Aatif Ahmad (B22AI002), Saumitra Agrawal (B22AI054)

Indian Institute of Technology, Jodhpur

CSL7370	Dependable AI
Assignment 01	3rd March, 2025

Explainability Assignment : Integrated Gradient

Table of Contents

References:

1. https://github.com/ankurtaly/Integrated-Gradients
2. https://medium.com/@kemalpiro/xai-methods-integrated-gradients-6ee1fe4120d8

A. Setup (These details are also present in the Readme.txt- file attached)

a. Kindly follow these steps to run, predict and explain the Integrated-Gradient method with AnnexML on IAPRTC dataset.

Assumptions: The Integrated-Gradient method is placed in a separate directory called “Integrated-Grad”. Following is a complete step-by-step guideline for the method and it assumes a directory called datasets with the following files : “iaprtc-train.svm” and “iaprtc-test.svm”.

To run these commands, make sure you are in the current working directory of Integrated Gradient Implementation.

i. Build: make -C src_integrated/annexml

ii. Training: src_integrated/annexml train annexml-config-ig.json

This saves the model with the file name “annexml-model-ig.bin” inside the datasets directory.

iii. Testing: src_integrated/annexml predict annexml-config-ig.json

This saves the results inside the datasets directory with the file name “annexml-result-ig.txt”.

iv. Evaluation: cat datasets/annexml-result-ig.txt | python scripts/learning-evaluate_predictions.py

v. Explanation: src_integrated/annexml explain annexml-config-ig.json

This saves the attribution scores in the file “ig_attributions.txt” in the current directory.

vi. Visualization of Attributions: python plot_attributions.py

This saves the plots inside the plots directory.

B. Implementation

a. The core implementation lies in the files “IntegratedGradients.cc” and “IntegratedGradients.h” along with a function for computing label scores inside the “AnnexML.cc” file.
b. These are the major implementation steps:

i. First, the size of the output vector which is going to store the feature attributions is determined by extracting the biggest feature ID.

ii. The baseline is a zero baseline which is a sparse vector of all zeros (same size as the input). {0.0, 0.0, 0.0, …… 0.0}

iii. The model is run to get the scores for the input and baseline.

iv. Gap between the input and baseline scores is determined.

v. For approximating gradients, a loop is run 50 times setting alpha value from 0.02 to 1.0.

$$IntegratedGrads_i^{approx}(x) ::= (x_i - x_i') \times \sum_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m}$$

Here, m defines the number of interpolation steps basically at what step of alpha value we are currently at (there will be a total of 50 steps).

vi. At each alpha value, baseline and input are mixed by interpolation. The model calculates the score change from the previous step. This change in score is multiplied by how much the feature has changed and added to the running sum.

vii. The final sum is appended to a list of attributions. Zero attribution scores are neglected.

viii. The attributions for all features are finally normalized to bring them in unit scale and the list with all normalized attributions is finally returned.

C. Analysis and Results

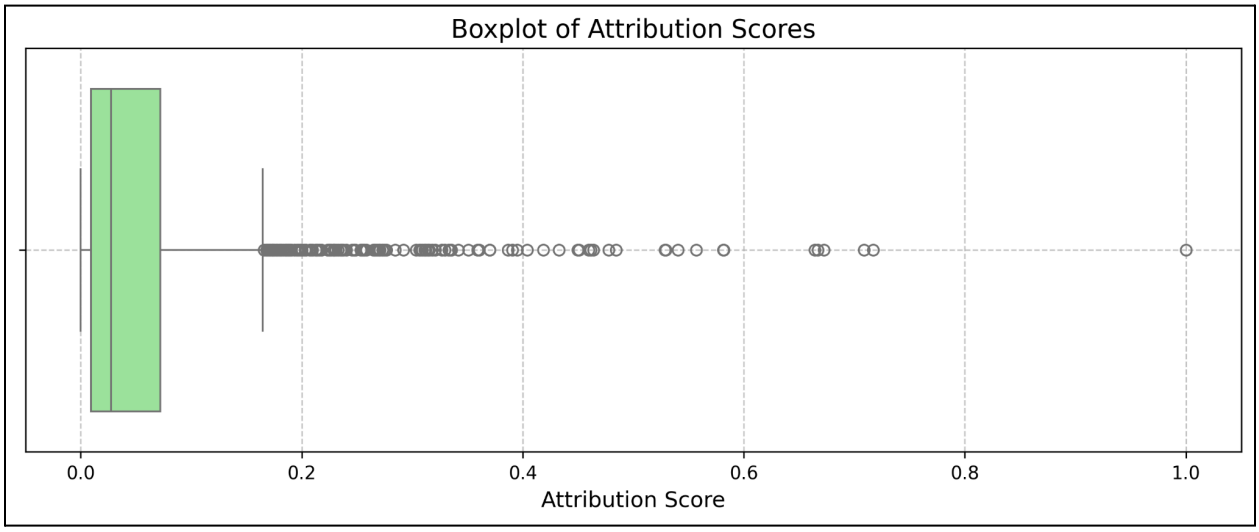
a. The following is a list of some of the attribution scores along with the feature indices for the prediction of the first sample.

528 0.01896
529 0.00943597
530 0.0672748
531 0.012438
532 0.103945
533 0.0031678
535 0.00117498
536 0.0103395
537 0.00250542

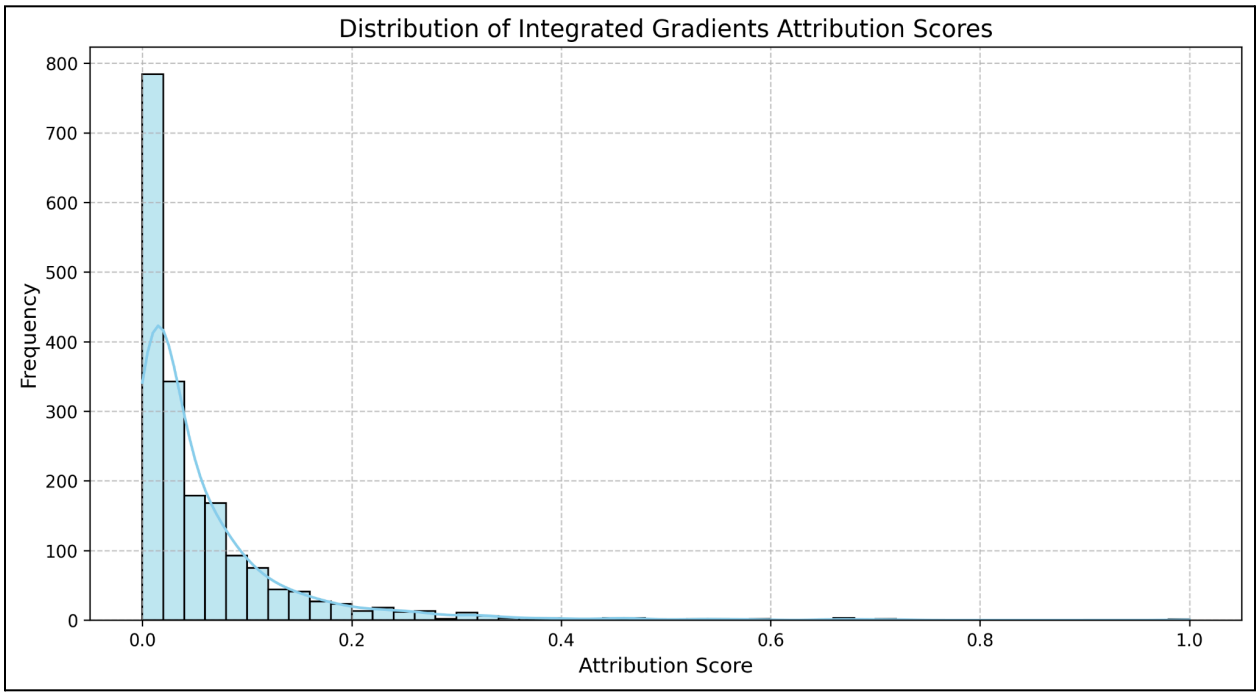
This shows that some features have a higher importance than others in predicting the output. The scores are normalized as per the implementation described in [B. Implementation].

b. All attribution scores after normalization are coming positive which suggests that the features have a positive contribution to prediction.

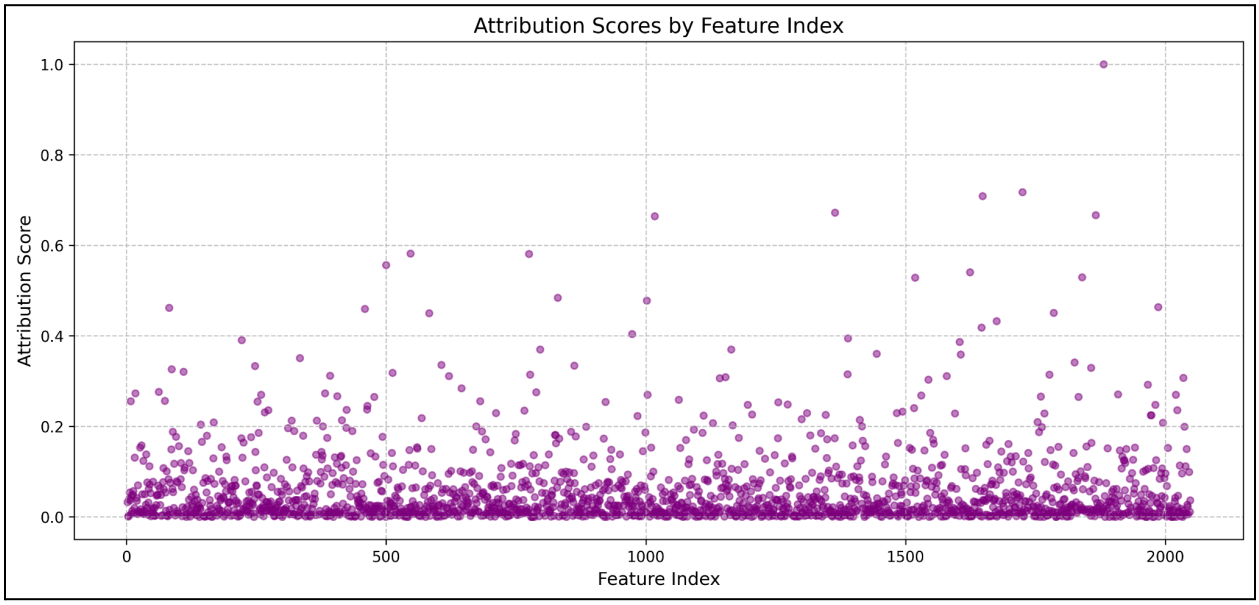
c. Lets analyze some visualizations to understand the results better:



This suggests that the majority of features have negligible impact on prediction and only those features whose attribution scores have been represented as outliers in the image above i.e. greater than 0.2 drive the result.

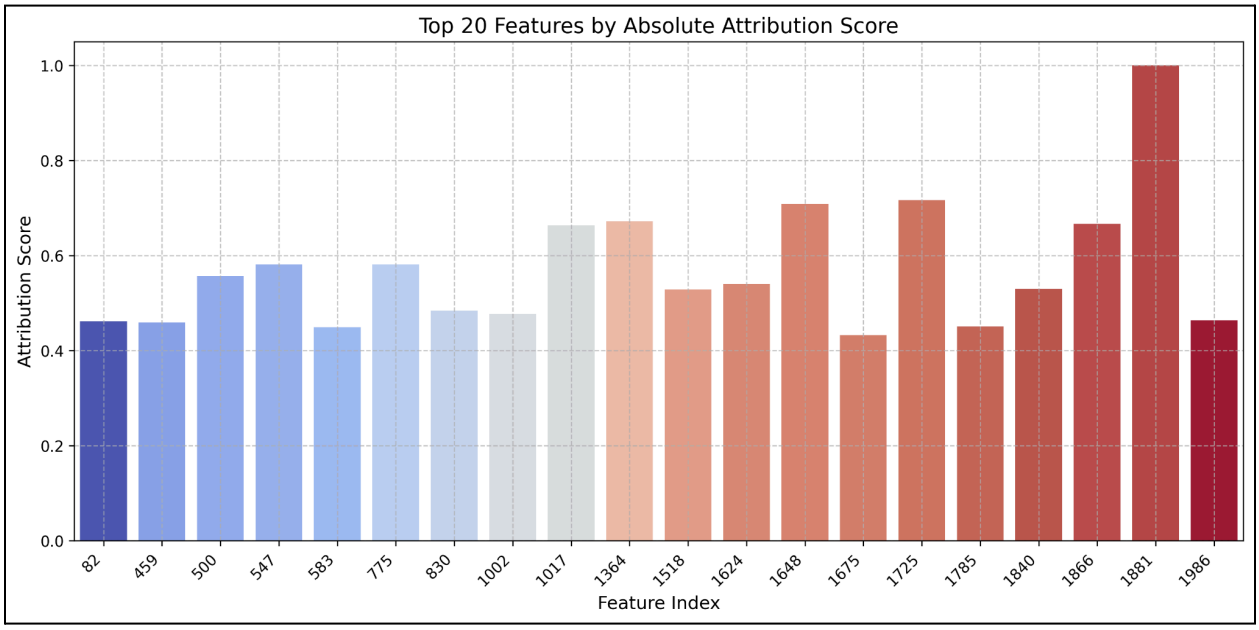


The previous observation is well supported with the above visualization. Several hundred features have attribution scores under 0.2 and few of them which have above 0.2 have significant impact on driving the result of prediction.



This is a complete distribution of attribution scores for each feature index. It is interesting to note the dense cluster between the 0.0 - 0.2 belt specifically

between 0.0 and 0.1. Moreover, there is a greater sparsity above 0,2 specifically above 0.4. It is also worth noticing that there is a certain feature index in the 1800-1900 feature index range whose attribution score has touched the top line (1.0).



The above visualization confirms the presence of a feature with index 1881 whose attribution score has touched the top-score of 1.0. This suggests that this feature has the most significant impact in determining the result of prediction.