Counterfactual Local Explanations via Regression (CLEAR)

CLEAR explains single predictions of machine learning classifiers. It is based on the view that a satisfactory explanation of a single prediction needs to both explain the value of that prediction and answer ’what-if-things-had-been-different’ questions. In doing this it needs to state the relative importance of the input features and show how they interact. A satisfactory explanation must also be measurable and state how well it can explain a model. It *must know when it does not know*

Prerequisites

CLEAR is written in Python 3.7 and Tensorflow 1.13. It runs on Windows 10. CLEAR requires the following Python libraries to be installed: tkinter, numpy, pandas, sympy, datetime, matplotlib.pyplot, scipy.signal, csv, json, jinja2 and math.

Installation

Download a copy of the CLEAR repository into a new directory on your PC. The file CLIME_settings.py contains the parameter variables for CLEAR. Open CLIME_settings and change the value of parameter *CLEAR_path* to the name of the directory you have created for CLEAR e.g. CLEAR_path='D:/CLEAR/'

Running CLEAR

First, CLEAR's parameters for the experiment should be set. These are all in CLIME_settings.py. The admissible values for each parameter are shown in the comment to the right of the parameter eg for *case_study* the admissible values are 'Census', 'PIMA', 'Credit Card', 'BreastC' and ‘IRIS’. The pdf file 'Input parameters for CLEAR' documents the input parameters.

CLEAR is then run by running CLEAR.py

CLEAR will generate a report explaining a single prediction if the parameters 'first_obs' and 'last_obs' are set to the same value e.g first_obs=7, last_obs=7 will generate a report explaining observation 7 in the test dataset. The report is entitled 'CLEAR_prediction_report.html'

There are two detailed csv files created for each run. The first file's name consist of the characters 'CLRreg_' and the date/time it was created eg 'CLRreg_20190522-1618.csv' This contains details of the regression for each observation e.g. adjusted R-squared score, coefficient weights and so forth. The second file's name consists of the characters 'wPerturb_' and the date time. This contains details of each w-perturbation for each observation. An error histogram is also created for each run, the name consisting of characters 'Hist_' and the date/time.