ONLINE NEWS POPULARITY

Presentation by 丁柏雄、顏嘉佑

Machine Learning and Econometrics | 2023

OUTLINE

MODEL INTRO

Causal Forest

REPLICATE ESSAY

Random forest
Navie bayes
KNN
SVM
Adaboost

OUR ANALYSIS

1.Rank Factor importance
2.Causal Forest
using Econml
3.Double machine learning
using doubleml

CAUSAL FOREST

- Causal forest → To maximum treatment effect

$$\tau(x) = \mathbb{E}\left[Y_i^{(1)} - Y_i^{(0)} \mid X_i = x\right]$$

$$\hat{\tau}(x) = \frac{1}{|\{i : W_i = 1,\ X_i \in L\}|} \sum_{\{i:W_i=1,\ X_i \in L\}} Y_i - \frac{1}{|\{i : W_i = 0,\ X_i \in L\}|} \sum_{\{i:W_i=0,\ X_i \in L\}} Y_i.$$

Treatmet

Leaves

CAUSAL FOREST

- $\hat{\tau}(x) = B^{-1} \sum_{b=1}^{B} \hat{\tau}_b(x)$

  $\hat{\tau}_b^*(x)$ is the treatment effect estimate given by the bth tree

- $$\widehat{V}_{IJ}(x) = \frac{n-1}{n} \left(\frac{n}{n-s} \right)^2 \sum_{i=1}^{n} \mathrm{Cov}_*[\hat{\tau}_b^*(x),\ N_{ib}^*]^2$$

  $\frac{n-1}{n} \left(\frac{n}{n-s} \right)^2$ finite-sample correction ;

  N_{ib}^* indicate whether or not the ith training example was used for the bth tree

- $\widehat{V}_{IJ}(x) / \mathrm{Var}[\hat{\tau}(x)] \rightarrow_p 1$
- $(\hat{\tau}(x) - \tau(x)) / \sqrt{\mathrm{Var}[\hat{\tau}(x)]} \Rightarrow \mathcal{N}(0,\ 1)$

EXAMPLE

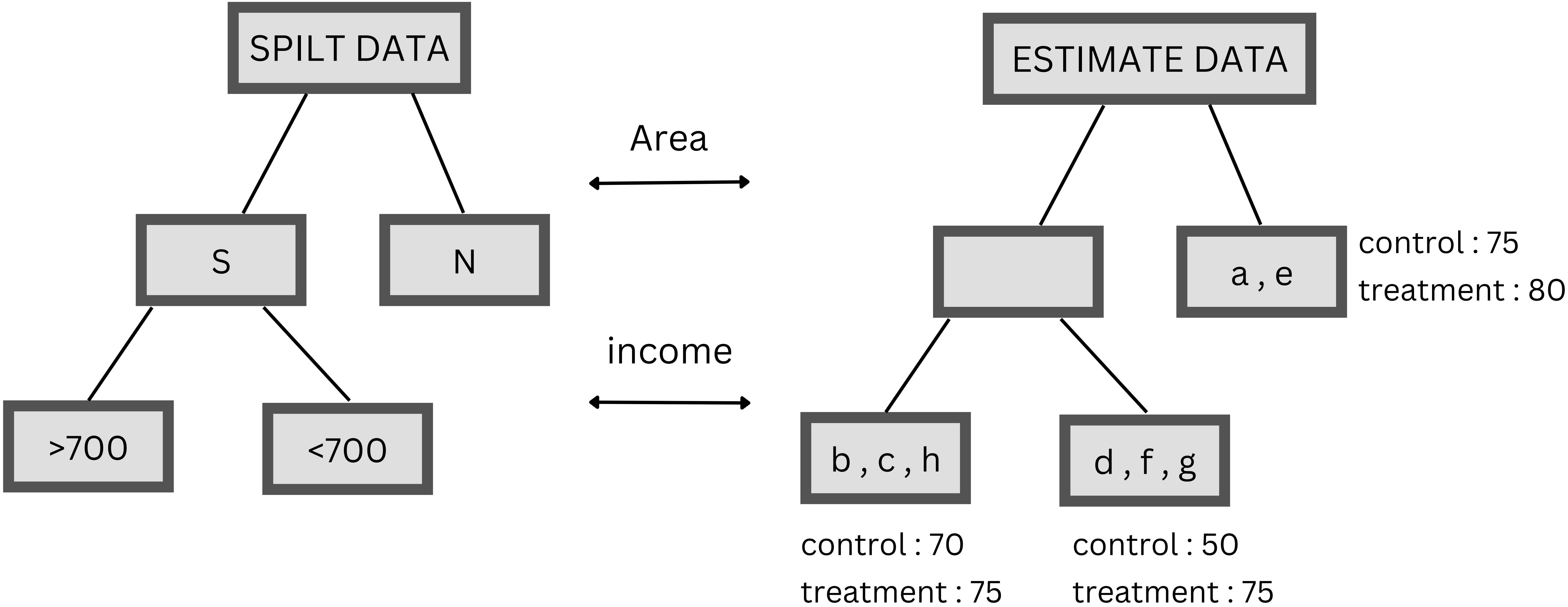
GOAL : How does canvassing behavior affect voters' evaluation of candidates

-To check the treatment on canvassing

Training data

SPILT	ESTIMATE

(Area , Income , Rating Score , canvassing)
a(N , 600 , 80 , 1) 、 b(S , 800 , 70 , 0) 、c(S , 800 , 80 , 1) 、d(S , 600 , 90, 1)
e(N , 800 , 75 , 0) 、 f(S , 600 , 50 , 0) 、 g(S , 600 , 60 , 1) 、h(S , 800 , 70, 1)



ESSAY REPLACTAE

Model	Accuracy	Precision	Recall	F1	AUC
Random Forest (RF)	**0.67**	0.67	**0.71**	**0.69**	**0.73**
Adaptive Boosting (AdaBoost)	0.66	0.68	0.67	0.67	0.72
Support Vector Machine (SVM)	0.66	0.67	0.68	0.68	0.71
K-Nearest Neighbors (KNN)	0.62	0.66	0.55	0.60	0.67
Naïve Bayes (NB)	0.62	**0.68**	0.49	0.57	0.65

In orginal essay, they use five ML models to estimate the puplarity of News
Their result showed that random forest beat the other four model.
We try to use the same five modes to replicate their result.

OUR RESULT

Model	Accuracy	Precision	Recall	F1	AUC
Random Forest (RF)	0.659	0.667	0.715	0.69	0.72
Adaptive Boosting (AdaBoost)	0.659	0.667	0.715	0.69	0.72
Support Vector Machine (SVM)	0.652	0.667	0.715	0.69	0.71
K-Nearest Neighbors (KNN)	0.61	0.667	0.715	0.69	0.64
Naïve Bayes (NB)	0.610	0.667	0.715	0.69	0.65

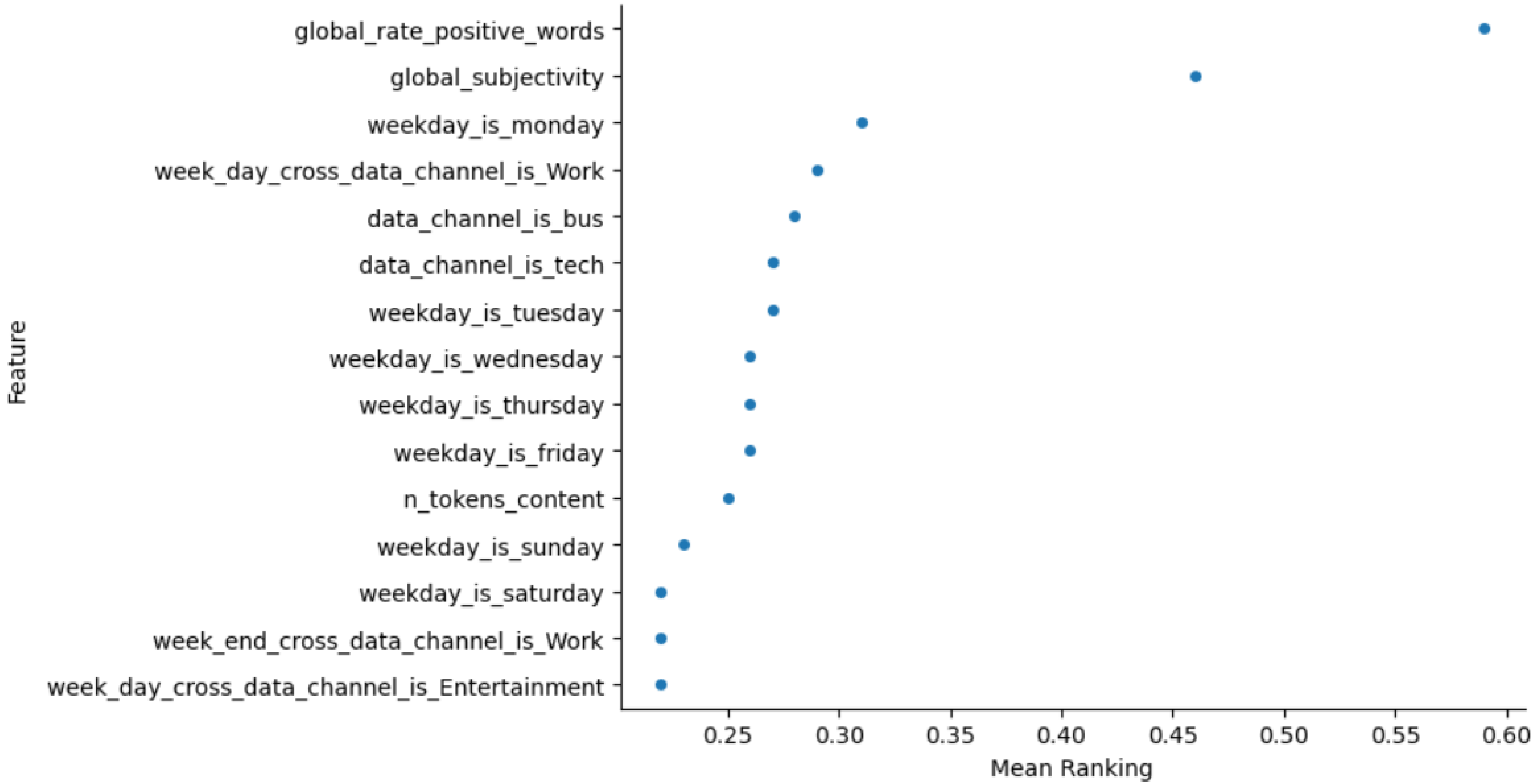
RANK FACTORS

CRITERIA

The average of coefficients produced by

1. Linear
2. Lasso
3. Ridge
4. Random Forest Feature Ranking

RANK FACTORS



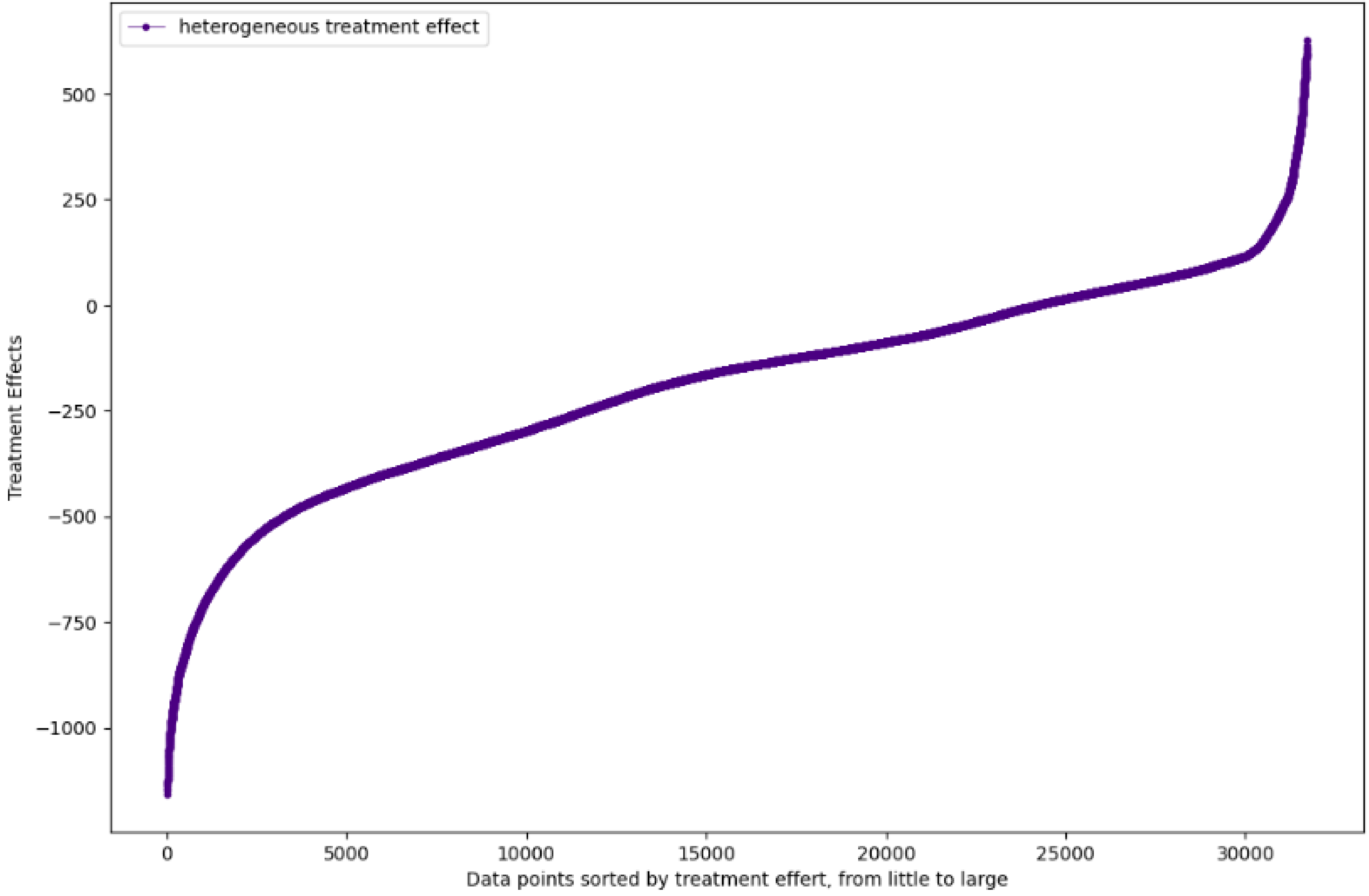
CAUSAL FOREST

- Package: econml.dml.CausalForestDML in EconML
- Developed by Microsoft, using the idea of Double machine learning
- Able to choose the estimator for 1.fitting the response to the featrues 2.fitting the treatment to the features
- After fitting the estimator, the package use causal forest to estimate heterogeneous treatment effect

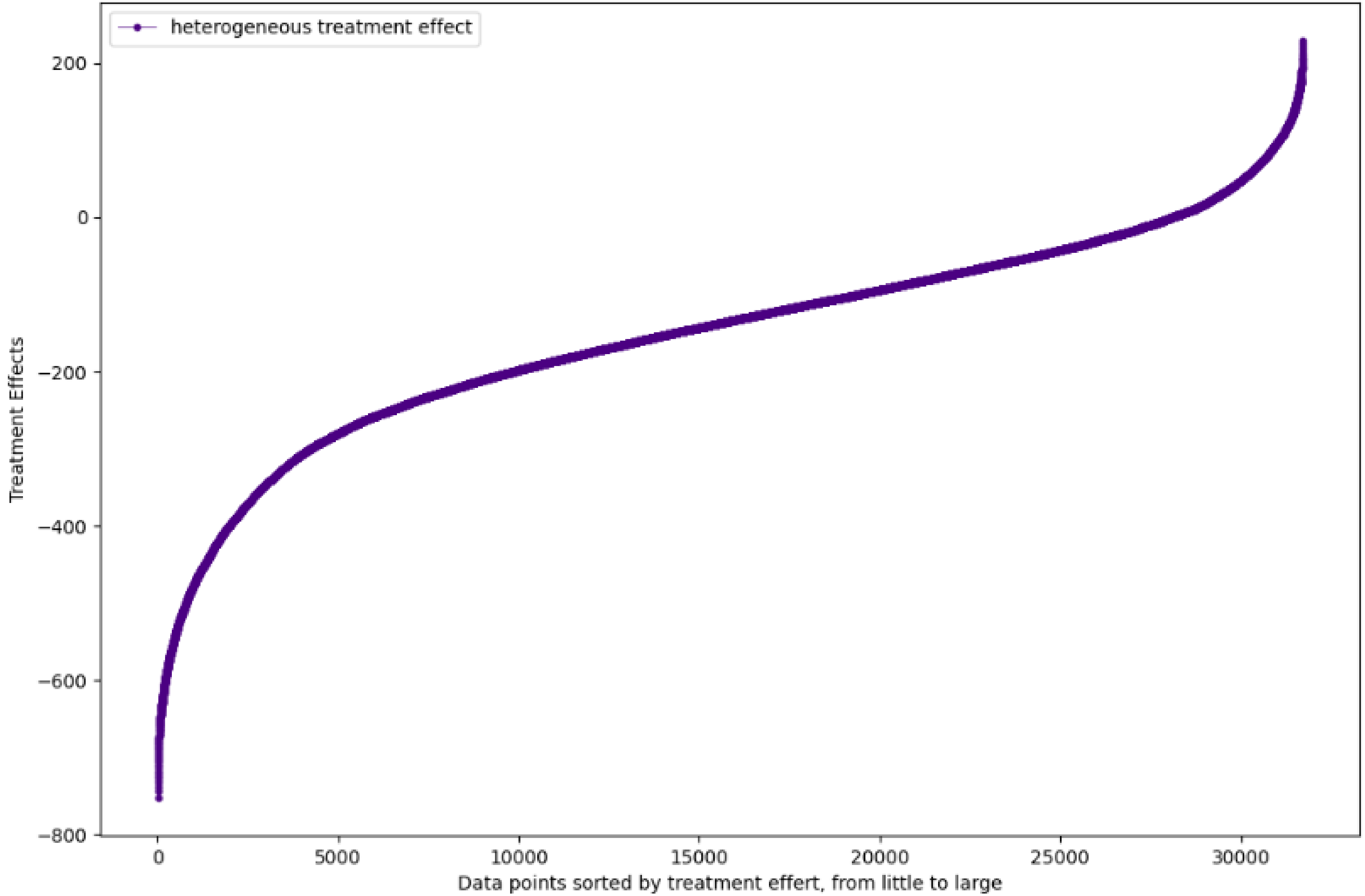
MODEL SETTING

n_estimators=10000,
max_depth=10,
model_t=DecisionTreeRegressor(),
model_y=DecisionTreeRegressor(),

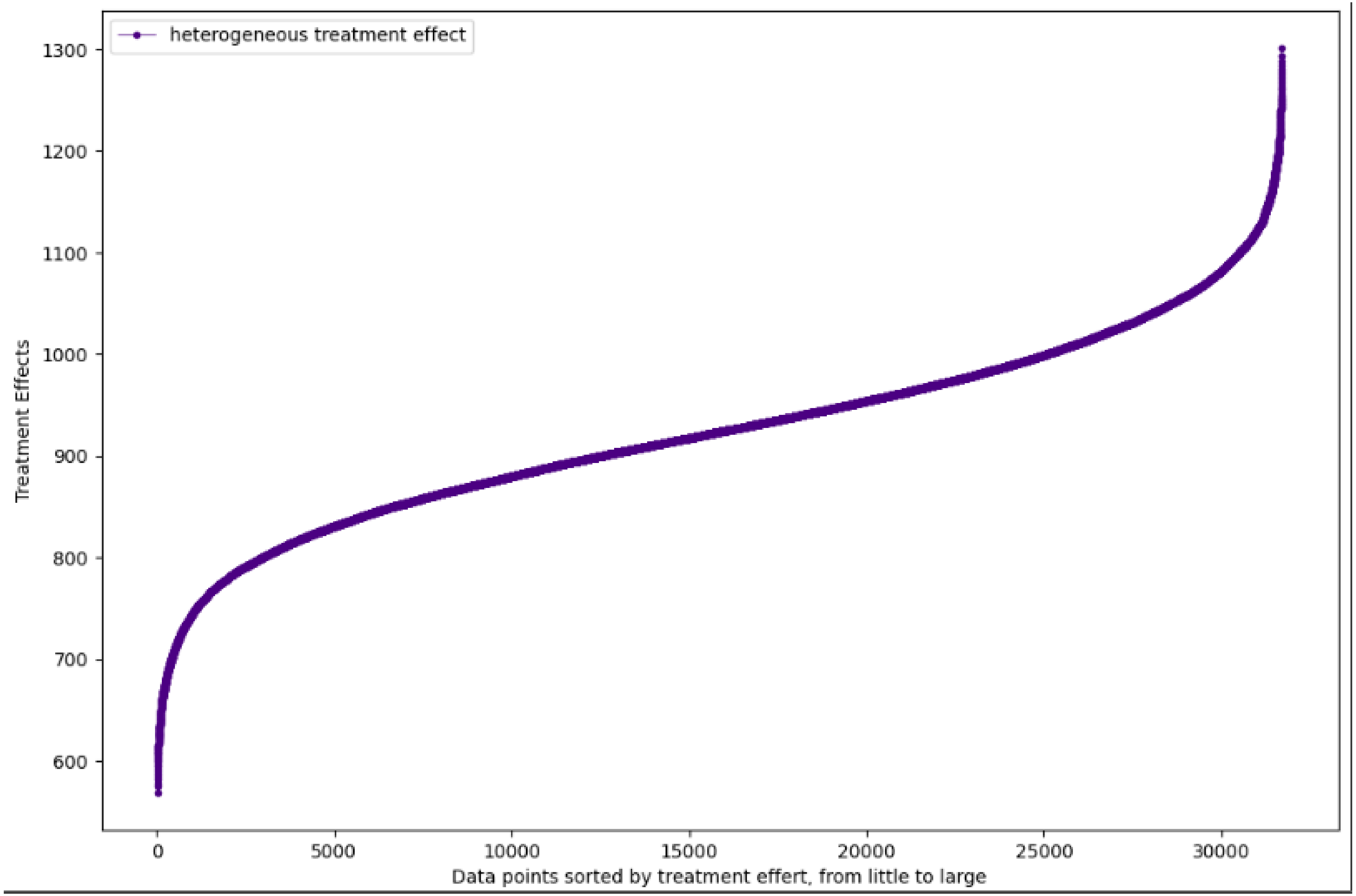
Treatment: number of words in title



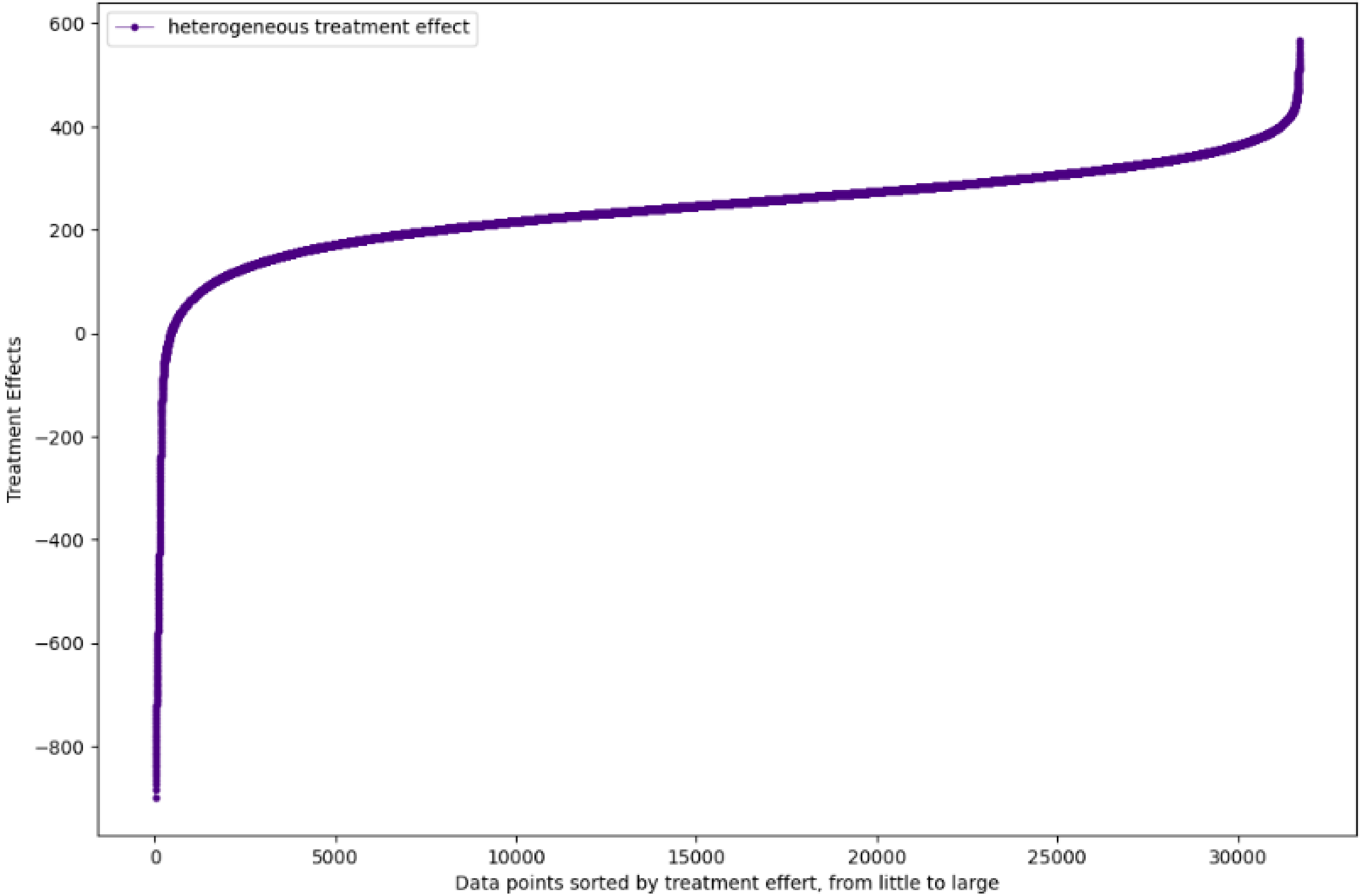
Treatment: number of words in content



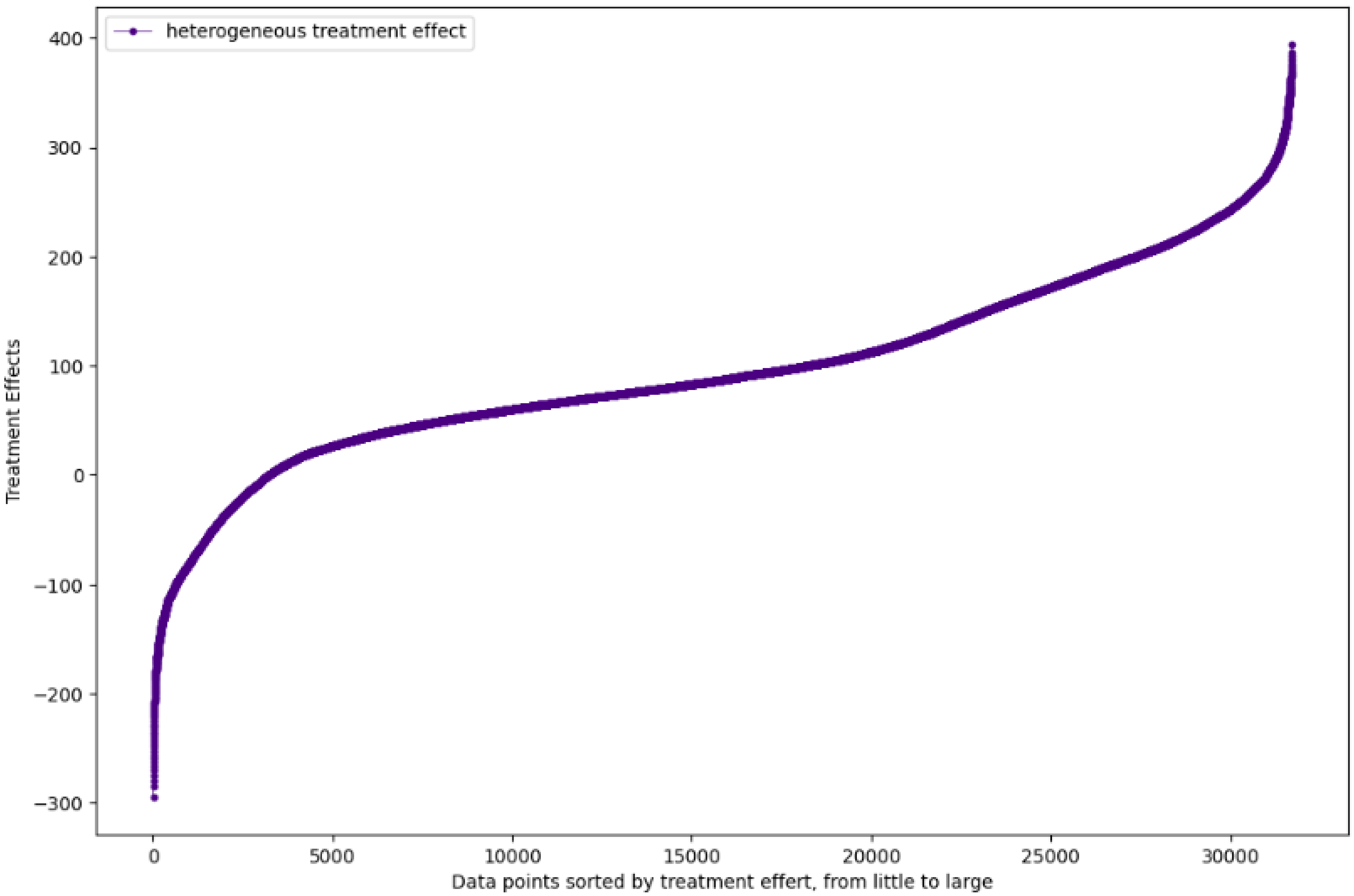
Treatment: number of image



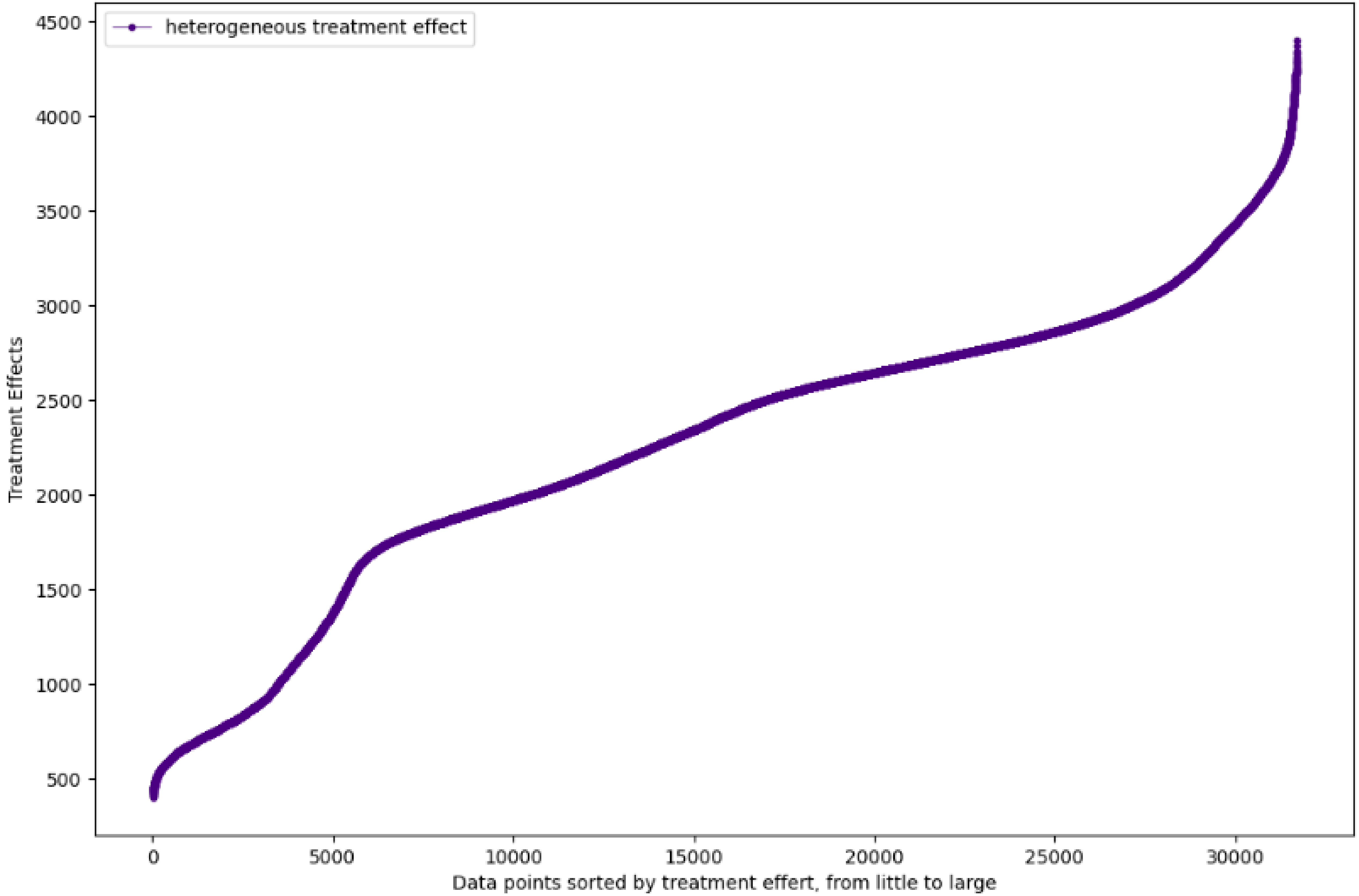
Treatment: number of video



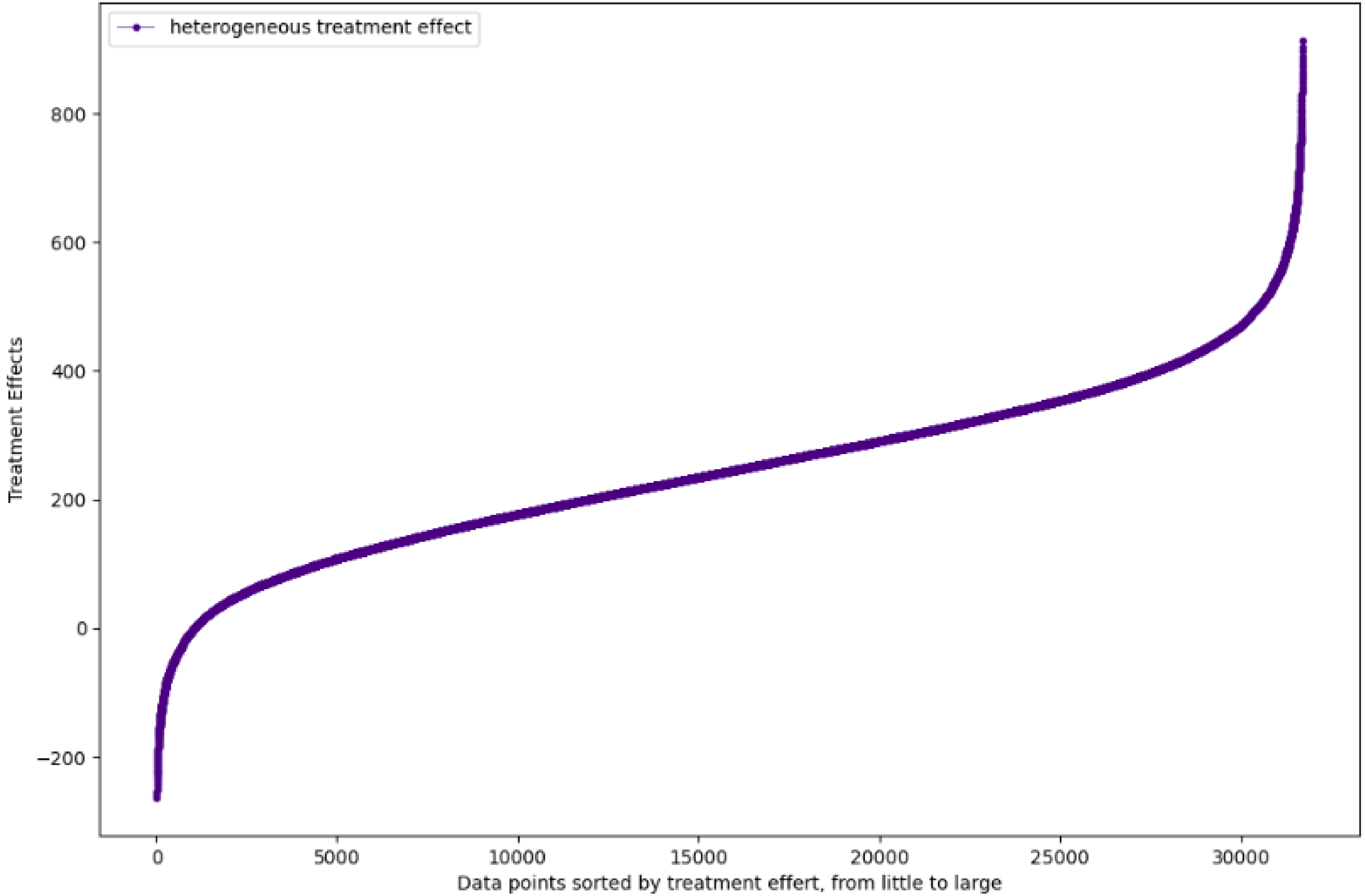
Treatment: number of keywords



Treatment: Global rate of positive words



Treatment: Average negative polarity



DOUBLE ML

- Package: DoubleMl for R and Python
- Provide an implementation of DML.
- The Python package is built on top of scikit-learn (Pedregosa et al.)
- The estimators we use for first stage are both RandomForestClassifier, with n_estimators=100, max_depth=10

MODEL SETTING

- We use DoubleMLQTE, which is able to analysis quantile treatment effect on the outcome variable ('shares')
- We select 0.25, 0.5 and 0.75 of the shares to analysis
- Example: data_channel_is_Work

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	92.0	8.476741	10.853228	1.925042e-27	75.385893	108.614107
0.50	300.0	15.291686	19.618504	1.074692e-85	270.028847	329.971153
0.75	600.0	55.588321	10.793634	3.689067e-27	491.048893	708.951107

Work or Entertainment

Treatment: data_channel_is_Work

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	92.0	8.476741	10.853228	1.925042e-27	75.385893	108.614107
0.50	300.0	15.291686	19.618504	1.074692e-85	270.028847	329.971153
0.75	600.0	55.588321	10.793634	3.689067e-27	491.048893	708.951107

Treatment: data_channel_is_Entertainment

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	-124.0	7.895859	-15.704434	1.410299e-55	-139.475599	-108.524401
0.50	-200.0	13.583944	-14.723265	4.570520e-49	-226.624041	-173.375959
0.75	-500.0	49.646961	-10.071110	7.413665e-24	-597.306255	-402.693745

Weekday or Weekend cross Work or Entertainment

Treatment: week_end_cross_data_channel_is_Work

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	663.0	33.431370	19.831673	1.586737e-87	597.475718	728.524282
0.50	900.0	73.256234	12.285644	1.081803e-34	756.420419	1043.579581
0.75	1200.0	134.788855	8.902813	5.444875e-19	935.818699	1464.181301

Treatment: week_day_cross_data_channel_is_Work

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	66.0	8.283023	7.968105	1.611258e-15	49.765573	82.234427
0.50	100.0	15.372044	6.505316	7.753027e-11	69.871348	130.128652
0.75	100.0	55.690812	1.795628	7.255365e-02	-9.151987	209.151987

Weekday or Weekend cross Work or Entertainment

Treatment: week_end_cross_data_channel_is_Entertainment

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	270.0	16.436012	16.427342	1.218846e-60	237.786008	302.213992
0.50	400.0	29.188369	13.704089	9.596463e-43	342.791848	457.208152
0.75	700.0	98.963356	7.073325	1.512645e-12	506.035386	893.964614

Treatment: week_day_cross_data_channel_is_Entertainment

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	-255.0	8.307390	-30.695562	6.523277e-207	-271.282185	-238.717815
0.50	-300.0	12.499810	-24.000365	2.756460e-127	-324.499177	-275.500823
0.75	-500.0	45.290534	-11.039835	2.454847e-28	-588.767815	-411.232185

The day of the week

Treatment: weekday_is_monday

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	-3.300000e+01	10.292937	-3.206082e+00	0.001346	-53.173786	-12.826214
0.50	-1.591616e-12	16.246209	-9.796844e-14	1.000000	-31.841985	31.841985
0.75	1.818989e-12	58.155070	3.127826e-14	1.000000	-113.981842	113.981842

Treatment: weekday_is_tuesday

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	-59.0	9.925113	-5.944517	2.772734e-09	-78.452864	-39.547136
0.50	-100.0	13.809150	-7.241576	4.435021e-13	-127.065436	-72.934564
0.75	-300.0	50.129925	-5.984449	2.171230e-09	-398.252847	-201.747153

The day of the week

Treatment: weekday_is_thuresday

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	-57.0	10.403005	-5.479186	4.272872e-08	-77.389516	-36.610484
0.50	100.0	64.415904	1.552412	1.205638e-01	-26.252852	226.252852
0.75	-300.0	79.304940	-3.782866	1.550326e-04	-455.434826	-144.565174

Treatment: weekday_is_friday

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	4.600000e+01	12.608817	3.648241e+00	0.000264	21.287172	70.712828
0.50	1.000000e+02	31.374456	3.187306e+00	0.001436	38.507196	161.492804
0.75	4.547474e-13	117.280146	3.877445e-15	1.000000	-229.864863	229.864863

The day of the week

Treatment: weekday_is_saturday

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	367.0	23.874357	15.372142	2.517077e-53	320.207121	413.792879
0.50	600.0	35.277170	17.008167	7.143814e-65	530.858017	669.141983
0.75	800.0	85.125243	9.397917	5.565368e-21	633.157590	966.842410

Treatment: is_weekend

	coef	std err	t	P>|t|	2.5 %	97.5 %
0.25	384.0	14.714181	26.097273	3.915149e-150	355.160735	412.839265
0.50	500.0	23.751877	21.050968	2.240579e-98	453.447176	546.552824
0.75	900.0	64.912810	13.864752	1.035837e-43	772.773230	1027.226770

CONCLUSION

- On weekends, news popularity increase, regardless of its category
- On monday and friday, people tend to watch more news, compared to the other weekdays
- The categories of the news itself are generally important
- Negativity does not help, while positivity does help

APPENDIX 1

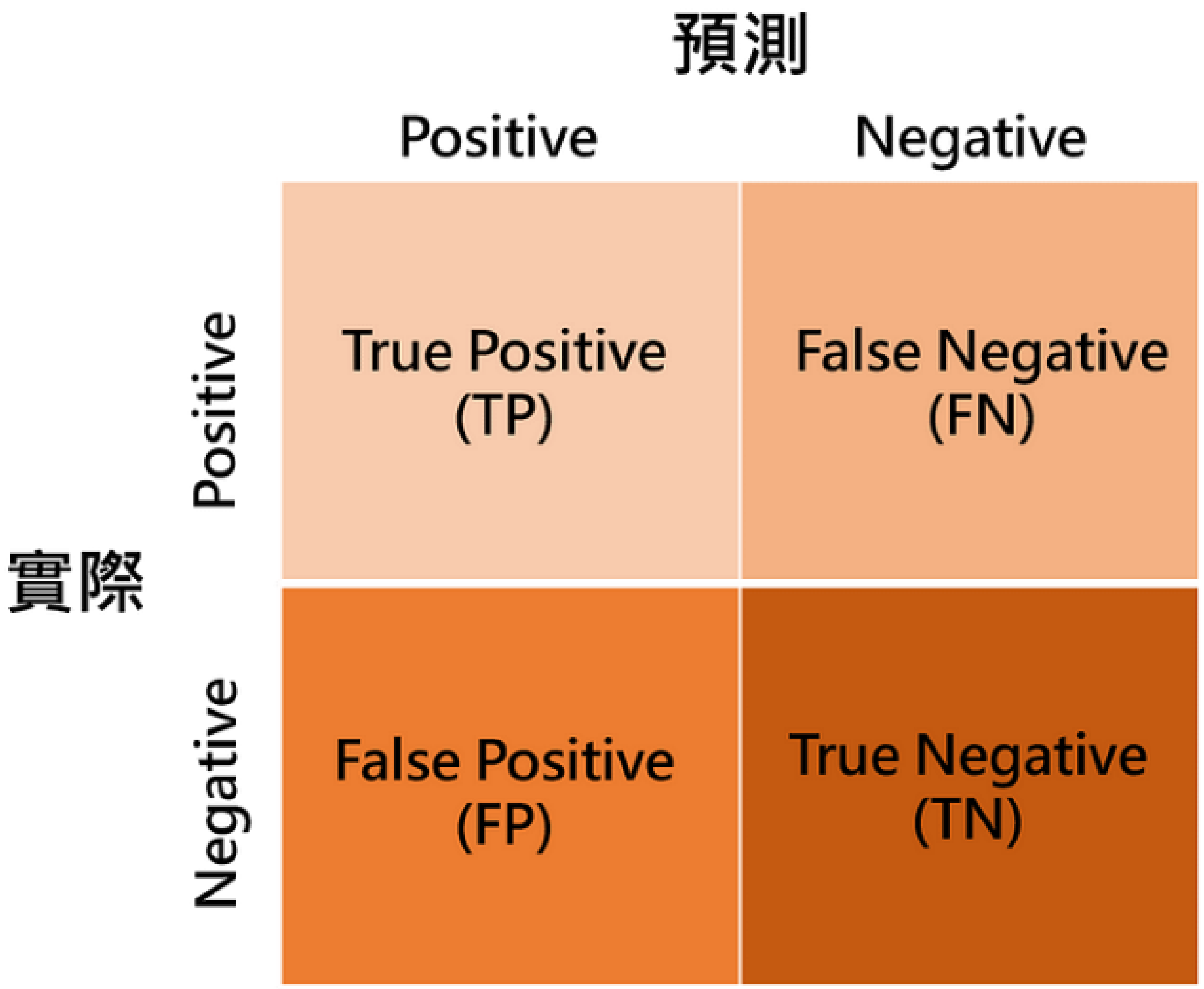
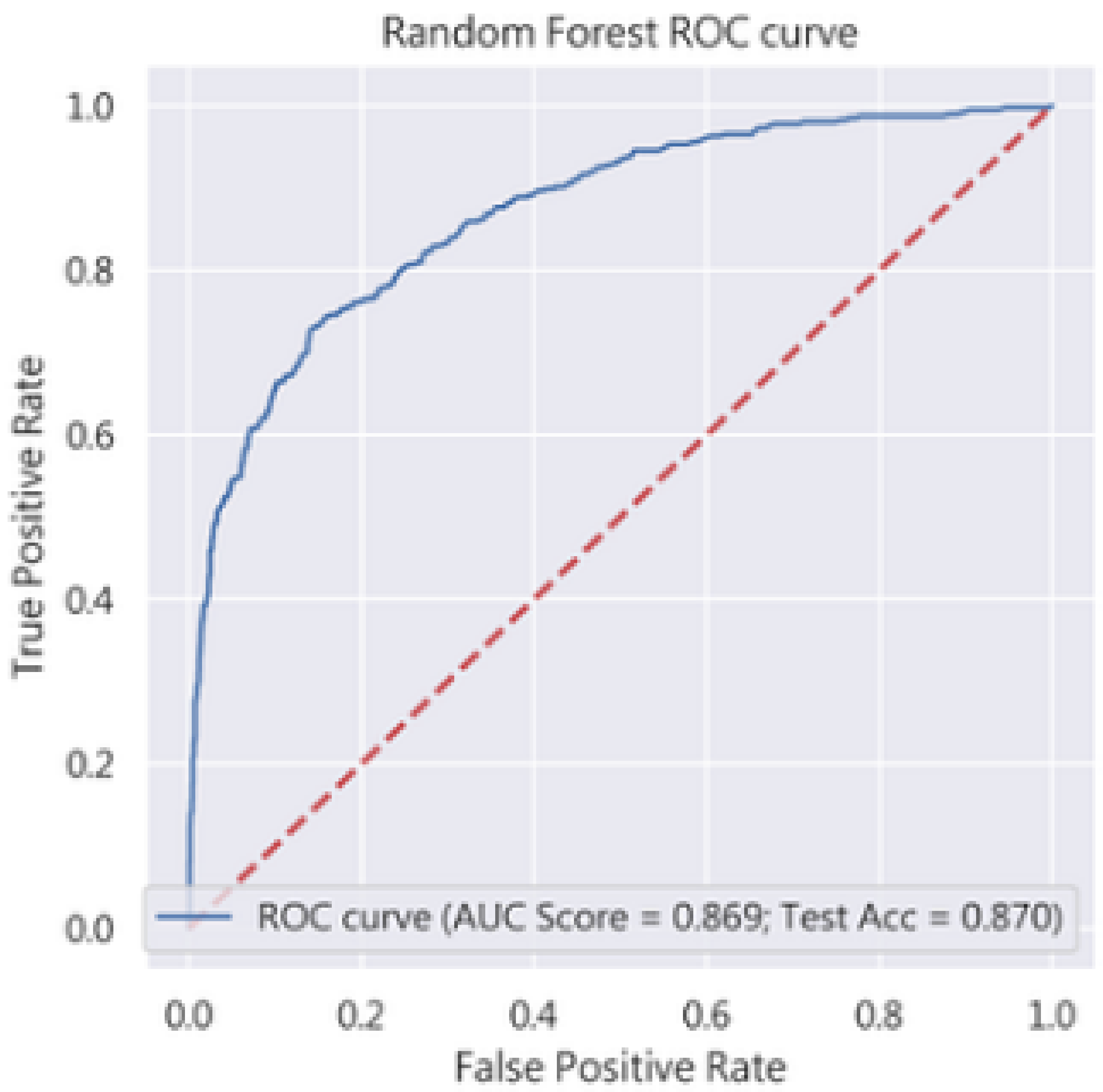


圖1 混淆矩陣

APPENDIX 1

- Accuracy score: The accurate classification ratio: (TP+TN)／(TP+TN+FP+FN)
- Precision score is tp / (tp + fp)
- Recall scroe is tp / (tp + fn)
- F1 = 2 * (precision * recall) / (precision + recall)
- AUC: Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.

APPENDIX 1



Random Forest ROC curve
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve (AUC Score = 0.869; Test Acc = 0.870)

APPENDIX 2

- **Naive Bayes**
- Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable.

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y),$$

APPENDIX 2

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

APPENDIX 2

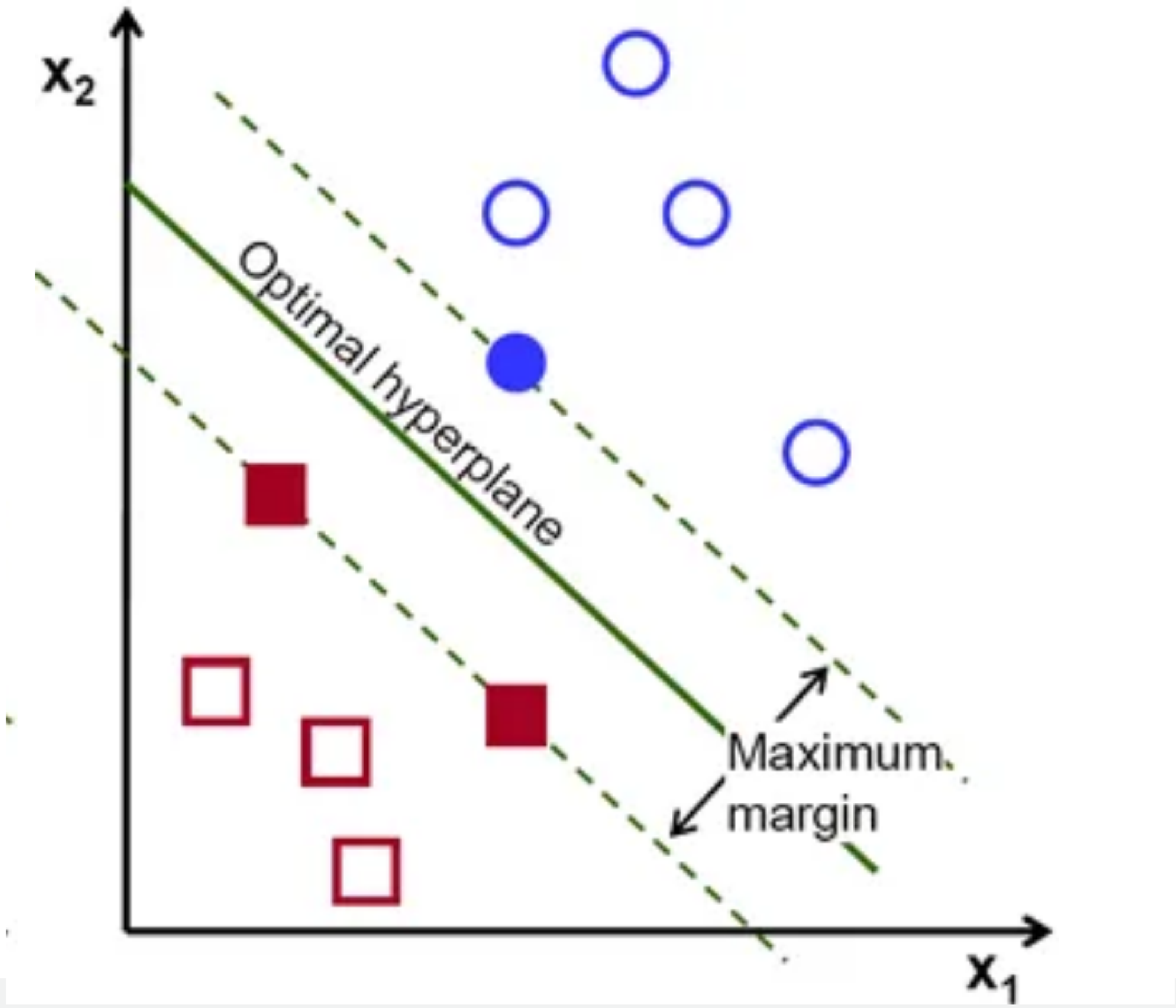
- **K-NN**
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm

APPENDIX 2

- **SVM**
- The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.
- We choose kernal to be 'rbf', while there are other kernals to choose

APPENDIX 2

- SVM



APPENDIX 2

- **AdaBoost**
- The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set. Adaboost should meet two conditions:
- The classifier should be trained interactively on various weighed training examples.
- In each iteration, it tries to provide an excellent fit for these examples by minimizing training error.

APPENDIX 2

- Initially, Adaboost selects a training subset randomly.
- It iteratively trains the AdaBoost machine learning model by selecting the training set based on the accurate prediction of the last training.
- It assigns the higher weight to wrong classified observations so that in the next iteration these observations will get the high probability for classification.
- Also, It assigns the weight to the trained classifier in each iteration according to the accuracy of the classifier. The more accurate classifier will get high weight.
- This process iterate until the complete training data fits without any error or until reached to the specified maximum number of estimators.
- To classify, perform a "vote" across all of the learning algorithms you built.

APPENDIX 3

- **Original DML Framework**
- **DGP**

$$
\begin{aligned}
Y &= \theta(X) \cdot T + g(X, W) + \epsilon & \mathbb{E}[\epsilon | X, W] &= 0 \\
T &= f(X, W) + \eta & \mathbb{E}[\eta \mid X, W] &= 0 \\
& \mathbb{E}[\eta \cdot \epsilon | X, W] = 0
\end{aligned}
$$

- **Estimation criteria**

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \mathbb{E}_n \left[(\tilde{Y} - \theta(X) \cdot \tilde{T})^2 \right]$$