GBDT4CTRVis: Visual Analytics of Gradient Boosting Decision Tree for Advertisement Click-Through Rate Prediction

Wenwen Gao, Shangsong Liu, Yi Zhou, Fengjie Wang, Feng Zhou, Min Zhu*

Fig. 1. The user interface of GBDT4CTRVis. GBDT4CTRVis assists advertising analysts in understanding the working mechanism of the GBDT-based CTR prediction model on multiple levels and facilitating the model tuning process. (1) Instance level: explore the prediction results of advertising data instances hierarchically through data overview (B), data statistics (C) and data details (D). (2) Feature level: analyze the importance of features (E) and the correlations (F). (3) Model level: display the most representative K decision trees (G), observe the evolution of the tree size (H), and present the evolution of the information gain during the model prediction (I). Analysts can also adjust hyperparameters and evaluate prediction performance through the Control Panel (A).

Abstract—Gradient Boosting Decision Tree (GBDT) is a mainstream model for advertisement click-through rate (CTR) prediction. Since the complex working mechanism of GBDT, advertising analysts often fail to analyze the decision-making and the iterative evolution process of a large number of decision trees, as well as to understand the impact of different features on the prediction results, which makes the model tuning quite challenging. To address these challenges, we propose a visual analytics system, GBDT4CTRVis, which helps advertising analysts understand the working mechanism of GBDT and facilitate model tuning through intuitive and interactive views. Specifically, we propose instance-level views to hierarchically explore the prediction results of advertising data, feature-level views to analyze the importance of features and their correlations from various perspectives, and model-level views to investigate the structure of representative decision trees and the temporal evolution of information gain during model prediction. We also provide multi-view interactions and panel control for flexible exploration. Finally, we evaluate GBDT4CTRVis through three case studies and expert evaluations. Feedback from experts indicated the usefulness and effectiveness of GBDT4CTRVis in helping to understand the model mechanism and tune the model.

Index Terms—Click-Through Rate Prediction, Gradient Boosting Decision Tree, Model Tuning, Visual Analytics

1 Introduction

Online advertising has been the major source of revenue for Internet businesses. In order to reduce advertisement (ad) costs and increase ad revenue, advertising analysts usually build click-through rate (CTR) prediction models to predict the probability of ads being clicked by users. The ads with higher probabilities are presented to users first, thus making the advertising more accurately exposed to target users.

Gradient Boosting Decision Tree (GBDT) [2, 14, 22, 31] is a widely used CTR prediction model. While tuning the GBDT-based CTR prediction model, advertising analysts should comprehensively analyze the model's prediction results, the features (i.e., the data attributes in the advertising CTR prediction dataset) involved in training, and the structural information of the model to understand the decision mechanism and further determine suitable hyperparameters. They often use several data analysis tools such as jupyter notebook [25, 30] to complete the above tasks through extensive coding and trial-and-error

• *Wenwen Gao, Shangsong Liu, Yi Zhou, Fengjie Wang, Feng Zhou and Min Zhu are with Sichuan University. E-mail: gaowenwen@stu.scu.edu.cn, piny@stu.scu.edu.cn, zhouyi2@stu.scu.edu.cn, wangfengjie@stu.scu.edu.cn, zhoufeng@stu.scu.edu.cn, zhumin@scu.edu.cn.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

experiments, which are time-consuming and cumbersome. In addition, due to the complex working mechanism of GBDT, advertising analysts always have difficulty analyzing the decision-making and the iterative evolution process of a large number of decision trees, and understanding the impact of different features on the model's prediction results.

Visual analytics have been demonstrated to be helpful in understanding the model mechanism and facilitating the process of model tuning [13, 34–36], which generally involves the analysis of input data features and model structure. In terms of features, the analysis is mainly about the importance. For example, INFUSE [9] designs a glyph to display the contribution of features to the prediction model. And RFSeer [40] helps users leverage features to understand the prediction results of random forest. However, these studies have not analyzed the correlations among features, and they fail to cover the need to distinguish the three main types of features in the field of online advertising: ad, medium, and user. Concerning the model structure, visualization studies of tree models related to our work, such as iForest [42] and BOOSTVis [16], can reveal the static structure of decision trees. But it is difficult for analysts to explore the dynamic evolution process of tree ensemble models, such as changes in information gain.

To address these issues, we propose GBDT4CTRVis, a visual analytics system that can help advertising analysts understand the working mechanism of GBDT and facilitate the process of model tuning. First, we construct a GBDT-based CTR prediction model and collect the log data generated during the training process. Second, we map the prediction results of data instances (i.e., a piece of advertising data) to a data overview view, and combine a data statistics view and a data list view to explore the advertising data instances hierarchically. Then, we use a list combined with dual-axis plot to display feature importance, feature distribution, and the impact of feature values on prediction results. We use a node-link chart to show the correlations among ad, medium, and user features. Next, we cluster decision trees and use icicle charts to display the structure of representative decision trees and use a streamgraph to express the temporal evolution of information gain during the model prediction process. We also provide multi-view interactions and panel control for flexible exploration. With this system, advertising analysts can quickly obtain insights into model tuning and perform validation. Finally, we verify the effectiveness and usefulness of GBDT4CTRVis through three case studies and expert evaluations.

In summary, our contributions include:

- We summarize the analytical requirements of advertising analysts in tuning the GBDT-based CTR prediction model, and propose a multi-level visual analytics pipeline to provide a visual analytics solution for GBDT model tuning.

- We propose a set of novel visualization methods that combine the three main participants (ad, medium and user) in online advertising campaigns to reveal the working mechanism of the GBDT-based CTR prediction model at three levels: instance, feature, and model.

- We implement GBDT4CTRVis, an interactive visualization system integrating the above methods, to help advertising analysts understand the working mechanism of the GBDT-based CTR prediction model and streamline the tuning process. We demonstrate the usefulness and effectiveness of GBDT4CTRVis through three case studies and expert evaluations.

2 RELATED WORK

In this section, we survey the most relevant papers to our work, including feature visualization and tree structure visualization.

2.1 Feature Visualization

Feature visualization aims to assist analysts in understanding the impact of different features. It can help analysts understand the importance of features and how models make decisions.

Regarding feature importance, INFUSE [9] performs feature selection by comparing different classification methods and designs a glyph to display the contribution of features to the model. Krause et al. [10] and Hohman et al. [3] investigate the effect of each feature on prediction results by changing the value of the feature. Additionally, there are methods such as LIME [24] and SHAP [17] that calculate feature importance from a local perspective.

In terms of understanding the model, Zhang [40] helps users understand the prediction results of random forest from the perspective of features by analyzing feature importance, distribution of feature segmentation points, and feature statistics. Rauber et al. [23] use t-SNE dimensionality reduction to downscale the high-dimensional feature maps of different layers of the network to a two-dimensional space for observation, which is used to understand and interpret multilayer perceptron and convolutional neural networks (CNN). Liu et al. [15] introduce a visual analytics system, CNNVis, which enables analysts to observe the most salient features learned by each type of neuron from images for the purpose of understanding CNN. Similar researches include RNNVis [19], DeepNLPVis [13], etc.

From the above research, existing studies can analyze feature importance and help understand the working mechanism of the model through features. However, they have not explained the relationship between features and their impact on prediction results. The calculation of feature importance depends on the analyzed model and cannot be directly transferred to GBDT. Moreover, the online advertising domain has to distinguish between the features of three subjects: ad, medium and user during feature analysis, yet existing studies have difficulty distinguishing the role of different categories of features in the model.

2.2 Tree Structure Visualization

Tree-based machine learning models include decision tree models and tree ensemble models. Decision tree models use a single tree to make decisions, while tree ensemble models are composed of multiple decision trees. Visualizing the structure of decision trees and tree ensemble models can help understand how the model makes decisions based on features and identify potential problems in the model.

The most common method for visualizing a single decision tree is intuitive node-link charts [16,32,33,38]. In addition, PaintingClass [28] uses a variant of the icicle chart to reveal local tree structures centered on nodes of interest. BaobabView [29] uses a node-link tree with confusion matrices to support interactive construction and analysis of decision trees. Mühlbacher et al. [20] use flow charts to represent the flow of samples on decision trees and embed pixel-level design in tree charts to help users evaluate the complexity and accuracy of decision trees. Jia et al. [6] design a foldable tree visualization to display data flow through a proxy decision tree. Wang et al. [32] develop an interactive tool based on the sunburst and node-link charts to display large decision trees.

For visualizing tree ensemble models, EnsembleMatrix [27] provides interactive visualization of the confusion matrix that enabled users to compare different classifiers at a class level. Höferlin et al. [4] visualize the class distribution of data at each stage of the Adaboosting model using a cascading scatter plot. Lee et al. [11] propose an interactive visualization framework that visualizes the model structure and prediction statistics at each step of the gradient-boosting tree learning process for users. BOOSTVis [16] clusters multiple decision trees and supports the interactive exploration of representative decision trees with different structures. iForest [42] provides information on all decision trees and reveals multiple decision paths to understand the generation process of final prediction results in random forests. RFSeer [40] uses sunburst charts as the main view to provide a multidimensional interpretation of the model structure of random forests.

The existing methods for visualizing decision trees and tree ensemble models are able to display the static structure and information of decision trees. However, they are difficult to explore the training progress of tree ensemble models and the evolution of prediction results. Therefore, they cannot meet the needs of advertising analysts to analyze the iterative evolution process of GBDT.

3 SYSTEM DESIGN

In this section, we describe the background, design requirements, system overview and data of GBDT4CTRVis.

3.1 Background

The core idea of the GBDT is to combine multiple decision trees in a sequential manner, so that each new tree can correct the errors of the previous trees, thereby improving the overall predictive performance. The construction process of the GBDT is as follows:

S1: Initialize the model. Estimate a constant value that minimizes the loss function to initialize a tree with only the root node.

S2: Compute residuals. Calculate the residuals by measuring the difference between the current model's predictions and the true labels.

S3: Iteratively build decision trees. In each iteration, GBDT constructs a new decision tree to further improve the predictive performance of the model. The construction process of this decision tree includes the following steps: **(a) Feature selection.** At each node, the best split feature is selected by calculating the information gain of each feature. The information gain represents the amount of information increase that can be brought by the selected features and is used to measure the importance of the features. In advertising CTR prediction, certain features may be more relevant to user click behavior, and information gain can better reflect this correlation and become a common splitting metric in the industry. **(b) Splitting nodes.** Based on the selected best splitting features, the samples in the nodes are assigned to the left and right child nodes. **(c) Termination conditions.** According to the stopping conditions, such as the maximum depth of the decision tree, the maximum number of leaf nodes, etc., determine whether to continue splitting the child nodes, and if the stopping conditions are satisfied, the node becomes a leaf node and stops splitting.

S4: Update the model. The newly constructed decision tree is combined with the current model using weighted aggregation, resulting in a more powerful model. This process involves multiplying the residuals (errors calculated in previous steps) by a learning rate (a constant) to dampen the influence of the new model. The multiplied residuals are then added to the previous model. This allows the new model to better fit the residuals that were not captured by the previous model.

S5: Iterative training. Repeat Steps 2-4 iteratively, continuously constructing new decision trees and updating the model. Each iteration attempts to reduce the residuals that were not fitted by the model in previous iterations. Through this iterative training process, the prediction of the model is gradually improved.

S6: Obtain the final model. Repeat the iterations until reaching predetermined stopping conditions, such as reaching the maximum number of iterations or achieving convergence of residuals. At that point, the final model is obtained.

During the training process of GBDT, a large number of decision trees are generated, which contain a wealth of information about the decision-making process. For subsequent visual analysis, this paper utilizes built-in methods of the model to export the logged data of the trained model, including feature importance and decision tree data. The importance score of each feature is calculated as the number of times that feature is selected as the splitting feature in all decision trees. The more frequently a feature participates in the splitting of decision tree nodes, the higher its importance. The decision tree data includes the node connection structure of each decision tree, the predicted values of the leaf nodes, as well as the splitting feature, information gain, and sample count for each node.

3.2 Design Requirements

To summarize the key pain points of advertising analysts in understanding the mechanism of GBDT and model tuning, we have thorough discussions with advertising analysts (E1-E4) from a large internet company. E1-E3 are advertising analysts in the computational advertising industry, their main responsibility is to combine machine learning technology with the advertising business to continuously optimize the effectiveness of the models for more accurate and efficient ad delivery. E4 is an advertising product manager, he needs to understand the results of the model constructed by the advertising analyst, devise a deployment strategy, and ensure that the advertising campaign achieves the expected performance (i.e., business metrics) at a lower cost. They all hope to understand the working mechanism and performance of the GBDT through visual analysis, analyze data distribution, feature importance, and feature correlations to facilitate model tuning and understand the driving factors behind advertising CTR. This will help advertising companies optimize their advertising strategies and improve campaign performance. Here, we summarize the following analysis requirements to guide the design of our system.

R1: Explore the model's prediction results at the instance level. Advertising analysts need to evaluate the model's performance by combining the original data instances and the model's prediction results to discover whether there are biases or errors in the model's predictions. They need to observe the model's applicability range from the data instances. They also need to identify the differences and similarities between instances to provide a basis for further customization of advertising delivery. However, current analysis tools such as Matplotlib, can not provide user-friendly interactions to meet the above requirements.

R2: Analyze model decision-making basis at the feature level. Advertising analysts need to identify the importance of features and discover correlations among them, which can help avoid inputting redundant features into the model and optimize the feature engineering. They also need to distinguish the features of the three categories in advertising campaigns: ad, medium, and user. Understanding the role of different types of features in the model helps to comprehend the decision-making basis, and internal operational mechanism of the model and determine appropriate model hyperparameters. Advertising analysts indicate that current feature analysis tools, such as correlation heatmaps, are difficult to satisfy the above analysis requirements due to their static and lack of interactivity, existence of data redundancy and visual confusion.

R3: Understand the model's decision-making mechanism at the model level. GBDT is an ensemble learning model that iteratively trains multiple decision trees and combines them for prediction. In each iteration, GBDT trains a new decision tree based on the residuals of the previous round's model to approach the true label value continuously. Understanding the decision-making mechanism of GBDT in each iteration, such as how to select features for node splitting, the evolution of information gain, and so on, can optimize and improve the model in a targeted manner. However, GBDT has numerous decision trees, and advertising analysts point out that existing tree model visualization tools, such as decision tree visualization, cannot effectively and intuitively present the dynamic information in a convenient and efficient way.

3.3 System Overview

According to the design requirements in section 3.2, we develop an interactive visual analytics system called GBDT4CTRVis to assist advertising analysts in understanding how the GBDT-based CTR prediction model works and simplifying the process of model tuning. The system pipeline is shown in Fig. 2 and consists of four main modules: (1) Data preprocessing, (2) Model construction, (3) Analysis and computation, and (4) Visualization and interaction.

The data preprocessing module cleans and down samples the advertising CTR prediction dataset before passing it to the model construction module for training. The log data generated during training is used by the analysis and computation module for dimensionality reduction by UMAP, decision tree clustering, feature importance and correlation calculation. Finally, the results of the analysis and computation module, along with the corresponding raw data, are mapped to the visualization and interaction module for analysis by the analysts. In the visualization and interaction module, users can adjust the model's hyperparameters through a control panel to obtain the model's re-training results and update the data and views. They can explore the model's predictive performance from the perspectives of instance, feature, and model, obtain insights for improving the model, and validate it.

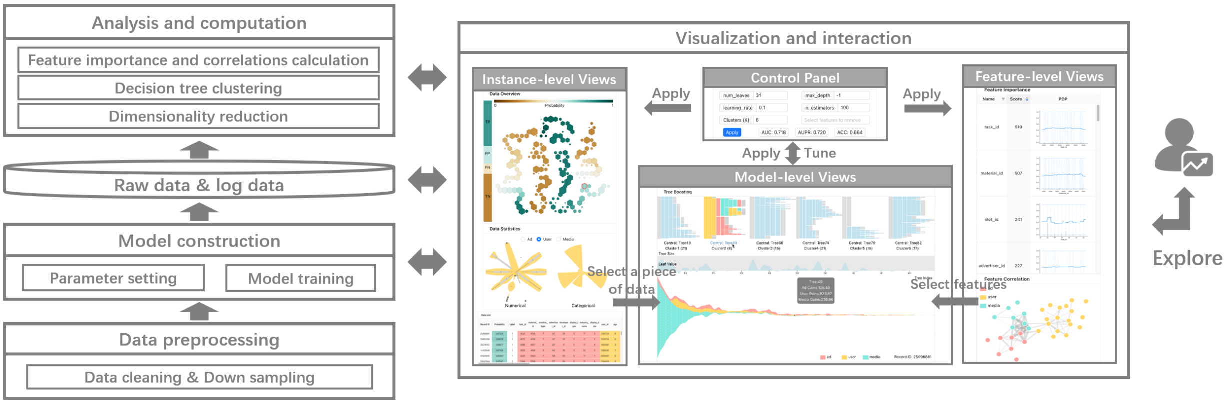


Fig. 2. Pipeline of GBDT4CTRVis. There are four modules included in the pipeline: Data preprocessing, Model construction, Analysis and computation, and Visualization and interaction. Our system supports the understanding of the working mechanism of the GBDT-based CTR prediction model from three levels: instance, feature, and model. Rich interactions are also provided to facilitate model tuning.

3.4 Data

3.4.1 Data Description

In this paper, we use the advertising CTR prediction dataset publicly available from the Huawei 2020 DIGIX algorithm competition [1], recommended by the domain experts we collaborate with. The dataset contains seven days of advertising delivery and clicks data, with approximately 41.91 million records, and each record has 36 fields.

Among the 36 fields, one is the label for advertising click behavior (with values of 1 or 0, representing whether the user clicked on the corresponding ad), and the remaining 35 fields can be divided into three categories of features: ad, medium, and user. These features include basic information about the displayed ad (such as the advertiser, industry, ad material, etc.), user information (such as demographic attributes, device name, etc.), and media information (such as application platform, advertising slot, etc.). In addition, the dataset has undergone discretization and desensitization processing through label encoding, where all feature values are numeric labels. To make better use of the original data in subsequent analysis, we divide the features into unordered discrete categorical data or ordered discrete numerical data based on their specific meanings.

3.4.2 Data Processing

GBDT4CTRVis processes the original dataset as follows: building a GBDT-based CTR prediction model and collecting the log data generated during the training process, using this log data to perform structural clustering on the multiple decision trees in the GBDT, and then combining the data instances and prediction results to calculate feature importance, correlation, and dimensionality reduction.

Data preprocessing and model construction. Before constructing the GBDT-based CTR prediction model, we first perform data cleaning and downsampling of the dataset [5], and then implement the GBDT model by LightGBM [8], which is an efficient implementation version of GBDT. Meanwhile, we enable users to adjust four hyperparameters of the model: the maximum number of leaf nodes of the decision tree (`num_leaves`), the maximum depth of the decision tree (`max_depth`), the number of decision tree (`n_estimators`), and the learning rate (`learning_rate`). For subsequent processing and analysis, we collect log data of the trained model, including decision tree data, feature importance information, and other details.

Decision tree clustering. Zhang-Shasha algorithm [39] is used in this paper. Based on the node connection structure of each decision tree collected in the previous section, the algorithm utilizes dynamic programming to calculate the tree edit distance (the minimum number of editing operations required to transform one tree into another) [1], thus quantifying the distance between decision trees and obtaining a tree distance matrix. The tree edit distance accurately measures the structural differences between decision trees, regardless of changes in node order, position, and splitting conditions. Compared to other methods that measure tree structural similarity, such as Tree Kernel, the tree edit distance is more suitable for understanding the decision mechanism of GBDT. Next, a more flexible and applicable K-Medoids algorithm [21] is applied to cluster the trees using this matrix. Users are allowed to customize the number of clustering centroids, K, based on their desired level of observation granularity. The clustering centroids represent representative decision trees. During the experimental process, we try different values of K and communicate with collaborating experts. Based on considerations of practical analysis requirements, cognitive cost, and computational efficiency, we define the default value of K as 6.

Feature importance and correlations calculation. The importance score of each feature can be obtained from the training log data of GBDT, which represents the number of times the feature is selected as the splitting feature in all decision trees. The more a feature participates in decision tree node splitting, the higher its importance score. In addition, we use Spearman's correlation coefficient [26] to measure the correlation between the features. Because Spearmen's correlation coefficient is non-parametric, it does not require the assumption that the data is normally distributed, does not consider specific numerical values, and is not sensitive to outliers. It can measure the correlation between variables that do not have a linear relationship. Compared to other correlation measures such as Pearson, the Spearman correlation coefficient is more suitable for the characteristics of the advertising CTR prediction dataset. The computation of the Spearman coefficient takes the training data samples as input and outputs a correlation coefficient matrix, which describes the degree of correlation between different features in the advertising CTR prediction dataset.

Dimensionality reduction. To facilitate the exploration of data instances by users, we employ a flexible nonlinear dimension reduction algorithm called UMAP [18]. The hyperparameters, `n_neighbors` and `min_dist`, which control the effectiveness of dimension reduction, are adjusted. Experimental results show that the default values for both parameters yield good results. Therefore, we use the default values: `n_neighbors=15` and `min_dist=0.1`. The data samples are reduced to a two-dimensional plane, and the resulting coordinate data is used in the subsequent visualization method of the hexagonal binning chart. We choose UMAP as the dimension reduction method because the advertising CTR prediction dataset typically contains numerous nonlinear relationships and complex interactions. Linear methods, such as PCA, assume linear relationships between data points and fail to capture the nonlinear structures in the dataset effectively. Additionally, compared to the nonlinear t-SNE dimension reduction method, UMAP can handle large-scale datasets more efficiently while maintaining the

[1] https://www.kaggle.com/datasets/louischen7/2020-digix-advertisement-ctr-prediction

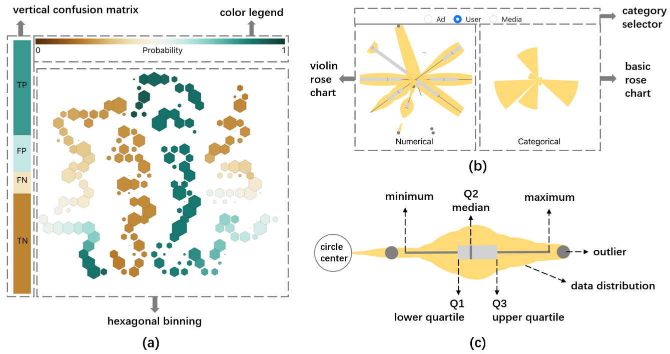


Fig. 3. Design details of Data Overview View and Data Statistics View.

quality of dimension reduction.

4 VISUALIZATION

We design a novel interactive visual analytics system, GBDT4CTRVis, to implement the above analysis requirements. GBDT4CTRVis can help advertising analysts understand the working mechanism of the GBDT-based CTR prediction model through the instance, feature and model level and facilitate the tuning process. As shown in Fig. 1, GBDT4CTRVis contains three levels of views: 1) Instance-level views, which use a hexagonal binning chart, two rose charts, and a table-based chart to explore advertising data instances from the perspectives of data overview, data statistics, and data details respectively **(R1)**; 2) Feature-level views, which use a list combined with dual-axis plot to display feature importance and use a node-link chart to show intra-class and inter-class correlations among three types of features: ad, medium, and user **(R2)**; 3) Model-level views, which use icicle charts to display the structure of representative decision trees and use streamgraph to express the temporal evolution of information gain in the model prediction process **(R3)**. GBDT4CTRVis also provides a control panel that allows users to adjust hyperparameters and validate their insights.

4.1 Instance-level Views

Instance-level views provide an overview of the prediction results and the instance clusters. By analyzing the distribution of feature values and detailed information of the instance cluster, instance-level views help to analyze the prediction results of the model, construct user profiles for different clusters, and further provide insights for model tuning at the data level **(R1)**. Instance-level views include Data Overview View (Fig. 1B), Data Statistics View (Fig. 1C), and Data List View (Fig. 1D).

Data Overview View (Fig. 1B) integrates the design of the confusion matrix and hexagonal binning chart, providing users with an overview of the prediction results of data instances. It consists of a color legend, a vertical confusion matrix, and a hexagonal binning chart. As shown in Fig. 3a, **the color legend** specifies the color encoding for Data Overview View, using two colors and their gradients to display the prediction scores of samples, where the prediction score of 0 is represented by dark brown, 1 is represented by dark green, and scores between 0 and 1 are linearly mapped using a gradient color line between. **The confusion matrix** consists of four rectangles arranged vertically from top to bottom, representing four classification results: true positive (TP), false positive (FP), false negative (FN), and true negative (TN). The height of each rectangle encodes the number of samples included in each type of classification result, with a longer rectangle indicating more samples. **The hexagonal binning** chart uses hexagons to aggregate similar data samples that fall within its boundaries after dimensionality reduction by UMAP. Hexagons located closely together representing data samples have similar features and prediction scores. The area of the hexagon encodes the number of samples, with a larger area indicating more samples. Compared to the scatter plots commonly used in traditional two-dimensional dimension reduction methods, the hexagonal binning chart aggregates data into coarser hexagons. This representation makes better use of visual space and alleviates visual clutter and browser pressure. Specifically, while a scatter plot with thousands or even tens of thousands of samples can be overwhelming, the hexagonal binning chart can present the same information using only a few hundred aggregated hexagons. The color and size of each hexagon can also encode different statistical information about the data samples it represents. Detailed information about the data within each hexagon can be viewed through the Data List View (Fig. 1D) for further analysis of the dimension reduction results.

Data Statistics View (Fig. 1C) uses two rose charts to display statistical information on numerical and categorical features. It consists of a category selector, a violin rose chart, and a basic rose chart. **The category selector** is used to switch the types of features to be presented in the rose charts, including three categories of features: ad, medium, and user, corresponding to the background colors of pink, yellow, and cyan in the rose charts. **The violin rose chart** is used to display statistical information on numerical features, consisting of several violin plots in a radial layout. Each violin plot encodes the shape of the distribution of one feature, arranged clockwise according to the order of the original data attribute fields. Each violin plot can be visualized as a "petal", as shown in Fig. 3c, integrating the design of the box plot and density plot. The position on the symmetric axis encodes the data value, with smaller values closer to the circle center. In the box plot, the side of the gray rectangle near the circle center represents the lower quartile $Q1$, the side far from the circle center represents the upper quartile $Q3$, and the line segment in the middle of the rectangle represents the median $Q2$. The proximal and distal ends of the line segment on the symmetry axis represent the minimum and maximum values in the normal data range, respectively. The upper and lower limits of data are determined using the interquartile range (IQR), i.e., $[Q1-1.5*IQR, Q3+1.5*IQR]$, and the minimum and maximum values are found within this range. Data points outside the upper and lower limits are considered outliers and represented by gray dots. In the density plot, the curvature encodes the contour of the data distribution, with higher curvature indicating denser data. The violin rose chart presents the density distribution of data through variations in shape. Compared to traditional box plots, it highlights the shape of data distributions, making it easier to identify the characteristic profiles that advertising analysts are interested in. Additionally, this chart has high visual aesthetics. Furthermore, compared to traditional horizontally-oriented violin plot, the radial layout design in this paper strikes a balance between information communication and visual appeal, while also optimizing the utilization of visual space. **The basic rose chart** is used to show statistical information on categorical features, where each petal represents a categorical feature, and the sector's radius encodes the frequency of the most frequently occurring value in that categorical feature.

Data List View (Fig. 1D) presents detailed information on raw data in a table format. As shown in Fig. 1D, each row in the table represents one sample in the dataset, with the first three columns representing its ID, prediction score, and actual label, respectively, and the following columns representing features. The background color of the second column represents the prediction score of the sample, using the same color mapping as in Data Overview View. The background color of the feature columns represents the category of the feature, using the same color mapping as in Data Statistics View with the rose diagrams.

Interaction: Analysts can click on the hexagons or confusion matrix of interest in (Fig. 1B) to update the data source for the Data Statistics View (Fig. 1C) and Data List View (Fig. 1D) with the data in that hexagons or confusion matrix. Then, analysts can analyze the user profile of the cluster, identify outliers in the cluster data through Fig. 1C, and view detailed information through Fig. 1D. When a data record in Fig. 1D is clicked, GBDT4CTRVis updates the streamgraph described in Section 4.3 for further analysis. We also reduce the visual burden and improve space utilization by employing techniques such as highlighting, tooltip, zooming, dragging (panning), filtering, etc.

4.2 Feature-level Views

Feature-level views are designed to help users analyze the importance ranking of all features, the impact of different values of a single feature on prediction results, as well as the intra-class and inter-class correlations between features, thereby providing insights for model tuning at the feature level **(R2)**. Feature-level views include Feature Importance View (Fig. 1E) and Feature Correlation View (Fig. 1F).

Feature Importance View (Fig. 1E) presents the importance ranking of features and the impact of individual features on prediction results, based on a list combined with dual-axis plot. As shown in Fig. 1E, this view consists of a scrollable list, with each column representing the feature name, its importance score, and a partial dependence plot (PDP) in a dual-axis plot. Each row represents a feature, sorted by default from highest to lowest feature importance. In the PDP, the lines represent the impact of the feature value on the model prediction results. The dark blue line represents the overall impact of the feature on the prediction results (across all samples), while a light blue line represents the prediction of a single sample on that feature. The grey rectangle represents the distribution of the feature, with categorical features represented by discrete bar charts and numerical features represented by continuous histograms. The "dual-axis plot" refers to the combination of the line chart with the bar chart or histogram. The traditional PDP only includes the dark blue lines in the plot and can analyze the overall impact of different feature values on model predictions. However, it has two limitations: first, it can only reflect the average level of the feature variable and overlook the influence of data heterogeneity on the results; second, it lacks information about the feature distribution, which may lead to overinterpretation of regions with scarce data and result in misjudgment. The PDP with dual-axis chart proposed in our paper addresses both of these issues.

Feature Correlation View (Fig. 1F) uses a node-link chart to explore the intra-class and inter-class correlations between features, consisting of a legend in the upper left corner and a node-link network on the right side. As shown in Fig. 1F, in the node-link network, each circle represents a feature, and the line between the two circles represents the correlation between the corresponding features. The width of the line encodes the strength of the correlation. And the larger the correlation, the thicker the line. When hovering over the corresponding element, GBDT4CTRVis displays the corresponding tooltip. The rectangle in the upper left corner indicates the color legend. The circles and rectangles corresponding to three types of features: ad, medium, and user, are also respectively colored pink, yellow, and cyan. Compared to the traditional heatmap of a feature correlation matrix used to display feature correlations, the node-link chart allows for intuitive identification of the thickness of the lines and facilitates rapid identification of highly correlated features through interactive techniques. It visually distinguishes and compares intra-class and inter-class feature correlations based on different node colors. Additionally, it effectively avoids data redundancy and reduces visual burden caused by the traditional heatmap of a feature correlation matrix.

Interaction: Feature Importance View displays all features by default, sorted by feature importance from highest to lowest score. Analysts can filter the displayed features by clicking the button to the right of "Name" and change the sorting rules by clicking the button to the right of "Score". When clicking a feature represented by a row in the Feature Importance View, GBDT4CTRVis highlights the corresponding rectangle of the node split by that feature in the Icicle View described in section 4.3. Meanwhile, GBDT4CTRVis magnifies the corresponding feature nodes in the Feature Correlation View, and pops up a tooltip showing the feature name for user analysis.

4.3 Model-level Views

Model-level views assist users in analyzing the structure of a single representative decision tree, observing the overall structural evolution of the decision trees, and exploring the evolution of the information gain of an instance in the model prediction process, so as to better understand the model's decision process, and further provide insights for model tuning at the model level **(R3)**. Model-level views include Icicle View (Fig. 1G), Area View (Fig. 1H), and Streamgraph View (Fig. 1I).

Icicle View (Fig. 1G) displays the most representative K decision trees obtained using the method described in section 3.4.2. As shown in Fig. 1G, K horizontally arranged icicle charts are displayed from left to right to represent the structures of the K decision trees. The two lines of text below each icicle chart indicate the index number of the cluster center and the number of trees contained in that cluster, respectively. In the icicle chart, each rectangle represents a node in the tree structure, with the leftmost rectangle representing the root node and the rightmost rectangles representing leaf nodes. The area of a rectangle encodes the number of samples passing through that node during model training, with a larger area indicating a greater number of samples. Icicle View has both normal mode and highlight mode. In normal mode, the rectangles representing the root and leaf nodes are coded in gray, while those representing the internal nodes are coded in blue. In highlight mode, the color of each rectangle corresponding to a non-leaf node is determined by the splitting feature of that node, with ad, medium, and user features being colored pink, yellow, and cyan, respectively. Compared to other tree visual representations, the node-link chart has a lower spatial utilization and its visual effect may not be as prominent. Although treemaps can effectively display the proportion of nodes, they may lack intuitive representation of hierarchy. The icicle chart we used can preserve the hierarchical structure while efficiently utilizing space to visually and effectively encode proportion information.

Area View (Fig. 1H) is located below the Icicle View and is used to observe the evolution of tree size. The horizontal axis represents the sequence number of all decision trees, and the vertical axis represents the size of each tree, i.e. the number of all nodes, with the area colored gray by default. When a user selects a certain instance, GBDT4CTRVis superimposes a layer of blue area chart on the Area View. The horizontal axis likewise represents the sequence number of all decision trees, while the vertical axis represents the values of the leaf nodes reached by the sample through each decision tree, with positive values located above the horizontal axis and negative values located below.

Streamgraph View (Fig. 1I) is used to present the evolution of the information gain during the model prediction process. As shown in Fig. 1I, from left to right, the white dotted lines represent the 0-th to $(n-1)$-th decision trees generated in sequence during the training process of the prediction model, among which the dotted lines representing cluster centers are thicker and more opaque. When a user moves the mouse over a certain dotted line, GBDT4CTRVis highlights the line in black and pops up a tooltip with the sum of the information gain from different classes splitting features when the data sample passes through that decision tree. At the corresponding position of each tree, the overall width of the river band indicates the sum of the information gain of all splitting nodes (non-leaf nodes) on the decision path as the data sample passes through that tree, and the different categories of splitting features are colored with the corresponding colors as described previously. Streamgraph is capable of displaying the changes in multiple factors, emphasizing the process of information transmission, and guiding attention. It also possesses a high level of visual aesthetics. This makes the streamgraph an effective tool that can assist advertising analysts in better understanding and analyzing the evolution of information gain during the model prediction process.

Interaction: GBDT4CTRVis provides highlight and tooltip interactions for each rectangle in the Icicle View, the text below the icicle charts, and each dotted line in the Streamgraph View. When a user moves the mouse over the first line of text label below, GBDT4CTRVis highlights the corresponding dotted line in the Streamgraph View at the appropriate position and a tooltip will be displayed. When a dotted line is clicked, GBDT4CTRVis highlights the decision path of the data sample in the Icicle View corresponding to that decision tree. When the mouse is moved over the second line of text label below, Streamgraph View marks all the positions of the decision trees in that cluster with black dotted lines.

4.4 Control Panel

We provide a control panel to assist analysts in model tuning and evaluating model performance. As shown in Fig. 1A, analysts can adjust the maximum number of leaf nodes of the decision tree (`num_leaves`), maximum depth of the decision tree (`max_depth`), number of decision trees (`n_estimators`), learning rate (`learning_rate`), number of representative decision trees (`clusters_k`), and features participating in model training. After clicking the button "Apply", analysts can check the adjusted model performance and the updated views. The control panel provides three commonly used evaluation metrics for CTR prediction models: AUC (Area Under the ROC Curve), AUPR (Area Under the Precision-Recall Curve), and ACC (Accuracy) [1] for analysts to refer to.

5 EVALUATION

To investigate the effectiveness of GBDT4CTRVis, we invited four experts who worked with us during the design process (E1-E4) mentioned in section 3.2, two machine learning model researchers (E5 familiar with GBDT, E6 not familiar with GBDT), and four visual analytics researchers (E7-E10, among them, two individuals have experience in model visual analysis, while the other two have experience in the visual analysis of advertising data.) to participate in the evaluation experiment. There are 5 males and 5 females, with an average age of 26.3, with an average domain experience of more than 3 years. They include industry experts, researchers, and graduate students. The entire evaluation process took an average of approximately 70 minutes.

The procedure for the study consisted of 4 steps. First, we spent 5 minutes briefly introducing the background and research content of our work to the 10 experts. Second, we spent 15 minutes demonstrating the visual encoding and interaction of GBDT4CTRVis through an example. Next, experts spent 40 minutes freely exploring the data described in section 3.4. We encouraged the experts to think aloud and say what they thought and did during their exploration. Finally, we invited the experts to complete a questionnaire to quantitatively evaluate the visual encoding, interaction design, and system functionality, and collect their feedback and suggestions.

We summarize three representative case studies found by our experts in Section 5.1 to fully illustrate how experts use GBDT4CTRVis to explore the GBDT-based CTR prediction model and tune the model. Then, we report the results of the questionnaires in Section 5.2.

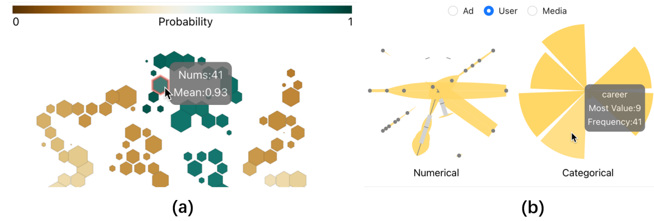
5.1 Case Study

5.1.1 Exploring Prediction Results of Data Instances

Exploring prediction results of data instances is a starting point for analyzing model performance and data characteristics. In this case, we describe how E1 uses GBDT4CTRVis combined with ad data instances to analyze the prediction results of the GBDT-based CTR prediction model and perform tuning (R1).

E1 started by analyzing the prediction results of the model with default hyperparameters (Fig. 1). He selected a hexagon with deeper color and larger area (Fig. 4a) in the Data Overview View (Fig. 1B) and analyzed the feature statistics information of this data cluster in the Data Statistics View (Fig. 4b). From the basic rose chart on the right side of the user type, E1 found that all samples in this data cluster have a career value of 9 and a gender value of 3. In addition, these samples have a net_type and purchase_tag value of 2, and most samples have a city value of 287 (these values are coded for anonymization). E1 further checked and validated the specific feature values of each instance by dragging the horizontal slider to the feature of user type in Data List View (Fig. 4c), and found that the values of these features were indeed largely consistent. E1 explained that the model performs well in predicting data with these data features. He also indicated that the same ads can be pushed to other similar users based on the features of the ad type data corresponding to these users.

Furthermore, E1 clicked on the first instance (record "4227959") to analyze its prediction result. As shown in Fig. 5a, E1 found that this sample could generate more information gain in the first 18 decision trees, and the subsequent decision trees corrected the deviation between the predicted value and the true value. However, this sample produced negative prediction scores between the 80th and 90th decision trees, but the evolution of information gain indicated that the deviation had not been fully corrected. E1 speculated that it might be due to insufficient basic learners. If there were more basic learners, perhaps more detailed information could be captured. E1 adjusted the number of



Record ID	Probability	Label	city	city_rank	device_name	device_size	career	gender	net_type	residence	emui_version
4227959	0.93191	1	287	2	52	193	9	3	2	11	24
33457928	0.93162	1	287	2	29	141	9	3	2	11	20
20761374	0.93099	1	287	2	29	141	9	3	2	11	20
2009611	0.92908	1	287	2	49	316	9	3	2	11	24
7989286	0.92905	1	287	2	77	316	9	3	2	11	24
5694577	0.92656	1	287	2	46	141	9	3	2	11	24

(c)

Fig. 4. Explore the prediction results of ad data instances hierarchically.

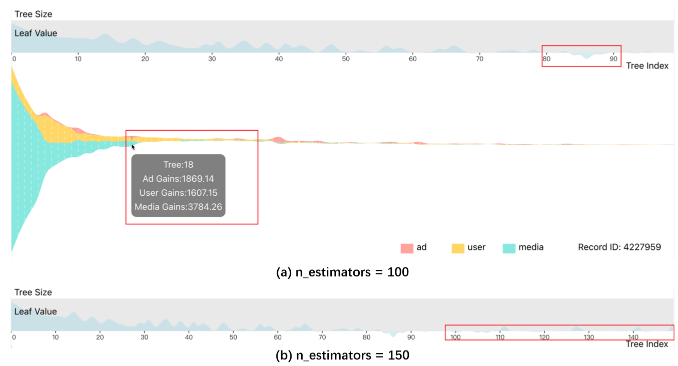


Fig. 5. Increase base learners to tune the model. (a) Analysis of leaf value and information gain evolution of an instance in prediction. (b) Area view after adjusting the hyperparameter `n_estimators`.

basic learners (n_estimators) from the default 100 to 150 in the control panel (Fig. 5b). After checking the predicted values of the leaf nodes in the updated blue area chart, E1 found that the new decision trees did learn more detailed information. And the model's evaluation metrics AUC increased from the original 0.718 to 0.722, AUPR increased from 0.720 to 0.724, and ACC increased from 0.664 to 0.666. In addition, E1 found that the fitting speed of predicted values to residuals was slow and suggested increasing the learning rate of the model. Then E1 verified his speculation through similar adjustment operations.

Overall, E1 was able to successfully combine ad data instances to analyze the prediction results of the GBDT-based CTR prediction model. He could understand the mechanism of the model well from the perspective of data, propose hypotheses for model tuning, and verify them through the control panel.

5.1.2 Analyzing and Selecting Features

Analyzing the importance of features in model prediction and the correlation between features can help analysts identify key factors that affect model performance, and select suitable features for model training. In this case, we describe how E2 used our system to analyze the role of ad data features in model prediction and select features for training (R2).

E2 found from the descending ranking (Fig. 6) that `task_id`, `material_id`, and `slot_id` had a significant impact on the model's prediction results. He clicked on these three features and found from the Icicle View (Fig. 7, where the rectangles corresponding to the features were highlighted) that these features were indeed used multiple times as the basis for decision tree node splitting. E2 also found from the Feature Correlation View (where the corresponding feature nodes were

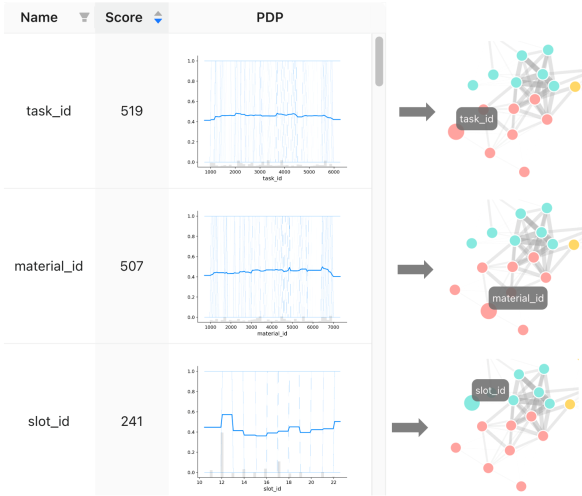


Fig. 6. Importance and correlation analysis of features.

Fig. 7. Split node distribution of important features.

enlarged) that these features were distributed around the periphery of the view and had no obvious correlation with other features, indicating that these features were more distinctive.

E2 analyzed the densely distributed nodes in the center of the Feature Correlation View and found that the "`app_score`" feature of the media type was strongly correlated with multiple features (Fig. 8). He clicked on one of the links and found that the correlation coefficient between "`app_score`" and "`app_first_class`" was 1, indicating that they had a very strong correlation. E2 also noticed from the corresponding PDP that they had the same data distributions and impacts on the model's prediction results, indicating that they were redundant features. E2 speculated that eliminating one of the features may be able to improve the model performance.

To verify his hypothesis, E2 removed the feature "`app_score`", which was strongly correlated with other features and had no contribution to the model's prediction results, from the control panel while keeping the other default model hyperparameters unchanged. He found that the model's performance improved slightly, with AUC increasing from 0.718 to 0.719, AUPR increasing from 0.720 to 0.721, and ACC remaining at 0.664.

In conclusion, E2 believed that GBDT4CTRVis provided an intuitive and effective view that could help him analyze the importance and correlation of features to some extent, thus gaining insights into feature selection.

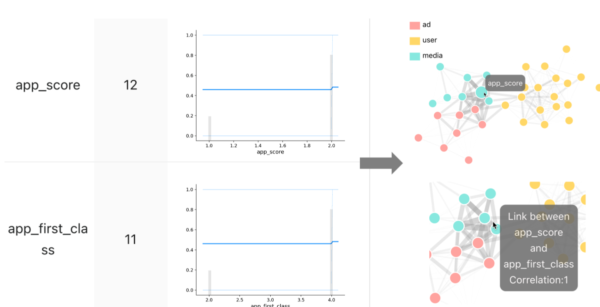
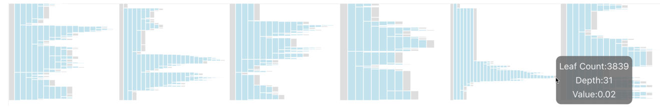
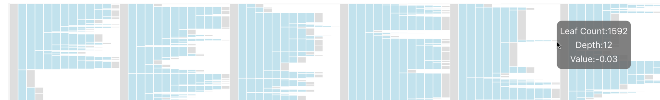


Fig. 8. Analysis of features with strong correlations.



(a) num_leaves=100, max_depth=-1



(b) num_leaves=100, max_depth=12

Fig. 9. Icicle view after adjusting the `num_leaves` and `max_depth`.

5.1.3 Analyzing and Tuning Model Structures

Analyzing the model structure can help to tune the model in a targeted manner. This case aims to describe how E3 used GBDT4CTRVis to diagnose and tune the model structure of the GBDT-based CTR prediction model (R3).

From the Icicle View (Fig. 1G) under default hyperparameters, E3 found that the overall model structure was relatively simple, and there was a large difference in the number of samples contained in the left and right nodes of most representative decision tree root nodes. E3 said this may cause the model to fail to fully learn the differences between samples, and he felt that the maximum number of leaves could be adjusted to increase the complexity and fitting ability of the model. To verify his hypothesis, E3 adjusted the maximum number of leaves (`num_leaves`) from the default 31 to 100 in the control panel and found that the evaluation indicators AUC, AUPR, and ACC of the model were all improved: AUC increased from the original 0.718 to 0.726, AUPR increased from 0.720 to 0.730, and ACC increased from 0.664 to 0.670. Then, E3 observed the structure of the model again and found a decision tree with a depth of 31 (Fig. 9). He said that too deep trees might cause overfitting of the model, reduce the speed of model operation, and lower the generalization ability of the model. E3 then set `max_depth` to 12 in the control panel and found that the model performance was consistent with before, but the depth of the representative decision tree did not exceed 12, and the distribution of nodes was more reasonable. E3 indicated that this hyperparameter combination simplified the structure of the base learner and improved computational efficiency while ensuring that the model evaluation indicators did not decrease.

After learning about the analysis content of E1 and E2, E3 set `num_leaves` to 100, `max_depth` to 12, `n_estimators` to 150, and `learning_rate` to 0.25 on the control panel, and removed redundant features. He found that the performance of the model was further improved, and the three evaluation indicators on the test set were higher than when the above hyperparameters worked alone. Specifically, AUC increased from 0.718 to 0.729, AUPR increased from 0.720 to 0.733, and ACC increased from 0.664 to 0.672. E3 said that in the field of ad CTR prediction with imbalanced positive and negative samples, a 0.1% improvement in these indicators could already bring enormous benefits. Existing research [14] has also demonstrated this point.

Therefore, E3 recognized the effectiveness of GBDT4CTRVis in analyzing and tuning the GBDT-based CTR prediction model structure. He also appreciated our system design and analysis pipeline. He believed that GBDT4CTRVis could indeed help advertising analysts understand the basic working mechanism of the GBDT-based CTR prediction model from the three levels of instance, feature, and model, and simplify the process of model tuning.

5.2 Expert Evaluation

We referred to related work in the field of visual analytics [7, 12, 37, 41] and designed 9 questions based on the system's usefulness and effectiveness, as shown in Table 1. Experts need to rate each question based on the 5-point Likert scale (1-5 represents "strongly disagree" to "strongly agree"). The results of the questionnaire are shown in Fig. 10.

Table 1. Questionnaire of the expert evaluation.

No.	Question
Q1	I think it's easy to learn the system
Q2	I think it's easy to understand the visual design of the system.
Q3	I think it's easy to interact with the system.
Q4	I am satisfied with the responsiveness of the system.
Q5	I think it's easy to explore the model's prediction results at the instance level.
Q6	I think it's easy to analyze the model decision-making basis at the feature level.
Q7	I think it's easy to understand the model's decision-making mechanism at the model level.
Q8	I think it's easy to perform model tuning through the system.
Q9	I think this system is very helpful in improving the effectiveness of ad CTR prediction models.

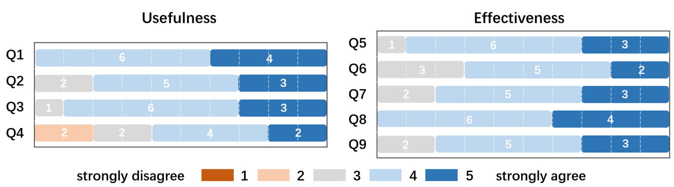


Fig. 10. Statistics of the questionnaire results, 1-5 represents "strongly disagree" to "strongly agree".

We report the results of the questionnaire survey and expert evaluation below from the perspectives of usefulness and effectiveness.

Usefulness. Almost all experts expressed that they can easily learn and use this system (Q1_mean=4.4, Q3_mean=4.2), and the design of the views is intuitive and easy to understand (Q2_mean=4.1). "The design of GBDT4CTRVis is very user-friendly, and I can quickly grasp the use of the system." -E8. "The visualization is divided from three perspectives: the instance, model, and feature are easy to understand and remember, and the color matching of the system is also quite identifiable." -E4. In terms of system response speed (Q4_mean=3.6), two machine learning experts, E5 and E6, expressed that further optimization is needed because they frequently modify model hyperparameters through the control panel to observe the effects, but the response time for the system to return calculation results exceeds their waiting threshold. Depending on the parameters set by the user, it usually takes about 2 minutes. They said it was tolerable and noted that it could be improved with higher performance computing resources. On this issue, experts in the visualization field gave high scores because they adjust model hyperparameters less frequently and attach more importance to the interaction of the system. The statistical results of Q1-Q4 indicate that the vast majority of experts gave high evaluations of the system's usefulness.

Effectiveness. The vast majority of experts believed that they could effectively explore the model's prediction results at the instance level through the system (Q5_mean=4.2). "GBDT4CTRVis met the need for analyzing the prediction results of data instances. I could quickly adjust model hyperparameters by analyzing these results," said E1. However, expert E9 suggested that the visualization design of the rose chart could be further optimized, such as using an adaptive layout according to the number of features. Most experts believed that the system could analyze the model's decision basis at the feature level, including the importance, correlation, distribution of features, and the impact of feature values on prediction results (Q6_mean=3.9). "I really liked the feature-level design because GBDT4CTRVis visualized all aspects of the features I cared about," said E2. However, E7 mentioned that the effect of some partial dependence plots was not significant enough, and complementary experiments could be conducted on datasets with real continuous features. In terms of understanding the model's decision mechanism at the model level, most experts believed that they could diagnose and tune the model through the system (Q7_mean=4.1). "I found multiple ways to improve the model while exploring the system, such as tuning the tree structure based on the insights from the Icicle View," said E3. "If there were other visualization views of the decision tree structure and node split, perhaps more effective insights could be obtained," E2 suggested. E2 also suggested supplementing the prompt box in the river chart to specify which features produced information gain. All experts acknowledged the effectiveness of this system in model tuning (Q8_mean=4.4). E1 and E3 both stated that GBDT4CTRVis allowed them to intuitively understand the model's working mechanism, discover directions for model tuning, and verify their findings through the control panel. Expert E5 mentioned that existing hyperparameter optimization solutions such as grid search could be integrated into the control panel to provide more references. Experts believed that the system was helpful in improving the accuracy of ad CTR prediction (Q9_mean=4.1), and E2 also hoped that the system could support the analysis of custom datasets.

In general, experts recognized the usefulness and effectiveness of the prototype system, believing that the system was easy to operate and could effectively meet the requirements described in section 3.2. Experts also made some constructive comments to guide our future work.

6 DISCUSSION

In this section, we first discuss the lessons learned from our design research and then discuss the limitation and generalizability of GBDT4CTRVis.

Lessons learned. Close communication and collaboration with domain experts provided valuable experience for developing GBDT4CTRVis. Inspired by domain experts, we designed visualization methods at three levels: instance, feature, and model to reveal the mechanism of the GBDT-based CTR prediction model. At the instance level, we used a hexagonal binning chart instead of traditional scatter plots to better utilize visual space and mitigate visual clutter. At the feature level, we addressed the issue of insufficient information encoding in traditional partial dependence plots (PDP) by using dual-axis chart to simultaneously encode the overall impact of different feature values on model predictions, the impact on individual sample predictions, and feature distribution. We also provided rich interaction through a node-link chart to explore the associations between features, avoiding data redundancy and the visual burden of matrix heatmaps. At the model level, we employed an icicle chart to present the structure of decision trees, effectively utilizing space to intuitively and efficiently encode proportion information while preserving the hierarchical structure. We also used streamgraphs to visually understand the evolution of information gain. This paper helps advertising analysts understand the mechanism of the GBDT-based CTR prediction model and simplifies the process of model tuning through these three levels of visualization methods.

Limitation. Case studies and expert evaluations confirm the usefulness and effectiveness of GBDT4CTRVis, but there are still some limitations. (1) System response time: When users modify hyperparameters to make the model structure more complex or lower the learning rate, it increases the computation of the system and makes the system response slower. In addition, calculating the tree edit distance also consumes a considerable amount of time. We can consider using GPUs to improve computational efficiency, and optimizing the time complexity of the tree edit distance algorithm to improve user experience. (2) Model tuning efficiency: The current model tuning solutions are relatively simple. We can further improve the efficiency of model tuning by combining common hyperparameter optimization solutions such as grid search and adding other model hyperparameters. (3) Visual and interactive design: We can optimize the layout design of the rose chart, increase the significance of some partial dependence plots, provide a magnifying glass function for the river chart, and provide a view interface for analyzing each decision tree or a custom dataset analysis interface for users to enrich the functionality and user experience of the system.

Generalizability. Our system has good scalability and can be easily applied to other fields that apply GBDT for binary classification on structured datasets, such as financial risk control and medical diagnosis. Before changing the domain dataset to be analyzed, users can categorize all features into three categories according to their own needs, or leave them uncategorized, in which case they will be treated as a single category by default. The system can then utilize GBDT to complete binary classification, and users can refer to the analysis cases provided in our paper to analyze the model's prediction results and perform model tuning.

7 CONCLUSION AND FUTURE WORK

We propose a visual analytics system, GBDT4CTRVis, which helps advertising analysts understand the working mechanism of the GBDT-based CTR prediction model from three levels: instance, feature, and model, thereby facilitating the process of model tuning. Based on the background of online advertising, this system combines the three main participants (ad, medium and user) in online advertising campaigns with the construction and tuning process of the CTR prediction model, allowing analysts to intuitively explore the ad data instances and their corresponding prediction results. Users can analyze the role of data features in model prediction from four aspects: the importance of features, the impact of feature values on prediction results, the distribution of feature values, and intra-class and inter-class correlations between features. Users can also understand the model's decision mechanism at the model level, analyze the model evolution, gain insight into model tuning, and validate it by controlling the panel. Our evaluation results show that GBDT4CTRVis can effectively help advertising analysts understand the working mechanism of the GBDT-based CTR prediction model and simplify the process of model tuning.

In future work, we plan to improve the response speed of the system by optimizing the tree edit distance algorithm and using GPUs for acceleration. We will also improve the system by providing hyperparameter optimization guidance and enriching the existing views to provide more functionality and more user-friendly interactions.

ACKNOWLEDGMENTS

We thank all our participants for their time and valuable input and appreciate Huawei's advertisement click-through rate prediction dataset which is publicly available at the 2020 DIGIX algorithm competition.

REFERENCES

[1] P. Bille. A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1):217–239, 2005. doi: 10.1016/j.tcs.2004.12.030

[2] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, and J. Q. n. Candela. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, ADKDD'14, p. 1–9. Association for Computing Machinery, New York, NY, USA, 2014. doi: 10.1145/2648584.2648589

[3] F. Hohman, A. Head, R. Caruana, R. DeLine, and S. M. Drucker. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–13. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300809

[4] B. Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, and G. Heidemann. Interactive learning of ad-hoc classifiers for video visual analytics. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 23–32, 2012. doi: 10.1109/VAST.2012.6400492

[5] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

[6] S. Jia, P. Lin, Z. Li, J. Zhang, and S. Liu. Visualizing surrogate decision trees of convolutional neural networks. *J. Vis.*, 23(1):141–156, feb 2020. doi: 10.1007/s12650-019-00607-z

[7] Z. Jin, Y. Wang, Q. Wang, Y. Ming, T. Ma, and H. Qu. Gnnlens: A visual analytics approach for prediction error diagnosis of graph neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 29(6):3024–3038, 2023. doi: 10.1109/TVCG.2022.3148107

[8] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[9] J. Krause, A. Perer, and E. Bertini. Infuse: Interactive feature selection for predictive modeling of high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1614–1623, 2014. doi: 10.1109/TVCG.2014.2346482

[10] J. Krause, A. Perer, and K. Ng. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, p. 5686–5697. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2858036.2858529

[11] T. Lee, J. Johnson, and S. Cheng. An interactive machine learning framework. *arXiv preprint arXiv:1610.05463*, 2016.

[12] H. Li, M. Xu, Y. Wang, H. Wei, and H. Qu. A visual analytics approach to facilitate the proctoring of online exams. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445294

[13] Z. Li, X. Wang, W. Yang, J. Wu, Z. Zhang, Z. Liu, M. Sun, H. Zhang, and S. Liu. A unified understanding of deep nlp models for text classification. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4980–4994, 2022. doi: 10.1109/TVCG.2022.3184186

[14] X. Ling, W. Deng, C. Gu, H. Zhou, C. Li, and F. Sun. Model ensemble for click prediction in bing search ads. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, p. 689–698. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017. doi: 10.1145/3041021.3054192

[15] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):91–100, 2017. doi: 10.1109/TVCG.2016.2598831

[16] S. Liu, J. Xiao, J. Liu, X. Wang, J. Wu, and J. Zhu. Visual diagnosis of tree boosting methods. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):163–173, 2018. doi: 10.1109/TVCG.2017.2744378

[17] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. 30, 2017.

[18] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[19] Y. Ming, S. Cao, R. Zhang, Z. Li, Y. Chen, Y. Song, and H. Qu. Understanding hidden memories of recurrent neural networks. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 13–24, 2017. doi: 10.1109/VAST.2017.8585721

[20] T. Mühlbacher, L. Linhardt, T. Möller, and H. Piringer. Treepod: Sensitivity-aware selection of pareto-optimal decision trees. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):174–183, 2018. doi: 10.1109/TVCG.2017.2745158

[21] H.-S. Park and C.-H. Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2, Part 2):3336–3341, 2009. doi: 10.1016/j.eswa.2008.01.039

[22] X. Qiu, Y. Zuo, and G. Liu. Etcf: An ensemble model for ctr prediction. In *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*, pp. 1–5, 2018. doi: 10.1109/ICSSSM.2018.8465044

[23] P. E. Rauber, S. G. Fadel, A. X. Falcão, and A. C. Telea. Visualizing the hidden activity of artificial neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):101–110, 2017. doi: 10.1109/TVCG.2016.2598838

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

[25] A. Rule, A. Tabard, and J. D. Hollan. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3173606

[26] C. Spearman. The proof and measurement of association between two

things. 1961.
[27] J. Talbot, B. Lee, A. Kapoor, and D. S. Tan. Ensemblematrix: Interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, p. 1283–1292. Association for Computing Machinery, New York, NY, USA, 2009. doi: 10.1145/1518701.1518895
[28] S. T. Teoh and K.-L. Ma. Paintingclass: Interactive construction, visualization and exploration of decision trees. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, p. 667–672. Association for Computing Machinery, New York, NY, USA, 2003. doi: 10.1145/956750.956837
[29] S. van den Elzen and J. J. van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 151–160, 2011. doi: 10.1109/VAST.2011.6102453
[30] F. Wang, X. Liu, O. Liu, A. Neshati, T. Ma, M. Zhu, and J. Zhao. Slide4n: Creating presentation slides from computational notebooks with human-ai collaboration. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23. Association for Computing Machinery, New York, NY, USA, 2023. doi: 10.1145/3544548.3580753
[31] X. Wang, G. Hu, H. Lin, and J. Sun. A novel ensemble approach for click-through rate prediction based on factorization machines and gradient boosting decision trees. In J. Shao, M. L. Yiu, M. Toyoda, D. Zhang, W. Wang, and B. Cui, eds., *Web and Big Data*, pp. 152–162. Springer International Publishing, Cham, 2019.
[32] Z. J. Wang, C. Zhong, R. Xin, T. Takagi, Z. Chen, D. H. Chau, C. Rudin, and M. Seltzer. Timbertrek: Exploring and curating sparse decision trees with interactive visualization. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pp. 60–64, 2022. doi: 10.1109/VIS54862.2022.00021
[33] M. WARE, E. FRANK, G. HOLMES, M. HALL, and I. H. WITTEN. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292, 2001. doi: 10.1006/ijhc.2001.0499
[34] W. Yang, X. Ye, X. Zhang, L. Xiao, J. Xia, Z. Wang, J. Zhu, H. Pfister, and S. Liu. Diagnosing ensemble few-shot classifiers. *IEEE Transactions on Visualization and Computer Graphics*, 28(9):3292–3306, 2022. doi: 10.1109/TVCG.2022.3182488
[35] J. Yuan, C. Chen, W. Yang, M. Liu, J. Xia, and S. Liu. A survey of visual analytics techniques for machine learning. *Computational Visual Media*, 7:3–36, 2021.
[36] J. Yuan, M. Liu, F. Tian, and S. Liu. Visual analysis of neural architecture spaces for summarizing design principles. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):288–298, 2023. doi: 10.1109/TVCG.2022.3209404
[37] C. Zhang, X. Wang, C. Zhao, Y. Ren, T. Zhang, Z. Peng, X. Fan, X. Ma, and Q. Li. Promotionlens: Inspecting promotion strategies of online e-commerce via visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):767–777, 2023. doi: 10.1109/TVCG.2022.3209440
[38] J. Zhang, L. Gruenwald, and M. Gertz. Vdm-rs: A visual data mining system for exploring and classifying remotely sensed images. *Computers and Geosciences*, 35(9):1827–1836, 2009. doi: 10.1016/j.cageo.2009.02.006
[39] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262, 1989. doi: 10.1137/0218082
[40] T. Zhang. Visual interpretation and analysis of random forest. Master's thesis, University of Electronic Science and Technology of China, 2021.
[41] J. Zhao, M. Fan, and M. Feng. Chartseer: Interactive steering exploratory visual analysis with machine intelligence. *IEEE Transactions on Visualization and Computer Graphics*, 28(3):1500–1513, 2022. doi: 10.1109/TVCG.2020.3018724
[42] X. Zhao, Y. Wu, D. L. Lee, and W. Cui. iforest: Interpreting random forests via visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):407–416, 2019. doi: 10.1109/TVCG.2018.2864475