Exercises List - Part 2: Explainability and Uncertainty

- Please submit the solution as a single file, using the naming convention 'your_last_name.py' OR 'your_last_name.ipynb', on the Moodle page of the course.
- Keep in mind to comment your code sufficiently and to include a demonstration.
- To avoid bugs due to conflicting libraries, it is recommended to set up an environment with the specific versions mentioned in the 'requirements.txt' files in moodle.
- You should not import any packages that violate the spirit of the exercise.
- It should suffice to execute your Python script once to verify that the exercise has been solved.

1 Permutation-based importance metrics

Consider the 'doubly-weighted' model developed in the Exercises List 1.

- Examine the permutation importance scores of the model with respect to the accuracy.
- Examine the permutation importance scores of the model with respect to the true positive rate.
- Examine the respective partial dependence functions (e.g: partial dependence and PartialDependenceDisplay from sklearn).
- Compare your different models with respect to the ones obtained in Exercises List 1.

Caltech-UCSD Birds-200-2011 Dataset

- Download the dataset from: `https://www.vision.caltech.edu/datasets/cub_200_2011/`. You will only need the images from this dataset. Additionally, you will need the 'annotation.csv' file from moodle.
- The dataset contains images of 200 different bird species taken from Flickr. The birds were further classified into landbirds and waterbirds.
- Alternatively, you can find all needed files at: `https://cloud.technikum-wien.at/s/5zZXEffzkEZ4TJS`

Model

- Two pretrained models are provided on the course page. One uses the ResNet18 architecture and needs less resources then the one based on the ConvNext-small architecture.
- Both models were trained to classify the bird images into the waterbird and landbird caterogies.
- Familiarize yourself with the model code and remember to use the preprocessing stepts provided in the transform field of the model classes.
- The preprocessing steps are resize to 256 pixel. Then center crop to 224 pixels, and then normalizing the image with mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$.

2 Model Specific: Integrated Gradients

- Use the implementation of the integrated gradients attribution method provided at moodle, under 'integrated-gradients.py' (use the requirements in the moodle page). Simple Riemann sums are applied to approximate the integral of the gradients. To get more accurate results (with fewer steps) you could also opt to use other quadrature methods like Gauss-Legendre quadrature.
- Use PyTorch's autograd capabilities to get the gradients. Important: You will need the gradient with respect to the input, not the model parameters.
- Visualize the integrated gradient attributions. Use a heatmap and appropriate color map.
- Only visualize the top $x\%$ of the integrated gradients
- Experiment with overlying the attributions onto an original misclassified image. Do not forget the scaling and cropping operations.

3 Model Agnostic: SHAP or LIME

- Choose between LIME (`https://github.com/marcotcr/lime`) or SHAP (`https://shap.readthedocs.io`), as a model agnostic approach.
- Select a correctly and incorrectly classified images of each class in the dataset.
- Compare the explanations provided by the model specific and agnostic methods. Summarize your findings about the methods.

Amazon Reviews Polarity Dataset

- Download the dataset in the file 'reviews.csv' from moodle.
- The dataset contains approximately 1 million user reviews from Amazon. The file contains 3 columns: polarity (1 for negative and 2 for positive), title (review heading), text (review body).

Model

- A code with a transformer-based model for sentiment analysis on the dataset is provided under 'transformer.py'.
- For a given review in text, the model classifies the sentiment expressed as positive or negative.
- The notebook allows you to sample different sizes of the dataset for training the model according to the power of the machine you are running the code. Use a minimum size of 2000 samples. Set the random seed in the code to be equal to the value of your student ID (eg. ma0844, $seed = 844$).

4 LIME

- Perform exploratory data analysis on the reviews dataset generated by your code.
- Select 2 reviews for each sentiment class and apply the LIME for text interpretation method.
- Generate a visualization of the results, displaying the most relevant parts of the LIME (`https://github.com/marcotcr/lime`) explanation.
- Choose a review for each category (positive and negative) and modify the sentences in order to change the model inference (and LIME interpretation). Eg: What words are necessary to be suppressed or modified for the model to provide a different sentiment output (from negative to positive, from positive to negative)?

5 Uncertainty and Robustness

- For your generated dataset, analyze the confidence of the model as a function of the length of the review. Discuss your results.
- Introduce noise to your data. Some ideas to generate noise are: replacing words with slang language, changing the polarity of words (eg: replace 'good' for 'bad', 'lot' for 'little', etc), suppressing negation, replacing words by synonyms.
- Compare the confidence of your results for the noisy dataset to the original. Highlight your interpretation for the model over/under-confidence.