teex: A toolbox for the evaluation of explanations

Jesús M. Antoñanzas\textsuperscript{a,c,}, Yunzhe Jia\textsuperscript{a}, Eibe Frank\textsuperscript{a,b}, Albert Bifet\textsuperscript{a,d}, Bernhard Pfahringer\textsuperscript{a,b}

\textsuperscript{a} AI Institute, University of Waikato, Hamilton, New Zealand
\textsuperscript{b} Department of Computer Science, University of Waikato, Hamilton, New Zealand
\textsuperscript{c} Department of Physics, Universitat Politècnica de Catalunya, Barcelona, Spain
\textsuperscript{d} LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France

\textbf{A B S T R A C T}

We present teex, a Python toolbox for the evaluation of explanations. teex focuses on the evaluation of local explanations of the predictions of machine learning models by comparing them to ground-truth explanations. It supports several types of explanations: feature importance vectors, saliency maps, decision rules, and word importance maps. A collection of evaluation metrics is provided for each type. Real-world datasets and generators of synthetic data with ground-truth explanations are also contained within the library. teex contributes to research on explainable AI by providing tested, streamlined, user-friendly tools to compute quality metrics for the evaluation of explanation methods. Source code and a basic overview can be found at \url{github.com/chus-chus/teex}, and tutorials and full API documentation are at \url{teex.readthedocs.io}.

\textbf{Code metadata}

Table 1 contains metadata for the code.

\section{Introduction}

Explainable Artificial Intelligence (XAI) is the field dedicated to making AI models human-understandable. An important part of this field are explainers methods, which, by generating explanations, give users a general overview of a model’s functioning (global explanation) or the reasoning behind a single prediction (local explanation). teex is a tool designed to evaluate explainer methods in XAI, particularly those that generate local explanations for classifications made by machine learning models (such as LIME [1]). teex provides an extensible collection of metrics that enable comparison between post-hoc and ground-truth local explanations. It also provides built-in support for multiple explanation types—saliency maps, decision rules, feature importance vectors, and word importance vectors—while aiming to be extensible in this regard. Although its use is not strictly bound to the availability of ground-truth explanations (e.g., it can be used to compare explanations generated by different methods), teex contains multiple, easy-to-access real-world and artificial datasets [2–5] with ground-truth explanations to enable benchmark comparisons 1. In the case of the real-world data included, expert annotations are provided as ground-truth explanations. To enable integration with related software, we provide wrappers for extraction and usage of local explanations from popular Python XAI libraries.

tee supports the usage of XAI evaluation methods in a (1) general, (2) extensible, and (3) simple way:

- By allowing evaluation of the most frequently used explanation types in a model- and explainer-independent manner.
- By clearly encapsulating functionality: evaluation and data generation methods exist within distinct modules, inside a sub-package for each explanation type. APIs are standardized between all modules and the architectural structure is clearly laid out.
- By providing single-line evaluation APIs (as shown in the example below) and comprehensive documentation, including tutorials and use cases. This enables seamless integration with evaluation pipelines.

\begin{verbatim}
from teex.saliencyMap.data import Kahikatea
from teex.featureImportance.eval import feature_importance_scores
X, y, exps = Kahikatea()[:]
metrics = ['f1score', 'cs', 'auc']
feature_importance_scores(exps, predExplanations, metrics)
# >>> [0.8, 0.7, 0.7]
\end{verbatim}

\subsection{Related software}

Evaluating the quality of explanations is a hard problem, mainly because there is no standardized set of metrics or methods to do so. In
Table 1

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Code metadata description</th>
<th>Please fill in this column</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Current code version</td>
<td>v1.1.3</td>
</tr>
<tr>
<td>C2</td>
<td>Permanent link to code/repository used for this code version</td>
<td><a href="https://github.com/chus-chus/teex">https://github.com/chus-chus/teex</a></td>
</tr>
<tr>
<td>C3</td>
<td>Permanent link to Reproducible Capsule</td>
<td><a href="https://github.com/chus-chus/teex/releases/tag/v1.1.3">https://github.com/chus-chus/teex/releases/tag/v1.1.3</a></td>
</tr>
<tr>
<td>C4</td>
<td>Legal Code License</td>
<td>MIT</td>
</tr>
<tr>
<td>C5</td>
<td>Code versioning system used</td>
<td>git</td>
</tr>
<tr>
<td>C6</td>
<td>Software code languages, tools, and services used</td>
<td>Python</td>
</tr>
<tr>
<td>C7</td>
<td>Compilation requirements, operating environments &amp; dependencies</td>
<td><code>tqdm</code>, <code>numpy</code>, <code>scikit-learn</code>, <code>scipy</code>, <code>sympy</code>, <code>Pillow</code></td>
</tr>
<tr>
<td>C8</td>
<td>If available Link to developer documentation/manual</td>
<td><a href="https://teex.readthedocs.io">https://teex.readthedocs.io</a></td>
</tr>
<tr>
<td>C9</td>
<td>Support email for questions</td>
<td><a href="mailto:jmariaan@waikato.ac.nz">jmariaan@waikato.ac.nz</a>, <a href="mailto:alvin.jia@waikato.ac.nz">alvin.jia@waikato.ac.nz</a></td>
</tr>
</tbody>
</table>

Fig. 1. Example observation from the Kahikatea dataset.

Table 2

Comparison of libraries that include functionality to evaluate explanations.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Captum</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>AIX360</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Torchray</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Quantus</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>teex</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

2. Software description

To evaluate the explanation quality, the required elements are demonstrated in Fig. 3, where $e$ is the explanation generated by an explanation method. $e$ explains the prediction $y$ made by a black box model $b$ for a given input $x$. To evaluate the quality of $e$, there needs to be a ground truth explanation $\hat{e}$. Now, given an evaluation function $Q$, we can compute $Q(e, \hat{e})$. See Fig. 2 for a concrete example of the evaluation process.

`teex` makes the evaluation process convenient by:

- Providing a collection of metrics $Q$ that are commonly used in the literature for evaluating explanations.
- Providing easy access to $\hat{e}$ for a collection of machine learning datasets, where the ground truth explanations for individual instances are available. This information is difficult to collect in practice and is not available in many traditional datasets.

2.1. Datasets

We provide datasets with ground-truth explanations for four explanation representations, including both real datasets and synthetic ones. All datasets share the same user API. In particular, `teex` includes, as of now:

- Image data with saliency maps as explanations.
- Text data with word importance as explanations.
- Tabular data with rules as explanations.
- Tabular data with feature importance as explanations.

2.1.1. Image data

We provide several image classification datasets with ground-truth saliency maps that are, e.g., suitable for the evaluation of the explanations of classifications obtained from convolutional neural networks:

- **Kahikatea**:1 contains images for Kahikatea classification. The Kahikatea is an indigenous plant in New Zealand. The data has

```
https://zenodo.org/record/5059769#.Y-OCsnZBwQ-
```
Fig. 2. Explanation evaluation procedure for a saliency map image explanation, given an expert explanation and an explanation generated by an external method. (1) First, the expert explanation is transformed into a binary 2D matrix, where each entry corresponds to a pixel, and is set to 1 or 0 depending on whether it contains the object or not. (2) Then, the generated explanation is transformed into a 2D matrix, where each entry is the normalized attribution (from 0 to 1) of the corresponding pixel. This matrix, depending on the quality metric that the user chooses, will need to be binarized by choosing a value threshold. (3) After this, both matrices are flattened into 1D vectors. (4) Finally, both vectors can be quantitatively compared using a selected metric $Q$.

Fig. 3. Evaluating explanation $e$ (generated with any explanation method) by contrasting it against ground truth $\hat{e}$.

human-annotated pixel-level explanations highlighting the tree pixels in individual images if Kahikatea trees are presented. It contains 519 images (232 images contain Kahikatea).

• CUB-200-2011\(^2\) and Oxford-IIIT Pet\(^3\) are well-known datasets frequently used for evaluating the accuracy of image classification techniques and are also available in teex. They exhibit over 19,000 images and 230 distinct classes.

An example image from the Kahikatea data and a corresponding explanation can be found in Fig. 1.

• The included synthetic image data generation method, adapted from [5] can produce an arbitrary number of images with pixel-level explanations of one class. An example can be found in Fig. 5. Given some parameters, first, a pattern image (yellow pixels in Fig. 5) is generated, then the images are generated with cells randomly colored in cyan, green or blue with black background. An image is considered positive if it contains the pattern and its true explanation consists of the pixels corresponding to the pattern.

2.1.2. Text data

For evaluation on text data, we provide, for now, a subset of the 20NewsGroup\(^4\) dataset with the word importance as explanations for individual articles:

2.1.3. Tabular data

Evaluating explanations on tabular data is another important task. We provide synthetic tabular data generation methods with two types of

\(^2\) https://www.vision.caltech.edu/datasets/cub_200_2011/
\(^3\) https://www.robots.ox.ac.uk/vgg/data/pets/
\(^4\) https://www.kaggle.com/datasets/crawford/20-newsgroups
of explanations – decision rules and feature importance – based on different underlying transparent models.

The example of feature importance is similar to the above example of word importance for text data. That is, an observation is a numerical list with an associated class, and its corresponding explanation is a list of numerical importance for each feature, bounded from −1 (inversely correlated with the observation’s class) to 1 (positively correlated). The data points are sampled from normally distributed clusters, and explained using their gradients. The observations are the same in the case of the synthetic data points are sampled from normal distributions, labeled by thresholding randomly generated linear functions, and explained using their decision rule explanations.

Example observation from the dataset.

Fig. 4. Example observation from the CUB-200-2011 dataset.

2.2. Metrics

Here we present an overview of the current quality metrics included in teex.

2.2.1. Feature importance

- Cosine Similarity [14]. If the explanations are vectors of feature importance, regardless of whether the values are binary or in the range [0, 1], we can measure the explanation quality using Cosine Similarity:

$$Q(e, \hat{e}) = \cosine(e, \hat{e}) = \frac{\|e \cdot \hat{e}\|}{\|e\| \cdot \|\hat{e}\|}$$

where $e \cdot \hat{e}$ is the dot product, and $\|e\|$ is the L2-norm of e. The closer the metric is to 1, the greater the explanation quality of $e$.

- Precision, Recall, $F_1$ score. For these metrics, both ground truth and prediction are binarized according to a user-defined threshold. Once this has been done, the explanation quality can be measured by the well-known precision, recall, and $F_1$ score metrics.

$$Q(e, \hat{e}) = Precision(e, \hat{e}) = \frac{|e \cdot \hat{e}|}{|\hat{e}|}$$

Precision measures how many selected features are truly important: its value is 1 if all features with non-zero importance in the generated explanation are also non-zero in the ground truth explanation.

$$Q(e, \hat{e}) = Recall(e, \hat{e}) = \frac{|e \cdot \hat{e}|}{|e|}$$

Recall measures how many truly important features are selected. Its value is 1 if all features with non-zero importance in the ground truth explanation are also non-zero in the generated explanation. Note that this can be easily achieved with an explanation that assigns non-zero importance to all features.

$$Q(e, \hat{e}) = F_1(e, \hat{e}) = \frac{2 \cdot Precision(e, \hat{e}) \cdot Recall(e, \hat{e})}{Precision(e, \hat{e}) + Recall(e, \hat{e})}$$

The closer the $F_1$ score, the harmonic mean of precision and recall, is to 1, the greater the explanation quality of $e$.

- AUC: The area under the ROC curve provides an alternative to the above metrics. In the case of this metric, only the ground truth is binarized and ground truth importance scores are used to obtain the ranking for the calculation of the area under the curve.

2.2.2. Saliency maps

In the context of image classification, a saliency map explanation $e$ for a prediction $f(x)$ is represented as a two dimensional array of the same size, where each entry in $e$ is a real number and provides the attribution of the corresponding pixel in $x$. For the evaluation of saliency maps, we provide the same metrics as in the case of feature importance. In this case, each pixel in an image is considered to be a feature: an saliency map explanation of size $(M, N)$ is flattened into a feature importance vector of length $M \times N$. 

---

**Fig. 5.** Example observation from the Synthetic image dataset generation method.
2.2.3. Decision rules

In the case of tabular data, \texttt{teex} can also process explanations in the form of decision rules, not just feature importance scores.

- Complete Rule Quality. Each rule explanation can be converted into a vector where, for the \( i \)th feature in the observed domain, the values of the lower and upper bounds \( v_i, l, u \) are reported. For example, the explanation \( x_0 > 5, x_1 < 2 \) can be converted to \( \{(x_0, 5), (x_1, -\infty), (x_0, -\infty), (x_1, 2)\} \). Then the explanation quality can be measured as:

\[
Q(e, \hat{e}) = \frac{1}{N} \sum_{i=1}^{|e|} \delta_i(e, \hat{e})
\]

where

\[
\delta_i(e, \hat{e}) = \begin{cases} 
1 & \text{if } |e_i - \hat{e}_i| \leq \epsilon \land |e_i| \neq \infty \land \hat{e}_i \neq \infty, \\
0 & \text{otherwise}.
\end{cases}
\]

Here, \( \epsilon \) is the similarity threshold, and \( N \) is the number of lower and upper bounds that are neither \( \infty \) nor \( -\infty \) in both \( e \) and \( \hat{e} \). This means that the closer a rule explanation’s lower and upper limits are to the real lower and upper limits, the better the explanation is. The more limits that are close, the closer the explanation is to being accurate.

- All metrics available for feature importance are also available for evaluating rules, where a transformation of the rule into a feature importance vector is performed first. The feature importance vector will have the same number of entries as the number of features in the domain. Each entry will be either 1 or 0, depending on whether the feature appears, or not, in the rule in question.

2.2.4. Word importance

For word importance, we have the same metrics as for feature importance, where the vocabulary is considered the feature space and a word importance explanation may or may not contain words from the vocabulary.

3. Experiments

To demonstrate \texttt{teex}’s usage, we present a benchmark comparison. A classification model (pre-trained SqueezeNet [15] from PyTorch [16]) has been fine-tuned on the Kahikatea dataset, obtaining 0.82 F1 score on the validation data and 0.65 F1 score on the test data when evaluating classification performance. From this model, we extract local explanations for the test set for 40 positively labeled test data when evaluating classification performance. From this model, the scores are not high, which indicates that the model has not entirely learned the particular features of the Kahikatea trees. Table 3.b reflects a characteristic of our evaluation procedure: the binarization threshold that is chosen for the evaluation directly influences the results. In this case, the cosine similarity scores indicate that explanations are almost identical to those of Integrated Gradients, but the other scores do not. This is due to how the threshold (0.5) interacts with the distributions of attributes, and needs to be taken into account by the user. This is particularly true if the explanations that are being compared seem to be very similar to each other.

For this simple experiment we have used our tool to quickly obtain relevant information about the model behavior, as well as compare the performance of various explainer methods on it. This could help iterate in the model-development process and allow for the selection of the right, and best-performing explainer for our particular use-case. Combining \texttt{teex} with other explanation evaluation libraries would bring us even more benefits.

4. Summary

\texttt{teex} is a Python library comprised of tools to help researchers and end-users evaluate the quality of local explanations against ground truth explanations for labels provided by human experts (or algorithmically in the case of synthetic data). It includes a comprehensive set of quality metrics that can be applied to different explanation types and also aims to serve as a hub for datasets with ground-truth local explanations, which are notoriously hard to find. \texttt{teex} has been conceived as an effort to help make XAI evaluation a more streamlined, reproducible, simple, and clear procedure, with ease-of-use and flexibility in mind, and can be used in tandem with other libraries for the generation and the evaluation of explanations.

5. Future improvements

Future improvements will be focusing on expanding access to more datasets with explanations, as well as more metrics. Beyond the immediate scope of the project, it may also be possible to extend the library to other predictive tasks such as regression or clustering.

CRediT authorship contribution statement

\textbf{Jesús M. Antoñanzas}: Conceptualization, Methodology, Software, Validation, Data curation, Writing – original draft, Writing – review & editing. \textbf{Yunzhe Jia}: Supervision, Methodology, Validation, Writing – original draft, Writing – review & editing. \textbf{Eibe Frank}: Supervision, Writing – original draft, Writing – review & editing. \textbf{Albert Bifet}: Supervision, Writing – original draft. \textbf{Bernhard Pfahringer}: Supervision, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
Data availability

All data used in this paper is openly available via teex or the original, referenced, sources.

Acknowledgments

teex has been developed as part of the TAIAO project (Time-Evolving Data Science/Artificial Intelligence for Advanced Open Environmental Science), funded by the New Zealand Ministry of Business, Innovation, and Employment (MBIE).

References