MIV 알고리즘 구현 상황 보고

2020.04.10

임승우

2020.04.10 현재 진행 상황

MIV 알고리즘 중 Regression 모델에 적용하기 위한
Python Class는 구현 및 검증 완료

Classifier 모델에 적용하기 위한 알고리즘은 현재 구상 중에 있음.

1. n개의 feature를 갖는 데이터 X를 uniform하게 생성

2. n개의 weight를 randint로 생성

3. n개의 weigh와 X를 dot product하여 Linear Regression Target 값 “Y” 생성

4. MinMaxScaler를 이용하여 데이터 X 스케일 변환

5. Regression Model을 선정하여 회귀모델 훈련.

(Linear Regression, SVR, KNeighbor Regression)

6. MIV Class로 Feature Selection 수행 및 Explained Variance Ratio 계산

7. Explained Variance Ratio와 weight를 Descending argsort하여 인덱스가 서로 같은 지 비교

Params:

1. Model : pre-trained Regression Model
2. threshold : cutting point to discard features
3. zeta : amount of change of X
4. score_ : Not Implemented
5. is_clf : flags regression or classification
6. is_fitted : flag represents MIV model is fitted for ***fit_transform*** method

Methods:

1. fit : from dataset X and target Y, calculate Impact value and MIV, then get explained variance ratio and selected feature.

   input : dataset X and target Y used in training original model.

   output : MIV class itself

2. transform : Returns shrunk dataset X′ using selected features from ***fit*** method.

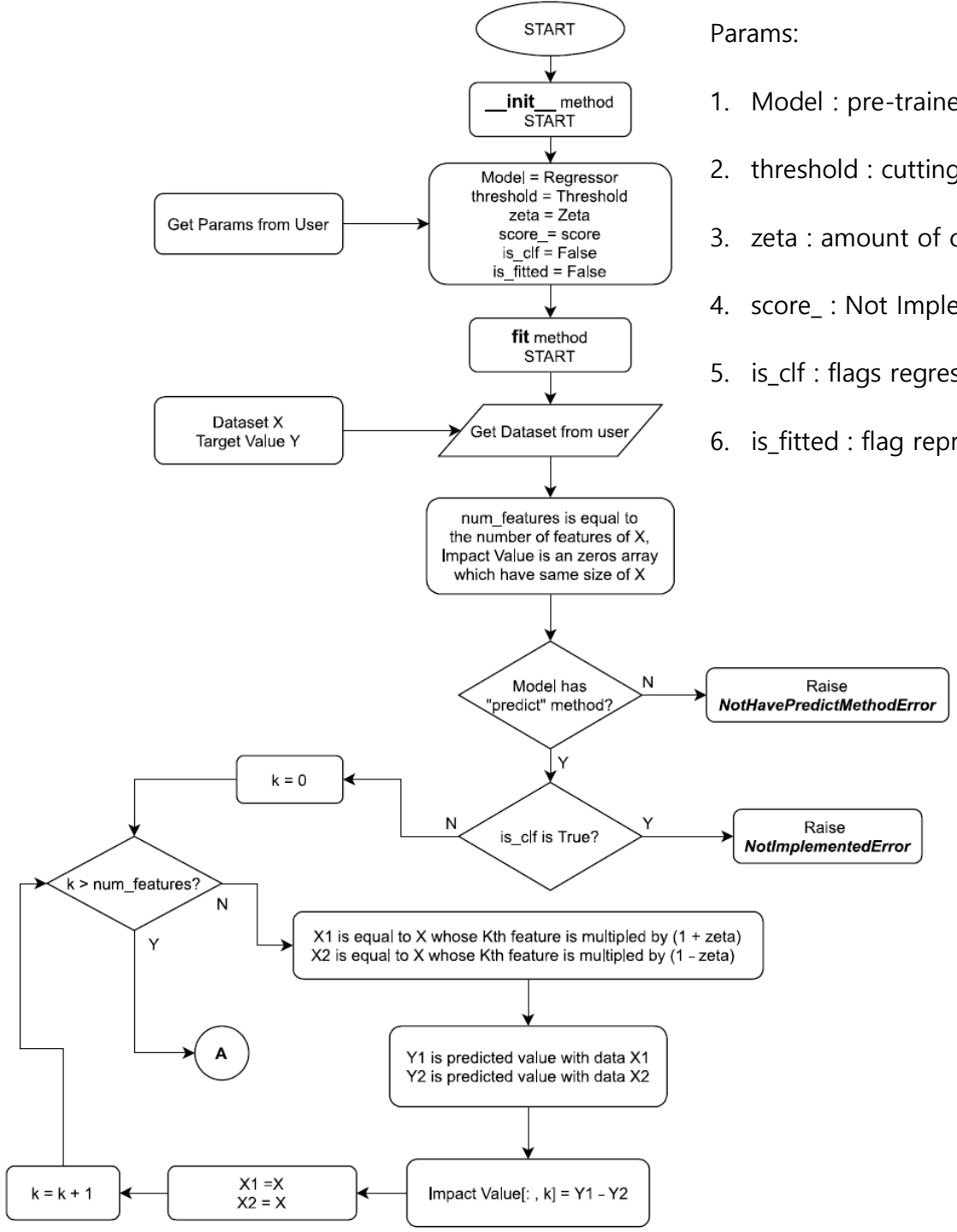
   input : dataset X and target Y used in training original model.

   output : shrunk dataset X′

3. fit_transform : fit the MIV model, and then transform dataset X.

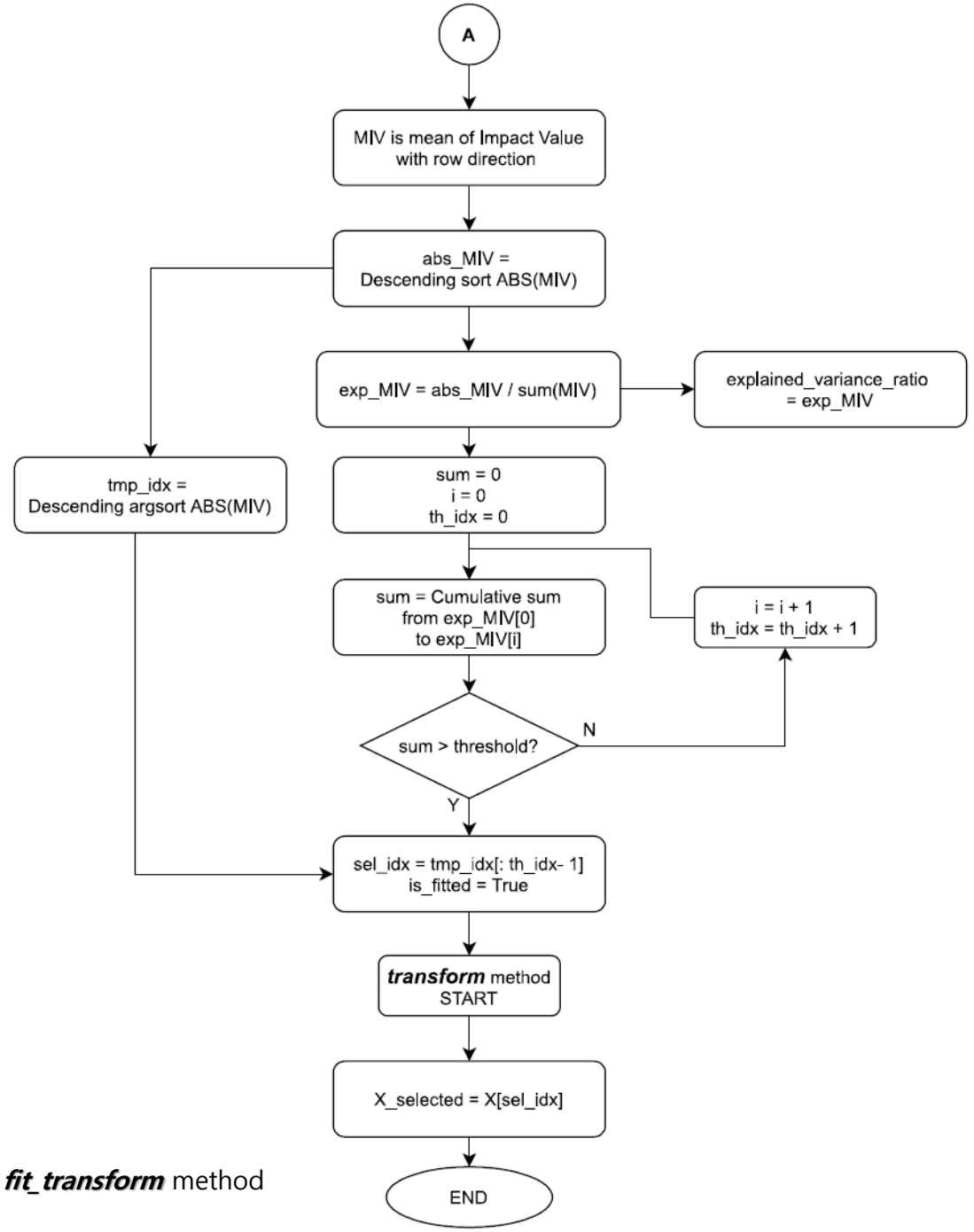
   input : dataset X and target Y used in training original model.

   output : shrunk dataset X′



Params:

1. Model : pre-trained Regression Model
2. threshold : cutting point to discard features
3. zeta : amount of change of X
4. score_ : Not Implemented
5. is_clf : flags regression or classification
6. is_fitted : flag represents MIV model is fitted for ***fit_transform*** method



Params:

1. Model : pre-trained Regression Model
2. threshold : cutting point to discard features
3. zeta : amount of change of X
4. score_ : Not Implemented
5. is_clf : flags regression or classification
6. is_fitted : flag represents MIV model is fitted for ***fit_transform*** method

1. Test Result From Linear Regression (num_features = 14)

<- Selected

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
EVR_idx	1	12	9	11	13	10	4	0	6	8	3	5	2	7
EVR_val	0.1390	0.1222	0.1054	0.1008	0.0862	0.0845	0.0835	0.0639	0.0581	0.0402	0.0393	0.0334	0.0241	0.0194
W_idx	1	12	9	11	13	10	4	0	6	8	3	5	2	7
W_val	288	251	217	208	178	173	173	132	120	83	81	69	50	40

의도대로 Weight가 큰 Feature가 중요도가 높아
Explained Variance Ratio가 크게 계산되었으며,
EVR과 Weight의 내림차순 정렬 인덱스가 일치하므로
Feature Selection이 제대로 이루어진 것을 확인하였다.

2. Test Result From SVR (num_features = 14)

<- Selected

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
EVR_idx	1	12	9	11	13	10	4	0	6	8	3	5	2	7
EVR_val	0.1392	0.1225	0.1054	0.1008	0.0862	0.0844	0.0833	0.0641	0.0585	0.0402	0.0391	0.0335	0.0237	0.0190
W_idx	1	12	9	11	13	10	4	0	6	8	3	5	2	7
W_val	288	251	217	208	178	173	173	132	120	83	81	69	50	40

이전과 마찬가지로, Weight가 큰 Feature가 중요도가 높아
Explained Variance Ratio가 크게 계산되었으며,
EVR과 Weight의 내림차순 정렬 인덱스가 일치하므로
Feature Selection이 제대로 이루어진 것을 확인하였다.

3. Test Result From KNeighborRegression (num_features = 14)

<- Selected

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
EVR_idx	1	12	11	9	4	10	13	0	6	8	3	5	7	2
EVR_val	0.1394	0.1207	0.1054	0.1001	0.0868	0.0837	0.0818	0.0712	0.0551	0.0385	0.0382	0.0363	0.0229	0.0198
W_idx	1	12	9	11	13	10	4	0	6	8	3	5	2	7
W_val	288	251	217	208	178	173	173	132	120	83	81	69	50	40

Linear Reg, SVR과는 달리 Weight와 내림차순 정렬 순서가 뒤바뀐 인덱스가 존재한다.
다만 큰 오차가 아니고 Weight가 크게 차이 나지 않은 2개의 Feature가 서로 순서가 뒤바뀐 정도이며,
Selection 된 Feature는 여전히 앞의 결과와 동일하다.

MIV ISSUE

MIV 이전의 모델을 훈련시키기 전에 데이터셋은 반드시 MinMaxScaler를 통해 스케일 되어야 한다.

Linear Model에서 Impact Value는

2 * zeta * weight[k] * X[:, k]로 계산되는데, X의 원소에 음수 값이 있으면

IV의 평균을 취해 MIV를 계산할 때 값이 상쇄되며 MIV값이 의도대로 나오지 않는 문제가 발생한다.

from

$$Y1 = a * (1 + zeta) * X1 + b * X2$$

$$Y2 = a * (1 - zeta) * X1 + b * X2$$

$$IV = Y1 - Y2 = a * 2 * \text{zeta} * \text{X1}$$

MIV는 IV의 평균 값인데, 평균을 계산하기 위한 값에 서로 다른 부호가 있으면

값이 서로 상쇄되어 MIV가 제대로 계산되지 않는다.

따라서 X의 모든 원소는 모두 같은 부호를 가져야 하며,

MinMax Scaler로 전처리하여 모든 원소가 [0, 1] 값을 가지도록 조절해야 한다.