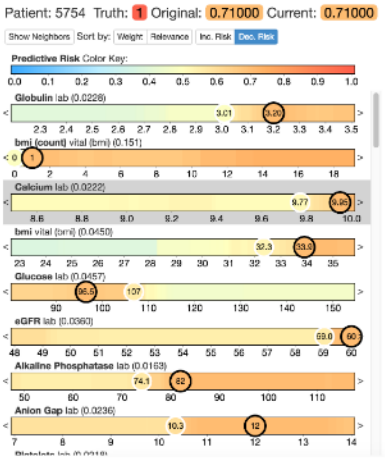

Figure 2. Identifying changes to features that reduce the risk of a high risk patient. The features are sorted by decreasing impact to make large sets of features manageable. The background color of each feature indicates the predicted risk for this value. Impossible values, if known, appear grayed out and the slider snaps back to the last possible value if selected.

> to make correct decisions and also how multiple models differ. This is another circumstance in which interpretability can play a positive role.

It is important to notice that many subjective properties play a role in model interpretation: *e.g.*, plausibility, comprehension, trustworthiness, actionability.

3. Model Transparency, Representation, and Interpretability

While a machine learning technique is often defined as being more or less interpretable, it is important to point out that it is hard to assign a level of interpretability to a whole class of techniques. For instance, *classification rules* are considered as a highly interpretable method but: how does one interpret the results of classification rules when the number of rules is very high (e.g., in the order of hundred or even thousands)? Similarly, *neural networks*, are commonly regarded as having low interpretability, yet, recent visualization methods allows to look into some of the internal decisions the network makes which increase the interpretability of the model (Yosinski et al., 2015; Xu et al., 2015; Zeiler & Fergus, 2013).

We therefore propose to make the following distinctions. First, we propose to distinguish between *model structure* and *model representation*. As we will explain with our examples, it is possible through visual analytics to represent a model even by not having access to its internal logic or structure. Therefore, we propose that its interpretability cannot be defined exclusively by what specific training method was used or what internal representation/structure has been derived. Second, we propose to describe models as having different degrees of transparency, from fully opaque to fully transparent and define their representations as more or less interpretable.

This observation is particularly important when we consider the commonly held belief that interpretability and accuracy in machine learning are at odds (Breiman, 2001). While we cannot definitely refute such statement here, we believe it is important to consider that this may actually be a false dichotomy. If models can be interpreted by looking at input/output behavior, disregarding what internal structures produce such behavior, then it is possible to reconcile the need of using highly accurate models and yet retain interpretability.

4. The Role of Visual Analytics in Interpretation

Visual analytics uses the power of visual perception as a way to involve humans in solving complex problems with data. Such human involvement, broadly speaking, can be desirable for two reasons: (1) because humans are better than machines in solving some tasks; (2) because human understanding and interpretation is desired. By designing perceptually effective visual representations (Munzner, 2014; Treisman, 1985; Ware, 2004) we can enable fast, accurate, and hopefully trustworthy interpretation of machine learning models.

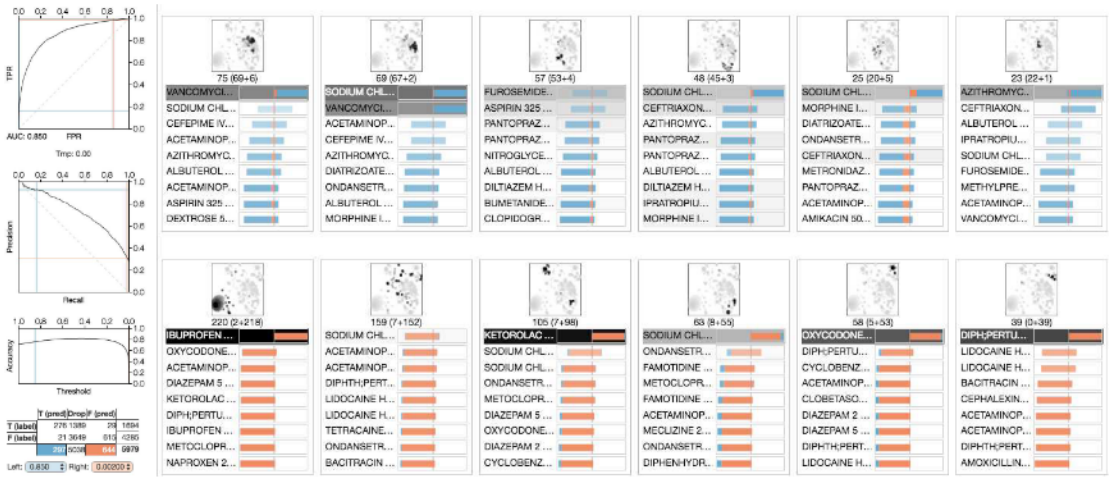

Figure 3. Using class signatures to describe groups of patients admitted to the hospital because of different medications. Each column represents one group (*cluster*-step) whereas each row shows the amount of patients in this group taking a particular medication (the bar from the middle towards the right shows the percentage of patients taking the medication; the bar towards the left shows not taking medication). The color of the bar shows the distribution of the true outcome labels as found in the input data. The background of the rows shows the discriminativeness of a medication (dark being more discriminative *wrt.* all other clusters; *rank*-step). Above each group a *t-SNE* projection of the items shows its relation to the other groups. ROC, precision-recall, and accuracy curves are shown on the left to facilitate selecting two thresholds to filter high signal patients (*contrast*-step).

We identify two generic modalities that can be used for model interpretation with visual analytics:

1. **Visualizing Model Structure (White-Box).** For transparent models, e.g., *decision trees*, one option is to use visualization to represent the structure built through the training method. Several examples exist in this area, especially for *decision trees* and *rules* (*eg.,* (van den Elzen & van Wijk, 2011; Yang, 2005; Liu et al., 2006)).
2. **Visualizing Model Behavior (Black-Box).** Another option is to use visualization as a way to look at the *behavior* of the model by looking exclusively at the relationship between input and output.

Following, we focus exclusively on the second case.

4.1. Visualizing Model Behavior (Input/Output)

Model interpretation through visualization of input/output behavior of a model has a number of interesting properties, and it has received so far, comparatively, less attention than white-box approaches.

The most interesting property of this approach is that it does not depend on what specific method has been used to train a model and, as such, it has the big advantage of being extremely flexible and generic. Furthermore, by creating methods that enable interpretation of models by looking exclusively at their behavior, we can study the role of interpretation independently from model representation.

We have identified three main mechanisms through which model behavior can be observed and analyzed:

1. **Item(s) to outcome.** In this modality, the user "probes" the model by creating input examples with desired properties and observing what output the model generates (e.g., an input image obtained from a camera or a patient descriptor obtained by specifying a given set of properties).
2. **Single feature to outcome.** In this modality, the user inspects and observes the relationship between one feature at a time and how its values relate (correlate) to the outcome. In our first example below (**Prospector**), we show how this can be done in practice.
3. **Multiple features to outcome.** Finally, in this modality, the visualization aims at representing the relationship between many features and their values and how they related to the outcome. In our second example below (**Class Signatures**), we show one way this idea can be realized in practice.

An additional aspect worth mentioning in relation to these three mechanisms, is how the data necessary to observe input/output behavior is obtained. Here, we have identified three main options (not mutually exclusive): *training data*, *test data (hold-out)*, *simulated data*.

While the first two types of data are very common in machine learning training and validation steps, we notice that simulated data is much less common.

With simulated data we mean data that is synthetically generated (and as such may not belong to training or test data) by letting the user specify its properties. One example of this situation is given in the example below (**Prospector**), in which, in a disease prediction task based on electronic health records, the user can specify a "fictional" patient by dragging sliders that define his or her values.

Being able to work with *simulated data* seems to be a particularly useful and promising direction of research when we consider the idea of probing machine learning models through visual analytics.

5. Prospector

Prospector is a novel visual analytics system designed to help analysts better understand predictive models (Krause et al., 2016). **Prospector** aims to support data scientists to go beyond judging predictive models solely based on their accuracy scores by also including model interpretability and actionable insights. It leverages the concept of partial dependence (Friedman, 2001), a diagnostic technique that was designed to communicate how features affect the prediction, and makes this technique fully interactive.

Figure 1 shows how partial dependence can be used to debug machine learning models in **Prospector**. In this example imputation of missing values created unexpected behaviour of the inspected classifier. Partial dependence is given by

$$pdp_f(v) = \frac{1}{N}\sum_{i}^{N} pred(x_i) \text{ with } x_{if} = v$$

where N is the number of rows in the input matrix x, $pred$ is the prediction function that takes one input row, a feature vector, and returns a prediction score, and f is the feature used to compute the partial dependence plot. The formula computes the average outcome over all input rows while changing the value of feature f to the input value v for each row x_i. The original input data is kept fixed. This allows for observing the influence of f on the prediction scores. Unlike generalized additive models, *eg.,* (Caruana et al., 2015), this technique is model agnostic.

Prospector also supports localized inspection, so users can understand why certain data results in a specific prediction, and even lets users hypothesize new data by tweaking values and seeing how the predictive model responds. Users can interactively tweak feature values and see how the prediction responds, as well as find the most impactful features using a novel model agnostic local feature importance metric that only depends on partial dependence.

Figure 2 shows the prediction inspection portion of the **Prospector** UI, which allows users to examine the features

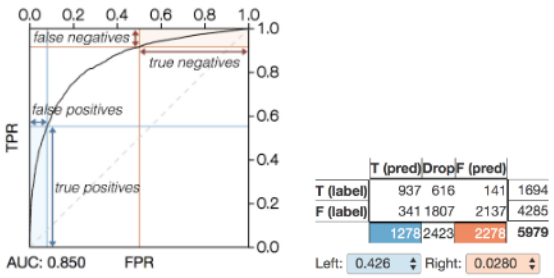

Figure 4. Using the receiver operating characteristics curve (ROC-Curve) for determining filter thresholds. The curve shows how the true positive rate (TPR; y-axis) and the false positive rate (FPR; x-axis) changes with decreasing thresholds (the threshold values are not shown in the graph). The extended contingency matrix on the right shows the actual numbers.

contributing to the prediction of a selected data point. All of the features partial dependence bars are shown in a scrollable list, with the data's feature values set with circular labels. Users can drag the circular label to change the value of any feature and see the prediction change in real-time. Users can change the sort order of the partial dependence bars by using the buttons at the top. In addition to sorting by the feature weight and relevance as determined by the predictive model if available, users can also sort according to our local feature importance and impactful changes. If impactful changes are chosen as the sort order, the suggested changes to each feature are indicated with a white circular label in the partial dependence bar.

6. Class Signatures

With **Class Signatures**, we propose a visual analytics workflow to interpret predictive associations between a large set of binary features and a binary target. For this we use a 4 step pipeline: *model*, *contrast*, *cluster*, and *rank*, and a visual analytics interface that allows the end user to detect and interpret such associations. After modeling the predictive associations using a binary classifier we leverage the prediction scores with two user defined thresholds, one for positive cases and one for negative cases, to focus only on data items with a strong predictive signal, increasing contrast. Then, we cluster both positive and negative examples *separately*. This groups together data points that have the same predicted outcome and a similar configuration of values. Finally, we rank each feature in the computed clusters using discriminative analysis across *all clusters*.

For interpreting the results with visual analytics, we use **Class Signatures** as shown in Figure 3. As our input features are binary in nature we show in the class signatures

how consistently present a feature is in a cluster. This is indicated by bars growing both to the right (percentage of which feature is present) and the left (percentage of which feature is not present). In combination with the discriminative measure of the features (computed as gini-importance; the shade of feature backgrounds is visually encoded, so darker means more discriminative) users can formulate rules that explain predictions for different sub-groups of data items.

The proposed workflow allows for a more fine-grained analysis of the driving factors of a predictive task than using commonly used feature importance techniques. This is due to the observation that many phenomena have multiple underlying reasons for the same result. Thus an explanation is needed that distinguishes which features were actually responsible for given data points. **Class Signatures** provide this distinction in the form of user interpretable rules.

7. Conclusion

We have provided an initial characterization of how visual analytics can be used for model interpretation, with a focus on visualizing input/output behavior rather than model structure. To exemplify these approaches, we presented two practical examples of systems we built to understand the reasoning of classification algorithms while treating them as black-box. As such, these examples provide initial anecdotal evidence that we do not necessarily need to accept lower prediction performance in order to gain in interpretability (Breiman identifies an inverse relation between model performance and interpretability of a machine learning model (Breiman, 2001)). Of course, there is much future work to do as the space of possible solutions in analyzing and understanding input-output relationships of machine learning models using visual analytics is largely unexplored. We identify two major research trajectories to develop these ideas further: (1) develop and validate more solutions that exploit this specific modality and (2) study human interpretation through user studies aimed at better defining interpretation and establishing metrics useful for comparison of alternative solutions.

References

Breiman, Leo. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231, Aug 2001.

Caruana, Rich, Lou, Yin, Gehrke, Johannes, Koch, Paul, Sturm, Marc, and Elhadad, Noemie. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 1721–1730, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3664-2.

Friedman, Jerome H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5): 1189–1232, Oct 2001.

Krause, Josua, Perer, Adam, and Ng, Kenney. Interacting with predictions: Visual inspection of black-box machine learning models. *ACM CHI 2016*, 2016.

Kuhn, M. and Johnson, K. *Applied Predictive Modeling*. Springer London, Limited, 2013.

Liu, Bing, Zhao, Kaidi, Benkler, Jeffrey, and Xiao, Weimin. Rule interestingness analysis using olap operations. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 297–306. ACM, 2006.

Munzner, Tamara. *Visualization Analysis and Design*. A K Peters/CRC Press, Natick, MA, USA, 2014. ISBN 9781466508910.

Treisman, Anne. Preattentive processing in vision. *Comput. Vision Graph. Image Process.*, 31(2):156–177, Aug 1985. ISSN 0734-189X. doi: 10.1016/S0734-189X(85)80004-9. URL http://dx.doi.org/10.1016/S0734-189X(85)80004-9.

van den Elzen, Stef and van Wijk, Jarke J. Baobabview: Interactive construction and analysis of decision trees. In *Visual Analytics Science and Technology (VAST), IEEE Conference on*, pp. 151–160, Oct 2011.

Ware, Colin. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004. ISBN 1558608192.

Xu, Kelvin, Ba, Jimmy, Kiros, Ryan, Cho, Kyunghyun, Courville, Aaron, Salakhudinov, Ruslan, Zemel, Rich, and Bengio, Yoshua. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 2048–2057, 2015.

Yang, Li. Pruning and visualizing generalized association rules in parallel coordinates. *Knowledge and Data Engineering, IEEE Transactions on*, 17(1):60–70, 2005.

Yosinski, Jason, Clune, Jeff, Nguyen, Anh, Fuchs, Thomas, and Lipson, Hod. Understanding neural networks through deep visualization. In *Deep Learning Workshop, International Conference on Machine Learning (ICML)*, 2015.

Zeiler, Matthew D. and Fergus, Rob. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. URL http://arxiv.org/abs/1311.2901.