Modified Causal Forest: Estimation, Optimal Policy Estimation

Section 1: General information

Welcome to the mcf estimation and optimal policy package.
This report provides you with a summary of specifications and results. More detailed information can be found in the respective output files. Figures and data (in csv-format, partly to recreate the figures on your own) are provided in the output path as well.

Output information for MCF ESTIMATION
All outputs: c:\Users\User\Desktop\06 June 2025 causal\24062025\output
Subdirectories with figures and data are named ate_iate, gate, and common support and contain the content related to their name.
Detailed text output: c:\Users\User\Desktop\06 June 2025 causal\24062025\output\txtFileWithOutput.txt
Summary text output: c:\Users\User\Desktop\06 June 2025 causal\24062025\output\txtFileWithOutput_Summary.txt

Output information for OPTIMAL POLICY ANALYSIS
All outputs: c:\Users\User\Desktop\06 June 2025 causal\24062025\output0
Detailed text output: c:\Users\User\Desktop\06 June 2025 causal\24062025\output0\txtFileWithOutput.txt
Summary text output: c:\Users\User\Desktop\06 June 2025 causal\24062025\output0\txtFileWithOutput_Summary.txt

BACKGROUND

ESTIMATION OF EFFECTS
The MCF is a comprehensive causal machine learning estimator for the estimation of treatment effects at various levels of granularity, from the average effect at the population level to very fine grained effects at the (almost) individual level. Since effects at the higher levels are obtained from lower level effects, all effects are internally consistent. Recently, the basic package has been appended for new average effects as well as for an optimal policy module. Effect estimation is implemented for identification by unconfoundedness as well as by instrumental variables. While unconfoundedness estimation can deal with multiple treatments, instrumental variable estimation is restricted to binary instruments and binary treatments.The basis of the MCF estimator is the the causal forest suggested by Wager and Athey (2018). Their estimator has been changed in several dimensions which are described in Lechner (2018). The main changes relate to the objective function as well as to the aggreation of effects. Lechner and Mareckova (2024) provide the asymptotic guarantees for the MCF and compare the MCF, using a large simulation study, to competing approaches like the Generalized Random Forest (GRF, Athey, Tibshirani, Wager, 2019) and Double Machine Learning (DML, Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, Robins, 2018, Knaus, 2022). In this comparison the MCF faired very well, in particular, but not only, for heterogeneity estimation. Some operational issues of the MCF are discussed in Bodory, Busshof, Lechner (2022). There are several empirical studies using the MCF, like Cockx, Lechner, Boolens (2023), for example.

References
- Athey, S., J. Tibshirani, S. Wager (2019): Generalized Random Forests, The Annals of Statistics, 47, 1148-1178.

- Athey, S., S. Wager (2019): Estimating Treatment Effects with Causal Forests: An Application, Observational Studies, 5, 21-35.
- Bodory, H., H. Busshoff, M. Lechner (2023): High Resolution Treatment Effects Estimation: Uncovering Effect Heterogeneities with the Modified Causal Forest, Entropy, 24, 1039.
- Bodory, H., F. Mascolo, M. Lechner (2024): Enabling Decision Making with the Modified Causal Forest: Policy Trees for Treatment Assignment, Algorithm, 17, 318.
- Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, J. Robins (2018): Double/debiased machine learning for treatment and structural parameters, Econometrics Journal, 21, C1-C68.
- Cockx, B., M. Lechner, J. Bollens (2023): Priority to unemployed immigrants? A causal machine learning evaluation of training in Belgium, Labour Economics, 80, Article 102306.
- Knaus, M. (2022): Double Machine Learning based Program Evaluation under Unconfoundedness, Econometrics Journal.
- Lechner, M. (2018): Modified Causal Forests for Estimating Heterogeneous Causal Effects, arXiv.
- Lechner, M. (2023): Causal Machine Learning and its Use for Public Policy, Swiss Journal of Economics & Statistics, 159:8.
- Lechner, M., J. Mareckova (2024): Comprehensive Causal Machine Learning, arXiv.
- Lechner, M., J. Mareckova (2025): Comprehensive Causal Machine Learning with Instrumental Variables, mimeo.
- Wager, S., S. Athey (2018): Estimation and Inference of Heterogeneous Treatment Effects using Random Forests, Journal of the American Statistical Association, 113:523, 1228-1242.

The optimal policy module offers three (basic) algorithms that can be used to exploit fine grained knowledge about effect heterogeneity to obtain decision rules. The current version is implemented for discrete treatments only.
There is also an option for different fairness adjustments.

The BEST_POLICY_SCORE algorithm is based on assigning the treatment that has the highest impact at the unit (e.g., individual) level. If the treatment heterogeneity is known (not estimated), this will lead to the best possible result. This algorithm is computationally not burdensome. However, it will not be easy to understand how the implied rules depends on the features of the unit. Its statistical properties are also not clear (for estimated treatment heterogeneity) and there is a certain danger of overfitting, which could lead to an unsatisfactory out-of-training-sample performance.

The BPS_CLASSIFIER classifier algorithm runs a classifier for each of the allocations obtained by the BEST_POLICY_SCORE algorithm. One advantage of this approach compared to the BEST_POLICY_SCORE algorithm is that prediction of the allocation of (new) observations is fast because it does not require to recompute the policy score (as it is the case with the BEST_POLICY_SCORE algorithm). The specific classifier is selected among four different classifiers from scikit-learn, namely a simple neural network, two classification random forests with minimum leaf size of 2 and 5, and ADDABoost. The selection is made according to the out-of-sample performance of the Accuracy Score of scikit-learn.

The POLICY TREE algorithm builds optimal shallow decision trees. While these trees are unlikely to lead to gloablly optimal allocations, and are computationally much more expensive, they have the advantage that the decision rule is much easier to understand and that some statistical properties are known, at least for certain versions of such decision trees (e.g., Zhou, Athey, Wager, 2023). The basic algorithmic implementation follows the recursive algorithm suggested by Zhou, Athey, Wager (2023) with three (more substantial) deviations (=extensions).
Extension 1: Since using One Hot Encoding for categorical variables may lead to rather extreme

leaves for such variables with many different values when building (shallow) trees (splitting one value against the rest), a more sophisticated procedure is used that allows to have several values of the categorical variables on both sides of the split.
Extension 2: Constraints are allowed for. They are handled in a sequential manner: First, an approximate treatment-specific cost vector is obtained and used to adjust the policy score accordingly. Second, trees that violate the constraints are removed (to some extent, optional).
Extensions 3: There are several options implemented to reduce the computational burden, which are discussed below in the section showing the implementation of the policy score.

References

-Zhou, Z., S. Athey, S. Wager (2023): Offline Multi-Action Policy Learning: Generalization and Optimization, Operations Research, INFORMS, 71(1), 148-183.

Modified Causal Forest: Estimation, Optimal Policy Estimation

Section 2: MCF estimation

METHOD
Standard MCF method used. Nearest neighbour matching performed using the Prognostic Score.
Feature selection not is used.
Local centering is used.
Common support is enforced.

VARIABLES
Outcome: jan_avg_inc_opv
Treatment: oct_asp_flag (with values 0 1)
Ordered confounders: oct_head_inc, oct_avg_inc_opv, oct_fam_cnt, oct_fam_cat, oct_com_re_own, oct_children, oct_disab_child, oct_overdue_debt, oct_gov_complaint, jul_overdue_debt, aug_overdue_debt, sep_overdue_debt, jul_head_inc, oct_high_edu_adults, oct_fam_debt, oct_payer_opv, jul_avg_inc_opv, aug_head_inc, aug_avg_inc_opv, sep_head_inc, sep_avg_inc_opv, oct_api_level, oct_high_edu_adults, oct_children, oct_disab_child
Unordered (categorical) confounders: oct_regions, oct_employment, oct_real_estate, oct_fndr_smallbiz, oct_regions
Ordered heterogeneity variables (few values, continuous variables are discretized): oct_high_edu_adults, oct_children, oct_disab_child
Unordered heterogeneity variables: oct_regions

EFFECTS ESTIMATED
Average Treatment Effect (ATE), Group Average Treatment Effect (GATE), Individualized Average Treatment Effect (IATE), Efficient IATE

NOTE on unordered variables:
One-hot-encoding (dummy variables) is not used as it is expected to perform poorly with trees: It may lead to splits of one category versus all other categories. Instead the approach used is analogous to the one discussed in Chapter 9.2.4 of Hastie, Tibshirani, Friedmann (2013), The Elements of Statistical Learning, 2nd edition.

Section 2.1: MCF Training

Training uses 19 CPU cores.

Section 2.1.1: Preparation of training data (mcf training)

METHOD
Variables without variation are removed.
Variables that are perfectly correlated with other variables are removed.
Dummy variables with less than 10 observations in the smaller group are removed.
Rows with any missing values for variables needed for training are removed.

Modified Causal Forest: Estimation, Optimal Policy Estimation

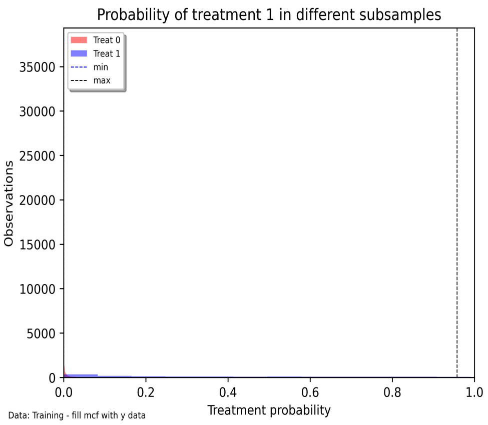
RESULTS
No relevant variables were removed.
Sample size of training data: 150000   (no observations removed).

Section 2.1.2: Common support (mcf training)

METHOD
The common support analysis is based on checking the overlap in the out-of-sample predictions of the propensity scores (PS) for the different treatment arms. PSs are estimated by random forest classifiers. Overlap is operationalized by computing cut-offs probabilities of the PSs (ignoring the first treatment arm, because probabilities add to 1 over all treatment arms). These cut-offs are subsequently also applied to the data used for predicting the effects.
Overlap is determined by the min / max rule.
Out-of-sample predictions are generated by 4-fold cross-validation.

RESULTS
Share of observations deleted: 0.04%
Number of observations remaining: 149944

Common support plots



Section 2.1.3: Local centering (mcf training)

METHOD
Local centering is based on training a regression to predict the outcome variable conditional on the

features (without the treatment). The regression method is selected among various versions of Random Forests, Support Vector Machines, Boosting methods, and Neural Networks of scikit-learn. The best method is selected by minimizing their out-of-sample Mean Squared Error using 4-fold cross-validation. The full set of results of the method selection step are contained in c:\Users\User\Desktop\06 June 2025 causal\24062025\output\txtFileWithOutput.txt.
The respective out-of-sample predictions are subtracted from the observed outcome in the training data used to build the forest. These out-of-sample predictions are generated by 4-fold cross-validation.

RESULTS
Out-of-sample fit for Random Forest of Ey|x (R2) for jan_avg_inc_opv: 44.86%

Section 2.1.4: Forest

METHOD and tuning parameters
Method used for forest building is MSE & MCE Penalty "MSE of treatment variable".
The causal forest consists of 1000 trees.
The minimum leaf size is 4.
The number of variables considered for each split is 8.
The share of data used in the subsamples for forest building is 41%.
The share of the data used in the subsamples for forest evaluation (outcomes) is 82%.
Alpha regularity is set to 10%.

jan_avg_inc_opv_lc is the outcome variable used for splitting (locally centered).
The features used for splitting are oct_head_inc oct_avg_inc_opv oct_fam_cnt oct_fam_cat oct_com_re_own oct_children oct_disab_child oct_overdue_debt oct_gov_complaint jul_overdue_debt aug_overdue_debt sep_overdue_debt jul_head_inc oct_high_edu_adults oct_fam_debt oct_payer_opv jul_avg_inc_opv aug_head_inc aug_avg_inc_opv sep_head_inc sep_avg_inc_opv oct_api_level oct_regions oct_employment oct_real_estate oct_fndr_smallbiz.

RESULTS
Each tree has on average 237.90 leaves.
Each leaf contains on average 65.2 observations. The median # of observations per leaf is 34.
The smallest leaves have 4 observations.
The largest leaf has 4739 observations.
22.35% of the leaves were merged when populating the forest with outcomes from the honesty sample.

NOTES
To reduce computational demands, data is randomly splitted in 2 folds. In each fold, forests and effects are estimated. Subsequently, effects are averaged over the {dic["folds"]} folds.
For the estimation of the "efficient" IATEs, the role of the samples used for building the forest and populating it are reversed. Subsequently, the two sets of estimates for the IATEs are averaged.

Modified Causal Forest: Estimation, Optimal Policy Estimation

Section 2: MCF estimation

Section 2.2: MCF Prediction of Effects

Training uses 19 CPU cores.

Section 2.2.1: Common support (mcf prediction)

Share of observations deleted: 0.12%
Number of observations remaining: 104874

Section 2.2.2: Results

GENERAL REMARKS
The following results for the different parameters are all based on the same causal forests (CF). The combination of the CF with the potentially new data provided leads to weight matrices. These matrices may be large requiring some computational optimisations, such as processing them in batches and saving them in a sparse matrix format. One advantage of this approach is that aggregated effects (ATE, GATE, BGATE) can be computed by aggregation of the weights used for the IATE. Thus a high internal consistency is preserved in the sense that IATEs will aggregate to GATEs, which in turn will aggregate to ATEs.

ESTIMATION
Weights of individual training observations are truncated at 5.00%. Aggregation of IATEs to ATE and GATEs may not be exact due to weight truncation.

INFERENCE
Inference is based on using the weight matrix. Nonparametric regressions are based on k-nearest neighbours.

NOTE
Treatment effects for specific treatment groups (so-called treatment effects on the treated or non-treated) can only be provided if the data provided for prediction contains a treatment variable (which is not required for the other effects).

Section 2.2.2.1: ATE

Modified Causal Forest: Estimation, Optimal Policy Estimation

RESULT

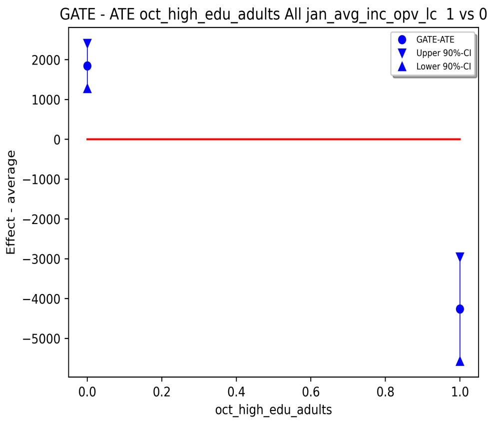
ATE for jan_avg_inc_opv

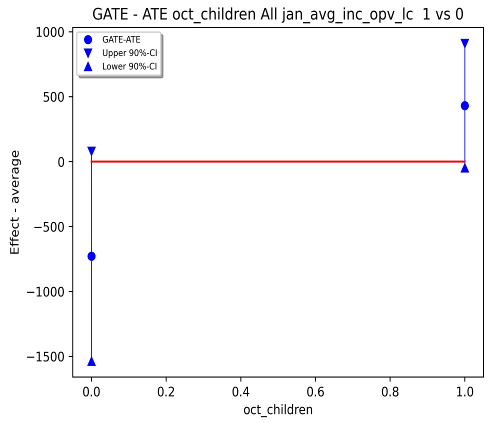
Comparison	*Effect*	*SE*	*t-value*	*p-value (%)*	*Sig.*
1 vs 0	-8413.399	1437.235	5.85	0.0	****

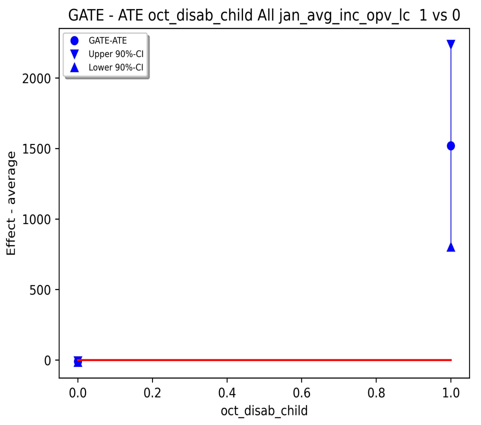
Note: *, **, ***, **** denote significance at the 10%, 5%, 1%, 0.1% level. The results for the potential outcomes can be found in the output files.

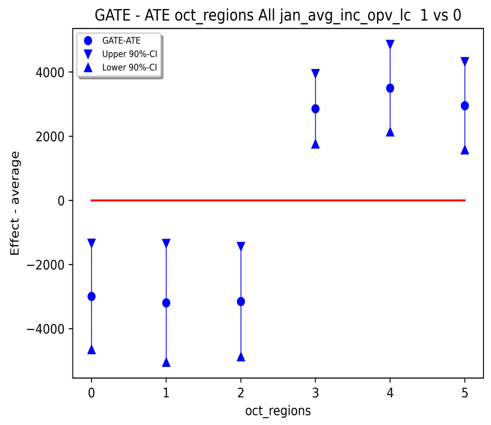
Section 2.2.2.2: GATE

GATE









Note: Detailed tables and figures for additional effects are contained in the output files and output directories.

Section 2.2.2.3: IATE

Modified Causal Forest: Estimation, Optimal Policy Estimation

This section contains parts of the descriptive analysis of the IATEs. Use the analyse method to obtain more descriptives of the IATEs, like their distribution, and their relations to the features.

METHODOLOGICAL NOTE
In order to increase the efficiency of the IATE estimation, a second set of IATEs is computed by reversing the role of the two samples used to build the forest and to populate it with the outcome information. The two IATEs are averaged to obtain a more precise estimator (which may be particulary useful when the IATEs, or the corresponding potential outcomes, are used as inputs for decision models).
The following descriptive analysis is based on the first round IATEs only.

RESULTS

Outcome variable: jan_avg_inc_opv

Comparison	Mean	Median	Std	Effect > 0	mean(SE)	sig 10%	sig 5%	sig 1%
1 vs 0	-7082.53608	-4087.70018	7892.71715	11.72%	3501.35081	52.59%	44.14%	29.05%

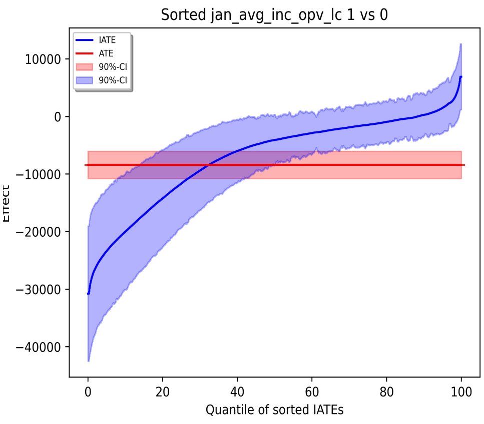
Modified Causal Forest: Estimation, Optimal Policy Estimation

Section 2: MCF estimation

Section 2.3: Analysis of Estimated IATEs

This section contains parts of the descriptive analysis of the IATEs. More detailed tables and figures are contained in the output files and directories. These additional results include variable importance plots from a regression random forest as well as and linear regression results. Both estimators use the estimated IATEs as dependent variables and the confounders and heterogeneity variables as features.

IATEs



K-MEANS CLUSTERING
The sample is divided using k-means clustering based on the estimated IATEs. The number of clusters is determined by maximizing the Average Silhouette Score on a given grid. The following table shows the means of the IATEs, potential outcomes, and the features in these clusters, respectively.

Number of observations in the clusters

Cluster 0:   4655
Cluster 1:   8241
Cluster 2:   8919
Cluster 3:   9913

Modified Causal Forest: Estimation, Optimal Policy Estimation

Cluster 4: 12899
Cluster 5: 23981
Cluster 6: 28613
Cluster 7: 7653

IATE

Comparison	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
jan_avg_inc_opv_lc1vs0_iate	-26588.36	-21094.78	-16164.59	-11447.95	-7128.79	-3659.06	-965.15	2588.57

Note: Mean of variable in cluster.

Potential Outcomes

Comparison	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
jan_avg_inc_opv_lc0_un_lc_pot	412356.15	336507.31	255103.29	154589.91	79392.74	30710.7	15426.13	17011.31
jan_avg_inc_opv_lc1_un_lc_pot	385767.79	315412.54	238938.7	143141.96	72263.95	27051.65	14460.98	19599.88

Note: Mean of variable in cluster.

Features

Comparison	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
oct_regions	6.34	7.34	9.11	9.96	10.92	11.99	13.24	14.36
oct_employment	2.64	2.54	2.57	2.59	2.46	1.68	1.41	1.66
oct_real_estate	1.7	1.81	1.86	1.71	1.59	1.44	1.27	1.14
oct_fndr_smallbiz	0.41	0.47	0.54	0.54	0.5	0.42	0.33	0.26
oct_head_inc	2.75	2.42	2.08	1.48	1.0	0.45	0.22	0.28
oct_avg_inc_opv	4.25	3.91	3.38	2.41	1.39	0.56	0.27	0.32
oct_fam_cnt	2.87	3.13	3.47	3.89	4.21	3.7	3.63	4.14
oct_fam_cat	2.98	2.93	2.91	2.77	2.49	1.95	1.63	1.64
oct_com_re_own	0.02	0.02	0.03	0.03	0.03	0.03	0.03	0.03
oct_children	0.56	0.55	0.63	0.7	0.74	0.6	0.59	0.71
oct_disab_child	0.0	0.0	0.0	0.01	0.01	0.01	0.01	0.01
oct_overdue_debt	0.14	0.15	0.18	0.24	0.27	0.31	0.27	0.31
oct_gov_complaint	0.04	0.12	0.31	0.31	0.3	0.25	0.23	0.19
jul_overdue_debt	0.14	0.14	0.17	0.23	0.26	0.3	0.26	0.29
aug_overdue_debt	0.14	0.14	0.17	0.23	0.26	0.3	0.26	0.29
sep_overdue_debt	0.14	0.15	0.18	0.24	0.27	0.31	0.27	0.3
jul_head_inc	2.55	2.33	2.1	1.63	1.05	0.41	0.16	0.12

oct_high_edu_adults	0.58	0.5	0.47	0.46	0.4	0.25	0.15	0.09
oct_fam_debt	0.97	0.97	0.98	0.99	0.99	0.98	0.99	0.99
oct_payer_opv	1.0	1.0	1.0	0.99	0.98	0.64	0.42	0.61
jul_avg_inc_opv	4.74	4.64	4.3	3.53	2.36	1.02	0.54	0.68
aug_head_inc	2.85	2.43	2.14	1.62	1.0	0.36	0.13	0.09
aug_avg_inc_opv	5.21	4.84	4.39	3.52	2.3	0.94	0.47	0.59
sep_head_inc	2.9	2.45	2.04	1.53	0.98	0.37	0.14	0.1
sep_avg_inc_opv	5.29	4.83	4.22	3.38	2.24	0.98	0.53	0.72
oct_api_level	4.96	4.88	4.76	4.47	3.77	2.58	1.68	1.71
oct_regions_0	0.24	0.27	0.22	0.19	0.17	0.18	0.12	0.06
oct_regions_2	0.26	0.25	0.22	0.2	0.17	0.14	0.13	0.14
oct_regions_10	0.38	0.24	0.22	0.21	0.19	0.12	0.13	0.07
oct_regions_17	0.08	0.08	0.1	0.13	0.16	0.16	0.2	0.31
oct_regions_19	0.01	0.07	0.13	0.13	0.15	0.16	0.24	0.21
oct_regions_20	0.02	0.08	0.12	0.14	0.16	0.23	0.18	0.2
oct_employment_0	0.0	0.0	0.0	0.0	0.01	0.2	0.15	0.05
oct_employment_1	0.0	0.0	0.0	0.01	0.01	0.16	0.43	0.34
oct_employment_2	0.36	0.46	0.43	0.39	0.48	0.39	0.29	0.5
oct_employment_3	0.64	0.54	0.57	0.6	0.5	0.24	0.13	0.1
oct_real_estate_0	0.38	0.31	0.3	0.34	0.37	0.4	0.47	0.55
oct_real_estate_1	0.02	0.03	0.04	0.04	0.03	0.04	0.04	0.04
oct_real_estate_2	0.28	0.34	0.33	0.33	0.34	0.38	0.31	0.2
oct_real_estate_3	0.17	0.2	0.18	0.16	0.15	0.11	0.11	0.16
oct_real_estate_4	0.14	0.11	0.13	0.11	0.1	0.07	0.06	0.06
oct_real_estate_5	0.01	0.02	0.02	0.01	0.01	0.01	0.01	0.0
oct_fndr_smallbiz_0	0.75	0.7	0.66	0.65	0.66	0.71	0.76	0.81
oct_fndr_smallbiz_1	0.13	0.18	0.2	0.23	0.23	0.2	0.17	0.14
oct_fndr_smallbiz_2	0.08	0.07	0.07	0.06	0.05	0.05	0.04	0.03
oct_fndr_smallbiz_3	0.04	0.05	0.07	0.06	0.05	0.04	0.03	0.02

Note: Mean of variable in cluster. Categorical variables are recoded to indicator (dummy) variables.

Section 3: Optimal Policy

METHOD
The assignment rule is based on allocating units using a shallow decision tree of depth 4.

VARIABLES provided
Policy scores: jan_avg_inc_opv_lc0_un_lc_pot_eff, jan_avg_inc_opv_lc1_un_lc_pot_eff
Treatment: oct_asp_flag
Identifier: id_mcfx
Oderered features of units: oct_children
Categorical / unorderered features of units: oct_regions

COSTS
No user provided costs of specific treatments.

RESTRICTIONS of treatment shares
Treatment shares are unrestricted.

FAIRNESS
No fairness adjustments performed.

Section 3.1: Optimal Policy: Training

COMPUTATION
19 logical cores are used for processing.
Continuous variables are internally split for best use of cpu ressources.

DATA PREPARATION
Variables without variation are removed.
Variables that are perfectly correlated with other variables are removed.
Dummy variables with less than 10 observations in the smaller group are removed.
Rows with any missing values for variables needed for training are removed.

COMPUTATIONAL EFFICIENCY
Optimal policy trees are computationally very demanding. Therefore, several approximation parameters are used.
Instead of evaluating all values of continuous variables and combinations of values of categorical variables when splitting, only 100 values are considered. These values are equally spaced for continuous variables and random combinations for categorical variables. This number is used for EVERY splitting decision, i.e. the approximation improves the smaller the data in the leaf becomes. Increasing this value can significantly improve the computational performance at the price of a certain approximation loss.
The depth of the tree is also a key parameter. Usually, it is very hard to estimate trees beyond the depth of 4 (16 leaves) with reasonably sized training data. There are two options to improve the computational performance. The first one is to reduce the depth (leading to loss of efficiency but a

gain in interpretability). The second option is to split the tree building into several steps. However, the second option is not used here.
Another parameter crucial for performance is the minimum leaf size. Too small leaves may be undesirable for practical purposes (and they increase computation times). The minimum leaf size in this application is set to 1311.
In addition, the user may reduce the size of the training data to increase speed, but this will increase sampling noise.

CATEGORICAL VARIABLES
There are two different approximation methods for larger categorical variables. Since we build optimal trees, for categorical variables we need to check all possible combinations of the different values that lead to binary splits. This number could indeed be huge. Therefore, we compare only 200 different combinations. The available methods differ on how these methods are implemented. In this application, at each possible split, we sort the values of the categorical variables according to the values of the policy scores as one would do for a standard random forest. If this set is still too large, a random sample of the entailed combinations is drawn.

STRUCTURE OF FINAL TREE (using data from Training PT data)

Leaf information for estimated policy tree

Depth of 1st tree: 3, depth of 2nd tree: 1, total depth: 4
- -
Leaf 00: oct_children <= 0.500 oct_regions In: 0 17 oct_regions In: 0
Alloc Treatment: 0 Obs: 6999
- -
Leaf 01: oct_children <= 0.500 oct_regions In: 0 17 oct_regions Not in: 0
Alloc Treatment: 0 Obs: 4258
- -
Leaf 10: oct_children > 0.500 oct_regions In: 17 19 oct_regions In: 17
Alloc Treatment: 0 Obs: 13046
- -
Leaf 11: oct_children > 0.500 oct_regions In: 17 19 oct_regions Not in: 17
Alloc Treatment: 0 Obs: 11151
- -
Leaf 20: oct_children <= 0.500 oct_regions Not in: 0 17 oct_regions In: 2 10 oct_regions In: 2
Alloc Treatment: 0 Obs: 7724
- -
Leaf 21: oct_children <= 0.500 oct_regions Not in: 0 17 oct_regions In: 2 10 oct_regions Not in: 2
Alloc Treatment: 0 Obs: 6360
- -
Leaf 30: oct_children <= 0.500 oct_regions Not in: 0 17 oct_regions Not in: 2 10 oct_regions In: 19
Alloc Treatment: 0 Obs: 6335
- -

Leaf 31: oct_children <= 0.500 oct_regions Not in: 0 17 oct_regions Not in: 2 10 oct_regions Not in: 19
Alloc Treatment: 0 Obs: 7342
- -
Leaf 40: oct_children > 0.500 oct_regions Not in: 17 19 oct_regions In: 0 10 oct_regions In: 0
Alloc Treatment: 0 Obs: 10617
- -
Leaf 41: oct_children > 0.500 oct_regions Not in: 17 19 oct_regions In: 0 10 oct_regions Not in: 0
Alloc Treatment: 0 Obs: 11175
- -
Leaf 50: oct_children > 0.500 oct_regions Not in: 17 19 oct_regions Not in: 0 10 oct_regions In: 2
Alloc Treatment: 0 Obs: 9687
- -
Leaf 51: oct_children > 0.500 oct_regions Not in: 17 19 oct_regions Not in: 0 10 oct_regions Not in: 2
Alloc Treatment: 0 Obs: 10180
- -
NOTE: Splitpoints displayed for ordered variables are midpoints between observable values (e.g., 0.5 for a variable with values of 0 and 1).

Section 3.2: Optimal Policy: Evaluation of Allocation(s)

Main evaluation results.

Note: The output files contain relevant additional information, like a descriptive analysis of the treatment groups. Also Qini-like plots are produced and saved in the same location as the text output. Those plots compare the optimal allocation to a reference allocation (3 allocations are used as such reference allocations, if available: (i) observed, (ii) random, (iii) the treatment with the highest ATE is allocated to everybody). They show the mean welfare when an increasing share of observations (starting with those who gain most from the optimal allocation compared to the reference allocation) is allocated using the optimal allocation rule.

Evaluation of treatment allocation

Allocation	*Share of 0 in %*	*Share of 1 in %*
All Policy Tree	100.0	0.0
All observed	91.05	8.95
All random	91.13	8.87
All best ATE	100.0	0.0
Switchers Policy Tree	100.0	0.0
Switchers random	50.25	49.75

Switchers best ATE	100.0	0.0

Note: Allocation analysed is the SAME as the one obtained from the training data.

Evaluation of treatment allocation

Allocation	*Share of 0 in %*	*Share of 1 in %*
All Policy Tree	100.0	0.0
All observed	90.9	9.1
All random	91.33	8.67
All best ATE	100.0	0.0
Switchers Policy Tree	100.0	0.0
Switchers random	51.32	48.68
Switchers best ATE	100.0	0.0

Note: Allocation analysed is DIFFERENT from the one obtained from the training data.