Implementation and Evaluation of Feature Attribution and Counterfactual Explanations

Deadline: 20th February 2024

Summary

You should submit a 3-page report and an implementation code (using the provided templates). For submission, zip your report and code into a single file called explanantion.zip. The main task of the assignment is to explore different properties of both feature attribution and counterfactual explanations in the context of machine learning models. The task is to compare different explanation strategies and compare them using metrics seen in the lectures.

Tools

You will be using several Python libraries, with Python versions from 3.9.0 to 3.10.12. To install all the dependencies, create a Python virtual environment, activate it and run `pip install -r requirements.txt` in the directory with the provided notebook and requirements.txt file.
Alternatively, you can install the required packages manually (e.g., if the provided requirements.txt file is incompatible with your environment):

- PyTorch (version 2.2.0, see the "install PyTorch" section at `https://pytorch.org/` to download the right version for your environment).
- Captum
- torcheval
- matplotlib (==3.8.2), scikit-learn (==1.4.0), seaborn (==0.13.2)
- jupyterlab, notebook, tqdm, tabulate

Tasks

1. Exploratory Data Analysis [10 marks]

(a) Based on the exploratory data analysis results provided in the notebook, answer the following questions:

(i) [**3 marks**] Considering the feature correlations and the distribution of the `sex` feature, are there any trends or patterns that you can identify in the data?

(ii) [**4 marks**] Without having access to any particular model or the associated explanations, which features would you expect to be the most and least important for a neural network trained on the dataset? How can you tell and how certain can you be of your assessment?

(iii) [**3 marks**] Apart from inspecting the above plots, is there anything else you could do as part of the exploratory analysis that would allow you to better understand the data and the behaviour of the models trained on it?

2. Feature Attribution Explanations [45 marks]

(a) [**15 marks**] Implement the SHAP method as introduced in the lectures. The code skeleton and more detailed instructions for this task can be found in the provided Jupyter notebook.

(i) Implement the `compute_coefficient` function computing the SHAP coefficient for a coalition.

(ii) Implement the `generate_coalitions` function generating the possible coalitions for a particular target feature.

(iii) Implement the `delete_features` function, which deletes the specified features for an input.

(iv) Implement the `shap_attribute` function computing the final SHAP attribution scores.

(b) [**10 marks**] Using your implementation of SHAP and the implementations of Shapley Value Sampling and DeepLIFT from the Captum library, compute feature attribution scores for 10 randomly sampled points from the Titanic test set and answer the following questions:

(i) Which features generally seem to be the most important and least important for the explained model according to each of the explanations?

(ii) Are there any substantial differences between the different attribution methods? What might be the possible reasons for the different methods returning different attribution scores?

(iii) Do the attribution scores match your expectations for the most/least important features from task 1(a)(ii)? What might be the reasons for a user's expected explanations differing from the computed attribution explanations?

(iv) Considering the insights gained from the exploratory data analysis and the feature attribution explanations, as well as the definitions of the explanations themselves, what are the potential advantages/disadvantages of each of these methods when trying to understand the behaviour of a model on a particular dataset?

(c) [**10 marks**] Perform a quantitative evaluation of the different attribution methods by computing their infidelity on the full Titanic dataset. Add a table summarising the results to your report and comment on the findings.

(d) **[10 marks]** Evaluate the computational efficiency of the different methods by taking the following steps:

(i) Preproccess the Dry Bean Dataset, similarly to what we have done for Titanic. You do not need to perform any exploratory data analysis for this dataset

(ii) Train an additional neural model on the preprocessed data. Briefly report the key performance metrics for the model in your report.

(iii) Compute the runtimes required to produce the attribution scores for the different methods when considering the first 200 samples in the Titanic and Dry Bean test sets. Report the results in a table in your report. Which methods seem to be the most/least computationally efficient?

3. Counterfactual Explanations [45 marks]

(a) **[12 marks]** Consider the same Titanic dataset and the corresponding PyTorch model used for the previous task. Design a suitable distance metric to use for measuring the proximity between points. Detailed instructions are in the Jupyter notebook.

(i) Briefly discuss the difference between standard L1 and normalised L1 and its effect on counterfactual explanations.

(ii) If we want to treat each feature equally in the original unprocessed dataset, how would you design the distance metric for the preprocessed dataset using L1-based distance? In the report, write down the details of your distance function for the preprocessed dataset and justify why each original feature is treated equally.

(iii) Implement your designed distance metric in the `distance_function` code block.

(b) **[5 marks]** Implement a Nearest-Neighbour Counterfactual Explainer (NNCE) as seen during the tutorial on Counterfactual Explanations. Complete the `compute_nnce` function.

(c) **[5 marks]** Implement the method by Wacther et al (WAC) using the code skeleton provided in the notebook. Complete the `CostLoss.forward` and `compute_wac` functions.

(d) **[10 marks]** Now use the implementations above to generate counterfactual explanations for 20 randomly selected test instances of the titanic dataset. Evaluate the results using the following metrics: proximity, validity and plausibility. Repeat the process 5 times, and report the mean and standard deviation of each evaluation metric for both methods.

(i) Follow the provided code skeleton in the Jupyter notebook and complete the experiments. Implement the `calculate_validity`, `calculate_proximity`, `calculate_plausibility` functions.

(ii) In the report, include the mean and standard deviation of each evaluation metric for both methods.

(e) [**13 marks**] Discuss in the report the differences between NNCE and WAC counterfactuals and their performances. Link any findings (even counterintuitive ones, if any) to their theories and/or implementations.

Details of the report

You are expected to write a report with up to 3-pages. Please use the provided latex or word template. Make sure to describe the key results, observations and conclusions in the main text. You may include supplementary plots in the appendix. References may go on an additional fourth page. You must also submit your source code. Please make sure it is possible to run your code as is.