**Explainability User Manual**

**Exploratory Data Analysis(EDA):**

**Line Plot**: The line plot shows the value of a selected feature for each data point.

**Histogram:** A particular bin in the x-axis represents a region in the sample space and the corresponding value in the Y-axis is nothing but the total no of samples in that region.

Non-technical: It helps us to understand what are the most frequent values, less frequent values, and rare values.

**Aggregation**: The process of summarising numbers into a single value is known as aggregation.

Example: Calculating the average age of people in the dataset.

**Probability density function:** A particular point in the x-axis represents a sample (in the sample space) and the corresponding value in the Y-axis can be understood as the probability that the random variable's(feature’s) value will be equal to that sample.

**Cumulative density function**: A particular point in the x-axis represents a sample (in the sample space) and the corresponding value in the Y-axis is the probability that the random variable’s(feature’s) value will be less than or equal to that sample.

**Box plot**: A box plot is a graphical representation of numerical data. A line inside the box represents the median using which we can understand the amount of spread and skewness in the data. Outliers are denoted by the points outside the box.

**Violin plot**: It is nothing but the combination of the box plot and probability density plot.

**Model Performance:**

**Classification report:** It shows the model’s score of commonly available classification metrics.

**Confusion matrix**: For each class, the confusion matrix tells us how many points are predicted as belonging to a particular class and how many of them actually belong to it. Each row indicates the actual class and each column represents the predicted class. we can infer which classes the model is confusing.

**Prediction probability chart**: The index of data points is represented on the X-axis, and the predicted probability is presented on the Y-axis.

**Individual conditional expectation(ICE)**: For each data point, we have a line(blue) indicating how the prediction varies when we change the selected feature's value.

**Partial dependence plot(PDP)**: It is nothing but the line(red) obtained by calculating the mean of all the lines returned by ICE. It shows how the feature affects the output of the model on average.

**Precision-recall curve**: Curve with precision-recall sums up the tradeoff between a true positive rate and precision with varying probability thresholds in a classifier for a particular class.

***Precision***: Out of all the data points predicted as positive how many are actually positive?

***Recall(True positive rate)***: How many positive data points are predicted as positive?

**ROC curve**: Similarly, the ROC curve sums up the tradeoff between a true positive rate and a false positive rate with varying probability thresholds in a classifier for a particular class.

***False positive rate***: Out of all the negative data points how many are predicted as positive? Simply: negative data points predicted as positive divided by the total no of negative data points.

**True positive rate(TPR) and false positive rate(FPR) at every threshold**: Threshold is represented on the X-axis, while TPR and FPR are represented on the Y-axis.

**Lift chart**: Lift is a statistic that compares a given model's performance to that of a random model. It is calculated by dividing the model's response by the average response. The data set is ordered based on the output probability and divided into bins. Companies may then examine each bin and determine whether to sell to that bin or not by evaluating the financial benefit against the cost.

The graph shows for a particular percentage of the data set, how many positive(favorable) data points are there compared to the random model(ie: compared to randomly ordered data points).

**Example for bin 1**:

Case 1: The company randomly chooses 10 people and approaches them.

Outcome: 1 person(favorable data point) buys the product.

Case 2: The company chooses the 10 most likely people to buy the product based on the model and approaches them.

Outcome: 5 people(favorable data points) buy the product.

Lift score is 5/1=5.

**Gain chart**: The gain chart also compares a given model's performance to that of a random model. Gain is the ratio between the cumulative number of positive(favorable) data points up to a percentage to the total number of positive data points in the data.

**Explainability:**

**Attribution:** The contribution of each feature in the model to the prediction of each data point is shown by feature attribution. Simply, Attribution = feature value \* feature importance.

***Usage:***

It helps end-users, domain experts, and data scientists to understand the model.

**Feature Importance(Global Analysis)**: we aggregate the attribution score to get the overall impact of each feature on the output.

***Usage:***

It helps data scientists to understand the importance of features and also helps domain experts to validate the trustworthiness of the model.

The attribution calculations are separated into three types which are defined as follows:

**Feature Importance(Primary Attribution)**: calculates the impact of each input feature on the output of a model.

***Usage:***

It helps end-users, domain experts, and data scientists to understand the model.

**Neuron Importance(Layer Attribution)**: calculates each neuron's impact in a selected layer on the model’s output.

***Usage:***

It helps data scientists to identify problems in the model and improve it.

**Feature Importance with respect to neurons (Neuron Attribution)**: calculates the impact of each input feature on the activation of a particular hidden neuron.

***Usage:***

It helps data scientists to identify problems in the model and improve it.

**Algorithms to calculate attribution:**

**Gradient-based Algorithms:**(all these algorithms are applicable for differentiable models only)

1**.Saliency**: simply calculates the gradient of the output with respect to the input. By default, it returns the absolute gradients.

***Drawback:***

Despite the fact that the output value for the input differs from the baseline, the prediction function may flatten at the input and so have a zero gradient. Also, It doesn’t consider input.

2.**Input X Gradient**: calculates the gradient of the output with respect to the input and multiplies it with input. The basic inspiration behind this idea is that the weights in the linear regression are nothing but gradients and hence input\*gradient gives attribution.

***Drawback:***

Despite the fact that the output value for the input differs from the baseline, the prediction function may flatten at the input and so have a zero gradient.

3.**Guided backpropagation and deconvolution**: calculates the gradient of the output with respect to the input but varies in the backpropagation of ReLU functions. In deconvolution, only positive gradients are backpropagated and in guided backpropagation, gradients are backpropagated only when both the gradient and input are positive. They were designed primarily for convolutional neural networks as the neurons function as detectors of a specific picture characteristic. As a result, we only focus on what visual features a neuron perceives, not what it doesn't.

***Drawback:***

It also suffers from the same problem wherein the attribution score will be zero even though the feature is useful.

4.**GradCAM:** calculates the gradient of the given convolutional layer(typically the final convolution layer) with respect to output and takes the global average for each channel. The average is multiplied with the corresponding layer activation and the values in the same position of different channels are summed together. Finally, negative values are converted to zero and the resulting values are returned as attribution scores. It is class discriminative, that is, it can localize the region respective to a particular class. It only gives the heatmap and not the pixel level contribution.

5.**Guided GradCAM**: GradCAM is combined with guided backpropagation to get class discriminative as well as pixel-level contribution. It is just the pointwise multiplication of attribution calculated using GradCAM and the guided backpropagation.

6.**Integrated gradients**: calculates the integral of the gradients along the straight-line path from the baseline to the input. Riemann Sum or Gauss Legendre quadrature rule is used to calculate the approximate area under the curve.

***Usage:***

It is not affected even when the prediction function flattens at the given input which is not the case with algorithms seen so far.

7.**Gradient SHAP**: adds Gaussian noise to each data point multiple times to get a few samples. It is the extension of integrated gradients. Instead of a single baseline, we integrate over all possible baselines represented by a distribution.

8.**DeepLIFT**: the slope between baseline and input is backpropagated instead of gradients.

***Usage:***

It is faster than integrated gradients as it needs only one backpropagation per data point.

9.**DeepLIFT SHAP**: calculates DeepLIFT attribution for each datapoint multiple times for different baseline values (chosen from a distribution) and returns average as the attribution.

10.**Conductance**: The conductance of a neuron is used to determine how important it is to the prediction of a particular data point. Conductance is nothing but the flow of attribution via a neuron. The attribution is calculated using integrated gradients.

11.**Internal Influence:** Gives the internal features of a neural network an importance score. It is similar to integrated gradients calculated with respect to a particular layer instead of input.

12.**Layer Activation:** is nothing but the activation of each neuron in the given layer.

**Permutation-based Algorithms:**(all these algorithms are useful for any kind of models but they are generally less efficient than gradient-based algorithms like Integrated gradients and DeepLIFT)

1.**Feature ablation**: masks(replaces) each feature(or group of features) of the given data point with baseline and returns the difference in output as attribution.

2.**Feature permutation**: takes a batch of data points as input, permutes the values of each feature, and calculates the difference in output or loss.

3.**Occlusion**: It is primarily used for images because of the similarity of pixels in a region. The values in the contiguous rectangular region are masked with the baseline value and returns the difference in output as attribution.

4.**Shapley value Sampling**: calculates the average marginal contribution of each feature(or a group of features) over a sample of random permutation of input features.

5.**Lime**: samples multiple data points around the given input and builds an interpretable surrogate model(linear model) to predict these points. It gives more weightage to the samples closer to the given data point. The coefficients of the surrogate model are returned as the attribution.

6.**Kernel SHAP:** is similar to LIME and returns the coefficients of the linear surrogate model as the attribution. The difference is that it samples from marginal distribution and also gives more weightage to small and large coalitions.

**Attribution Effectiveness Score:**

We have developed a novel metric called attribution effectiveness score to choose the best attribution algorithm for the given dataset and model.

***Usage:***

It automatically chooses the best attribution algorithm saving enormous time spent in manual evaluation. We can also compare the scores and do trade-offs between time complexity and performance.

**Plots:**

**Attribution Visualization:**

The attribution scores are plotted in a bar plot, with each bar representing a feature and the height of the bar representing the feature's attribution score.

***Usage***:

It makes the comparison between features much easier because it is ordered and presented neatly.

**Histogram:**

Generally, the attributions are calculated for each data point and then averaged to get the global score. The histogram shows the frequency distribution of each feature’s attribution scores.

***Usage***:

It helps us to verify how well the aggregation(average) represents the distribution.

**SHAP bar plot**: plots the contribution of the features for a particular data point’s prediction. We can select any data point from the dataset using the drop-down menu.

**SHAP average bar plot**: plots the average contribution of the features by calculating the mean over all data points.

**SHAP scatter plot**: displays how a particular feature affects the model’s prediction. Each dot represents a data point whose x coordinate is the feature value and the y coordinate is the corresponding SHAP value.

**SHAP heat map**: on the x-axis are instances, on the y-axis are model inputs, and on the color scale are SHAP values. The samples are arranged by their similarity of SHAP values using hierarchical clustering. Samples with the same result due to the same reason are grouped together.

**SHAP swarm/violin plot**: for each feature, the violin plot and swarm plot of SHAP values are displayed. In the swarm plot, each dot represents a data point.

**SHAP waterfall plot**: depicts the prediction for a specific data point where each row represents a feature and its attribution. Begin at the bottom of the plot and add the attribution score of each feature to arrive at the model prediction. The red color denotes negative attribution, while the blue color denotes positive attribution.

References:

1.<https://christophm.github.io/interpretable-ml-book/>

2.<https://arxiv.org/abs/1704.02685>

3.<https://arxiv.org/abs/1703.01365>

4.<https://arxiv.org/abs/1805.12233>

5.<https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html>

6.<https://arxiv.org/abs/1412.6806>

7.[https://glassboxmedicine.com/2019/10/06/cnn-heat-maps-gradients-vs-deconvnets-vs-guided-backpropagation/#:~:text=Deconvolution%20is%20defined%20as%20convolving,pooling%20layer%20for%20the%20reconstruction](https://glassboxmedicine.com/2019/10/06/cnn-heat-maps-gradients-vs-deconvnets-vs-guided-backpropagation/" \l ":~:text=Deconvolution is defined as convolving,pooling layer for the reconstruction)

8.<https://arxiv.org/abs/1311.2901>

9.<https://arxiv.org/abs/1312.6034>

10.<https://arxiv.org/abs/1602.04938>