Evaluated Effectiveness of Interventions for Individuals in Infelicitous Eventualities

An attempt at solving a pair of Causal Inference problems.

2100816

April 28, 2022

Registration number:	2100816
Project:	Causal inference - assignment 2
Link to GitHub:	https://github.com/11BelowStudio/ce888

Executive summary (max. 250 words)	128
Introduction (max. 600 words)	321
Data (max. 300 words/dataset)	353
Methodology (max. 600 words)	600
Results and Discussion (max. 1000 words combined)	877
Conclusions (max. 500 words)	470
Total word count	2749

Table 1: Word counts for each section.

Contents

Abstract

This document contains details on an attempted causal inference investigation into the the IHDP[1] [2] and JOBS[3][4][5] datasets.

This document briefly discusses these datasets, explains the methodology used to approach the causal inference problems, and the findings which were made along the way.

Regrettably, no meaningful conclusions regarding the central causal questions for these datasets were reached (although some mildly tangential conjectures are mentioned), and a brief discussion of reasons for these shortcomings, for purposes of identifying them so they can be avoided in the future, concludes this document.

Issues with the methodology used (such as the author's focus on implementation over results), and the general lack of forethought in the approach to this task have been identified as key contributing factors to the failure to produce meaningful findings.

1 Introduction

This project involves two datasets: the Infant Health and Development Program (`IHDP`)[1][2] and `JOBS`[3][4][5].

Both of these datasets contain information about individuals (`x`), whether or not the individuals received some 'treatment' (`t`), and a `y` outcome for the individuals. The task I have been given for these datasets is to find the causal relationships within these datasets, to assess whether or not the treatments (`t`) given to the individuals have had any effect on the outcomes (`y`).

To evaluate this, there are a series of preliminary steps which must be taken.

Firstly, as stated by Fernández-Loría and Provost, it is vital to first be able to estimate `y`, given `x` and `t`[6], as, without the ability to model the known, factual, outcomes, we cannot start to consider how we can simulate the counterfactual outcomes.

Then, to help minimize bias, the data must be balanced (between treatment and control groups), ideally via Inverse Propensity Score Weighting (based on a modelling of whether or not a sample would be likely to be assigned to the treatment or to the control group), such as is discussed by Mitchell et. al.[7].

Once we have achieved this, we can then start trying to work on the overarching task of 'causal inference'; or, in other words, working out if (and how) outcomes may be affected by certain 'causal variables', in turn allowing us to, as described by Hill and Stuart, compare 'potential outcomes' given certain values for these causal variables[8].

This is what I intend to do with this task, specifically, by trying to work out if the 'causal variable' of the 'treatment' `t` in these datasets has an impact on the outcomes `y` for the individuals in these datasets, for purposes of assessing if the treatments are actually providing any benefit to the people that they are supposed to be helping.

The Python packages `scikit-learn`[9], `pandas`[10][11], `shap`[12], `numpy`[13], and `econml`**`econml`** were used to produce this research.

2 Data

2.1 High-level overviews

Both datasets contain `x` background information (all numeric, various scales), factual treatment `t` (0 or 1), and factual `y` outcome data (`yf` in `IHDP`, `y` in `JOBS`). This permits all evaluations of `yf` predictions given `x` and `t`, along with `t` predictions given `x`. `JOBS` contains experimental and observational data, with experimental samples indicated in an `e` column. `IHDP` does not explicitly contain this; however, `IHDP` only contains experimental data, meaning that we can consider `e=1` for all of it. This permits 'absolute treatment effect on the treated' and 'policy risk' evaluations on both datasets.

2.2 IHDP - The Infant Health Development Program

`IHDP` comes from a study into the effects of providing additional support to families of premature babies on the development of the aforementioned babies, via IQ tests[1][2]. The `yf` values from this dataset come from this study, whilst the counterfactual `ycf` values were simulated. These `y` values are continuous, meaning that a regression approach would be appropriate for this dataset. From the abstract given for this dataset, the data collection strategy was somewhat comprehensive; however, I do not know if these features are all present in the provided IHDP data. It appears that t assignments were random, but stratified on birth weight, but children with a weight over 2.5kg were ineligible for treatment[1]. None of the provided features appear to indicate birth weight directly, so this may be an unknown confounder.

2.3 JOBS

JOBS consists of data regarding job-seekers and their success in finding jobs, with the treatment t being whether or not an individual was provided with support in their job search[3][4][5]. This data is a combination of observational (e=0) and experimental (e=1) data; however, only individuals in the experimental group were potentially able to receive the treatment, potentially working as a confounder. Additionally, in this study, the treatment was randomly assigned, but only to *qualified* individuals who applied to the program[3]; these barriers to entry, despite being minor, will have had a somewhat confounding effect on treatment assignment, and may have an impact on the outcomes as well. As the outcome y for this dataset is binary (1 or 0), a classification approach is appropriate.

3 Methodology

This research followed a rather simple methodology, nearly identical for each dataset (besides a few specifics). A RNG seed of 42 is used for everything that uses a random seed, for consistent experimental results[1].

3.1 Loading the data

The data pre-preprocessing can be seen in the `datasets_to_csv.ipynb` notebook. This merely reads the datasets (presently in .npz) format, and converts them to .csv files, with labelled x column names, an extra tcf column (indicating the opposite of the treatment received by the individual), and. for the IHDP dataset, not only does the produced csv have a counterfactual y column, it also has a t0 and t1 column, holding the measured/simulated y outcome for the t=0 and t=1 case[2]

3.2 Train/test splits

IHDP uses 10% of the factual data in a held-out validation set, whilst JOBS uses 20%. This is selected via stratification; in IHDP, this is stratified based on t, but in JOBS, this is stratified based on y, e, and t. IHDP uses a smaller sized factual validation set due to the smaller dataset size and due to counterfactuals all being in the validation set.

3.3 non-CATE learners

All of the non-CATE learners use `HalvingGridSearchCV` (with 10-fold cross-validation) to find the optimal hyper-parameter configurations to maximise R2 scores on the training set (receiver operating characteristic area-under-curve for classifiers), and are then compared to each other based on their performance for that metric on the test set. Each of these learners are in a pipeline with a `QuantileTransformer` ahead of them, to ensure that the inputs to the learners are scaled within a reasonable range. The learners with the optimal hyper-parameter configurations for each task are then compared against each other, to find the optimal learner.

Additionally, each of the optimally-configured learners have three versions of their feature importances plotted. There are bar graphs for the importances obtained by the `feature_importances_` properties of the learners which have them (using the `coeff_` property of the learners which don't have `feature_importances_`) as well as the importances returned by `scikit-learn''s` permutation_importance' method[9] (calculated on the test set). Furthermore, there is also a beeswarm plot, indicating the shapley

[1] editable within /a2_utils/seed_utils.py

[2] taking it from the appropriate y column, given

values (relative impact on the output which each feature has depending on its value, plotted per-sample) for the learner, via the `SHAP` library[12][3].

3.3.1 Learners for Y given XT

`RandomForestRegressor`, `ARDRegressor`[4], `SGDRegressor`[5], and `AdaBoostRegressor` instances[9] (with Adaboost boosting each of the aforementioned learners) are trained for this task[6].

3.3.2 Learners for T given X (IPSW learning)

Instances of `RandomForestClassifier`, `SGDClassifier`, and `AdaBoostClassifier`[9] (boosting the aformentioned `RandomForestClassifier` instances) are trained for this task. This is constrained by the requirement for these learners to have a `predict_proba` method for use with the propensity weight scoring method later on (hence the lack of `ARDClassifier`). In `JOBS`, these learners are only trained/ evaluated (still using 10-fold cross-validation) on a train/test set consisting of the `e=1` samples, as the `e=0` samples all have `t=0`.

3.3.3 Learners for Y given X and IPSW weights

Instances of `RandomForestRegressor`, `ARDRegressor`, `SGDRegressor`, and `AdaBoostRegressor`[9] (using the aforementioned instances) are trained, using the 'best' IPSW learner from the prior step to produce sample weights[7].

3.4 CATE (Conditional Average Treatment Effect) estimators

`CausalForestDML`, `ForestDR`, `DMLIV`, and `ForestDRIV`**econml** estimators are trained on the data, using the best `Y|X` (and `T|X` and `Y|XT`) learners produced earlier on in their construction. `IHDP` performance is assessed by Precision of Estimation of Heterogeneous treatment Effect, `JOBS` is assessed by the absolute Accuracy for Treatment effects on the Treated (due to no counterfactuals). Feature importances (shapley values[12]) are plotted for each `CATE` estimator, along with tree visualizations of how features affect the outputs, and policies for whether or not it would be best to treat/not treat an individual for the best outcome based on their `X` features[14].

4 Results and Discussion

4.1 IHDP

4.1.1 Simple Learners (Y given XT)

The learner with the best R2 score was the `RandomForestRegressor`, with an R2 score on the test set of 0.70. As can be seen from Figure 1, the features with the strongest impact on the outcome appear to be `t`, `x5`, and `x14`, in that order.

The lack of any influence from the other features is somewhat unexpected.

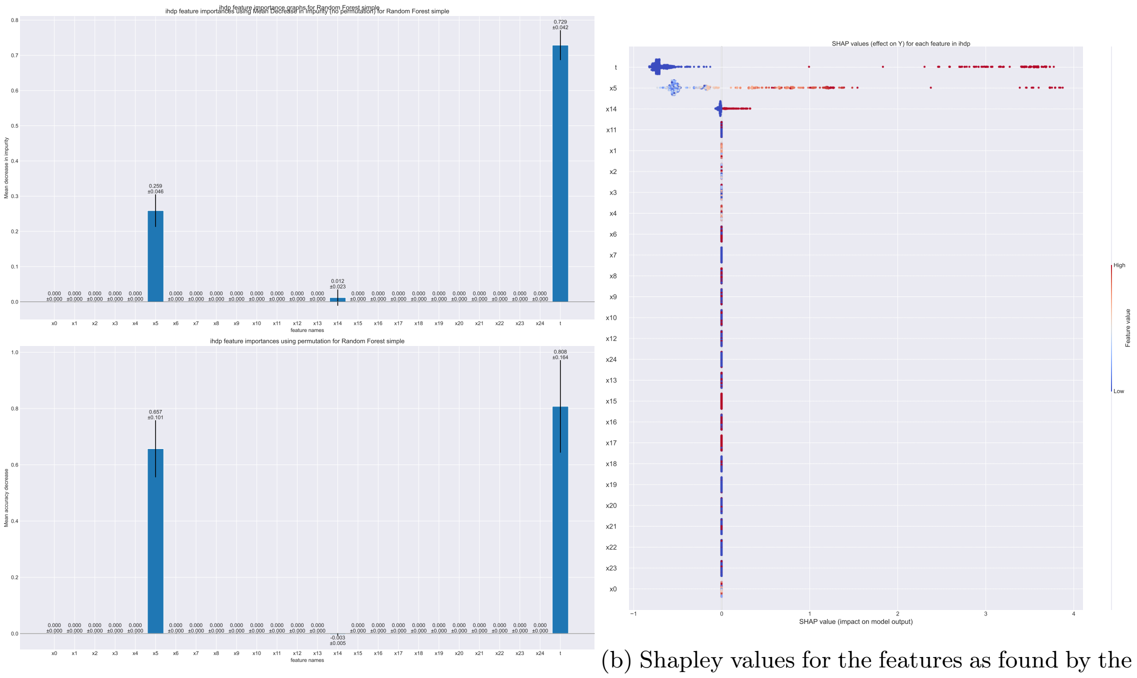
[3] In other words, it not only shows the importances of the features, but also how the higher/lower values of the features could have an impact on the output.

[4] Bayesian ridge regression with Automatic Relevance Detection (ARD)

[5] Stochastic Gradient Descent

[6] Classifier versions are used instead for the JOBS dataset.

[7] Classifier versions are used instead for the JOBS dataset.



(a) Feature importances (from feature_importances_ and permutation_test) for IHDP simple RandomForest

(b) Shapley values for the features as found by the IHDP Y|XT RandomForest predictor

Figure 1: Feature importances found by the IHDP Y|XT learner

Looking at Figure 1b, t=0 has a somewhat negative impact on the outcome, dwarfed by the positive effect of t=1. x5 has a less obvious boundary for when the impacts turn positive, however, higher x5 results in a higher y, whilst a lower x5 results in a somewhat lower y, but the negative effects are less dramatic than the positives. x14 appears to have no/negligibly negative impact on y when x14=0. (x14 is a binary value (only 0 or 1), so blue=0, red=1, with potential for a slightly less negligible positive impact when x14=1.)

4.1.2 IPSW learners (T given X)

The best estimator for T|X for IHDP was the SGD classifier, with a ROC AUC[8] score of 0.71 on the test set. The feature importances of this classifier (Figure 2) appear to be completely different to the IHDP Y|XT regressor, with x5 and x14 having rather low importances, whilst several other features all compete to have the highest importance, with wildly fluctuating standard deviations for the importance predictions visible in Subfigure 2a.

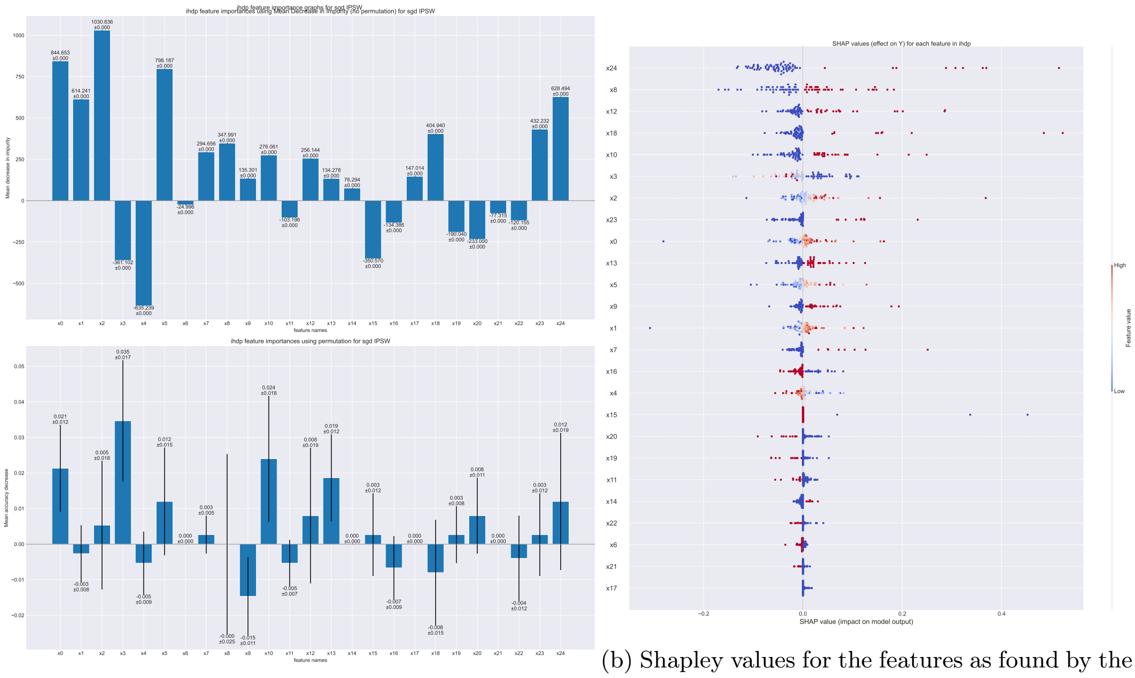
The shapley values (Subfigure 2b) are somewhat more coherent; for all of the binary features, values of 0 and 1 have opposing impacts on the final t value[9]. The continuous values are a bit more fuzzy in this regard, but they mostly follow this rule of 'bigger has opposite sign effect to smaller'. The only real exception is with x3; the high and slightly low values of x3 have negative impacts, and only the particularly small x3 values have a positive impact.

Considering this unusual overlap, the rather high importance granted to x3 in Subfigure 2a, and the fact that children with a birth weight above 2500g were not eligible for inclusion in the treatment group in the IHDP study [1], this suggests that x3 might indicate 'birth weight'[10].

[8] Receiver Operator Characteristic Area Under Curve

[9] By this, I mean that 'for the ones where xN=0 has a negative effect, xN=1 has a positive effect, and vice versa'

[10] With the higher x3 values resulting in a lower t due to the ineligibility for treatment. Then again, this is just conjecture



(a) Feature importances (from feature_importances_ and permutation_test) for IHDP IPSW SGD

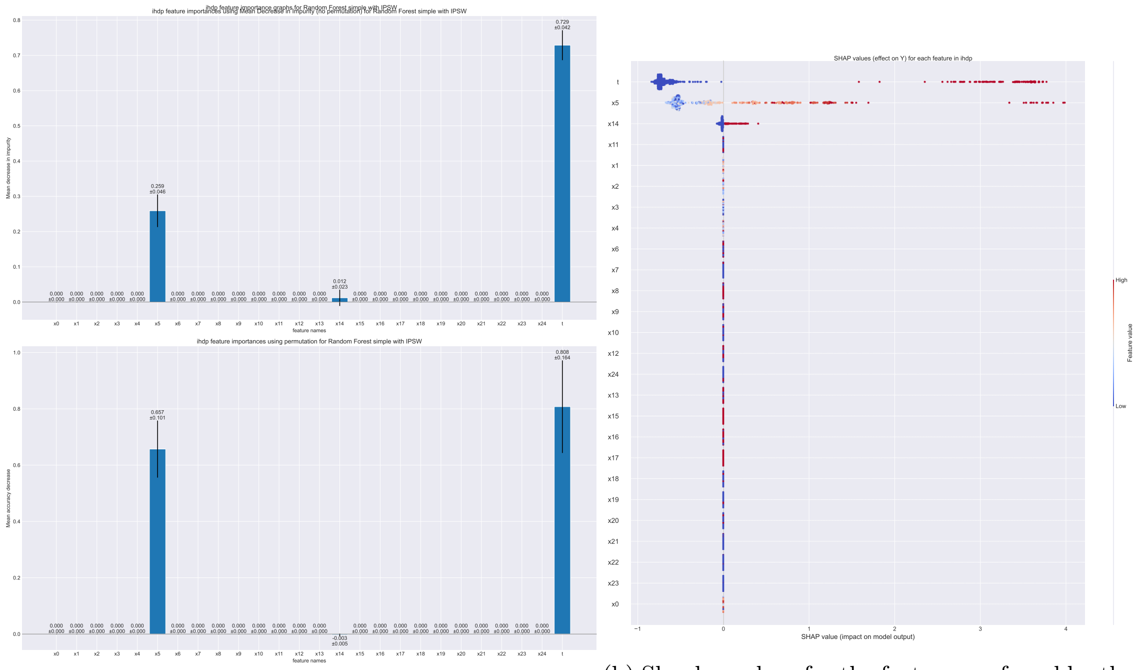
(b) Shapley values for the features as found by the IHDP T|X SGD predictor

Figure 2: Feature importances found by the IHDP T|X learner

4.1.3 IHDP simple learners with IPSW (Y given XT*IPSW(X))

Once again, the `RandomForestRegressor` had the best result (`r2` score of 0.70), and, once again, as shown in Figure 3, `t` is the most important feature, followed by `x5`, `x14` is questionably important[11], and then everything else being almost entirely unimportant.

[11] The estimator's 'feature_importances_' property gave it a somewhat positive result, but scikit-learn's 'permutation_test' functionality gave it a mostly negative result, but the shapley values still indicate some significance



(a) Feature importances (from feature_importances_ and permutation_test) for IHDP simple+IPSW RandomForest

(b) Shapley values for the features as found by the IHDP Y|XT*IPSW RandomForest predictor

Figure 3: Feature importances found by the IHDP Y|XT*IPSW learner

Seeing as the `T|X` predictor, shown in Figure 2, was used to help fit this predictor, I would have assumed that the importances of the inputs to `T` derived from that would have had some tangible impact on this learner's importances, but their lack of an impact is somewhat concerning. This either suggests underfitting, the impacts of `T` effectively outweighing the impacts of the contributors to `T`, or both. The most probable explanation would likely be underfitting, as the other 'best' learners (being ARDRegressor, SGDRegressor, and Adaboosted versions of those and RandomForest) all had somewhat more complex importance/shapley graphs, yet lower **`r2`** scores (everything but the RandomForest/Adaboosted Random Forest having r2 scores around 0.5), implying that they overfit to the point of becoming less accurate than this underfit regressor.

4.1.4 IHDP CATE estimator

The best CATE estimator for `IHDP` was the `CausalForestDML` estimator, with a PEHE[12] value of 4.8. However, looking at the outputs of it, I suspect that the somewhat underfit models produced in the previous stage, used as inputs to this model, may have resulted in this model not being fitted correctly either, resulting in the unusual results visible in Figures 4 and 5.

The shapley values for one of the other CATE models, which more closely resembles what one may have expected to see considering the inputs so far, can be seen in Figure 11, although that other model had an atrocious PEHE score, so that other model may be a case of garbage-in-garbage-out, with this model ultimately being more accurate overall.

The unexpected shapley values could be due to this model finding out that x2 and x0 were confounders, however, as the inputs to this model were massively underfit, I cannot confidently assert that to be the case.

[12] error in Precision of Estimation of Heterogeneous Effect

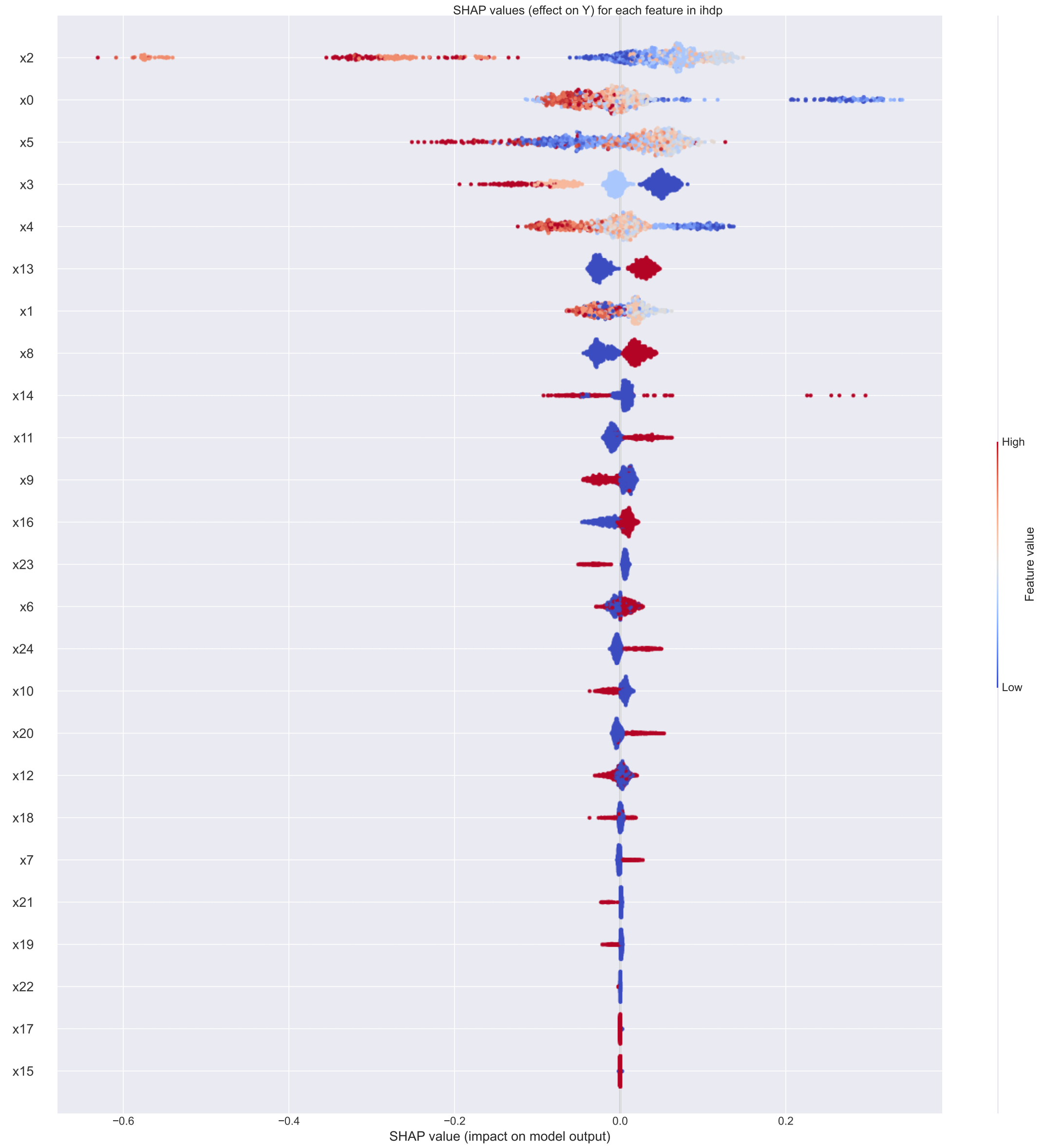
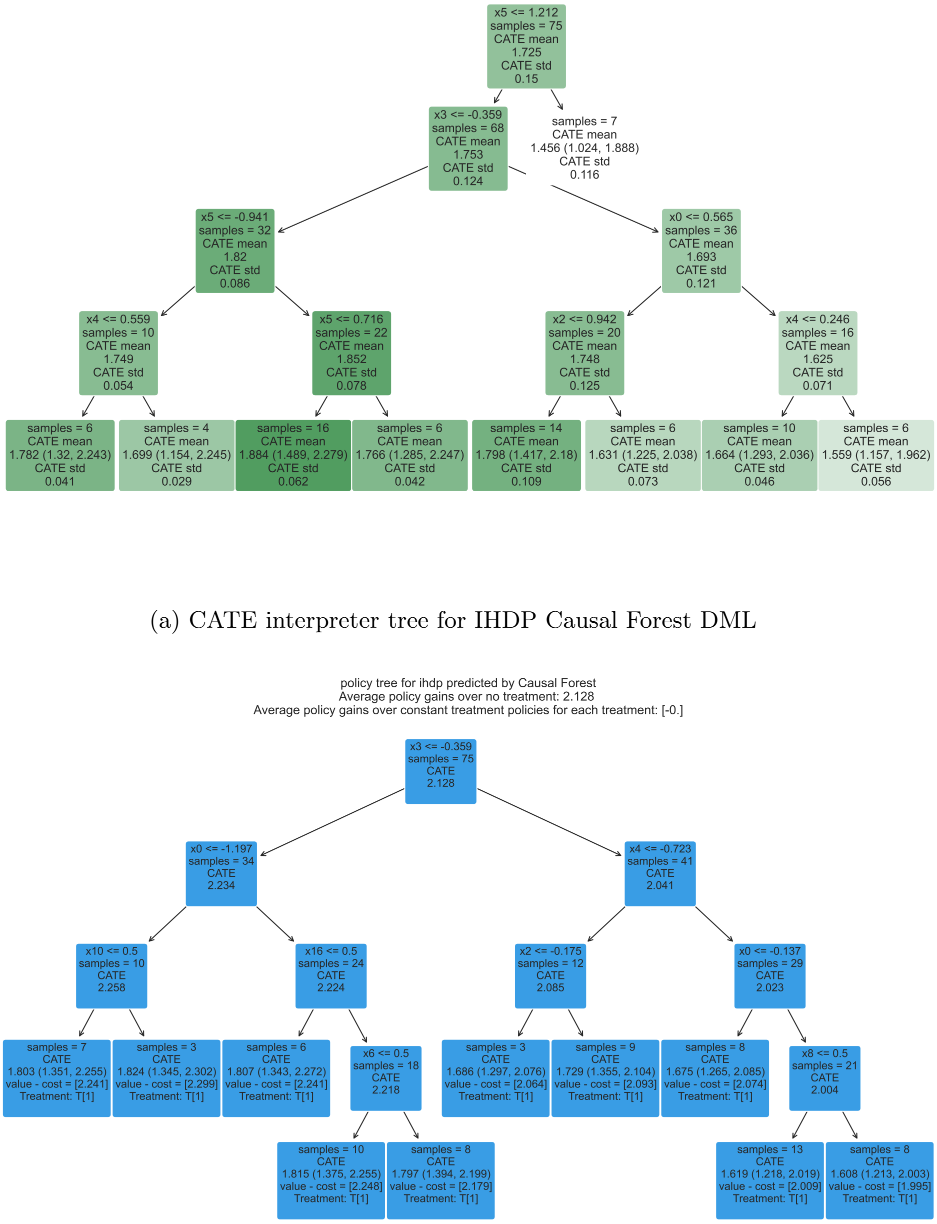


Figure 4: Shapley values for the features in IHDP Causal Forest DML

These shapley values are somewhat unexpected, considering the lack of any resemblance they have to the shapley values for the earlier predictors, especially the sudden importance given to x2. The impacts of x2 are also somewhat unusual. It appears that higher x2 values have a strong negative effect on y, with very low x2 having a negligible positive effect, whilst x2 values in the middle appear to have somewhat positive effects on y. This same pattern of effects can also be seen in x5 (median values->slightly positive, extreme high/low->somewhat negative) and x1.



(a) CATE interpreter tree for IHDP Causal Forest DML

(b) Policy interpreter tree for IHDP Causal Forest DML

Figure 5: Interpreter trees for the IHDP Causal Forest DML

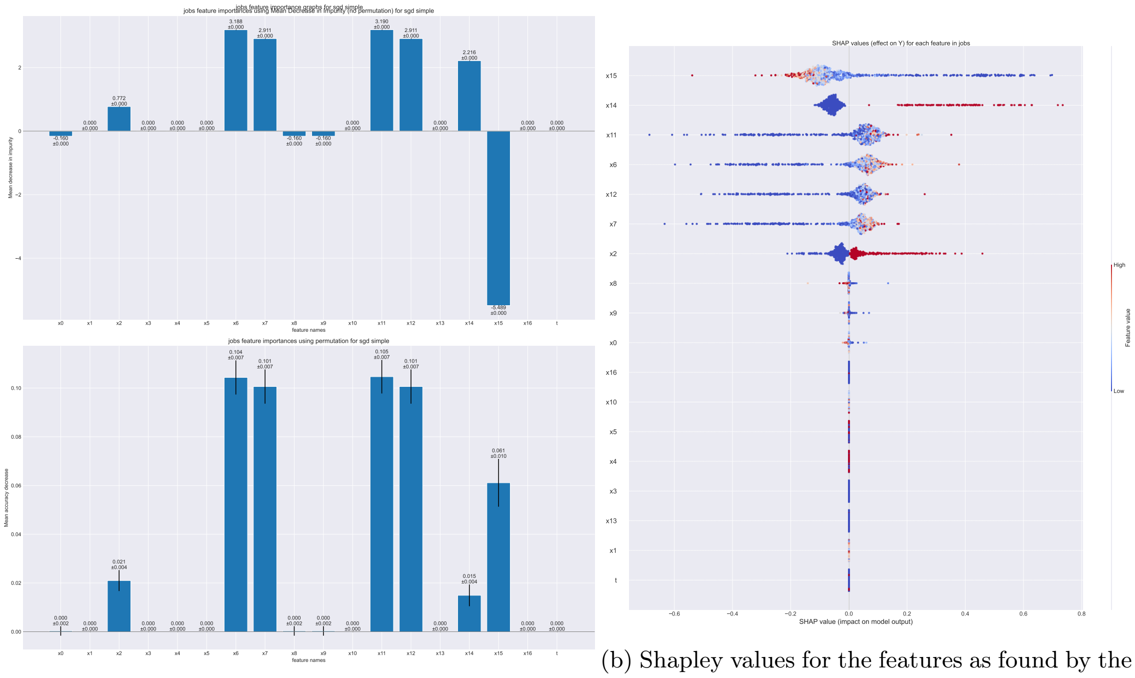
These trees are interpretations of the inner workings of the Causal Forest DML CATE estimator (with these trees being constructed by [14]), and all give the impression of a rather optimistic outlook on how the treatment effects the individuals involved. Granted, looking at the raw data for IHDP, there are very few cases where an individual has a worse outcome for t=0 than t=1, so the policy illustrated in 5b to always treat (only with the extent of the treatment benefit fluctuating) does make sense. This is also reflected in the CATE interpreter tree (5a), as, whilst there are a few outlier cases in the full dataset where the effect is negative, none of those samples are in the test set, therefore, once again, it's just the scale of the benefit that fluctuates.

4.2 JOBS

4.2.1 Simple Learners (Y given XT)

The learner with the best `ROC AUC` score was the `SGDClassifier`, with an `ROC AUC` score on the test set of 0.76. As can be seen from Figure 6, the features with the strongest impact on the outcome appear to be `x6` and `x11`, then `x7` and `x12`, whilst `t` has a negligible importance.

The lack of any influence from `t` is somewhat unexpected.



(a) Feature importances (from coeff_ and permutation_test) for JOBS simple SGDClassifier

(b) Shapley values for the features as found by the JOBS Y|XT SGDClassifier

Figure 6: Feature importances found by the Jobs Y|XT learner

Looking at Figures 6a and 6b, `t` appears have no impact on the `y` outcome for an individual, indicating that the treatment may not be having any impact at all. However, if we remember that JOBS is a combination of experimental and observational data, and that the relative size of the treatment group is particularly small (leading to the vast majority of the data being about untreated individuals), it's understandable how `t`, being only applicable to a minority, would not be seen as having any bearing on the outcome for the majority.

That said, it is interesting how, in 6b, the four 'most important' features, `x6`, `x7`, `x11`, and `x12` all have a shape in this graph somewhat reminiscent of a pike (or some other big stick with a big pointy bit at the end), specifically with how higher values lead to a higher `y`, then the most common 'lower' values also have a (lesser) positive `y` effect, with the `y` effect generally getting worse as these features degrade further. Features `x14` and `x15` have somewhat similar shapes, but reversed (`x14`: the lower values are the 'spike' and are negative, higher is the 'stick' and is positive; `x15`: literally just the reverse of `x6`,`x7`,`x11`,and `x12`). These similar distributions imply that there may be a causal link between these features.

4.2.2 IPSW learners (T given X)

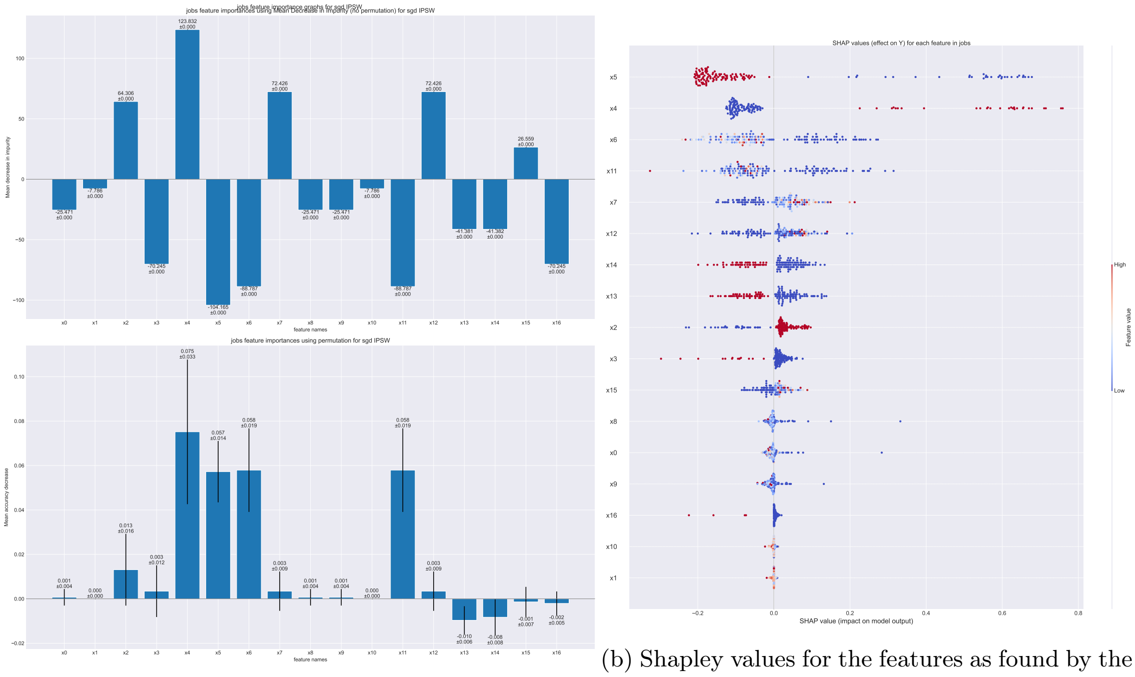
The best estimator for `T|X` for `JOBS` was also the SGD classifier, with a `ROC AUC`[13] score of 0.62 on the test set. The feature importances of this classifier (Figure 7) appear to be completely different to the IHDP `Y|XT` classifier, with `x5` and `x14` having rather low importances, whilst several other features all compete to have the highest importance, with wildly fluctuating standard deviations for the importance predictions visible in Subfigure 7a.

The shapley values (Subfigure 7b) are somewhat more coherent; for all of the binary features, values

[13] Receiver Operator Characteristic Area Under Curve

of `0` and `1` have opposing impacts on the final `t` value[14]. The continuous values are a bit more fuzzy in this regard, but they mostly follow this rule of 'bigger has opposite sign effect to smaller'. The only real exception is with `x3`; the high and slightly low values of `x3` have negative impacts, and only the particularly small `x3` values have a positive impact.

The rather extreme effects of `x5` and `x4` on `t`, combined with the consistent direction of the impacts, indicates that `x5` and `x4` could potentially be related to an individual's eligibility to have been included in the treatment group (whether or not that it would have lead to the individual being treated is somewhat questionable). However, it's also possible that the other features where the high/low values for that feature were all present on the same side as each other could have been related to eligibility.



(a) Feature importances (from feature_importances_ and permutation_test) for JOBS IPSW SGD

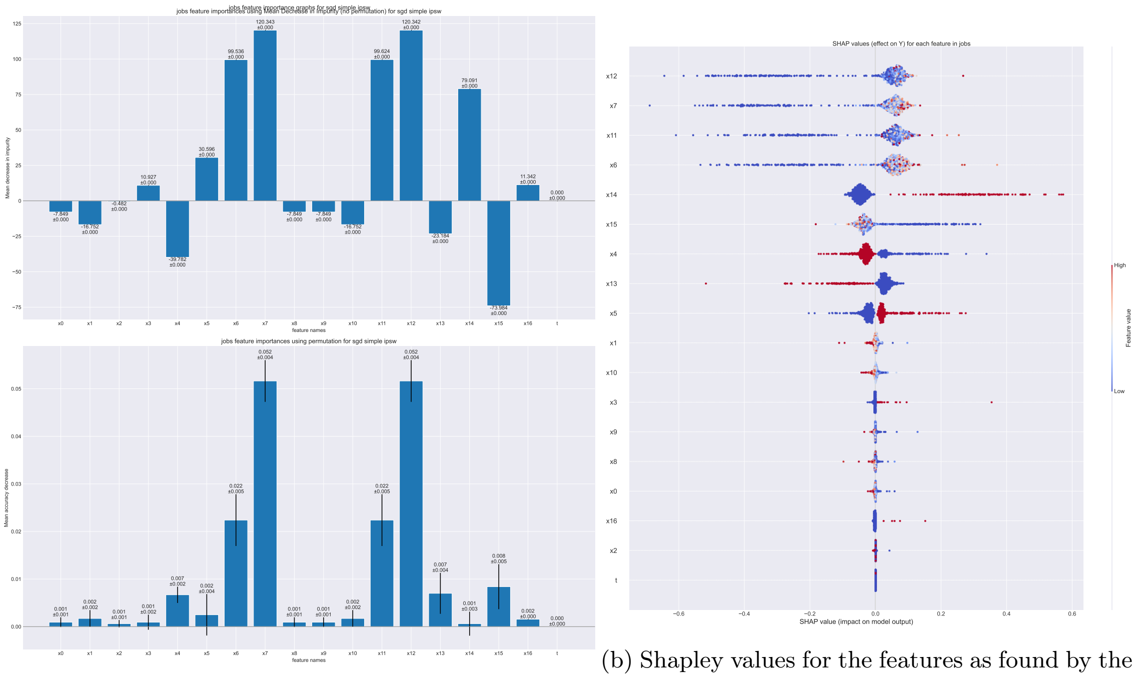
(b) Shapley values for the features as found by the IHDP T|X SGD predictor

Figure 7: Feature importances found by the JOBS T|X learner

4.2.3 JOBS simple learners with IPSW (Y given XT*IPSW(X))

Once again, the `SGDClassifier` had the best result (`roc_auc` score of 0.76), and, once again, as shown in Figure 8, `x6`, `x7` ,`x11`, and `x12` have once again been identified as the most important features, with `t` still being classed as unimportant (despite the IPSW scores being used to apply sample weights during the training process), presumably due to the same issues regarding the lack of samples with `t=1` for `t` to be considered to have had an impact.

[14] By this, I mean that 'for the ones where xN=0 has a negative effect, xN=1 has a positive effect, and vice versa'



(a) Feature importances (from feature_importances_ and permutation_test) for JOBS simple+IPSW SGD

(b) Shapley values for the features as found by the JOBS Y|XT*IPSW SGD predictor

Figure 8: Feature importances found by the JOBS Y|XT*IPSW learner

Seeing as the `T|X` predictor, shown in Figure 7, was used to help fit this predictor, it would have made sense if any of the other features relevant to the training of `T|X` would have been considered to have had more of an impact for this estimator. However, the outcome of this is, for the most part, functionally identical to the outcomes of 6.

4.2.4 JOBS CATE estimator

The best CATE estimator for `JOBS` was the `CausalForestDML` estimator, with an ATT[15] value of 0.349. Once again, however, I suspect that it may not be fitted properly, due to the rather unexpected shapley values and questionable tree interpretations (see Figures 9 and 10).

[15] Absolute error in Average Treatment effect on the Treated

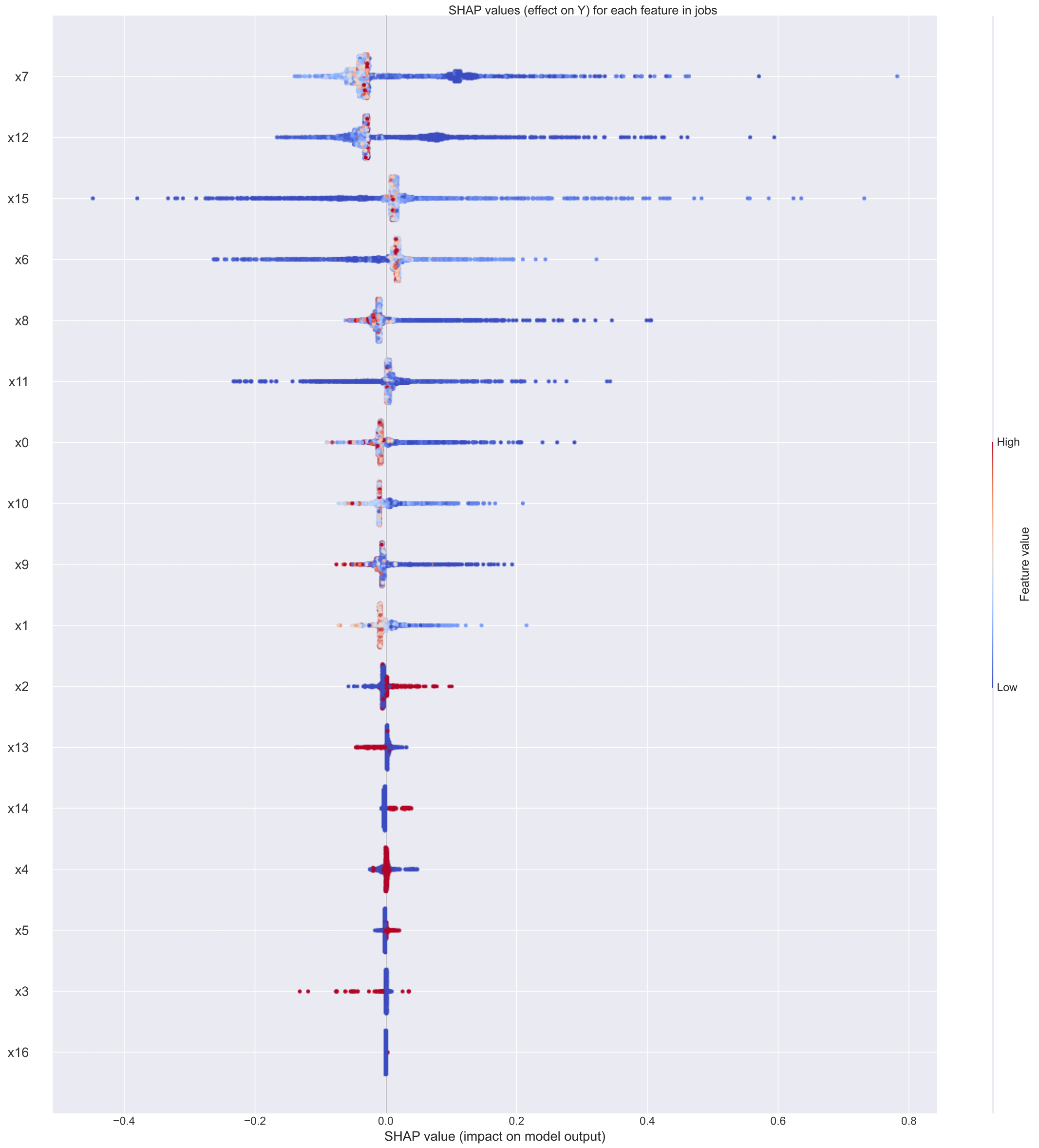
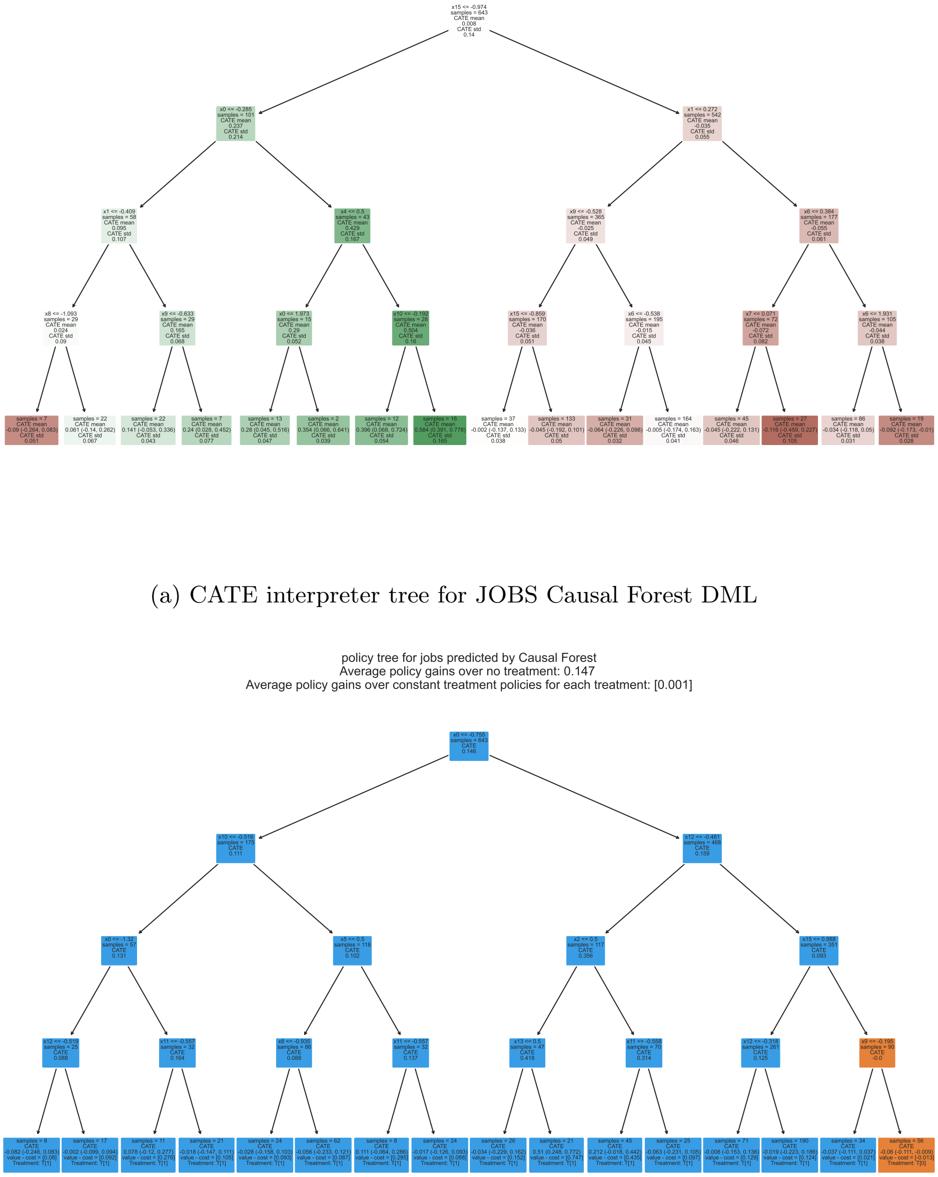


Figure 9: Shapley values for the features in JOBS Causal Forest DML

These shapley values are somewhat unexpected, considering the lack of any resemblance they have to the shapley values for the earlier predictors, and the seemingly nonsensical mixture of high values and low values on either end of the `increase y` and `decrease y` sides of the chart, implying the presence of a confounder, causing these seemingly completely arbitrary `y` outcomes for somewhat similar `x` values. However, from this graph, it appears that, if one wants to boost their chances of getting employed, they need to ensure that their `x12`, `x0`, and `x8` isn't incredibly low (bringing that to the `+y` side), but to drastically lower one's `x15` and `x6` values (bringing those firmly into the `+y` side as well)


x15 <= -0.974
samples = 643
CATE mean
0.008
CATE std
0.14
policy tree for jobs predicted by Causal Forest
Average policy gains over no treatment: 0.147
Average policy gains over constant treatment policies for each treatment: [0.001]
x0 <= -0.755
samples = 643
CATE
0.146

(a) CATE interpreter tree for JOBS Causal Forest DML

(b) Policy interpreter tree for JOBS Causal Forest DML

Figure 10: Interpreter trees for the JOBS Causal Forest DML

These trees are interpretations of the inner workings of the Causal Forest DML CATE estimator (with these trees being constructed by [14]).

These trees both appear to suffer from poor fitting. The CATE tree in 5a appears mostly normal (with a somewhat helpful indicator of the conditions where the treatment may have a negative effect on the treated), however, the projections of CATE values far outside of the true `y` range of `0-1` indicates that the CATE prediction model may be somewhat underfit (making assumptions of CATE values that may not actually be possible). Despite this, it does still provide an indication of which samples might respond negatively to treatment.

The policy tree in 5b is either underfit, rather poorly-thought out, or both (or I'm just misreading it), due to how it advises treating individuals with a seemingly negative CATE (and therefore would probably not benefit from the treatment), instead of going further to divide the leaf nodes further into clear 'treat' or 'don't treat' leaves.

5 Conclusions

This project has not been a success. Neither investigation produced any meaningful results, mostly due to the poorly-thought out methodology I used to approach this task.

Whilst the program structure used may, in theory, be applicable to a normal machine learning problem, and the requirements of having simple estimators, IPSW estimators, Simple+IPSW weighting estimators, and CATE estimators were met, the initial aims for this project were not met. I spent too long working out the particular program architecture used to get from the start to the point where the causal inference was supposed to happen, and, only after I had finally ran everything from the simple estimators to the CATE estimators (with all the graphs produced along the way), I realized that I had completely neglected to consider how to extract the actual answers from the data.

Certain aspects of my approach reeked of 'seemed like a good idea at the time', prompting further delays to reaching that point of realization. For example, I made a major blunder with the training set/test set. Whilst, yes, I did keep the full training data and the validation data completely distinct, and did not use any of the validation data (or the counterfactual data) within the training process, (relying on 10-fold cross validation for the training), I used a single validation set repeatedly for all of the validation (including the validation when working out which model was the 'best' and worthy of being used for the later modelling stages), meaning that the produced models were effectively indirectly trained on the validation set. Furthermore, the usage of passing the seed, `42`, as a number to every `scikit-learn` method which could be given a seed, instead of passing a `RandomState` with a seed of 42 down the callstack, may have lead to some stagnation in the models, such as by how all of the simple models were trained on the exact same sequence of KFold indices every time. Of course, there are still merits to passing in a number instead when consistency is important for reproducability of results, but defaulting to a RandomState from a specified start position each time would still be reproducable.

On a less negative note, the `.py` files in `assignment.a2_utils` are somewhat reusable (with it being somewhat feasible for some of these classes to be adapted to more general data science tasks, potentially allowing some time to be saved with setting up some of the code for these tasks). However, the glaring omission of anything that can actually be used to perform any meaningful causal analysis will need to be addressed.

To summarize, due to my poor approach to and mismanagement of my work on this task, the only conclusion I can make is that I have not made any findings significant enough to warrant any meaningful conclusions.

A Appendix: Further visualizations of note

These are some supplementary figures, not intended to replace the main content, but merely to supplement them.

A.1 IHDP CATE

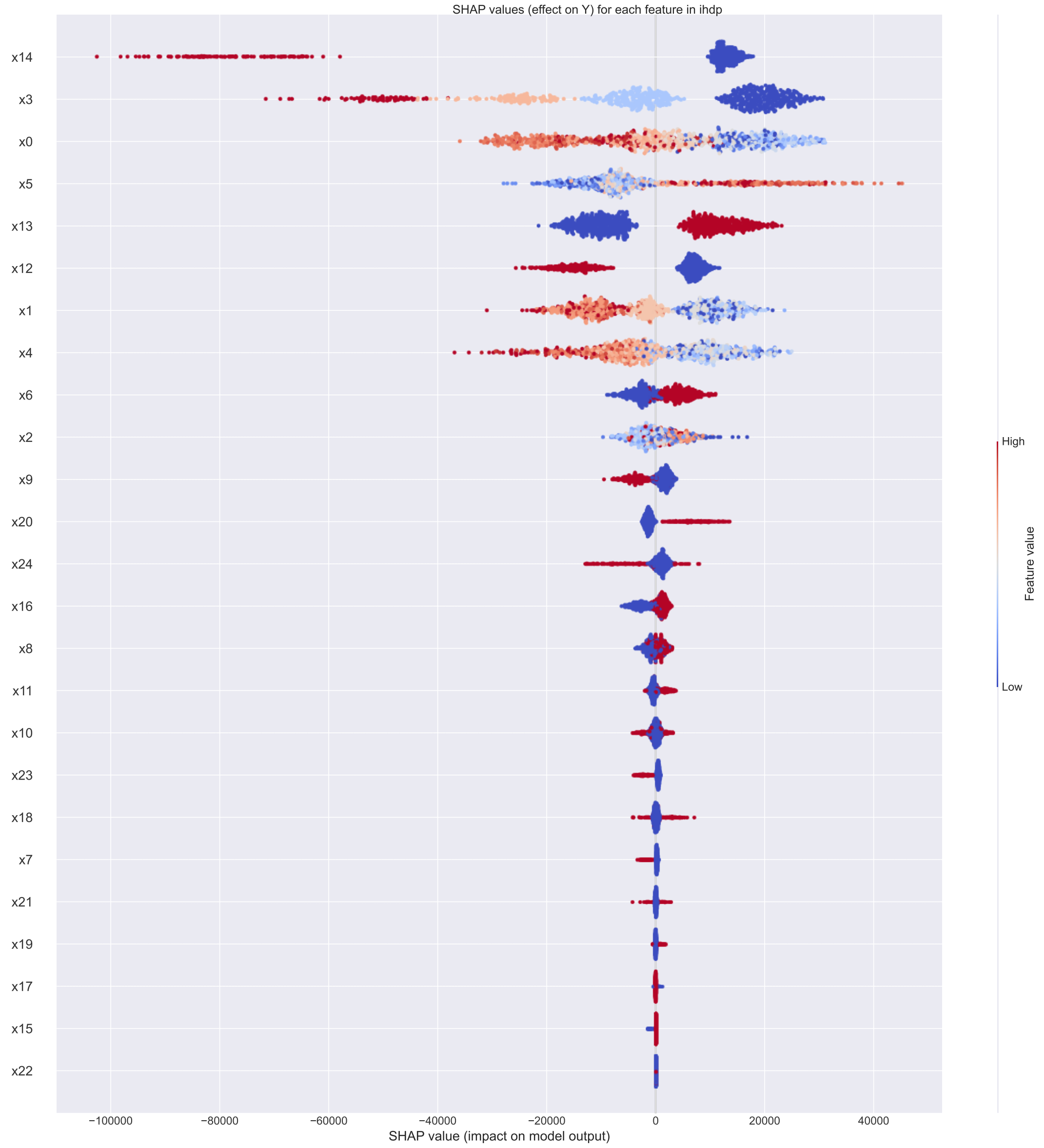


Figure 11: Shapley values for the IHDP ForestDR CATE estimator

These shapley values for the ForestDR learner appear to more closely resemble what one would anticipate to see from the CATE estimators for IHDP, considering the inputs found along the way, significantly moreso than the CausalForestDML results, visible in Figure 5 Considering the fact that the Y estimators (somewhat illustrated by Figures 1 and 3) granted some semblance of importance to x14, it makes sense that it might be seen as having some potentially drastic impact on the predictions made by this estimator, which uses those estimators as part of its inputs. Additionally, x5 (also identified as important by those predictors) is also shown as having relatively high importance. Furthermore, x3 (indicated as a key contributor towards `t` by the estimator illustrated in Figure 2) is also shown as being rather important in this estimator, as expected.

However, in practice, these shapley values are from an estimator with a PEHE (Precision of Estimation of Heterogeneous Treatment Effects) value of 68332.93913561162.

PEHE is supposed to be minimized.

Which is another way of saying 'these shapley values are utterly incorrect in practice'. This either means that the learner itself was poorly fitted (garbage-in-garbage-out), or it could mean that there was a rather large unknown confounder interfering with the outputs of this learner, but, considering the rather underfit learners which lead up to this point, it is most likely the former (garbage-in-garbage-out), not the latter.

References

[1] R. T. Gross *et al.*, *Infant health and development program (ihdp): Enhancing the outcomes of low birth weight, premature infants in the united states, 1985-1988*, 1993. DOI: 10.3886/ICPSR09795.v1. [Online]. Available: https://doi.org/10.3886/ICPSR09795.v1.

[2] J. Brooks-Gunn, F.-r. Liaw, and P. K. Klebanov, "Effects of early intervention on cognitive function of low birth weight preterm infants," *The Journal of Pediatrics*, vol. 120, no. 3, pp. 350–359, 1992, ISSN: 0022-3476. DOI: https://doi.org/10.1016/S0022-3476(05)80896-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022347605808960.

[3] R. J. LaLonde, "Evaluating the econometric evaluations of training programs with experimental data," *The American Economic Review*, vol. 76, no. 4, pp. 604–620, 1986, ISSN: 00028282. [Online]. Available: http://www.jstor.org/stable/1806062.

[4] R. H. Dehejia and S. Wahba, "Propensity score matching methods for non-experimental causal studies," National Bureau of Economic Research, Working Paper 6829, Dec. 1998. DOI: 10.3386/w6829. [Online]. Available: http://www.nber.org/papers/w6829.

[5] J. A. Smith and P. E. Todd, "Does matching overcome lalonde's critique of nonexperimental estimators?" *Journal of Econometrics*, vol. 125, no. 1, pp. 305–353, 2005, Experimental and non-experimental evaluation of economic policy and models, ISSN: 0304-4076. DOI: https://doi.org/10.1016/j.jeconom.2004.04.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030440760400082X.

[6] C. Fernández-Loría and F. Provost, *Causal decision making and causal effect estimation are not the same... and why it matters*, 2021. arXiv: 2104.04103 [stat.ML].

[7] J. D. Mitchell, B. F. Gage, N. Fergestrom, E. Novak, and T. C. Villines, "Inverse probability of treatment weighting (propensity score) using the military health system data repository and national death index," en, *J. Vis. Exp.*, no. 155, Jan. 2020.

[8] J. Hill and E. A. Stuart, "Causal inference: Overview," in *International Encyclopedia of the Social Behavioral Sciences (Second Edition)*, J. D. Wright, Ed., Second Edition, Oxford: Elsevier, 2015, pp. 255–260, ISBN: 978-0-08-097087-5. DOI: https://doi.org/10.1016/B978-0-08-097086-8.42095-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080970868420957.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[10] T. pandas development team, *Pandas-dev/pandas: Pandas*, version latest, Feb. 2020. DOI: 10.5281/zenodo.3509134. [Online]. Available: https://doi.org/10.5281/zenodo.3509134.

[11] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[12] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[13] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[14] S. M. Lundberg, G. Erion, H. Chen, *et al.*, "From local explanations to global understanding with explainable ai for trees," *Nature Machine Intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.