

## **Given: source x, source y (yield GT), target x (we assume target y is not provided), domain label (source vs. target)**

Predict: Domain source; features are ranked using domain source as signal (this could be changed to rank features based on domain source + GT prediction)

Key points:

(\*) Model is trained to concurrently predict: 1. domain (source vs. target) + 2. yield GT (only for source data)

(\*) decorrelation loss (lines 521-526) is introduced to help decorrelate features during training

(\*) We backprop on domain loss back to input features, and then average these gradients over holdout dataset to generate domain-shift feature importance

High-level Workflow: (1) Data load / pre-processing -> (2) run lightgbm to identify top-k features (e.g., k=300) -> (3) Using these top-k features, train NN to concurrently predict domain + yield GT, rank input features for data-shift importance

## **Lines 284-416:**

(\*) data loading,

(\*) pre-processing: using simpleimputer for missing values (line 309), filled with -100; normalizing by max (line 312)

(\*) results: (1) POR\_X (X source), (2) POR\_Y (Y source), (3) POR\_Y\_domain (source vs. target) ->

**POR\_train\_X, POR\_test\_X, POR\_train\_Y** (2-D array, first entry is yield prediction, second is domain; 0=source, 1=target)

HDR\_X (X target), (2) HDR\_Y\_domain ->

**HDR\_train\_X (X target), HDR\_test\_X (X target), HDR\_train\_Y, HDR\_test\_Y**

## **Lines 423-452:**

(\*) Run lightgbm to identify top-k features wrt source yield GT (this could be changed to a different Y label, such as domain label)

## **Lines 490-535:**

(\*) train simple NN with decorrelation regularization to predict domain and GT (for source)

(\*) domain\_y\_hp (line 518) is hp to balance domain prediction vs GT prediction in loss function

## **Lines 537-605:**

(\*) Model evaluation

### **Lines 613-623:**

(\*) Gradient-based feature ranking, backpropagate wrt to domain prediction (line 621: could be amended to propagate wrt domain + yield prediction)

(\*) line 622: choice to accumulate gradients using absolute value (this is standard, but could remove absolute value to accumulate wrt raw gradients)

### **Lines 631-669:**

(\*) Write top features to txt file; generate pdf for visualization of top-ranked features