Submission for **Code with SAP Labs India - Discover, Design, Deliver LIVE**

Team : ACES

Team Members

1. Yashwanth Venkat (Leader)
2. Niharika

Requirements for Submission :

1. **Document(Presentation/Text) on the solution that can contain:**
   a) **Intuition behind the solution**

   The project has been designed with the idea to provide basic interpretation to data to even a newbie who has little to no knowledge about machine learning. Shapley values help in simply interpretting the results by assigning values that tell us how much each variable contributed towards the final results.

   The project has been created as GUI dashboard with multiple ML algorithms to test how each algorithms provide different explanations for local and global level for the dataset. Our main reference was “Interpretable Machine Learning” book by “Christoph Molnar”.

   b) Logic and Working

   For Logic refer to section 5.10 of “Interpretable Machine Learning” book by “Christoph Molnar”.

   The process to run the codes has been mentioned in Readme.md file of the project.
   Github Link : https://github.com/d2Anubis/Code-with-SAP-Labs-India

   c) Different Components

   The submission consists of following components:
   a) Jupyter files to generate datasets, trained models and shapley values
   b) Streamlit python files to run and create a GUI dashboard to test the outputs

   d) Architecture

   Refer Readme.md file for architecture of project.

   e) Documentation on each class and its need

   Not Applicable

   f) Open source algorithms used in the submission

We have use SHAP (SHapley Additive exPlanations) from
https://github.com/slundberg/shap/

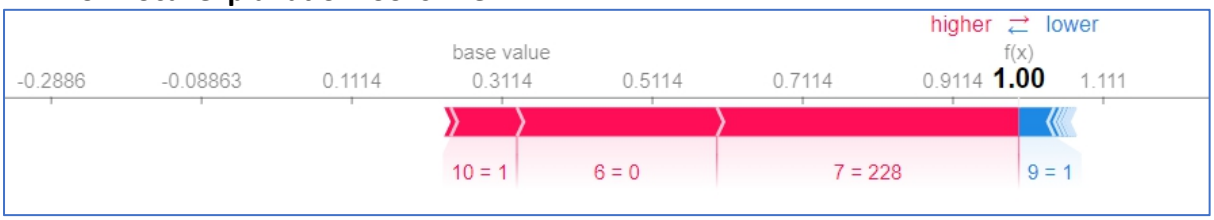
g) How to : detailed explanation on how the solution should be run for

i. Different datasets : Credit (Loan) dataset, MNIST dataset

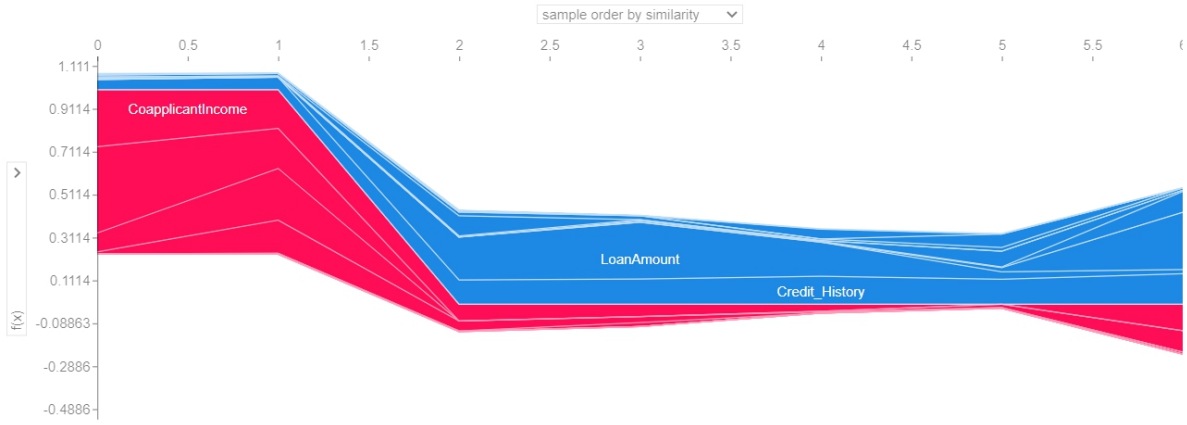
ii. **Different algorithms** : Shapley, KNN, SVM, Logistic Regression, Decision Tree, Random Forest, Neural Network

h) Results section should contain:

i. How local explanation looks like



ii. How global explanation looks like



iii. More information relevant and unique advantages (eg. Visualization)

Refer diagrams above. We have also added utilities to test ones own test data through input within the dashboard itself.

2. The code for framework :

a) You have to develop the application on your local system and submit it on HackerEarth in tar/zip file format along with instructions to run the application and source code.

Functions expected in template submission

Let's call the XAI class instance 'Xai'

Naming:

- * on a particular argument indicates that it is an optional argument
- .* in argument list indicates that the students can use more arguments if they want. Provided they clearly state the need for it

Mandatory functions:

To generate local predictions, outputs the explanation
Xai.explain_local([row(s) for explanation], classification_model*, metadata*, .*)

- Classification_model : the instance of classification algorithm, that has been trained
- ([row(s) for explanation] : is an array of rows for which explanation needs to be given in the same format on which the models were trained
- Metadata : any other metadata required for the functioning such as algorithm type,datatypes etc

To generate global predictions, output the explanation
Xai.explain_global(classification_model, training data*, metadata*)

- Classification_model : trained or untrained classification_model
- Training_data: training data, if needed
- Metadata : any other metadata required for the functioning such as algorithm type,datatypes etc

Optional Functions

If the framework needs to be trained before starting to make the prediction
Xai.fit(classification_model, training data,metadata*)

- Classification_model: trained or untrained classification model (pleace specify which one)
- training_data : training data
- Metadata : any other metadata required for the functioning such as algorithm type,datatypes etc