DOI: 10.1111/cgf.14525
Eurographics Conference on Visualization (EuroVis) 2022
R. Borgo, G. E. Marai, and T. Schreck
(Guest Editors)

Volume 41 (2022), Number 3

ModelWise: Interactive Model Comparison for Model Diagnosis, Improvement and Selection

Linhao Meng[1], Stef van den Elzen[1] and Anna Vilanova[1]

[1]Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

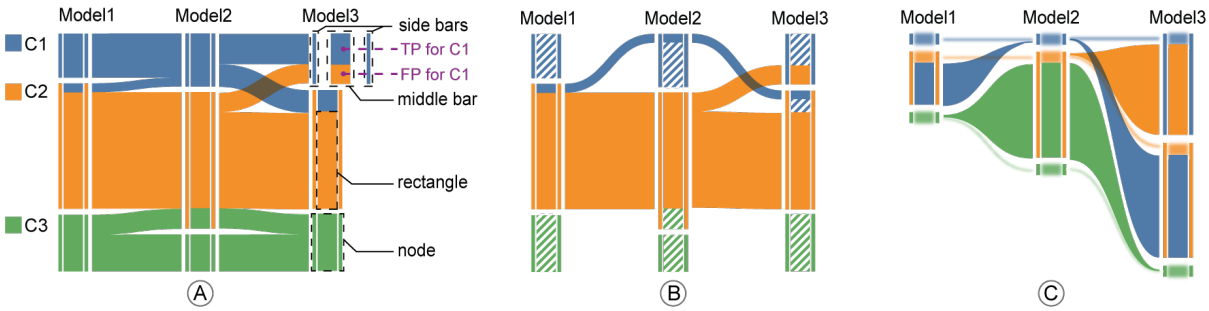

Figure 1: *Confusion Sankey design showing the same data as in Figure 4. (A) Confusion Sankey and its components; (B) Confusion Sankey with selected instances classified as C2 by Model1; (C) Confusion Sankey focusing on misclassification patterns.*

Abstract

Model comparison is an important process to facilitate model diagnosis, improvement, and selection when multiple models are developed for a classification task. It involves careful comparison concerning model performance and interpretation. Current visual analytics solutions often ignore the feature selection process. They either do not support detailed analysis of multiple multi-class classifiers or rely on feature analysis alone to interpret model results. Understanding how different models make classification decisions, especially classification disagreements of the same instances, requires a deeper model understanding. We present ModelWise, a visual analytics method to compare multiple multi-class classifiers in terms of model performance, feature space, and model explanation. ModelWise adapts visualizations with rich interactions to support multiple workflows to achieve model diagnosis, improvement, and selection. It considers feature subspaces generated for use in different models and improves model understanding by model explanation. We demonstrate the usability of ModelWise with two case studies, one with a small exemplar dataset and another developed with a machine learning expert with real-world perioperative data.

CCS Concepts

• ***Human-centered computing*** → ***Visualization; Visual analytics;*** • ***Computing methodologies*** → *Supervised learning by classification;*

1. Introduction

In the development of classification models, data scientists typically try various models. These models are built with different algorithms, hyperparameters, or selection of data features. Through model comparison, data scientists aspire to identify the best-fit model for the task at hand, which associates with the goal of **model selection**. In addition, they are interested in understanding how different models make classification decisions. In general, model understanding facilitates **model diagnosis** and yields insight for **model improvement**. Particularly for multiple models, data scientists can learn from well-performing models to improve a preferred target model (e.g., an intrinsic interpretable model is favored for clinical applications). For example, the insight from the well-performing models may be: *which data features contribute most to correct predictions in well-performing models but are not used in the target model?* To achieve such goals, it requires careful model comparison in terms of model performance and interpretation.

Currently, performance comparison of multiple multi-class classification models is conducted mainly according to summary statistics such as accuracy, precision, and recall. Although this overall

comparison provides clues for model selection, a detailed examination is required for gaining insight into model diagnosis and improvement. For example, even with a model having the highest overall accuracy, data scientists need to understand when the model fails. Detailed performance comparison at the class level helps to understand the strengths and weaknesses of different models. Furthermore, data scientists need to examine performance on different subsets of the data, from which they can derive where the model performs well or does not and where models agree or disagree on the classification results. Examination of the disagreement of classification results potentially helps correct model misclassifications by learning from the models with correct predictions.

As for interpretation, previous research relies on data and feature analysis to interpret and diagnose classification results [ZWM*19, PLHL19, JZHA22]. However, understanding how models make classification decisions, especially classification disagreements among different models, requires the interpretation of model behavior. Feature values reveal the feature discriminative power between classes, while it cannot discern if models capture the underlying information from the data features they use. Models can only be debugged and improved when the models themselves are interpreted [Mol19]. This is why XAI (eXplainable Artificial Intelligence) techniques are needed to disclose the model inner-workings by generating model explanations. But how to properly compare model explanations to boost insight in the context of model comparison is an open problem.

In this paper, we present **ModelWise** (Figure 3), a visual analytics method to assist data scientists in analyzing and comparing classification **model**s **wise**ly. It considers feature subspaces generated for use in different models and improves model understanding by model explanation. Our work is inspired by the real needs of data scientists who develop multiple classifiers with different characteristics (i.e., algorithms, features, and hyperparameters) and expect punctilious comparison to understand the difference of model results and how the models make classification decisions. Based on a review of related literature and discussions with two domain experts, we map the three domain goals to eight user tasks. All user tasks are related to exploring the three components, feature space, model performance, and model explanation. Our method integrates these three components and enables users to initiate exploration from any of them to achieve their goals, **model diagnosis**, **improvement**, and **selection**. In summary, the main contributions of this work are:

- The design and implementation of ModelWise, a visual analytics method for analyzing and comparing multiple multi-class classifiers. ModelWise integrates model performance, model explanation, and feature space for multi-perspective exploration while supporting multiple workflows to achieve domain goals of model diagnosis, improvement, and selection.
- Adapted visualization and interaction strategies to support user tasks. A Confusion Sankey is designed to achieve an effective class-based performance comparison of multiple multi-class classifiers. It enables users to trace instances across multiple models simultaneously. With interaction, we support subset selection from any perspective: model performance, model explanation, and feature space.

In the following section, we present related work on classifier performance comparison and model interpretation. Domain goals and tasks for model comparison are summarized in Section 3. We introduce the design in Section 4 and demonstrate its usability with two case studies in Section 5. Finally, we discuss our method in Section 6 and conclude our work in Section 7.

2. Related Work

Model performance and interpretation are two main perspectives of our model comparison work. In this section, we review relevant work on visual designs for classifier performance comparison. We also discuss various interpretation methods considering the eligibility of corresponding explanations for model comparison.

2.1. Classifier Performance Comparison

There exist a number of visual designs for performance comparison of two or multiple classifiers. Alsallakh et al. [AHH*14] design a confusion wheel that supports the performance comparison of two probabilistic classifiers. This confusion wheel is limited to pairwise comparison and not easily extendable to multiple models. Sugeerth et al. [MMD*19] build a hierarchical structure based on misclassification patterns and use treemaps to show the overall performance of two classifiers. This hierarchical structure is difficult to discern for multiple multi-class classifiers. Zhang et al. [ZWM*19] propose Manifold that utilizes a scatterplot-based visual technique to support pairwise model comparison based on model confidence. However, with this pairwise design, users tend to lose the performance relationship of multiple models when three or more models are compared simultaneously. Targeting multiple classifiers, Park et al. [PKL20, PLHL19] design a performance ranking visualization to evaluate models at the class level. The main goal of their system is to assist ML practitioners in selecting the appropriate classifier based on comparison from a high-level perspective, and thus performance examination below the class level such as customized subset or instance level is not well supported. In contrast, Boxer [GBYH20] is a performance comparison tool that provides a way to specify interesting subsets and assess performance. It uses an interactive Confusion Matrix Grid to show performance details, but the separated matrix design makes instance tracking and performance comparison of multiple classifiers difficult. Similar confusion matrix-like designs for performance comparison also appear in Confusionflow [HRS*20] and EnsembleMatrix [TLKT09].

Some VA tools support performance comparison but are not tailored towards classification models. In these tools, typical characteristics of classification models such as class confusions or classification patterns among models are hardly presented. For example, Jamonnak et al. [JZHA22] compare performance and predictions of autonomous driving models using aligned bar charts and a tabular list, while their design is unable and hard to be extended to unveil class confusions or complex classification patterns. DFSeer [SFC*20] as a model selection tool for demand forecasting models supports performance analysis, particularly on different products and time periods. Other designs that concern model performance comparison of multiple models mostly appear in visual analytics systems for model ensembling [SJS*21, CMKK21a, CMKK21b, XXM*19] and AutoML [WMJ*19, NZL*21].

High-level goals			Low-level user tasks
G1	G2	G3	
✓		✓	T1 [Performance] Compare performance metrics and classification results of different classifiers
✓			T2 [Explanation] Explore instances with similar model explanations
✓	✓	✓	T3 [Explanation] Compare feature importance-based explanations of different classifiers
✓			T4 [Feature] Explore instances with similar feature values
✓	✓	✓	T5 [Feature] Identify feature sets used by different classifiers
✓			T6 [Feature] Identify feature values of different classes for one feature
✓	✓	✓	T7 [Feature] Compare feature discriminative power between classes
✓	✓		T8 [Performance, Explanation, Feature] Select instance subset of interest

Table 1: *Three high-level goals (**G1**: Model Diagnosis, **G2**: Model Improvement, **G3**: Model Selection) and eight low-level user tasks (**T1-8**).*

Recent work employs a tabular layout to display the same label assignments in rows and show multi-label results of different algorithms or sources in columns [KAGB21]. This tabular design is adequate to inspect relationships of assigned labels. However, instance numbers for assigned labels are identified by text, making it difficult to compare the class-level performance of multiple models. Our visual design for classifier performance comparison builds upon the same idea of using a Sankey-based diagram similar to InstanceFlow [PHS20]. InstanceFlow uses the flow view mainly for temporal and individual performance analysis. In contrast, we employ it to compare performance and distinguish (dis)agreement patterns of multiple classifiers.

2.2. Model interpretation

Interpreting model results can be accomplished using two approaches: feature analysis and XAI methods. Most existing VA studies are dedicated to applying feature analysis to evaluate and diagnose ML models for tabular data [WXC*21, CEH*19, ZWM*19, AHH*14]. Generally, feature values are summarized and visualized according to customized subgroups, which yields insight into relationships between feature values and model results. We apply this similar idea in our work to visualize feature values based on user-defined instance subsets.

As the field of XAI evolves, various techniques have been proposed to explain the rationale behind model results. We summarize three requirements an XAI method should fulfill for our work: (1) model-agnostic and post hoc to enable comparing diverse models according to model results; (2) support model understanding at both global and local scope; (3) interpretation results of different models should be comparable. The above requirements inspire us to use local model-agnostic interpretation methods, with which model behavior interpretability of various scopes can be derived.

Different local model-agnostic interpretation methods produce different interpretation forms. Anchors [RSG18] generates a rule-based explanation given a prediction for one instance. Similar rule-based methods for global model behavior understanding can be found in RuleMatrix [MQB19] and GLocalX [SGM*21]. Although rules facilitate interpretation, rule-based explanations of multiple models are difficult to compare. Other interpretation forms of explanation results include contrastive explanations [DCL*18] and counterfactual explanations [WMR18, GHYB20, CMQ21]. The explanation results of these forms are hard to summarize for a subset or global scope and thus difficult to compare.

Compared to the interpretation forms mentioned above, feature importance-based explanation can be easily summarized for model understanding at various scopes and model behavior comparison. Local Interpretable Model-agnostic Explanations (LIME) [RSG16] is a widely-used feature importance-based interpretability technique. It explains individual predictions by training a local surrogate model around a given prediction. LIME is unified into the class of addictive feature attribution methods by Lundberg et al. [LL17]. According to cooperative game theory, Lundberg et al. show that there is a unique optimal explanation approach in the class, the Shapley values, that satisfies three essential properties: local accuracy, consistency, and missingness. They propose a unified measure of feature importance named SHapley Additive exPlanation (SHAP) values to approximate the Shapley values. They demonstrate the effectiveness of using SHAP values to promote model understanding with a clinical application [LNV*18] and propose a modified version of SHAP specifically for tree-based ML models [LEC*20]. In this paper, we use SHAP values. However, other methods that satisfy the requirements described in this section would also apply. A detailed summary of XAI methods can be found in Linardatos et al. [LPK21] and Vilone et al. [VL21].

3. Domain Goals and Tasks

Our work is inspired by the real needs of data scientists who develop multiple multi-class classifiers for tabular data and need to compare these models. Model comparison is not solely used to identify the best-fit model. The potential of learning from well-performing models to facilitate model diagnosis and improvement is also vital, however, difficult to explore. Furthermore, analysis results of model diagnosis and improvement directly guide model selection. Concisely, the domain goals for comparing multiple multi-class classifiers are to support model diagnosis, model improvement, and model selection.

Based on related literature [SSSEA20, CMJ*20, LFC*20] and several rounds of informal discussions with two domain experts, we map the three high-level domain goals (**G1-3**) to a set of concrete low-level user tasks (**T1-8**) summarized in Table 1. Each high-level goal is associated with several user tasks. Note that all low-level user tasks are related to exploration from three perspectives: *Performance*, *Feature*, and *Explanation*. These three perspectives correspond to three kinds of data generated during the machine learning modeling phase: feature space including all the feature attributes and feature values from the given set of data (*Feature*),

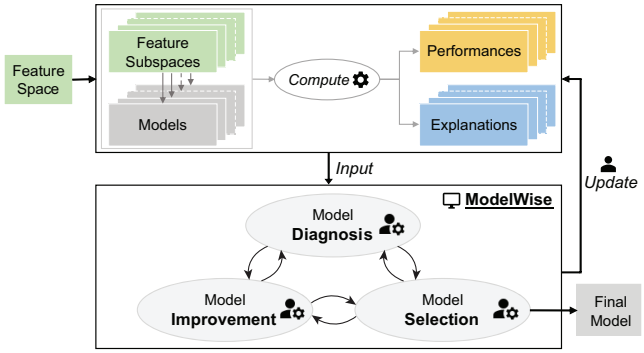

Figure 2: *Schematic representation of ModelWise components. The three kinds of data, feature space (Feature), model performance metrics and classification results (Performance), and model explanations (Explanation) are input to ModelWise. Through model analysis and comparison in ModelWise, users achieve the three high-level goals, model diagnosis, improvement and selection.*

performance metrics and classification results (*Performance*), and derived feature importance-based explanations (*Explanation*), see Figure 2. Data scientists can achieve the three domain goals through model comparison from these three perspectives.

During **model diagnosis**, data scientists aim to analyze model performance and understand how the models make decisions, especially where and why the errors are made [CBME16]. In the context of multiple models, their concerns include: *What are performance differences of different models? How do different models make classifications? Where and why do different models (dis)agree with the classification results of the same instances?*

To answer these questions, they need to inspect model performances, model explanations, and features from a global view. The summary performance metrics and class-based performance comparison provide a quick overview of model strengths and weaknesses (**T1**). Model explanations help understand model rationales to determine if one model works as expected (e.g., how does the model work for different instances? Does the model treat instances of different classes similarly?) (**T2**). Effective alignment of explanations of multiple models boosts model comparison from the view of model inner workings (**T3**). In addition, data scientists typically trace back to features to determine whether models capture the underlying structure of the data. They generally compare the selected features used by different models, look up feature values, or explore instances with similar feature values (**T4-6**). Analyzing feature values can disclose the discriminative power of features, which enables data scientists to conclude whether model errors can be corrected by using appropriate features (**T7**).

Data scientists also select instance subsets for detailed diagnosis [KDS*17, GBYH20] (**T8**). The examination of instance subsets narrows the analysis space and facilitates focused diagnosis. Data scientists consider instance subset selection across three perspectives. First, from the view of performance, they are interested in instances with different classification results for different models. Insights can be gained from models with correct classification results (e.g., to correct the errors made by lesser performing models). Second, from the perspective of model explanation, instances of different classes but with similar model explanations are worth further analysis. Last, in terms of feature, starting from instances with similar feature values is another angle to analyze and diagnose model results. After identifying the subset of interest, data scientists can examine the subset in detail from all three perspectives.

While model diagnosis emphasizes identifying and analyzing the errors in the model results, **model improvement** targets producing direct insight to enhance model performance. Generally, model diagnosis reveals the potential ways to improve the models. For instance, model errors caused by imbalanced data or unrepresentative instances can be remedied by improving data quality. Here, for model improvement we only consider direct insights about feature usage such as *Which features are helpful for the classification task? Are the features used appropriately by the models?*

We argue that these insights can be derived from two perspectives, explanation and feature. From the explanation perspective, data scientists identify important features appropriate to use in the final model by comparing feature importance-based explanations of different features in one model. By comparing feature importance-based explanations of the same feature in different models, data scientists detect if some models use this feature appropriately (**T3**). From the feature perspective, feature importance can be inferred by comparing feature usage and the corresponding model performance of different models (**T5**). Additionally, feature discriminative power between classes reveals the actual feature importance to the classification task (**T7**). Features with a high discrimination power tend to be helpful for the classification task. Combining these two perspectives with domain knowledge about models, data scientists can select the appropriate feature set, or adjust model parameters or architecture, to improve the models. The process of gathering insights can be done from the global or subset scope (**T8**).

Model selection is the process of selecting a final model from a collection of candidate models [HWN18]. The question it relates to is: *Which model is the best fit for the classification task?* However, this is not simply picking the model with the highest accuracy. It includes selecting the most predictive or less costly feature set (**T7**), choosing the appropriate algorithm and hyperparameters. By comparing model performances, explanations, and features usage, data scientists evaluate "best-fit" from different aspects (**T1, T3, T5**).

We aim for a visual design and interaction that supports users in achieving the three high-level goals, model diagnosis, model improvement, and model selection (see Figure 2).

4. ModelWise

Based on Figure 2, we design ModelWise, a visual analytics solution, to support model comparison of multiple multi-class classifiers. As shown in Figure 3, (A) the Overview provides a summary of the classification task and allows users to choose the models for detailed analysis. Then users explore the selected models in global and subset scopes through coordination of the four major views, (B) the Projection View, (C) the Classification View, (D) the Feature View, and (E) the Explanation View. Users can save the instance subsets of interest in (F) the Subset View for later analysis and comparison during the analysis process.

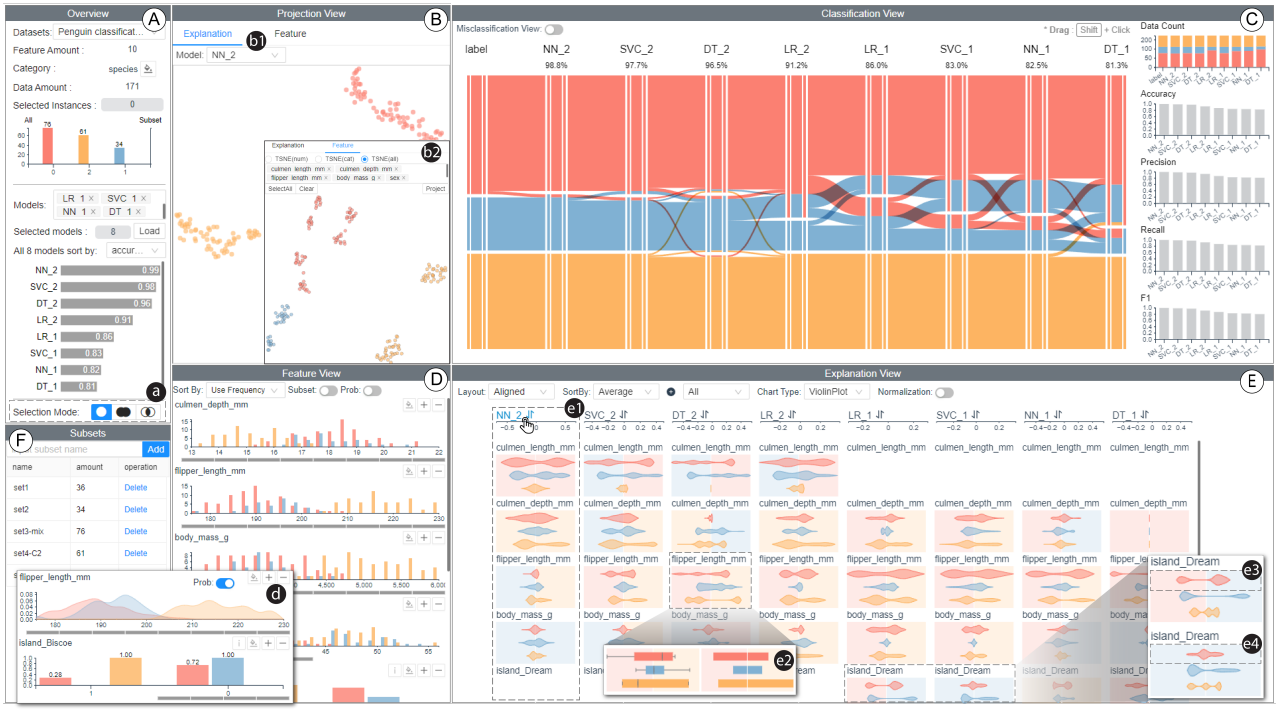

Figure 3: *The interface of ModelWise. (A) The Overview shows basic information about the classification task and enables users to select models for further analysis; (B-E) The four major views support the detailed comparison of multiple classifiers from different perspectives; (F) The subset view enables users to save the instance subset of interest for later analysis.*

As discussed in Section 3, all user tasks pertain to exploration from three perspectives: *Performance*, *Explanation*, and *Feature*. The corresponding three kinds of data, feature space (*Feature*), performance metrics and classification results (*Performance*), and model explanations (*Explanation*) are the pre-computed input to ModelWise. To generate feature importance-based explanations of all evaluation instances, we use a model-agnostic approximation of SHAP values called Kernel SHAP [LL17] (see Section 2.2). It assigns a contribution value for each model-feature-instance pair. Computing local explanations across all instances enables model interpretability from the global and subset scope.

Furthermore, these three perspectives directly guide the interface design of ModelWise. All four major views address one or two perspectives. More precisely, the Classification View visualizes performance information for *performance* comparison of multiple models. The Explanation View summarizes the derived local explanations for *explanation* comparison. The Feature View enables data *feature* exploration. In addition, the Projection View supports analysis of the high-dimensional *feature* and *explanation* space using projection techniques.Through model comparison from these three perspectives, data scientists derive the knowledge concerning the three high-level goals to update their models for the next development iteration or select the final model for deployment.

ModelWise enables multiple workflows for the exploration to achieve the three high-level goals. Users can start from any major view to compare models or select instance subsets for further analysis from specific perspectives. The subset selection from any of the three perspectives also reflects this flexibility.

4.1. Classification View

The Classification View (Figure 3C) visualizes performance metrics and classification results of multiple models to support model comparison and subset selection from the performance perspective (**T1, T8**). Summary metrics provide a quick summary of model performance, while closer examination of model classification patterns requires class-level or instance-level performance display based on the specific classification results. We design a Sankey-based diagram to align multiple model classification results and name it Confusion Sankey. A list of aligned bar charts is presented on the right side for quantitative performance comparison regarding classification results, accuracy, precision, recall, and F1 measure.

During earlier design iterations, we considered several design alternatives. As discussed in Section 2.1, separated confusion matrices are not effective for performance comparison and instance tracking, i.e., tracking classification results of instances in different models. To facilitate classification result comparison of multiple models, we combined classification results of multiple models into one confusion matrix with a bar chart in each cell (Figure 4A). Yet this design cannot show the relations of classification results through models. We also designed confusion matrices combined with contingency tables (Figure 4B). In this design, each confusion matrix on the diagonal axis displays the classification results of one model, and each contingency table shows the classification result relations of two models. Although pairwise classification relations are precise, identifying classification relations through multiple models by comparing separated matrices is challenging. We considered a tabular design alternative to summarize the classifi-

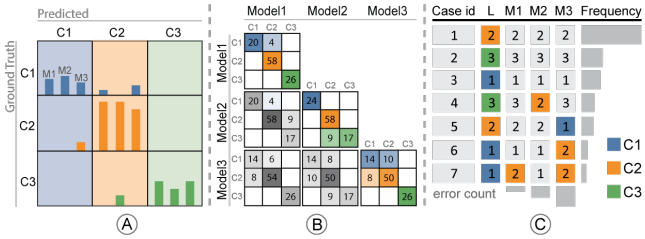

Figure 4: *Design alternatives for performance analysis of multiple classes and models, showing the same data as in Figure 1 : (A) A confusion matrix design; (B) Confusion matrices with contingency tables; (C) A tabular form design.*

cation results of multiple models (Figure 4C). This tabular design displays all the classification patterns in a list view, with each row representing classification results of the same instances in different models. However, it is hard to get a quick overview of the class-level performance of multiple models and requires more effort to search for particular classification patterns through models.

In Confusion Sankey (Figure 1), classification results of the models are summarized in terms of predicted class and laid out in parallel. This summarization and layout enable simultaneous class-level performance comparison of multiple models. We use the position-related channel to encode the model and class, which helps users compare class-level performance and locate models or classes of interest. As shown in Figure 1A, the y-axes that are sorted by model accuracy by default correspond to the classification results of the same instances of different models. Each y-axis is divided into several nodes according to the predicted classes in the Model axis. The height of a node is proportional to the number of instances corresponding to the node. Confusion Sankey enhances the traditional Sankey diagram by splitting nodes to represent true positives (TPs) and false positives (FPs) per class. Each node comprises a wide middle bar and narrow side bars on both sides. Each node represents an instance set with the same predicted class encoded in color hue on the side bars. The middle bar is divided into several rectangles colored by their corresponding actual classes. The rectangles with the same color as the side bars represent TPs, while others imply FPs. To help users identify classification patterns among the models, we use links colored by the actual class to connect the same instances between y-axes, which guides the tracking of classification results through multiple models. The width of links is proportional to the number of corresponding instances. In addition, we enable users to reorder the y-axes with drag and drop interaction. This flexible reordering facilitates performance comparison and instance tracing among models of interest.

To enable instance subset selection from the performance perspective, we extend Confusion Sankey for subset selection by clicking on the rectangles in the middle bars. Once one rectangle is clicked, the instance set represented by the corresponding rectangle is selected. Then the Classification View is updated to show only the classification results of the selected instances, as shown in Figure 1B where the instances classified as C2 in Model1 are selected by clicking on the corresponding rectangles in succession in the union selection mode. Specifically, the selected instances are highlighted while other parts are occluded with white-line textures in the middle bars. Links are updated only to show the connections of the selected instances among the model axes. The right-side bar charts display the performance metrics of the selected subset.

Confusion Sankey enables users to get an overview of the class-level performance between models. It also reveals model performance strengths and weaknesses for different classes and guides to discern (dis)agreement patterns among the model results. For example, in Figure 1A, Model2 performs well for C1 and C2, yet poorly for C3. On the contrary, Model1 and Model3 show their strength in the classification for C3 but confuse some instances of C1 or C2. With instances predicted as C2 by Model1 selected (Figure 1B), it can be seen that these misclassified instances are also misclassified by Model3. The above information cannot be easily obtained from any of the designs presented in Figure 4.

In general, TPs are less interesting compared to FPs. Misclassification patterns of the models are salient for deeper evaluation and diagnosis. However, in some cases, especially for unbalanced data, most space is taken by TPs, and rectangles in the middle bars representing FPs may be too short to discern. To enhance the visibility of FPs, we introduce a misclassification mode of Confusion Sankey (Figure 1C). All the rectangles representing TPs are collapsed into the same height in this mode. To avoid confusion and indicate that the height no longer encodes information, the rectangles are blurred [KMH01]. The associated links between TPs are also blurred to indicate that the width of the lines is not encoding any information. There are many approaches possible to convey uncertainty. Apart from applying blur, we considered others, e.g., texture or the use of color luminance. Blur is chosen since it guides user attention to the rectangles representing FPs through the semantic depth of field effect [KMH01]. This compression creates space to better show model FPs and facilitates selection.

4.2. Feature View

In the Feature View (Figure 3D), all features are organized in a vertical feature list to show feature distribution per class (**T6**). We use histograms to show the frequency for numerical features and apply bar charts for categorical features. To avoid the obscurity of some classes in the case of unbalanced data, we provide the probability distribution display option (Figure 3 d). It normalizes the frequency, thus enhancing the distribution comparison among different classes. When an instance subset is selected, the distribution chart of the subset will be overlaid on the original chart with darker colors (Figure 5D). Users can toggle the Subset button to show only the subset distribution chart to reduce visual clutter. To fulfill **T5**, we use equally divided bars below the feature distribution chart to indicate feature usage. Each bar represents one model, and the gray-filling bar represents that model uses this feature. The order of the bars is the same as the model order in Confusion Sankey.

To support **T7**, we define per feature discriminative power between classes to sort feature rows. To be specific, we use Bayes minimum error rate [DHS01] to measure the feature ability of class separation. Compared to some other distribution difference measures that only support the comparison of two distributions such as Kullback–Leibler divergence [KL51], Bayes minimum error rate, $P(\varepsilon)$, computes the probability of making an error, ε. It enables the consideration of the distributions of multiple classes simultaneously. We let w_i where $1 \leq i \leq c$, denote the finite set of c classes.

Given the class-conditional probability distribution $p(f|w)$ of a categorical feature f with n discrete values or a numerical feature f discretized into n equal-size bins, we compute Bayes minimum error rate to measure the discriminative power of feature f as

$$P(\varepsilon) = \sum_{j=1}^{n} P(\varepsilon|f_j) \cdot p(f_j) = 1 - \sum_{j=1}^{n} max(p(f_j|w_i) \cdot P(w_i) : i = 1, ..., c)$$

Since Bayes minimum error rate can be skewed for unbalanced classes, we put the prior probability $P(w_i)$ of each class as $1/c$ in actual implementation to magnify feature distribution differences between classes. Feature with a lower $P(\varepsilon)$ has more discriminative power. Sorting features based on their corresponding $P(\varepsilon)$ of the selected instances is provided. In addition, features can be sorted by the model use frequency to identify rarely used features.

4.3. Explanation View

The Explanation View displays feature importance-based explanations in a tabular manner to assist users in comparing explanations of different classifiers (**T3**). In the aligned layout, each row represents a feature, and each column represents a model consistent with the y-axis of the Confusion Sankey, as shown in Figure 3 e1. Each cell summarizes the feature importance-based explanations of selected instances for the corresponding feature and model. The corresponding feature cell is empty if the model does not use the feature. This layout facilitates the comparison of model usage.

In this work, we use SHAP values as feature importance-based explanations (see Section 2.2). SHAP values are computed per instance for each model and give an attribution value of each used feature per class. To extract global information, we use the distribution of SHAP values of each class for each feature among the evaluation instances. Thus each feature cell contains rows of distribution information of SHAP values of the corresponding feature for each class. To present distributions and facilitate comparison, we consider three design alternatives: box plots, violin plots, and bar charts (Figure 3 e2). Although the box plot is commonly used to show statistical summaries and is effective for comparing several groups of data sets, it can be misleading when, for example, the distribution is not unimodal. Actual explanation distributions are valuable because they reflect model behavior more precisely. For example, as shown in Figure 3 e3 and e4, the same feature behaves differently in the two models in terms of SHAP values of the same feature for the same class. In e3, SHAP values mainly reside on the edges of positive and negative effects, meaning that for most instances, the feature has a high effect, positive or negative, in the classification for this class. Yet in e4, the distribution of SHAP values is centered around 0, which means this feature has little attribution for most instances of this class. The bar chart is an extra aggregation showing one statistical value per class. It is useful to compare a single metric (e.g., average). It is also a good resource when only one instance is selected and there is no distribution information. Considering the respective advantages of the three kinds of plots, we enable users to switch between these plot types.

Apart from feature use differences among models, another concern is which features are more important per model and feature importance differences between models. To answer this question, it requires feature importance ranking per model. However, there is no unique way to decide this ranking. Therefore, we define several criteria to sort feature cells per model according to various summarized metrics of SHAP values of the corresponding model. These metrics are average, extreme value, median, and standard deviation, which reflect feature importance from different aspects. For the average and extreme value metrics, we compute the absolute values of average or extreme values for positive and negative values separately and use the bigger one of the two for sorting. In addition, data scientists are concerned with feature importance for a specific class. Suppose that one model misclassifies instances of C1 as C2, and the question would be which features are responsible for the classification result of C2 and which features are contributable to the actual class C1. Thus we include class selection in the sorting criteria so that users can decide to compare SHAP values of the selected class. By default, SHAP values of all classes are compared to achieve an overall comparison of feature importance. To facilitate the comparison of SHAP values for different classes for each feature cell, we use the background color of the positive and negative sides to encode for which classes the feature has the most effect on different sides. This encoding is only applied when the average or extreme value metric is selected (Figure 3E), otherwise, the background is grey as shown in Figure 6D.

In the aligned layout, one model column is selected for feature cell sorting. Feature cells in other columns are aligned with it. However, this layout cannot show the most contributing features for different models simultaneously. Thus we provide a compact layout in which feature cells are sorted per model column. To facilitate the comparison of the same feature in this layout, we highlight associated feature cells when hovering on a feature cell, as shown in Figure 5E where the feature cells of island_Dream are highlighted.

4.4. Projection View

A limitation of the Feature View (Section 4.2) and the Explanation View (Section 4.3) is that they show the variables per feature independently. Relations between multiple features and model explanations cannot be discovered easily, as well as the instance similarity in the high-dimensional feature space and model explanation space. The model explanation space is formed by considering the dimensions of SHAP values for all classes and features. The Projection View (Figure 3B) enables this multi-feature exploration. It allows users to visually inspect instance similarity based on feature values in the feature projection view b2 (**T4**) or feature importance-based explanations in the explanation projection view b1 (**T2**). It also serves as an entry point for users to select instances of interest based on feature and explanation similarity (**T8**).

We apply t-Distributed Stochastic Neighbor Embedding (t-SNE) [vdMH08] with Euclidean distance to reduce the dimensionality of the model explanation space, as well as the feature space, to two dimensions. We plot the instances as 2D points colored by their actual classes in the Projection View. We choose the t-SNE algorithm because it is adequate to preserve local information and identify patterns (e.g., clusters) within the data [LS19]. Specifically, we use a GPGPU implementation of the gradient descent linear tSNE [PTM*20] for fast real-time calculation. The fast computation enables users to interactively change the feature space used for the dimensionality reduction, for example, selecting the features

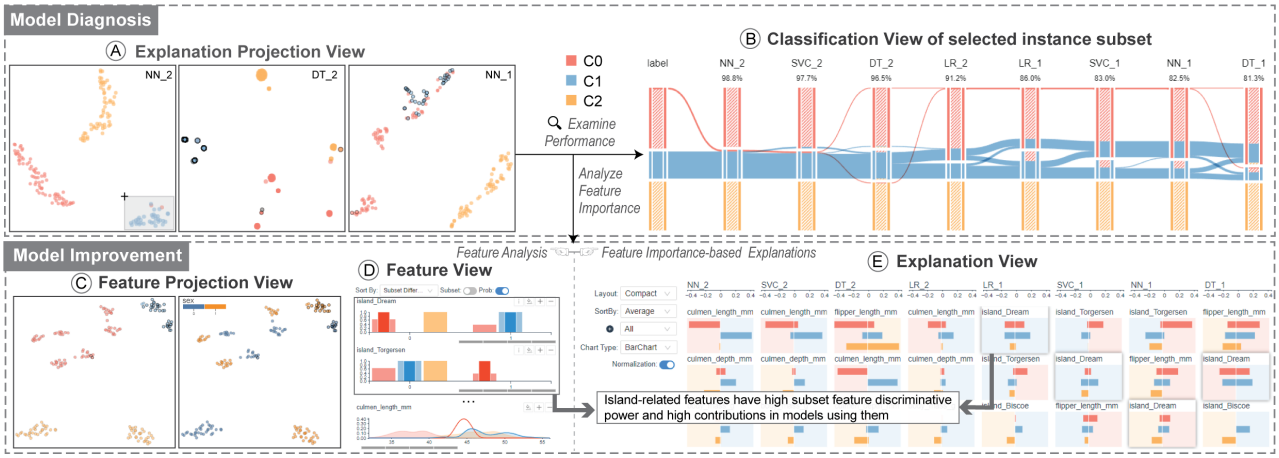

Figure 5: *Model comparison process with ModelWise to facilitate model diagnosis and improvement. (A) One instance cluster with mixed actual classes in the explanation projection view of NN_2 is selected. (B) Users evaluate performance differences in the Classification View. Users explore useful features to classify the instances of this subset from the perspectives of (C)(D) feature space and (E) model explanation.*

that one model uses. It also enables interactive exploration of different selected spaces. For model explanation projection, once one model is selected, SHAP values of all instances for all classes for this model are used to compute the projection result. Specifically, let v be a vector of SHAP values of one instance for the selected model, v can be represented as $v = (\phi_1, ..., \phi_j, ..., \phi_m)$ where ϕ_j is SHAP values of the jth feature and $\phi_j = (\phi_{j,1}, ..., \phi_{j,c})$ for a classification task of c classes. In the explanation projection view, instances with similar SHAP values are expected to be close to each other, implying the features are used similarly for these instances. Thus, the model more likely assigns the same classification results to them. To improve the interpretability of projection results, users can color instance points according to feature values of the features selected in the Feature View (Figure 5C).

4.5. Interaction

We design three main interaction strategies to support user tasks: selection & filtering, sorting, and coordinated views. First, selection & filtering can be conducted based on models, features, or instances. For example, users can select models to analyze in the Overview, and then information about the selected models is shown in the four major views. For subset selection, except for creating a new subset each time, we provide two other kinds of selection modes (a) corresponding to two set operations: union and intersection. These selection modes enable users to build more complex instance subset selections (e.g., FP instances in one model but TP in others). Next, sorting can be applied to models and features. Feature sorting is supported in both the Feature view and the Explanation View, as illustrated above. Last, coordinated linked views are implemented to support navigation and assist user exploration.

5. Evaluation

This section demonstrates the effectiveness of ModelWise with two case studies, one with a small exemplar dataset and another developed with a data scientist with real-world perioperative data.

5.1. Case Study - Penguin Species Classification

We use the penguin dataset [GWF14] to showcase the use of ModelWise and describe how different perspectives are incorporated in ModelWise to achieve the three domain goals. The dataset contains data for 342 penguins of 3 species (C0, C1, and C2). The task is to classify penguin species. We randomly partition the dataset into a training and test set (both 50%) and train 8 models with a combination of four algorithms, neural network (NN), support vector classifier (SVC), decision tree (DT), and logistic regression (LR), and two feature sets (s1 and s2). We evaluate these models on the test set and generate the data needed for our method (see Figure 3).

Model Diagnosis: *What are the performances of different models? How do models make different classification decisions?* As shown in Figure 3C, the models using feature set s2 (with name suffix 2) are in front and have fewer and shorter FP rectangles, which represents that their performances are better than the others. Performance differences indicate that some features in s2 are rather important for the classification, whereas s1 misses them. By comparing feature importance and usage in the Explanation View, we find that culmen_length_mm is an important feature in s2 but missed in s1 (Figure 3E). We further examine the models using the same feature set s2 but giving different classification results and want to understand why. First, we compare the explanation projection view of different models. As shown in Figure 5A, there are three obvious clusters in the explanation projection view of NN_2, each of which corresponds to one classification result of NN_2. The instances clustering together represent they have similar model explanations. However, why are some instances of different actual classes treated similarly by NN_2? We select the instances of one cluster representing the classification result of C1 but including instances with the actual class C0 by brushing on the projection view. The classification results of these instances are shown in Figure 5B. From the explanation projection view of DT_2, we see that the selected instances are scattered into various clusters. The difference in the explanation projections implies that different algorithms use different classification strategies, which explains the differences of

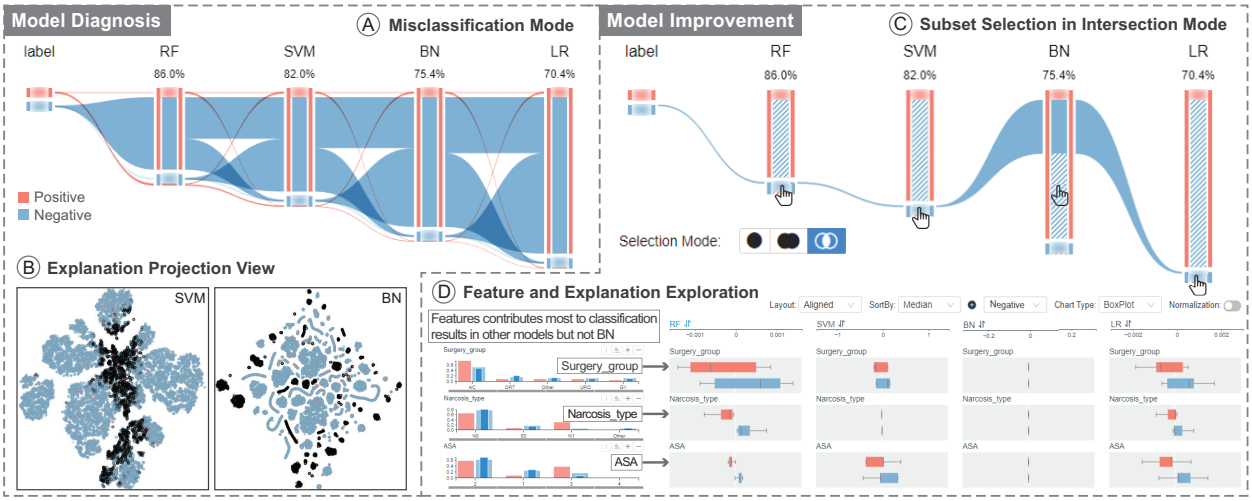

Figure 6: *User exploration in use case perioperative deterioration prediction. The user examined (A) model performance in misclassification mode and compared (B) model explanations in explanation projection view. To seek insight into model improvement, the user selected (C) an instance subset representing classification disagreement of models and then explored (D) feature space and model explanations.*

classification results between the models. Even for the same algorithm NN_1, the instances of different classes are in different clusters. This result implies that some features in NN_1 contribute to distinguishing the two classes for the selected instances, although there are different misclassifications in NN_1. Further examination on what classification strategies are for different clusters in terms of model explanations can be conducted in the Explanation View.

Model Improvement: *Which features are helpful to improve the final model?* We first project all the instances in the feature projection view with all the features selected (Figure 5C). Instances of different classes are scattered into different clusters, which indicates that existing features contain enough information to distinguish different classes. In addition, clusters of the same class are separated into two or more clusters. By coloring the instances using other features, we find that the separations of clusters of the same class are mainly due to the feature attribute sex. Thus it is probably a superfluous feature for this classification task. Since NN_2 is the model with the best performance, we try to identify which features are helpful to improve NN_2. With the above-mentioned instance subset selected, we sort the feature rows according to subset feature discriminative power (Figure 5D). The island features rank at the top, and the instances of different classes distribute into different islands. We also examine the corresponding feature importance-based explanations in the models that use island-related features (Figure 5E). These features have relatively high feature importance for these models. Further examination of the instances of different classes in that cluster proves that island-related features have opposite contributions to different classes. We retrain models with a new feature set that adds an island-related feature based on s2. Compared with the models trained on s2, the new models have an average accuracy increase of 2%, in which LR improves the most with 7%.

Model Selection: *Which model may be the best fit?* Based on the above analysis, we conclude the NN model with suitable features is a good fit. Despite being a black-box model, it shows the best performance and uncomplicated classification strategy regarding the explanation projection. SVC and LR are also good candidates.

5.2. Case Study - Perioperative Deterioration Prediction

This case study has been developed with a data scientist. His goal is to predict patients with a normal recovery (negative) versus patients with a potential unplanned ICU admission (positive) after being admitted to the ward. The dataset comprises 44 variables related to preoperative screening, surgery, or recovery room. The distribution of labels is very unbalanced, with 21257 negatives and 179 positives. He built four models with different algorithms, random forest (RF), support vector machine (SVM), Bayesian network (BN), and logistic regression(LR). Each of them uses different feature sets. BN is preferred since it permits incorporating domain knowledge and is more understandable for clinicians. However, the overall performance of BN is relatively low. Therefore, the data scientist wants to investigate these models to understand why BN is inferior to some models and derive knowledge by model comparison to improve it. We explained and gave a demo to the data scientist before he took control and managed ModelWise himself. We performed a screen recording and asked him to think aloud during the session.

Model Diagnosis: *How does BN perform compared to other models? How does BN make classifications?* The data scientist loaded all models and projected all instances in the explanation projection view based on SHAP values of BN. Since the dataset is very unbalanced, he switched the Confusion Sankey to misclassification mode to enlarge misclassification patterns (Figure 6A). At first glance, he found that BN has more FPs than SVM and RF. However, the TP rate of BN is the highest. The explanation projection shows many dispersed clusters representing different classification strategies of BN. In the Explanation View, he noticed that the most contributing features of BN are all recovery room related. He was curious about how BN makes classifications for different classes. He selected all the instances predicted as positives by BN by clicking on the corresponding rectangles of the Confusion Sankey in the union selection

mode. As shown in Figure 6B, these instances group into specific clusters in the explanation projection view, which indicates that BN generates clear separate classification strategies for two classes. In contrast, the explanation projection view of SVM does not show such separation between different classification results of SVM. He noticed that most clusters containing the FPs have mixed instances of different actual classes in the explanation projection view of BN. The possible correction method is to identify which features can separate these two classes of instances in these clusters.

Model Improvement: *What can be learned from the other models to improve BN?* The data scientist was interested in comparing BN with other models to get insights into improving BN. He focused on the instances of FPs (i.e., normals detected as ICU admissions) of BN that are TPs in other models. These instances show disagreement between BN and all other models and thus should be more likely to be corrected by incorporating the knowledge learned from other models. In the intersection mode, he selected the corresponding rectangles to make this subset selection (Figure 6C). He first noted that most of the contributing features are still recovery room related. By sorting the features cells according to the median of SHAP values of negative class in the Explanation View, he identified some features listed on the top such as surgery group, Narcosis_type, and ASA that contribute most to the correct classification in other models but have little attribution for BN, as shown in Figure 6D. He then checked the feature distribution of these features in the Feature View. These features show the relevance to the classes in terms of the actual feature distributions. They are also ranked high considering feature discriminative power between the selected negative subset and all positive instances. Both the explanation and feature views prove the importance of these features. He also mentioned that "*These features should have an influence on the classification according to clinical expert knowledge. For example, ASA (American Society of Anesthesiology) score is an important health metric.*". However, BN captures little information from them. The data scientist realized he needed to refine the BN structure by enhancing these important features. Through further exploration, he identified some features not used by BN but possibly important for the classification, which might be helpful for the construction of BN. Some other features used by BN but having little effect on the classification were also identified. These features could be removed to simplify the BN structure. After the exploration, the data scientist updated the BN model based on the above insights. First, the features identified as irrelevant were removed from the original BN model. After training, the model showed unchanged performance compared to the original one. This result implies that the BN model is simplified by removing non-contributing features but maintains the original performance. Secondly, the scientist changed the architecture to add new features, still removing the features that showed little effect. The new model showed 91% training accuracy, which is 16% higher than the original model. However, the danger of overfitting comes with this type of analysis and newly added nodes and edges in the new BN model. The current training and analysis with Modelwise were based on the whole dataset. The next step would be to test the model with an independent dataset.

The data scientist commented that *"I think it is a very nice way to explore what your model is doing. It gives you insight on model improvement, especially if you have different models to compare."*

6. Discussion

This section discusses scalability issues of ModelWise and potential extension of the explanation projection view.

Modelwise is able to deal with the scale of often occurring real-world model comparison analysis; however, **scalability** issues arise as the number of classes, models, features, and instances grows. There are mainly two aspects to consider: visual scalability and computational performance. ModelWise aims to support 2-10 class classification tasks with up to 10 models. Although ModelWise provides model filtering and a misclassification mode to improve the visual scalability of Confusion Sankey, the problem of a potentially large number of links and their crossings requires further study. Additionally, the aggregation of feature values and explanation results, and the tabular design are scalable to large data, while it may be overwhelming for users to conduct effective studies when too many features, classes, and models are considered. Computational performance issues arise with respect to real-time dimensionality reduction when the size of instances and features increases. Furthermore, currently, we precompute and store SHAP values for use in ModelWise considering the long computation time. We leave it as future work to reduce the computation time of model explanations so as to integrate ModelWise into a real-time ML pipeline.

The **explanation projection view** aims to show instance explanation similarity and reflect model inner-workings. Instances clustered together share similar feature importance-based explanations. Each cluster can be seen as a specific model decision strategy. The ideal result is that instances of different classes fall into different decision strategies. During experimentation, we try to project instances using the combination of SHAP values of multiple models. For the penguin classification case, an interesting finding is that even though projection results of each model cannot separate the classes well, that of combinations of some models show clear separations for different classes. This result may imply that these models are complementary and thus appropriate for model ensembling. It needs further analysis on how these findings could be exploited.

7. Conclusion

In this work, we abstract low-level users tasks for data scientists to achieve three high-level domain goals, model diagnosis, improvement, and selection, in the context of comparing multiple multi-class models. These user tasks are related to the exploration of feature space, model performance, and model explanation. We present ModelWise with adapted visual encodings to address these tasks. For example, Confusion Sankey allows the performance comparison of multiple multi-class models in ways not possible before. ModelWise supports rich interaction and multiple workflows to explore and compare feature space, model performance, and model explanation. Two case studies show that ModelWise yields insights into model diagnosis, improvement, and selection. We plan as future work to conduct a formal study about task abstraction for model comparison and solve the scalability issues of ModelWise.

Acknowledgments

We thank Simona Turco and Tom Bakkes (Dept. of Electrical Engineering, TU/e) for insightful discussions and help with case studies.

References

[AHH*14] ALSALLAKH B., HANBURY A., HAUSER H., MIKSCH S., RAUBER A.: Visual methods for analyzing probabilistic classification data. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1703–1712. doi:10.1109/TVCG.2014.2346660. 2, 3

[CBME16] CHEN D., BELLAMY R. K. E., MALKIN P. K., ERICKSON T.: Diagnostic visualization for non-expert machine learning practitioners: A design study. In *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (2016), pp. 87–95. doi:10.1109/VLHCC.2016.7739669. 4

[CEH*19] CABRERA A. A., EPPERSON W., HOHMAN F., KAHNG M., MORGENSTERN J., CHAU D. H.: Fairvis: Visual analytics for discovering intersectional bias in machine learning. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2019), pp. 46–56. doi:10.1109/VAST47406.2019.8986948. 3

[CMJ*20] CHATZIMPARMPAS A., MARTINS R. M., JUSUFI I., KUCHER K., ROSSI F., KERREN A.: The state of the art in enhancing trust in machine learning models with the use of visualizations. *Computer Graphics Forum 39*, 3 (2020), 713–756. doi:https://doi.org/10.1111/cgf.14034. 3

[CMKK21a] CHATZIMPARMPAS A., MARTINS R. M., KUCHER K., KERREN A.: Stackgenvis: Alignment of data, algorithms, and models for stacking ensemble learning using performance metrics. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2021), 1547–1557. doi:10.1109/TVCG.2020.3030352. 2

[CMKK21b] CHATZIMPARMPAS A., MARTINS R. M., KUCHER K., KERREN A.: Visevol: Visual analytics to support hyperparameter search through evolutionary optimization. *Computer Graphics Forum 40*, 3 (2021), 201–214. doi:https://doi.org/10.1111/cgf.14300. 2

[CMQ21] CHENG F., MING Y., QU H.: Dece: Decision explorer with counterfactual explanations for machine learning models. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2021), 1438–1447. doi:10.1109/TVCG.2020.3030342. 3

[DCL*18] DHURANDHAR A., CHEN P.-Y., LUSS R., TU C.-C., TING P., SHANMUGAM K., DAS P.: Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (2018), NIPS'18, p. 590–601. 3

[DHS01] DUDA R. O., HART P. E., STORK D. G.: *Pattern Classification*, 2 ed. Wiley, New York, 2001. 6

[GBYH20] GLEICHER M., BARVE A., YU X., HEIMERL F.: Boxer: Interactive Comparison of Classifier Results. *Computer Graphics Forum* (2020). doi:10.1111/cgf.13972. 2, 4

[GHYB20] GOMEZ O., HOLTER S., YUAN J., BERTINI E.: Vice: Visual counterfactual explanations for machine learning models. In *Proceedings of the 25th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2020), IUI '20, Association for Computing Machinery, p. 531–535. doi:10.1145/3377325.3377536. 3

[GWF14] GORMAN K. B., WILLIAMS T. D., FRASER W. R.: Ecological sexual dimorphism and environmental variability within a community of antarctic penguins (genus pygoscelis). *PLOS ONE 9*, 3 (03 2014), 1–14. doi:10.1371/journal.pone.0090081. 8

[HRS*20] HINTERREITER A., RUCH P., STITZ H., ENNEMOSER M., BERNARD J., STROBELT H., STREIT M.: Confusionflow: A model-agnostic visualization for temporal analysis of classifier confusion. *IEEE Transactions on Visualization and Computer Graphics* (2020), 1–1. doi:10.1109/TVCG.2020.3012063. 2

[HWN18] HÖGE M., WÖHLING T., NOWAK W.: A primer for model selection: The decisive role of model complexity. *Water Resources Research 54*, 3 (2018), 1688–1715. doi:https://doi.org/10.1002/2017WR021902. 4

[JZHA22] JAMONNAK S., ZHAO Y., HUANG X., AMIRUZZAMAN M.: Geo-context aware study of vision-based autonomous driving models and spatial video data. *IEEE Transactions on Visualization and Computer Graphics 28*, 1 (2022), 1019–1029. doi:10.1109/TVCG.2021.3114853. 2

[KAGB21] KRAUSE C., AGARWAL S., GHONIEM M., BECK F.: Visual Comparison of Multi-label Classification Results. In *Vision, Modeling, and Visualization* (2021), Andres B., Campen M., Sedlmair M., (Eds.), The Eurographics Association. doi:10.2312/vmv.20211367. 3

[KDS*17] KRAUSE J., DASGUPTA A., SWARTZ J., APHINYANAPHONGS Y., BERTINI E.: A workflow for visual diagnostics of binary classifiers using instance-level explanations. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2017), pp. 162–172. doi:10.1109/VAST.2017.8585720. 4

[KL51] KULLBACK S., LEIBLER R. A.: On Information and Sufficiency. *The Annals of Mathematical Statistics 22*, 1 (1951), 79 – 86. doi:10.1214/aoms/1177729694. 6

[KMH01] KOSARA R., MIKSCH S., HAUSER H.: Semantic depth of field. In *IEEE Symposium on Information Visualization, 2001. INFOVIS 2001.* (2001), pp. 97–104. doi:10.1109/INFVIS.2001.963286. 6

[LEC*20] LUNDBERG S. M., ERION G., CHEN H., DEGRAVE A., PRUTKIN J. M., NAIR B., KATZ R., HIMMELFARB J., BANSAL N., LEE S.-I.: From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence 2*, 1 (January 2020), 56—67. doi:10.1038/s42256-019-0138-9. 3

[LFC*20] LI Y., FUJIWARA T., CHOI Y. K., KIM K. K., MA K.-L.: A visual analytics system for multi-model comparison on clinical data predictions. *Visual Informatics 4*, 2 (2020), 122–131. PacificVis 2020 Workshop on Visualization Meets AI. doi:https://doi.org/10.1016/j.visinf.2020.04.005. 3

[LL17] LUNDBERG S. M., LEE S.-I.: A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (2017), NIPS'17, p. 4768–4777. 3, 5

[LNV*18] LUNDBERG S. M., NAIR B., VAVILALA M. S., HORIBE M., EISSES M. J., ADAMS T., LISTON D. E., LOW D. K.-W., NEWMAN S.-F., KIM J., ET AL.: Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature biomedical engineering 2*, 10 (2018), 749–760. doi:https://doi.org/10.1038/s41551-018-0304-0. 3

[LPK21] LINARDATOS P., PAPASTEFANOPOULOS V., KOTSIANTIS S.: Explainable ai: A review of machine learning interpretability methods. *Entropy 23*, 1 (2021). doi:10.3390/e23010018. 3

[LS19] LINDERMAN G. C., STEINERBERGER S.: Clustering with t-sne, provably. *SIAM Journal on Mathematics of Data Science 1*, 2 (2019), 313–332. doi:10.1137/18M1216134. 7

[MMD*19] MURUGESAN S., MALIK S., DU F., KOH E., LAI T. M.: Deepcompare: Visual and interactive comparison of deep learning model performance. *IEEE Computer Graphics and Applications 39*, 5 (2019), 47–59. doi:10.1109/MCG.2019.2919033. 2

[Mol19] MOLNAR C.: *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/. 2

[MQB19] MING Y., QU H., BERTINI E.: Rulematrix: Visualizing and understanding classifiers with rules. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 342–352. doi:10.1109/TVCG.2018.2864812. 3

[NZL*21] NARKAR S., ZHANG Y., LIAO Q. V., WANG D., WEISZ J. D.: Model lineupper: Supporting interactive model comparison at multiple levels for automl. In *26th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2021), IUI '21, Association for Computing Machinery, p. 170–174. doi:10.1145/3397481.3450658. 2

[PHS20] PÜHRINGER M., HINTERREITER A., STREIT M.: Instanceflow: Visualizing the evolution of classifier confusion at the instance level. In *2020 IEEE Visualization Conference (VIS)* (2020), pp. 291–295. doi:10.1109/VIS47514.2020.00065. 3

[PKL20] PARK C., KIM H., LEE K.: A visualization system for performance analysis of image classification models. *Electronic Imaging 2020*, 1 (2020), 375–1–375–9. doi:doi:10.2352/ISSN.2470-1173.2020.1.VDA-375. 2

[PLHL19] PARK C., LEE J., HAN H., LEE K.: Comdia+: An interactive visual analytics system for comparing, diagnosing, and improving multiclass classifiers. In *2019 IEEE Pacific Visualization Symposium (PacificVis)* (2019), pp. 313–317. doi:10.1109/PacificVis.2019.00044. 2

[PTM*20] PEZZOTTI N., THIJSSEN J., MORDVINTSEV A., HÖLLT T., VAN LEW B., LELIEVELDT B. P., EISEMANN E., VILANOVA A.: Gpgpu linear complexity t-sne optimization. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 1172–1181. doi:10.1109/TVCG.2019.2934307. 7

[RSG16] RIBEIRO M. T., SINGH S., GUESTRIN C.: "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2016), KDD '16, Association for Computing Machinery, p. 1135–1144. doi:10.1145/2939672.2939778. 3

[RSG18] RIBEIRO M. T., SINGH S., GUESTRIN C.: Anchors: High-precision model-agnostic explanations. *Proceedings of the AAAI Conference on Artificial Intelligence 32*, 1 (2018). URL: https://ojs.aaai.org/index.php/AAAI/article/view/11491. 3

[SFC*20] SUN D., FENG Z., CHEN Y., WANG Y., ZENG J., YUAN M., PONG T.-C., QU H.: *DFSeer: A Visual Analytics Approach to Facilitate Model Selection for Demand Forecasting*. Association for Computing Machinery, New York, NY, USA, 2020, p. 1–13. URL: https://doi.org/10.1145/3313831.3376866. 2

[SGM*21] SETZU M., GUIDOTTI R., MONREALE A., TURINI F., PEDRESCHI D., GIANNOTTI F.: Glocalx - from local to global explanations of black box ai models. *Artificial Intelligence 294* (2021), 103457. doi:https://doi.org/10.1016/j.artint.2021.103457. 3

[SJS*21] SCHNEIDER B., JÄCKLE D., STOFFEL F., DIEHL A., FUCHS J., KEIM D.: Integrating data and model space in ensemble learning by visual analytics. *IEEE Transactions on Big Data 7*, 3 (2021), 483–496. doi:10.1109/TBDATA.2018.2877350. 2

[SSSEA20] SPINNER T., SCHLEGEL U., SCHÄFER H., EL-ASSADY M.: explainer: A visual analytics framework for interactive and explainable machine learning. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 1064–1074. doi:10.1109/TVCG.2019.2934629. 3

[TLKT09] TALBOT J., LEE B., KAPOOR A., TAN D. S.: *EnsembleMatrix: Interactive Visualization to Support Machine Learning with Multiple Classifiers*. Association for Computing Machinery, New York, NY, USA, 2009, p. 1283–1292. URL: https://doi.org/10.1145/1518701.1518895. 2

[vdMH08] VAN DER MAATEN L., HINTON G.: Visualizing data using t-sne. *Journal of Machine Learning Research 9*, 86 (2008), 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html. 7

[VL21] VILONE G., LONGO L.: Classification of explainable artificial intelligence methods through their output formats. *Machine Learning and Knowledge Extraction 3*, 3 (2021), 615–661. doi:10.3390/make3030032. 3

[WMJ*19] WANG Q., MING Y., JIN Z., SHEN Q., LIU D., SMITH M. J., VEERAMACHANENI K., QU H.: *ATMSeer: Increasing Transparency and Controllability in Automated Machine Learning*. Association for Computing Machinery, New York, NY, USA, 2019, p. 1–12. URL: https://doi.org/10.1145/3290605.3300911. 2

[WMR18] WACHTER S., MITTELSTADT B., RUSSELL C.: Counterfactual explanations without opening the black box: automated decisions and the gdpr. *Harvard Journal of Law and Technology 31*, 2 (2018), 841–887. 3

[WXC*21] WANG Q., XU Z., CHEN Z., WANG Y., LIU S., QU H.: Visual analysis of discrimination in machine learning. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2021), 1470–1480. doi:10.1109/TVCG.2020.3030471. 3

[XXM*19] XU K., XIA M., MU X., WANG Y., CAO N.: Ensemblelens: Ensemble-based visual exploration of anomaly detection algorithms with multidimensional data. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 109–119. doi:10.1109/TVCG.2018.2864825. 2

[ZWM*19] ZHANG J., WANG Y., MOLINO P., LI L., EBERT D. S.: Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 364–373. doi:10.1109/TVCG.2018.2864499. 2, 3