Feature Selection on Deep Learning Models: An Interactive Visualization Approach

Zhaoyu Zhou
Fudan University
Shanghai, China
21210980099@m.fudan.edu.cn

Jason Kamkwai Wong
HKUST
Hong Kong, HKSAR, China
kkwongar@connect.ust.hk

Kerun Yu
Fudan University
Shanghai, China
19307110540@fudan.edu.cn

Guozheng Li
Peking University
Beijing, China
guozheng.li@bit.edu.cn

Siming Chen
Fudan University
Shanghai, China
simingchen@fudan.edu.cn

Abstract

In the era of data explosion, deep neural networks are widely used for prediction tasks in various scenarios. However, the complex feature spaces in high-dimensional data pose challenges to model training. While the common practice is to perform feature selection, existing approaches are generally designed for non-deep models. Additionally, deep models, considered as black-boxes, lack interpretability in utilizing features. This paper presents a visual analytics approach to facilitate the feature selection process of deep learning models by introducing human experience and decisions in integrated classical statistical methods. Our visual analytics system includes a Data Filter Component and an Interactive Verification Component. The former identifies and filters irrelevant and redundant features, while the latter supports the fine selection by understanding the features' contribution. Furthermore, iterative exploration is supported to gain a proper feature subspace. We use two case studies and an expert study to demonstrate the effectiveness of our approach.

Keywords: ***Visualization, Feature Selection, Deep Learning Model, Feature Interaction, SHAP.***

1. Introduction

Feature selection is an important technique in machine learning. The quality of data features determines the upper bound of machine learning models' accuracy. In real-world scenarios, the true distribution of data cannot be known, so we can only obtain its empirical distribution from massive data collected; However, big data is usually sparse. Thus, it is of great benefit to pick the feature set that represents the data appropriately. Specifically, removing unnecessary features helps (1) alleviate the curse of dimensionality and reduce the difficulty of the learning task; (2) avoid overfitting and enhance the generalization of the model; (3) and avoid the noise introduced by unnecessary features. In practice, massive data is generated and grows together with the feature space. Feature selection becomes an essential routine for many practitioners to balance the cost of feature production and computation. For example, aggregating correlated features can save considerable storage resources, relieve the computational pressure, and shrink the latency of operations.

However, most existing feature selection and evaluation methods are generally designed for non-deep learning models with low complexity. For instance, recursive feature elimination method [12] and forward selection algorithm [27] make use of the high interpretability of non-deep models to select a proper set of features. These methods can be applied to typical non-deep models such as linear regression [24] and SVM [6], which both have intuitive explainable features for the corresponding parameters. Likewise, the generation process of tree models [29] inherently incorporates feature-related information gain, which clearly shows the role played by features in the decision-making process. These are in stark contrast with deep learning models, which use multiple hidden layers to gradually extract higher-level feature interactions from the original input [7]. Due to their nonlinear and complex internal structure, conventional feature selection methods cannot be easier adapted and applied. Moreover, in terms of model complexity, the non-deep model consumes fewer training resources so that the feature contribution can be verified by efficiently testing against multiple hyperparameters. On the contrary, deep models require huge amount of parameters to capture high-level representations, limiting the feasibility of the retraining approach.

Some feature importance assessment methods are applicable to deep models, such as permutation importance [2] and SHAP [23]. However, they mainly focus on individual features but pay negligible emphasis on the interaction effects. The deep and nonlinear structure of deep models learns the hierarchies behind different levels of representations. This means some features are gradually combined to produce the final output. Therefore, mining significant feature interactions helps understand the models in balancing between the flexibility of model structures and the intuitiveness of feature interaction mining.

In this paper, starting from the characteristics and needs of supervised deep learning, we integrate a series of classical and effective feature evaluation and analysis methods with an interactive visualization approach, and keep human-in-the-loop in the feature selection process. We design a workflow and tool that helps users interactively explore, analyze and select features from multiple perspectives. Along with feature selection, it also provides interpretability for deep models from the feature perspective, and applicational insights. The main contributions of our paper can be summarized as follows:

- **Propose a multi-perspective and hierarchical feature selection, analysis approach and workflow.** From the model perspective, both model-independent and model-dependent feature statistics are evaluated; In terms of feature relationship, both the individual features and the interactions between feature pairs are considered; From the sample perspective, both holistic and local analysis are supported. Users can locate to a subset of data of interest.

- **Combine visualization and human-computer interaction to implement an efficient system.** Visualization can show dense information to effectively materialize complex, summarize high-dimensional data, and reduce the user's thinking load. With human-computer interactions, expert knowledge can be incorporated to facilitate better decision making.

- **Provide interpretability for deep models from the feature perspective.** The interactive system supports exploration on how the feature contributes to the model with global and local samples, which provides interpretability to the output of deep models from the feature perspective.

2. Related Work

The related work of our paper can be categorized into three parts: deep neural network visualization, numerical methods for feature selection and feature interaction.

2.1. Deep Neural Network Visualization

Deep learning has taken artificial intelligence to a new level, which in turn leads to their greater reliance for real business scenarios. However, due to the "black box" nature of neural networks, their interpretability has received extensive and close attention, especially in fields such as precision medicine, law enforcement, and financial investment, where important decisions are involved. Interactive visualization plays an important role in improving the interpretability of deep learning models, and interactive visualization of deep neural networks has become a hot research topic in recent years. Sacha et al. [28] provides an overview of the current state and supporting role of visual analytics in machine learning, and Choo et al. [4] provide an overview of the potential challenges and future research directions of visual analytics, information visualization, and machine learning interpretability.

Explainable deep learning contains three main research directions: model understanding, debugging, and refinement.

Model understanding aims to explain the principles behind predictions and the inner workings of deep learning models. For example, previous studies investigate visualizing the pixels that contribute most to the prediction results [40], explaining the predictions of convolutional neural networks in terms of agent decision trees [14], and understanding models by visualizing activation states [16]. CNNVis [22] analyzes convolutional neural networks applied to images by visualizing the values of the learned neurons and their interactions. LSTMVis [34] support retrieve similar sentences and paragraphs in the corpus by visualizing the hidden states of recurrent neural networks.

Model debugging aims to identify and resolve defects or problems within deep learning models. TensorBoard [38] is an open-sourced visualization toolkit for such a purpose. Gaining knowledge on the training data about their interrelationships enhance the efficiency of the debugging process. Some methods, such as CNNVis [22], also indirectly reveal the structure of the data. Although it aims to improve and debug models, the visualization results provide some ways of knowing the structure of the training dataset, such as the underlying characteristics of images and different categories (e.g., cats and dogs).

Model refinement refers to the process of improving and refining deep learning models by interacting the user with deep learning training process, so as to introduce expert knowledge interactively. ReVACNN [5] supports the user to dynamically remove neuron nodes and filter data. DeepEyes [26] helps users to remove low activation nodes by highlighting stable nodes. They both monitor and interactively guide the training of the deep model in real time to optimize the training process. In this paper, we focus on deep learning models to help understanding and refinement,

increasing its interpretability from the feature perspective.

2.2. Numerical Methods for Feature Selection

Chandrashekar et al. [3] have reviewed the numerical feature selection techniques classified them into three types: filtering method, wrapping method, and embedding method. In the **filtering** method, variance [30] is used to measure the amount of information contained in the features. Pearson correlation coefficient [1] is used to measure the linear correlation between variables. Mutual information [9] is used to measure the independence between variables and reflect the nonlinear correlation between variables. In the **wrapping** method, the forward selection algorithm (SFS) [27] adds the feature with the greatest gain each time and compares the model effect. The genetic algorithm (GA) [10] uses the idea of the evolutionary algorithm to find the optimal subset of features. The **embedding** method incorporates feature selection into model training, such as adding regular terms to the objective function to obtain a sparse solution [25].

Despite the popularity of these approaches, their application on deep models is largely limited. The filtering method lacks the evaluation of feature importance in a specific model. The wrapping method is more suitable for non-deep models with simple structures, as it is not very feasible to evaluate features by multiple training for complex deep learning models. The embedding method is implicit in the feature selection process and does not give an intuitive explanation. Additionally, there are some feature importance evaluation methods that are more applicable to deep learning models. A gradient-based feature importance ranking was formulated by Wojtas and Chen [37] to predict optimal subset features, using a stochastic local search on two jointly trained deep neural networks. The permutation importance method [2] estimates the importance of features by shuffling the data by features and measuring the impact it causes on the model. Lundberg and Lee [23] are inspired by the Shapley value of the cooperative game [31] and propose the concept of SHAP value, which computes the contribution of features to the prediction. Based on SHAP value, they further propose DeepLIFT [32] algorithm, which is more suitable for deep learning models. Our approach integrates effective mathematical methods such as stochastic gradient descent, mutual information, and SHAP, ensuring reliability.

2.3. Feature Interaction Mining

Machine learning models has the ability to extract the interactions from the original features without large human intervention. However, the generated feature combination is often incomprehensible. Researchers use some specially designed model structures to understand feature interaction and extract optimal feature subsets. For example, deepFM [11] is designed to learn sophisticated feature interaction for both low- and high-order features. To analyze specific feature combinations and their contribution, Deep&Cross [35] and xDeepFM [21] are proposed to investigate the features at the bit and vector level. These methods perform a relatively explicit feature interaction, but their interpretability is still under-explored.

AutoInt [33] introduces the attention mechanism into modeling the contribution of feature interactions. Wojtas and Chen [37] propose a dual network architecture to learn optimal feature subset during the training process. While these approaches focus on investigating feature interactions alone, Dinh and Ho [8] have laid out a theoretical foundation on the benefits brought by combining the investigation with correlation analysis. Knitte et al. [17] introduce regularization technique to amplify the differences between neurons and filter out important feature interactions. They also visualize the neurons to provide interpretation and discover the correlations between features. However, their goal is to analyze and explain the datasets. Our approach seeks to identify the optimal feature subset that can improve model's performance.

3. Background and Motivation

3.1. Feature Selection: Mathematical Definition

Traditional supervised machine learning deals with a collection of fixed-length feature vectors. A feature vector is a sample that consists of the value l of the label L and a value set $f = \{f_1, f_2, ..., f_n\}$ of corresponding feature set $\mathcal{F} = \{F_1, F_2, ..., F_n\}$. The objective is to obtain a model that accurately predicts the labels with the feature vectors. Thus, the model is determined by the features. Koller and Sahami [18] describe the goal of feature selection statistically as finding a minimum feature subspace $G \in \mathcal{F}$ such that $\mathbf{P}(L \mid G = f_G)$ is the same or very similar to $\mathbf{P}(L \mid F = f)$, where f_G is the value set of G, $\mathbf{P}(L \mid G = f_G)$ is the probability distribution of the label L given f_G as the priori knowledge and $\mathbf{P}(L \mid F = f)$ is the true conditional distribution of the label L with respect to the full feature set F. Since using exhaustive methods will have to perform 2^n evaluations, we consider the relationship between features and labels instead to imrpove efficiency. Specifically, we break down feature selection into two subtasks: removing irrelevant features and removing redundant features.

Relevance. Let $\mathcal{F}$ denote the full feature set, F_i denote the feature i, and $S_i = \mathcal{F} - \{F_i\}$, John, Kohavi, and Pfleger [15] classify features into three disjoint categories, i.e., strongly relevant($\mathbf{P}(L \mid F_i, S_i) \neq \mathbf{P}(L \mid S_i)$), weakly relevant($\mathbf{P}(L \mid F_i, S_i) = \mathbf{P}(L \mid S_i)$, and $\exists S_i' \subset S_i$, such that $\mathbf{P}(L \mid F_i, S_i') \neq \mathbf{P}(L \mid S_i')$), and irrelevant features($\forall S_i' \subseteq S_i, \mathbf{P}(L \mid F_i, S_i') = \mathbf{P}(L \mid S_i')$).

The strong relevance of a feature indicates that the feature is always necessary for the optimal subset. The weak relevance indicates that the feature is only indispensable for the optimal subset under some conditions. The irrelevance indicates that the feature has no effect on the label distribution. An optimal subset should include all strongly relevant features as well as some weakly relevant features and not contain any irrelevant ones.

Redundancy. The analysis of feature relevance provides basic principles for feature selection, but we cannot derive which subset of weakly relevant features should be selected. Therefore, redundancy between relevant features needs to be defined and analyzed. Feature relevance describes the relationship between features and labels, while the concept of feature redundancy elaborates the relationship between features. Yu and Liu [39] formulate feature redundancy based on the Markov blanket of features [18]. Given a feature F_i, $M_i \subset \mathcal{F}(F_i \notin M_i)$, M_i is said to be a Markov blanket for F_i iff $\mathbf{P}\left(\mathcal{F} - M_i - \{\mathcal{F}_i\}, L \mid \mathcal{F}_i, M_i\right) = \mathbf{P}(\mathcal{F} - M_i - \{\mathcal{F}_i\}, L \mid M_i)$. The Markov blanket describes a feature subset that are adequate to infer the output variable. Combining the two concept, a feature is redundant and should be removed from a subset G iff it is weakly relevant and has a Markov blanket M_i in G.

Since the redundancy of the removed features to the current feature subset always exists, removing redundant features can be performed independently and sequentially for individual features.

Based on the feature relevance and redundancy, the full feature set can be conceptually divided into four basic disjoint parts: irrelevant features, redundant features, weakly relevant but non-redundant features, and strongly relevant features. The optimal feature subspace can be obtained when all irrelevant and redundant features are removed.

3.2. Design Requirements

Human-in-the-loop(HITL) helps to ensure that AI systems are transparent, accurate, and ethical, while also making the best use of human expertise. We introduce HITL via a visual analytic system where human input or oversight is required at certain points in a process. To design it, apart from surveying for the statistical criteria of feature selection, we also have interviewed domain experts for their need in applications, and summarized five design requirements.

R1 Detect irrelevant and redundant features. Irrelevant and Redundant features features are costly and have a negative impact towards model training, which should be mined out and removed.

R2 Evaluate features with the specific models. The diverse model structures focus differently on feature extraction. Thus, features may contribute differently in different models. Since there is an interaction between the features and models, it is necessary to combine them for analysis.

R3 Avoid unnecessary training. To support the huge size of features and control the model version iteration cost, it is infeasible to evaluate features by testing them one by one as is commonly done with non-deep models. The optimal solution should require as few model training as possible.

R4 Explicitly mine feature interactions. Mining strong feature combinations is useful, such as helping engineers find which features should be co-added when they are ported to other models to ensure a promising result.

R5 Locate sub-datasets of interest. Different tasks may focus on different parts of the dataset. For example, in the disease prediction task, the researcher will be more interested in how the features work in positive samples compared to negative samples, to summarize important feature combinations as self-test suggestions to patients.

4. Visualization Design

4.1. System Interface

The system contains two main components, namely, data filter component and interactive verification component. The data filter component displays information before introducing the model, performing fast feature filter in an efficient but brief manner by analyzing basic statistics. Also, redundant features are filtered here to exclude noise as the prerequisites for the feature evaluation followed. The interactive verification component mainly shows the features' contribution in the given model structure, evaluating and analyzing features in a more refined and sophisticated way.

4.1.1 Data Filter Component

This component (Fig.1-I) consists of three views: Meta View (Fig.1-A), Correlation View (Fig.1-B), and Slider Control (Fig.1-C).

Meta View (Fig.1-A) is a scrollable table with good scalability and allows users to view sequentially. It provides basic information of each feature for feature selection, such as name and category, and metrics from the perspective of feature-task relevance. Two main metrics are used to assess feature relevance: fill rate and variance. Fill rate is the non-null ratio of a feature in all samples. Features with low fill rates generally have already deviated severely from the true distribution and cannot provide sufficient information. Similarly, for variance, features with variance close to zero have almost no discriminative power. These features satisfy

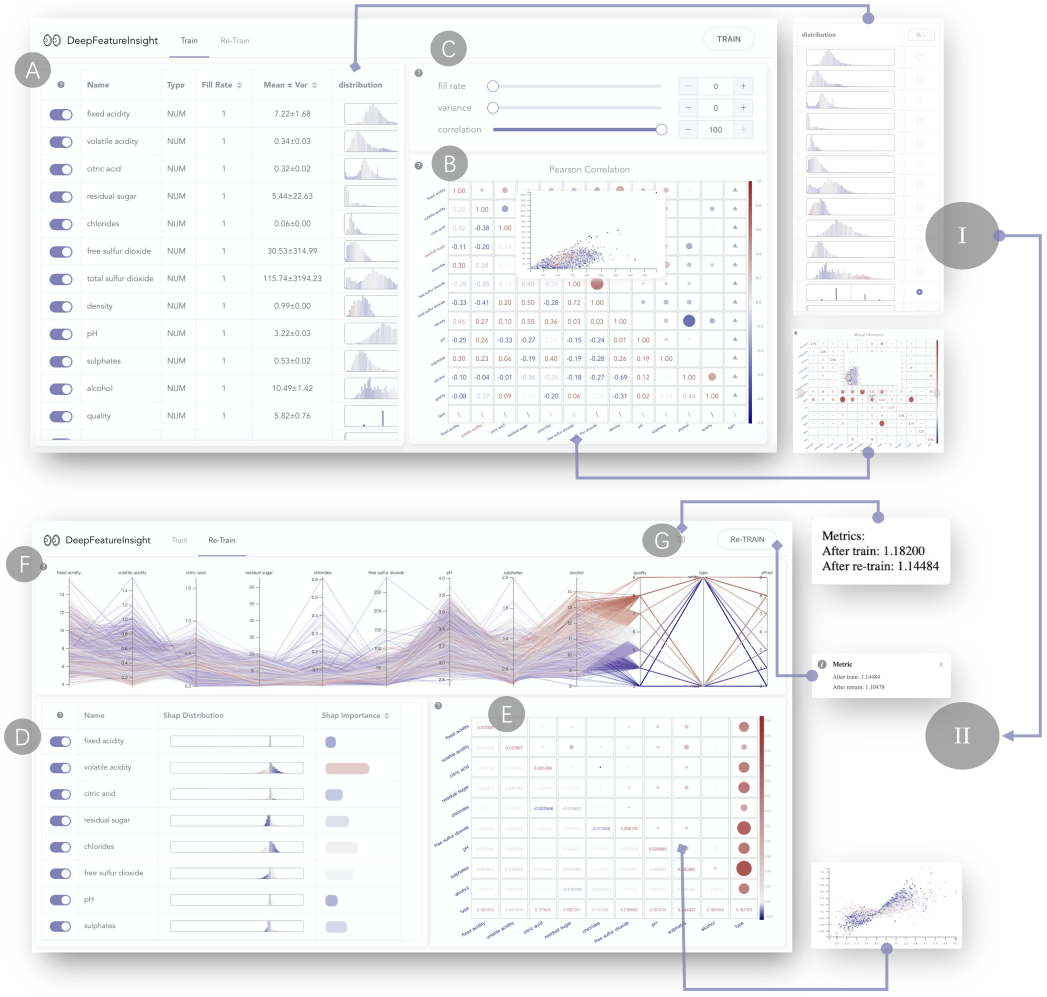

Figure 1. The user interface: (I) Data Filter Component for fast filter of features: (A) Meta view for basic information display; (B) Correlation View for features' correlation display; (C) Slider Control for thresholds setting. (II) Interactive Verification Component for fine selection: (D) SHAP Information View for SHAP values display; (E) Interaction Information View for feature interaction mining; (F) Parallel Coordinate Control for sub-datasets analysis; (G) Metric Box for metrics changes record.

the definition of irrelevant features (**R1**). Therefore, they should be filtered based on empirical thresholds.

The feature's probability distribution are also displayed, in which the color of the bar encodes the value of the target variable selected by the user, ranging from red to blue corresponds to the value from high to low. The target variable related encoding helps the user intuitively determine whether the feature is discriminative. For example, a unified color implies not. Outliers are also highlighted to provide additional information about variance. Some important features may have a small variance because of a low overall value interval, and the distribution allows further verification of these extreme cases. Thus, the probability distribution with a target variable panel provides information about the feature relevance by showing its discriminative power (**R1**).

The view supports manually switching the feature's status, sorting and arranging for fill rate and variance. The distribution supports hovering to display the specific values, and the search box in the last column supports searching features of interest among all for exploration and analysis.

Correlation View (Fig.1-B) shows a feature pair's correlation, helps filter features from the redundancy perspective. It also displays the relationship between features and the target variable, taking relevance into account (**R1**). A feature pair with correlation greater than a threshold are weakly relevant features, and one of them should be deleted as a redundant feature. Pearson correlation coefficient and mutual information are the two most commonly used methods for linear and nonlinear correlation detection, respectively. Mutual information can compensate the deficiency of Pearson coefficient in dealing with nominal features as it is suitable for discrete variable pairs. For continuous variables, discretization is needed for estimation. We adopt a method proposed by Kraskov [19] et al., which is an entropy estima-

tion based on k nearest neighbor distance as an alternative to the traditional splitting box.

This view combines the two linear and nonlinear methods above into one view with a carousel format, with good feature type compatibility. Correlation values are displayed as a matrix. The feature names are on the left and below. The upper triangular matrix contains elements of circles (continuous values) or triangles (discrete values) to allow the user to visually capture the correlation differences, and the symmetrical element in the lower matrix is the specific value, by which the circle radius, color, and text color are coded. For Pearson correlation coefficient, the color is coded from blue to red with the value $[-1, 1]$, while for mutual information, the value ranges in $[0, max\_value]$. Moreover, the view contains a scatter plot of each feature pair, with the horizontal and vertical axes indicating the values of the two variables, respectively, whose point is color-coded by the value of the target variable. A scatter plot containing a clear trend or pattern implies a strong correlation between features.

Hovering over a circle will highlight its corresponding numeric text, as well as the feature name, and hovering over the numeric text will do the same. Clicking on a circle/triangle or numeric text will draw a scatter plot of this pair of features in order to provide a quantitative understanding of the numeric values based on the correlation matrix as well as a visual understanding of the relationship distribution of the combination of features of interest.

Slider Control (Fig.1-C) supports setting filter thresholds for fill rate, variance, and correlation, according to which the on/off status will be automatically changed. As for the feature pair filtered out by correlation, the system will take both redundancy and relevance into consideration and prioritizes retaining the one with higher relevance with the target variable.

4.1.2 Interactive Verification Component

The Interactive Verification Component (Fig.1-II) contains three main views: SHAP Information View (Fig.1-D), Interaction Information View (Fig.1-E) and Parallel Coordinate Control (Fig.1-F), as well as a model metrics display box (Fig.1-G), which records the changes in the model evaluation metric after training and retraining, to help users compare the model performance.

SHAP Information View. (Fig.1-D) shows the probability distribution of SHAP values and SHAP-based importance in the specific model (**R2**) for each feature, with the intention of selecting the superior ones and discarding the inferior ones from the perspective of feature-task relevance (**R1**).

SHAP considers all features as "contributors". It quantifies the concrete contribution of a particular feature towards the model prediction. DeepSHAP is a SHAP value estimation method proposed by the authors for deep models, which completes the importance calculation in the model structure by backward propagation. Note that since SHAP values are additive, the SHAP value of the nominal feature can be obtained by summing the SHAP values obtained in each class after one-hot encoded. Moreover, since SHAP values are calculated specifically at the sample level, the global and local importance of a feature can both be calculated based on the SHAP values of according samples by $Importance_j = \sum_i |f(x_{ij})|$. In the presence of highly correlated features, the SHAP values give inaccurate results, but the redundant features are already roughly removed in the Data Filter Component, thus ensuring that the SHAP values are accurate and convincing.

SHAP score is a powerful metric to evaluate differences between features themselves of samples, stressing less significance to the ground truth in task-oriented situation. Therefore, FIR score obtained in preliminary dimension reduction is introduced to complement this characteristic of SHAP importance.

To be more specific, the jointly trained networks used for preliminary feature selection are closely combined via their objections as follows:

$$\mathcal{L}_O\left(\mathcal{D}, \mathcal{G}; \theta\right) = \frac{1}{|\mathcal{G}|\,|\mathcal{D}|} \sum_{g \in \mathcal{G}} \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}} l(\boldsymbol{x}_g, \boldsymbol{y}; \theta),$$

$$\mathcal{L}_S\left(\mathcal{G}; \varphi\right) = \frac{1}{2\,|\mathcal{G}|} \sum_{g \in \mathcal{G}} \left(f_S(\varphi; \boldsymbol{g}) - \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}} l(\boldsymbol{x}_g, \boldsymbol{y}; \theta) \right)^2.$$

Here, D is the dataset and G is the feature subset, with a fixed dimension of each element. $l(\boldsymbol{x}_g, \boldsymbol{y}; \theta)$ is a customized loss function on instance level, trained on the reduced feature subspace, and $\boldsymbol{x}_g$ denotes the dimensional-reduced data points with the specific features g . The first network (operator net) will learn to minimize its objection $\mathcal{L}_O$ via specific feature candidates provided by the second network (selector net), to optimize the parameter θ; while the selector network receives the performance feedback $l(\boldsymbol{x} \otimes \boldsymbol{m}, \boldsymbol{y}; \theta)$ from the operator, and optimize the parameter φ in its objection $\mathcal{L}_S$. After learning, a gradient-based score $\mathrm{Score}\,(g) = \frac{\partial f_S(\varphi; g)}{\partial \boldsymbol{g}}\Big|_{\boldsymbol{g}=g}$ gives us an insightful information of each feature's contribution to the task.

After calculating SHAP and FIR, SHAP Information View lists all features in a scrollable table and shows their SHAP distribution. The uniform coordinate scale makes their distributions comparable. Features with a more dispersed SHAP distribution generally have higher importance, and vice versa. The color of the histogram encodes the value of the feature to help users observe their relationship, ranging from blue to red encoding the SHAP value

from small to large. The left-hand side of the distribution displays the exact FIR score from the dual networks, while the right-hand side of it shows the feature importance based on SHAP values, encoded by color as well as the length of the column, helping the user perceive the feature importance intuitively.

The SHAP distribution supports hovering to display the current feature value. The importance column supports hovering to display specific values, and also supports sorting to facilitate users to quickly select features based on SHAP importance.

Interaction Information View. (Fig.1-E) shows the interaction of feature pairs. Interaction with other features is also an important aspect of feature relevance (**R1**). Strong interactions reflect strong relevance. In addition, mining out feature interactions is an important requirement for enhancing machine learning interpretability in conjunction with real-world business. The quantitative metric of interaction is given by the permutation importance and the qualitative understanding is given by the SHAP-based scatter plot (**R4**).

Permutation importance considers any feature important if the error of the model prediction increases significantly after shuffling the feature values. Similar to the SHAP values, the permutation importance requires the independence of features and is susceptible to highly correlated features [13], which is ensured in Data Filter Component. This method is frequently used for importance assessment of individual features, and this paper draws on the idea to evaluate feature interactions. The algorithm flow is shown in 1.

Algorithm 1 Feature interaction estimation based on permutation importance

Input: Model $Model$; Dataset $X_{\mathcal{F}}$, $\mathcal{F} = \{f_1, f_2, ..., f_n\}$
Output: Feature pairs' interaction matrix $PI \in \mathbb{R}^{n \times n}$

$e_{\mathcal{F}} \leftarrow$ error($Model(X_{\mathcal{F}})$, y_{true})
$R \leftarrow$Number of iterations
$PI_r \leftarrow$ Matrix of $r-th$ calculation
while $r < R$ **do**
  for f_i in $\mathcal{F}$ **do**
    for f_j in $\mathcal{F}$ **do**
      $X_{copy} \leftarrow X_{\mathcal{F}}.copy()$
      $X_{copy}[f_i, f_j] \leftarrow$ shuffle$(X_{\mathcal{F}}[f_i, f_j])$
      $PI_r(f_i, f_j) \leftarrow$ error$(Model(X_{copy}), y_{true})$ - $e_{\mathcal{F}}$
$PI \leftarrow \frac{1}{R}\sum_r PI_r$
return PI;

To reduce the uncertainty, the algorithm takes the multi-calculation average. The diagonal is the permutation importance of each feature, and the (i, j) and (j, i) elements of the matrix are equal, both value for the F_i and F_j pair. The algorithm provides an estimation of the interaction importance of feature pairs, giving a preliminary judgment by comparing the matrix elements. With the quantitative reference, it can be further qualitatively verified whether the feature pair has a significant interaction by the SHAP-based scatter plot.

The main function of Interaction Information View (Fig.1-E) also shows the relationship of features, so it is in the matrix. The upper triangular element is a circle, which is graphically intuitive for users to capture importance differences, and the lower triangular element provides specific values. The circle size, color, and text color are coded by the estimation results of the permutation importance, and blue to gray to red corresponds to $[min\_value, 0, max\_value]$ to help the user visually distinguish between positive and negative effects.

Additionally, the view contains a scatter plot of feature interactions based on SHAP values, with the vertical and horizontal axes indicating the value and SHAP value of feature F_i, respectively, and the color is coded by the value of feature F_j. A scatter plot containing distinct patterns or clusters implies significant feature interactions, while uniformly distributed scatters' color indicates no distinct feature interactions.

Hovering over a circle highlights the corresponding text, the feature name, and the corresponding two features on the diagonal, and hovering over the text does the same, making it easy for the user to locate the feature pair with obvious interaction. Clicking on a circle or text displays the SHAP-based interaction scatter plot, further adding an intuitive perception of the interaction distribution.

Parallel Coordinate Control (Fig.1-F) is a parallel coordinate plot (PCP) that displays high-dimensional data, allowing users to intuitively find the relationships between multiple variables. This control support locating sub-datasets for analysis (**R5**). Each axis of the PCP represents a feature, which scales to a uniform distribution of minimal to maximal feature values. Consistent with the other views, the color of the fold line is coded by the value of the target variable, ranging from red to blue corresponds to the values from large to small. Encoding the target variable values into the PCP helps the user to visually explore whether the values of other features show a distinct pattern when fixing the range of the target variable.

One obvious disadvantage of PCP is that when there are too many axes, the relationships between axes that are far away are difficult to observe. To alleviate this problem, the view supports the rearrangement of axes by dragging. The view also supports brushing on a single axis, and also on multiple axes simultaneously for users to explore local datasets of interest.

As a control, the PCP is linked with other views. After a user brushes at the Parallel Coordinate Control (Fig.1-G),

the SHAP Information View (Fig.1-D) and the Interaction Information View (Fig.1-E) will be updated accordingly to explore how features act and which features and feature pairs play a significant role when locating at the datasets of interest.

4.2. Workflow

Following the design requirements derived in 3.2, we propose an interactive visualization workflow (Fig.2) including the front-end and back-end. The front-end is mainly for the user's interactive exploration, and the back-end is mainly for the machine's data processing. First, the raw data are carefully examined and pre-processed with a state-of-the-art feature selection algorithm at the very beginning [37]. The following feature selection process is divided into two stages: data filter and interactive verification, which are performed before and after the introduction of the models (**R3**), respectively.

We construct a custom dataset to illustrate the functionality and prove the effectiveness of the workflow.

The dataset contains nine features: four features $\{a, b, c, d\}$ relevant to the target variable y and five irrelevant features $\{f_1, f_2, f_3, f_4, f_5\}$, with a sample size of 10,000. The target variable y is randomly generated with equal probability by one of the three ways followed:

$$y = \begin{cases} a + b + ab & method1 \\ b + c + bc + 2 & method2 \\ d + 5 & method3. \end{cases}$$

where $a, b, c \sim uniform(0, 1)$, $d, f2 \sim norm(0, 1)$, $f_1 \sim norm(1, 10^{-4})$, $f_3 \sim power(1)$, $f_4 = f_2^2$, $f5 \sim norm(f_3, 10^{-3})$. We can know that $y \in [0, 3]$ with method 1, $y \in [2, 5]$ with method 2, and $y \sim norm(5, 1)$ with method 3. We design the following tasks based on customized dataset: First, to discover redundant features for linear correlation (f_3, f_5) and nonlinear correlation (f_2, f_4) (**R1**); Second, to mine the feature pairs (a, b), (b, c) with interaction (**R4**); Third, to discover the variation in the features' contribution in the model to different value intervals of y (**R2, R5**). In addition, the dataset can also be used to verify the interference of redundant features and noise features.

Irrelevant and redundant features. In the Data Filter Component, after selecting the target variable y, the feature f_1 is found to have zero variance as an irrelevant feature, which is further confirmed in the distribution (Fig.3-a1). Meanwhile, the Pearson correlation matrix (Fig.3-a2) detects strong linear correlations for f_3 and f_5, but fails at the nonlinear-correlated pair f_2 and f_4, for which is made up by the mutual information matrix (Fig.3-a3), and the parabolic pattern is found in the scatter plot. Fig.3-b1 is the scatter plot of feature a and b. Since a and b are generated independently, the points are uniformly distributed. The feature interaction of a and b need to be further explored in the subsequent stage. Fig.3-b2, 3-b3, 3-b4 are the scatter plots of features b, d and f_3 with the target variable y, respectively, and it can be seen that b, d are only partially correlated, and the noise variable f_3 shows no correlation with the target variable.

Mining Interaction. The feature interactions (a, b), (b, c) present are detected as two clear and symmetric circles in the permutation importance matrix (Fig.3-c1). Clicking on the circle with coordinates (c, b) to observe the SHAP-based scatter plot (Fig.3-c2) reveals a clear pattern. The horizontal and vertical coordinates indicate the values and SHAP values of the feature b, respectively, and the color indicates the values of feature c. The absolute value and the sign on the vertical indicate the contribution of the feature b, and whether it's positive or negative. It can be seen that the small value of c reduces the positive or negative pull of feature b on the predicted value, while the pulling effect of feature b is amplified when c takes a high value. Analyzing from how the data are constructed, as the contribution of b contains $b \cdot c$ interaction term, it will be scaled by the value of the feature c, which is consistent with the results observed. As a comparison, observing the SHAP scatter plot for features d, b without interaction (Fig.3-c3), no clear color pattern is found.

Locating sub-dataset. Fig.4-a shows the global SHAP importance of the features. We can find that the noise features f_4, f_5 with low importance, and features a, c with similar importance as expected. Features b, d show the highest importance, and since feature b playing a role in two generation methods and feature d determining the value of y alone in the third one, it is in line with expectations.

In addition to being a control, the PCP can also explore patterns. Fig.4-b shows the strong linear correlation with feature d when brushing high values of y, and that when brushing low value of y and high value of feature b, feature a is found to take significantly low values, while the other features not. This also taps into the strong correlation between the high values of b and y, as well as the existence of feature interactions of a, b.

Linking PCP and SHAP Information View, we can observe the contribution of features in the sub-datasets. Fig.4-c demonstrates that when y takes high values, the feature d shows significantly higher importance than a, b, and c. While in Fig.4-d, when y takes intermediate values, since most of the data are generated with method 1 and 2, the feature b exhibits significant importance, the features a, c are similar, and the contribution of the feature d has a significant decrease.

Linking PCP with Interaction Information View, we can observe feature interactions in the sub-datasets. Fig.4-e shows that when brushing low values of y, the SHAP-based scatter plot of features b, c shows significant color cluster-

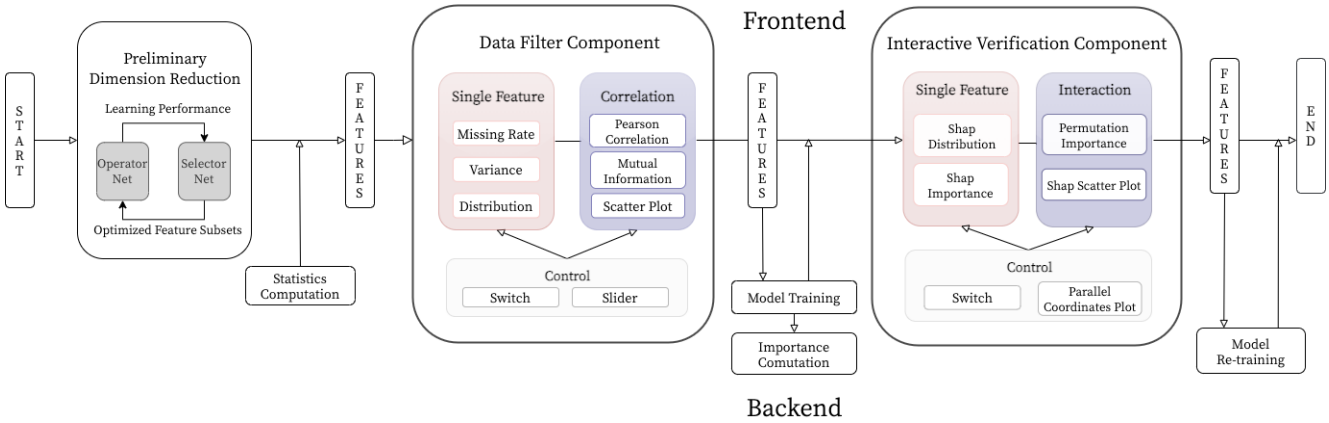

Figure 2. The interactive visualization workflow. The workflow contains front-end and back-end collaboration. A preliminary dimension reduction is performed first. The following feature selection process is divided into two main stages: Data Filter Component and Interactive Verification Component, supporting fast filter and a fine selection before and after model training respectively. Both components contain the analysis of single features and feature relationships.

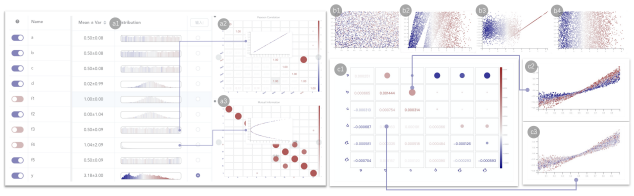

Figure 3. Detect irrelevant, redundant features and mine interaction. (a1) Feature f_1 has the variance close to 0; (a2) the linear correlation of features f_3 and f_5; (a3) the nonlinear correlation of features f_2 and f_4; (b1) the independence of features a and b; (b2), (b3), (b4) show the scatter plot of features b, d and f_3 with y, respectively; (c1) the obvious interactions of features a, b and b, c; (c2) the SHAP-based scatter plot for the feature pair b, c with obvious interactions; (c3) the scatter plot for the feature pair b, d without interaction.

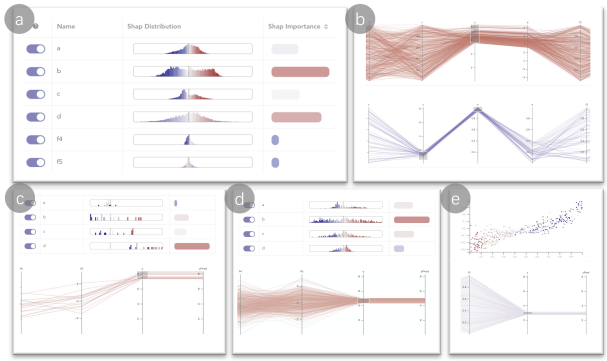

Figure 4. Sub-dataset analysis. (a) high importance of features b, d and low importance of noise features f_4 and f_5; (b) the correlation of d, y, and$a$, b, y; (c) feature d shows the highest importance when y takes high values; (d) feature b shows the highest importance when y decreases; (e) when y takes the values around 3, there is a clear pattern of features b, c.

ing, indicating a relationship between b, c that, one and only one can take the high value at the same time in most of the samples, and that their distributions are roughly negatively correlated, which is indeed the sufficient condition for $b + c + bc + 2$ to be valued near 3.

Figure 5. Redundant features' impact verification. (a) the presence of linearly correlated features f_3, f_5; (b) the presence of nonlinearly correlated features f_2, f_4.

Verifying impact of redundant features Fig.5-a, b shows the global SHAP importance and permutation importance when the highly linear-correlated feature pair (f_3, f_5) and nonlinear-correlated pair (f_2, f_4) exists, respectively. It can be seen that both the individual and interaction importance evaluations are significantly affected, showing that highly correlated feature pairs tend to have high importance and strong interactions. This finding validates the removal of redundant features as a necessary prerequisite.

The process supports the analysis of features with and without the model, the analysis of single features and feature relationships, and the analysis of global and local datasets of interest, providing a comprehensive explorations of features.

5. Evaluation

We use two case studies and an expert study to demonstrate and validate the system.

5.1. Case study

5.1.1 Wine quality dataset

We use a real-life scenario to illustrate the usability of the system. Let's assume that Tom is an algorithmist in the

data analysis department of a winery. He is given a wine dataset containing 6497 samples, consisting of 12 physical and chemical properties of wine and the wine quality rated by experts as the prediction target, and is required to construct a deep learning model. As the model will be used to replace experts in the tedious work of wine quality assessment, he needs to ensure the accuracy. Moreover, to communicate and give advice to the production department, he needs to evaluate existing features (**R3**), and interpretability (**R4, R5**) is also in need.

Tom input the data, defined a DNN model and the evaluation metric, and then entered the feature selection process.

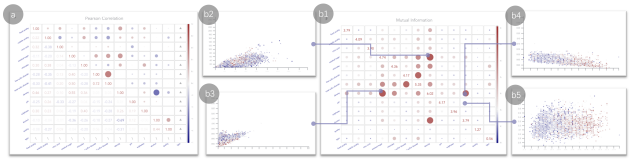

Figure 6. Correlation analysis. (a) linear correlation between (density, alcohol), (free sulfur dioxide, total sulfur dioxide); (b1) nonlinear correlation between (density, residual sugar); (b2), (b3), and (b4) show the clear trends of the three nonlinearly correlated features (density, residual sugar, alcohol); (b5) irregularity of the less correlated feature pair(alcohol, pH).

First, Tom explored at the Data Filter Component. In Meta View, Tom found no apparently uniformly distributed features. In Correlation View, he caught two clearly linearly correlated pairs of features (density, alcohol) and (free sulfur dioxide, total sulfur dioxide) in the Pearson correlation coefficient matrix (Fig.6-a2). From Tom's experience, he knew that chemically, the more the alcohol, the more the ethanol, leading to lower density. Also, he thought the correlation between free sulfur dioxide and total sulfur dioxide is also intuitive to understand with the concept of solubility. Sliding into the mutual information matrix (Fig.6-b1), he found a nonlinear correlated pair (density, residual sugar). After reviewing relevant information online, he learned that residual sugar fermentation produces ethanol, while high ethanol concentration inhibits yeast survival, thus affecting fermentation. The correlation between the two is stronger than linear. Since the initial sugar content can be affected by many factors such as grape variety and sugar content, and yeast survival rate is affected by various chemicals such as free sulfur dioxide, residual sugar shows a closer correlation with density, an intuitive physical indicator, than with alcohol. He further verified the findings from the scatter plots (Figures 6-b2, 6-b3, 6-b4), which show more obvious patterns compared to the no-correlation feature pairs(e.g., Fig.6-b5). Tom removed the two features, total sulfur dioxide and density, and then click the train button for training.

Next, the interface jumped to Interactive Verification Component. After clicking on the SHAP importance column in descending order, Tom found that alcohol matters most, volatile acid, and type followed (Fig.7-a1). He further

Figure 7. SHAP importance and interaction. (a1) features with high global importance; (a2) features with low global importance; (b1) a clear interaction between sulfate and type; (b2) the SHAP-based scatter plot shows a distinct pattern; (b3) the scatter plot of the feature pair with low interaction, showing a more uniform color distribution.

learned from the SHAP distribution that the more the alcohol, the greater the positive pull on wine quality. He found it consistent with the feedback from the market that wine with high alcohol shows more popularity. Although high-quality wine cannot be separated from some other indicators, high alcohol content often means riper grapes and fuller fermentation. Additionally, fixed acid, pH, sulfate, and citric acid showed low importance (Fig.7-a2), being considered as alternative features to be removed. He then performed the global interaction exploration. Tom noticed that the wine type showed clear interactions with other features (Fig.7-b1), and he thought that's because wine type essentially distinguishes its various physicochemical properties in relation to quality. He found the interaction between sulfate and type especially obvious. He clicked the circle and found a clear left-red-right-blue color pattern in the SHAP-based scatter plot (Fig.7-b2) compared to feature pair with weak interaction (e.g., Fig.7-b3), which reflected a positive effect of sulfate on quality in red wines but a negative effect in white wines, while the effect of alcohol on wine quality was more even in both wine types. This reflects the subtle preferences shown by the tasters in the two wines.

Figure 8. Brushing the red wines, the importance of the sulfate increases significantly.

To further verify the interaction, Tom brushed the red and white wine data separately in the Parallel Coordinate Control (Fig.8), and saw a significant increase in the importance of sulfate in the red wine data, and therefore, he decided to retain it.

To better suit the interests of company, Tom should attach importance to wine samples with extremely high and low quality. Compared to the global importance9-a, brushing the high-quality wines (Fig.9-b), an increase is found in the importance of the acidity (marked by the blue box). He learned that acidity makes the flavors of the wine more

Figure 9. Sub-datasets analysis. (a) the global SHAP importance; (b) a significant increase in the importance of pH when brushing high-quality wines; (c) an obvious increase in the importance of free sulfur dioxide and volatile acids when brushing low-quality wines; (d) fixed acidity is important for white wine.

clearly identifiable, making it pivotal in the high-quality bracket where the details make the difference. Brushing low-quality data (Fig.9-c), Tom found free sulfur dioxide and volatile acidity most important, whose high values showed a significant negative pull on quality. He was confused and searched for help online, and learned that high concentration of both free sulfur dioxide and volatile acid can cause olfactory discomfort to the taster when tasting, thus affecting the score. Moreover, both acidity and fixed acidity are found to increase obviously, especially the fixed acidity in the white wine (Fig.9-d). Therefore, he decided to retain the two. In contrast, the citric acid consistently shows low importance during the exploration process, so Tom made the final choice to remove it.

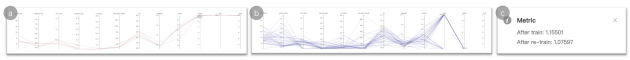

Figure 10. (a) and (b) are the feature patterns of high-quality and low-quality wines, respectively. (c) the model metric is improved after completing the feature selection process.

Tom wanted to explore whether there is a pattern of features for high-quality wine, he brushed wine of extremely high quality on the PCP, and was delighted to obtain a clear pattern (Fig.10-a): mostly white wine with about 7 fixed acidity, 0.25 volatile acidity, 0.4 citric acid, 3 residual sugars, 0.02 chloride, 25 free sulfur dioxide, 3.3 pH, 0.5 sulfate and 12.5 alcohol. He would give advice to the production department for production environment adjustment. He also brushed the extremely low part and found there is no clear pattern (Fig.10-b). He would also give the warning that each deviation in physicochemical properties may result in a low-quality wine.

After completing the feature selection and retraining, the results are returned and displayed in the pop-up box (Fig.10-c) that the error of the model is reduced. He is satisfied with the results, as well as inspired by the observations during the process.

5.1.2 GM12878 (200dp) dataset

As an ideal case, the wine quality dataset suits our system well. However, the data in the real world is more likely to be of high dimension and low quality. For example, the great information capability of DNA generates multiple combinations of genetic features, resulting in a troublesome problem for biologists to recognize different regions and functions of DNA sequences. Therefore, in this case, our system will display its capability in helping our scientist Alice distinguish meaningful DNA fragments and understand their characteristics at the same time.

Classifying enhancers and promoters in a given DNA sequence is crucial for disease identification and medical research. Motivated by GM12878 (200dp) [20], a real-world dataset sampled from annotated DNA regions of the GM12878 cell line, with 7 classes and 102 features and 3,000 instances each class, we followed the guidance of the provider of the dataset, reduce the classes of raw data to 3: Active-Enhancer regions, Active-Promoter regions, and background (a pool of Inactive-Enhancers, Inactive-Promoters, Active-Exons, and unknown regions); since they are what biologists care about most. Moreover, to balance the distribution of labels, we only sample 3000 instances from every class, which assists models in classification.

DNA is informative, but it is also known for copious non-functional and highly correlated regions, leading to the 102 features faced by the biologist. Fortunately, the aforementioned dual networks perform well in high dimensional data; the gradient-based feature selection algorithm works by stochastic gradient descent and local search, choosing features that contribute most to a higher score in the specific task, with their instance-wise feature ranking score (FIR score)saved and used later. Specifically, we applied the architecture suggested in [37], an operator net configured as a Multi-layer perceptron as $204 \rightarrow 300 \rightarrow 200 \rightarrow 50 \rightarrow 3$, also a selector net as $102 \rightarrow 500 \rightarrow 250 \rightarrow 100 \rightarrow 1$; the size of feature subsets is 20. Therefore, we are confident to guarantee that the data sent to Alice is lossless and tractable for her to understand.

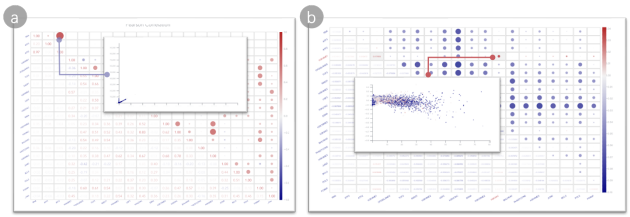

Figure 11. (a) Two highly correlated features(RNA, ATF2); (b) Two features with possible interactions(H3K4ME3, H3K9AC).

After preprocessing, the Pearson correlation coefficient matrix in the Data Filter Component will catch Alice's eyes,

owing to the significant correlation between RNA and ATF2 (Fig.11-a). It is amazing because both RNA and ATF2 have very high FIR scores. To figure out the reason behind it, Alice might move to the Interactive Verification Component to get more information.

Figure 12. (a) Remove ATF2, FOXM1 that contributing less; (b) the improvement of the classification accuracy.

Although having a high score given by the dual networks, the SHAP importance of ATF2 is pretty low, which means that ATF2 seldom distinguishes samples of DNA regions from each other. Alice realizes that this feature might be crucial to both enhancers and promoters, and even background, it seldom brings special information when we jump out of the task of classification.

Similarly, another feature with both low SHAP importance and FIR score is FOXM1, amounting to its low contribution to the classification task and its discrimination from other samples. Consequently, Alice tries to ignore ATF2 and FOXM1, and then re-train the model, finding that the accuracy has been improved from 92.40% to 92.46%(Fig.12).

Another striking symbol in the Interaction Information Matrix is the red cycle between H3K79ME2 and H3K4ME1. Those more common cycles in blue tell Alice that most DNA features can function well by themselves, whereas the read one is suggesting some latent effect, which can make sense only when H3K79ME2 and H3K4ME1 appear together. As a qualified biologist, Alice might be aware that H3K4ME1 is enriched in Active-Enhancers, and thus she might probe into the realm of H3K79ME2 and bring a new research topic.

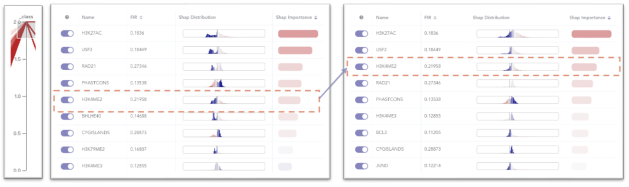

Figure 13. The importance of H3K4ME2 increases when choosing the Active-enhancer class.

As opposed to other explainable feature ranking systems, not only do we put forward multi-perspective analytic metrics, the SHAP importance, and FIR score, but also we allow the users to explore the performance of the same features in different classes. Specifically, Alice can click the label of Active-enhancer, where she will notice the feature H3K4ME2 contributing further than the other two categories(Fig.13). As a result, Alice turns to an academic survey and finds that H3K4ME2 defines one important binding region in common enhancers. [36]

5.2. Expert Study

We invited an expert in machine learning as a real user to use and evaluate our system. She is working on a Knowledge Graph(KG) system based on encyclopedia. As the update of entities is costly, experts need to construct a model to predict the entity's fresh degree. The probability of freshness predicted is used for ranking entities and then selecting the freshest ones. The expert wanted to filter out irrelevant and redundant features as crawling and storing them are time and resource consuming (**R1**). Besides, since it is not feasible to make predictions about the freshness of all entities with an enormous size, in the practical knowledge graph update system, rules are used first to recall entities into the alternative pool for model prediction, the expert wanted to evaluate the features' contribution (**R2**) in the model to find out the important ones. Developing rules based on such important features helps improve the efficiency of the KG update system. Also, finding important features is inspiring for the maintenance of other encyclopedia-based knowledge graphs. She was dealing with the data with 11 features and the 0-1 target variable "fresh or no" in the sample size of 86927.

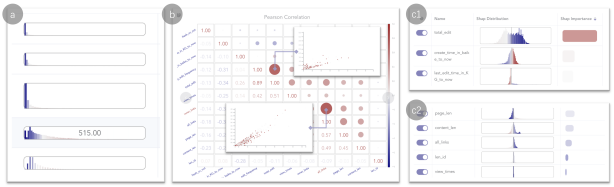

Figure 14. Expert Study. (a) right-skewed distributions found in Meta View; (b) (*history_edit_frequency*, *total_edit*) and (*inner_links*, *all_links*) are significantly linearly correlated. (c) the most importance features (c1) and less contributing ones (c2).

We used case two as a demo to illustrate our system to the expert, and then put the data into the system for her to explore freely. The expert got a general understanding of the features' distribution, finding that most of them are right-skewed, with a long tail (Fig.14-a). The expert immediately captured two highly correlated feature pairs (*history_edit_frequency*, *total_edit*) and (*inner_links*, *all_links*) in correlation view, clicked the scatter plots on and found an obvious linear pattern in both of them (Fig.14-b). No feature was found strongly relevant to the target variable, and their importance needed to be analyzed with the trained model. She dragged the slider for correlation threshold setting to 0.85, and the two features *history_edit_frequency* and *inner_links* are removed automatically. In Interactive Verification Component, she found the importance of feature *total_edit* significantly outweigh others (Fig.14-c1), while the contribution of *len_id* and *view_times* are significantly low

(Fig.14-c2), and the findings held when brushing samples of "fresh" and "not fresh".

After the feature selection and the feature exploration process, the expert said she got a deeper insight into the features. She may consider designing a rule about the feature *total_edit* to the update system for efficiency improvement. Moreover, she was thinking about further testing to see if she can stop crawling and storing features *history_edit_frequency*, *inner_links*, *len_id* and *view_times*.

For the visual analytics system, she thought that the interface was aesthetically pleasing and clearly laid out. She also found that the workflow was clear and the integrated methods and views supported a comprehensive analysis of features, which was effective. Also, there were some consistent findings with the previous sampling and analysis results, as well as some new inspirations. However, she also suggested us that the system support model boundary exploration, because she found important features and decided to add them to the recall stage. However, she got no idea how to set the thresholds to make reasonable rules.

6. Discussion & Conclusion

Concerning the characteristics of supervised deep learning and the practical requirements, we summarize and analyze the existing classical and effective mathematical methods for feature selection and analysis. We integrate and link them using visualization methods and techniques to introduce human decision-making in the feature selection process. We design and implement a visualization process and system for supervised deep learning models to select and analyze features. The system has a regular and harmonious layout and interface. It supports multi-level analysis from three perspectives: before and after model introduction, single and multiple features, global and local dataset. This allows users to interactively explore and interpret the deep models from the feature perspective.To handle real-world data with high dimensions, we adopt a dual network to implement preliminary feature reduction and provide a comprehensive view of feature importance, including SHAP importance and feature importance ranking score.

We use two case studies and an expert study to demonstrate the effectiveness of our approach. We first construct a customized dataset with a special structure to illustrate and verify the effectiveness of the system in identifying linear and nonlinear redundant features, mining feature interactions, and analyzing sub-dataset tasks. We then use two real datasets to explain and substantiate the rationality of the feature selection system in the context of real-life scenarios, and also demonstrate and illustrate the system's improvement of model interpretability from the feature perspective and the inspiration of exploring feature patterns for real-life business.

There might need future exploration in the following aspects. Firstly, it would be more automatic if the system could recommends reference values in the process. Secondly, the importance of each sample might be of great interest. Besides, the current design relies on tables with sorting function, matrix and PCP, which could be out-scaled by datasets with an extremely large size. The presentation and interaction can be further optimized by introducing zooming, supporting different levels of details, and supporting the rearrangement of matrix. Another direction concerns the applicability of the overall approach on other data types. Although our feature selection approach is applicable for general model structure in supervised deep learning, the system is more concerned with the tabular data input. For input containing special structures such as sequence information and hierarchical structure, it will be presented in a tiled way, which will lose structure information. The system can explore a richer and more flexible form in the presentation of feature information. In addition, extending the methodology to unsupervised tasks is interesting and should be carried out next.

Acknowledgement

The authors want to thank reviewers for their suggestions. This work is supported by National Natural Science Foundation of China (NSFC No.62202105), Shanghai Municipal Science and Technology Major Project (No. 2018SHZDZX01, 2021SHZDZX0103), General Program (No. 21ZR1403300), Sailing Program (No.21YF1402900) and ZJLab.

References

[1] J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, Berlin, German, 2009. 3

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 2, 3

[3] G. Chandrashekar and F. Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014. 3

[4] J. Choo and S. Liu. Visual analytics for explainable deep learning. *IEEE computer graphics and applications*, 38(4):84–92, 2018. 2

[5] S. Chung, S. Suh, C. Park, K. Kang, J. Choo, and B. C. Kwon. Revacnn: Real-Time visual analytics for convolutional neural network. 2016. 2

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 1

[7] L. Deng and D. Yu. Deep learning: methods and applications. *Foundations and trends in signal processing*, 7(3–4):199–200, 2014. 1

[8] V. Dinh and L. S. T. Ho. Consistent feature selection for analytic deep neural networks. *arXiv preprint arXiv:2010.08097*, 2020. 3

[9] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual information analysis. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 426–442, Berlin, German, 2008. Springer. 3

[10] D. E. Goldberg. Genetic algorithms in search. *Optimization, and MachineLearning*, 1989. 3

[11] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017. 3

[12] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002. 1

[13] G. Hooker and L. Mentch. Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:1905.03151*, 2019. 7

[14] S. Jia, P. Lin, Z. Li, J. Zhang, and S. Liu. Visualizing surrogate decision trees of convolutional neural networks. *Journal of Visualization*, 23(1):141–156, 2020. 2

[15] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Morgan Kaufmann, San Francisco, CA, 1994. 3

[16] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. Chau. A cti v is: Visual exploration of industry-scale deep neural network models. *IEEE transactions on visualization and computer graphics*, 24(1):88–97, 2017. 2

[17] J. Knittel, A. Lalama, S. Koch, and T. Ertl. Visual neural decomposition to explain multivariate data sets. *IEEE Transactions on Visualization and Computer Graphics*, 2020. 3

[18] D. Koller and M. Sahami. Toward optimal feature selection. Technical report, Stanford InfoLab, Stanford, CA, 1996. 3, 4

[19] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004. 5

[20] Y. Li, C.-Y. Chen, and W. W. Wasserman. Deep feature selection: Theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5):322–336, 2016. PMID: 26799292. 11

[21] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1754–1763, New York, NY, 2018. Association for Computing Machinery. 3

[22] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE transactions on visualization and computer graphics*, 23(1):91–100, 2016. 2

[23] S. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017. 2, 3

[24] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to linear regression analysis*. John Wiley & Sons, Hoboken, NJ, 2021. 1

[25] M. Y. Park and T. Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007. 3

[26] N. Pezzotti, T. Höllt, J. Van Gemert, B. P. Lelieveldt, E. Eisemann, and A. Vilanova. Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE transactions on visualization and computer graphics*, 24(1):98–108, 2017. 2

[27] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125, 1994. 1, 3

[28] D. Sacha, M. Kraus, D. A. Keim, and M. Chen. Vis4ml: An ontology for visual analytics assisted machine learning. *IEEE transactions on visualization and computer graphics*, 25(1):385–395, 2018. 2

[29] S. R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991. 1

[30] H. Scheffe. *The analysis of variance*, volume 72. John Wiley & Sons, Hoboken, NJ, 1999. 3

[31] L. S. Shapley. *17. A value for n-person games*. Princeton University Press, Princeton, NJ, 2016. 3

[32] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153, Cambridge MA, 2017. PMLR. 3

[33] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1161–1170, New York, NY, 2019. Association for Computing Machinery. 3

[34] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE transactions on visualization and computer graphics*, 24(1):667–676, 2017. 2

[35] R. Wang, B. Fu, G. Fu, and M. Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pages 1–7. Association for Computing Machinery, New York, NY, 2017. 3

[36] Y. Wang, X. Li, and H. Hu. H3k4me2 reliably defines transcription factor binding regions in different cells. *Genomics*, 103(2-3):222–228, 2014. 12

[37] M. Wojtas and K. Chen. Feature importance ranking for deep learning. *arXiv preprint arXiv:2010.08973*, 2020. 3, 8, 11

[38] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mane, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg. Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE transactions on visualization and computer graphics*, 24(1):1–12, 2017. 2

[39] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, 2004. 4

[40] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017. 2