BART for Prediction and Causal Inference

Jeremy Lee

May 7, 2025

Contents

1 Summary

In this project, we apply Bayesian Additive Regression Trees (BART) to a dataset consisting of used cars sold in the United States. The goal is to evaluate BART's performance as a prediction method as well as for causal inference. To do so, we split the project into two major portions.

The first portion is a prediction task. Here we train a BART model to predict the price of a used car and compare its performance to a linear regression baseline. We find that BART offers significant improvements in RMSE and R^2 score, making it a viable method for regression. Additionally, we use cross-validation to select hyperparameters for a second BART model. This model shows further improvements over the first model, and we are able to draw conclusions about the optimality of hyperparameter values.

The second portion is the application of BART to causal inference. BART has been noted for its suitability to causal inference tasks (Hill et al, 2020), and we test this by using BART to evaluate the causal effect of odometer mileage on price. We apply two methods: a counterfactual method using the trained BART models from the prediction task and a binary treatment method as described by Hill, 2011. Both methods result in causal effect values that provide insight into the data.

2 Project Description

The dataset for this project is a used cars dataset referenced in the Appendix. It contains 426,880 records of used cars sold on Craigslist in the United States, each with 26 features. This is a large dataset, so during the cleaning and preprocessing step, we favor removing problematic observations completely rather than modifying them. This simplifies the analysis and reduces the size of the dataset, which is still large enough for useful inference.

We begin by selecting only the relevant features. There are eleven relevant features total: year, manufacturer, fuel, odometer, title status, transmission, paint color, state, latitude, and longitude. We then remove all entries with missing values and all entries with price or odometer values less than 1000. We also use the interquartile range method to remove outliers. Finally, we remove all observations with unique categorical feature values. This is because for some features, like manufacturer or paint color, a single observation from a niche car manufacturer or unique paint color does not provide as much generalizable information and is more difficult to work with.

We use R for our implementation due to the language's robust selection of libraries for BART, described later. As a result, we are able to use R's factor encodings to create levels for categorical variables, such as an individual level for each manufacturer. This encoding works well with many R packages by default, including the ones we use for BART.

After preprocessing, the dataset has 231,475 observations with eleven features. We use an 80-20 train/test split to obtain a training dataset with 185,180 observations. The distributions of the more notable numerical features of the training dataset are shown in Figure 1. We observe

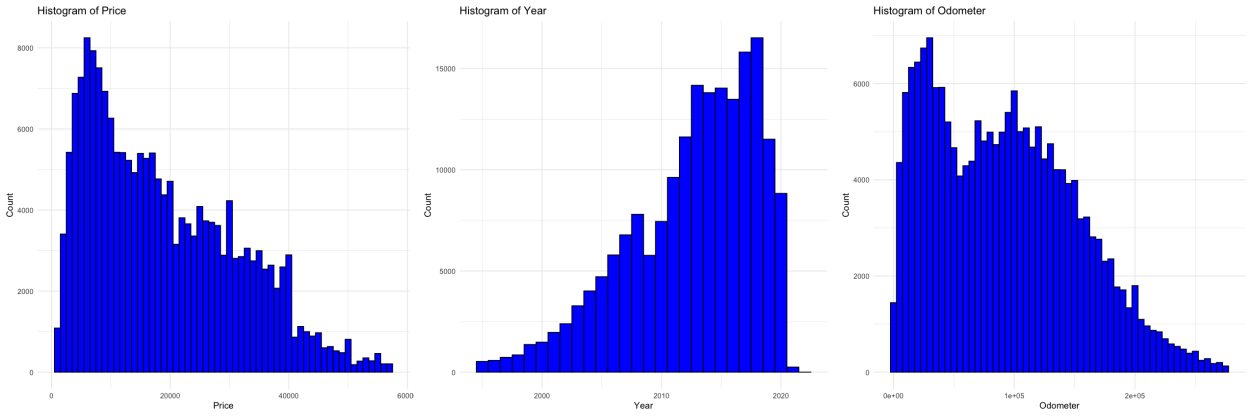

Figure 1: Numerical Feature Distributions

that price is centered around $10,000 with a long tail, and many of the cars in the dataset are newer, made between 2010 and 2020. Additionally, odometer mileage is spread out with two major peaks: one around 25,000 and one around 100,000. We do not transform the data, as "the sum-of-trees specification is adept at capturing both nonlinearities and interactions without the researcher having to explicitly add interaction terms or transformations of x or specify a limit on the level of interaction" (Hill, 2011).

We also observe feature correlations in order to confirm expected behavior. Correlations for the five numerical features are shown in Table 1.

	Price	**Year**	**Odometer**	**Latitude**	**Longitude**
Price	1.0000	0.6389	-0.6173	-0.0236	-0.0479
Year	0.6389	1.0000	-0.6834	-0.0569	0.0196
Odometer	-0.6173	-0.6834	1.0000	0.0498	-0.0216
Latitude	-0.0236	-0.0569	0.0498	1.0000	-0.0812
Longitude	-0.0479	0.0196	-0.0216	-0.0812	1.0000

Table 1: Feature Correlations

The correlations are as we would expect. Year and odometer are positvely correlated, since higher years indicate newer cars with lower mileage. Price is positively correlated with year and negatively correlated with odometer. Meanwhile, latitude and longitude are not significantly correlated with any other features.

One additional modification to the datset is used for the binary treatment model. Here we do not use the full dataset but rather three smaller subsets for comparison purposes. More details on this aspect are given in the next section when we discuss the binary treatment model.

For the implementation of the BART models, we use two R packages. The first is `dbarts`, which is a general purpose implementation of BART designed for efficiency. We use `dbarts` for the prediction models and the counterfactual model. The second package is `bartCause`, which implements the causal inference model described by Hill, 2011. We use this for the binary treatment model.

The overall structure of the project is as follows. First we train a standard BART model and cross-validated BART model using `dbarts`, as well as a linear regression model, to predict the prices of used cars. We evaluate and compare these models on the test set.

Next, we apply these trained models to counterfactual data and compute the average causal effect of 10,000 odometer miles on price, as described in the next section. The goal here is to answer the question of how driving a car for an additional 10,000 miles affects the price it can be sold for.

Finally, we train a separate binary treatment model using `bartCause`, since the structure of this model differs from the previous ones. We create a new binary variable that is equal to 1 if the odometer is over 100,000 and 0 otherwise. We view "odometer over 100,000" as a binary treatment, which makes Hill's method for causal inference applicable. We train the model on three smaller subsets of the data and estimate the average treatment effects, as described below.

3 Methods

3.1 Overview of BART

We now explain the basic structure of the BART model. BART was proposed by Chipman et al, 2007. This explanation and that of the next subsection largely follows Chipman et al, 2010. Bayesian Additive Regression Trees (BART) is a nonparametric learning method. It is similar to other ensemble methods, where the contributions of individual weak learners are combined to produce a final result. However, unlike some ensemble methods, BART is Bayesian; prior distributions are defined on several elements of the model, which are used to generate samples from a posterior distribution.

The BART model is a sum-of-trees model. For a tree T with terminal node values $M = \{\mu_1, \dots, \mu_b\}$, let $g(x; T, M)$ be the value $\mu_i \in M$ that T assigns to x. For a sum-of-trees model with m trees, the output Y is given by

$$Y = \sum_{j=1}^{m} g(x; T_j, M_j) + \epsilon.$$

Here the error ϵ is distributed as $\epsilon \sim N(0, \sigma^2)$ for some $\sigma > 0$. To predict the value of Y corresponding to input x, each tree T_j assigns a $\mu_{ij} \in M_j$ value to x. The sum of these values over all trees is the prediction for Y.

The model is fit by a backfitting Markov Chain Monte Carlo (MCMC) algorithm. Prior distributions are assigned to the trees T_j, the terminal node values $\mu_{ij} \mid T_j$, and the error standard deviation σ. The algorithm is similar to a Gibbs sampler, which successively draws (T_j, M_j) from the distribution

$$(T_j, M_j) \mid T_{(j)}, M_{(j)}, \sigma, y$$

and σ from the distribution

$$\sigma \mid T_1, \dots, T_m, M_1, \dots, M_m, y.$$

Here $T_{(j)}$ and $M_{(j)}$ are all trees and terminal values except T_j and M_j. More details on the specifics of this algorithm are given by Chipman et al, 2010.

The end result of the MCMC algorithm is a sequence of draws $y_1^*, \ldots, y_K^*$ from the posterior distribution. For this project, we use the mean of the K draws as the prediction for Y. Other values such as the median can also be used.

3.2 Hyperparameters and Priors

Next we describe the hyperparameters and priors of the BART model in more detail. There are three prior distributions that form the model. To simplify the model, we assume that the priors for each T_j and M_j are independent and symmetric. The priors are designed to regularize the model by reducing the impact of individual trees. For our default BART model, we use hyperparameter values based on the work of Chipman et al. and R's `dbarts` library. We also use five-fold cross-validation to select optimal parameter values for a second BART model, which we call XBART.

The first prior is the tree prior. The probability that a node at depth d is nonterminal is given by

$$\alpha(1+d)^{-\beta}$$

for base $\alpha \in (0,1)$ and power $\beta \in [0,\infty)$. This prior discourages trees with large depth, and changing the parameters changes the penalty for large depth. We use $\alpha = 0.95$ and $\beta = 2.0$ for our default BART model.

The second prior is for each terminal value on each tree: $\mu_{ij} \mid T_j$. It is given by

$$\mu_{ij} \sim N(0, \sigma_\mu^2),$$

where $\sigma_\mu = 0.5/k\sqrt{m}$ and m is the total number of trees. The hyperparameter k affects the variance σ_μ^2; increasing k reduces the variance of μ_{ij}, which shrinks the values toward zero. The number of trees m also affects the variance, in addition to its effect on the structure of the model. For our default model, we use $m = 75$ trees and $k = 2$.

The third prior is for the error variance σ^2. We use the inverse chi-square distribution

$$\sigma^2 \sim \frac{\nu\lambda}{\chi_\nu^2}.$$

We define a hyperparameter q and choose λ such that $P(\sigma < \widehat{\sigma}) = q$, where $\widehat{\sigma}$ is the least squares estimate of σ. Thus the pair of hyperparameters (ν, q) affects the error variance. We use a default $(\nu, q) = (3.0, 0.90)$.

Finally, we use a fixed number $K = 1000$ draws from the posterior distribution.

3.3 Cross-Validation

For the XBART model, we leave k, ν, and q at their default values and use five-fold cross-validation to select the values of m, α, and β. In particular, we choose values $m \in \{50, 75, 200\}$, $\alpha \in$

$\{0.90, 0.95, 0.99\}$, and $\beta \in \{1.5, 2.0, 2.5\}$ that minimize RMSE. The choice of which hyperparameters to leave fixed is motivated by the goal of testing how tree structure affects performance. The hyperparameter m is the number of trees, and the hyperparameters α and β affect how deep trees can grow. Focusing on these hyperparameters allows us to determine how the number and depth of trees interact with each other and affect the model performance, regardless of the terminal node values and error variance.

Figure 2 is a heatmap of all 27 hyperparameter combinations and the resulting RMSE computed through cross-validation. We see that the lowest RMSE is achieved by the model with $m = 200, \alpha = 0.99$, and $\beta = 1.5$. We use these values for the XBART model.

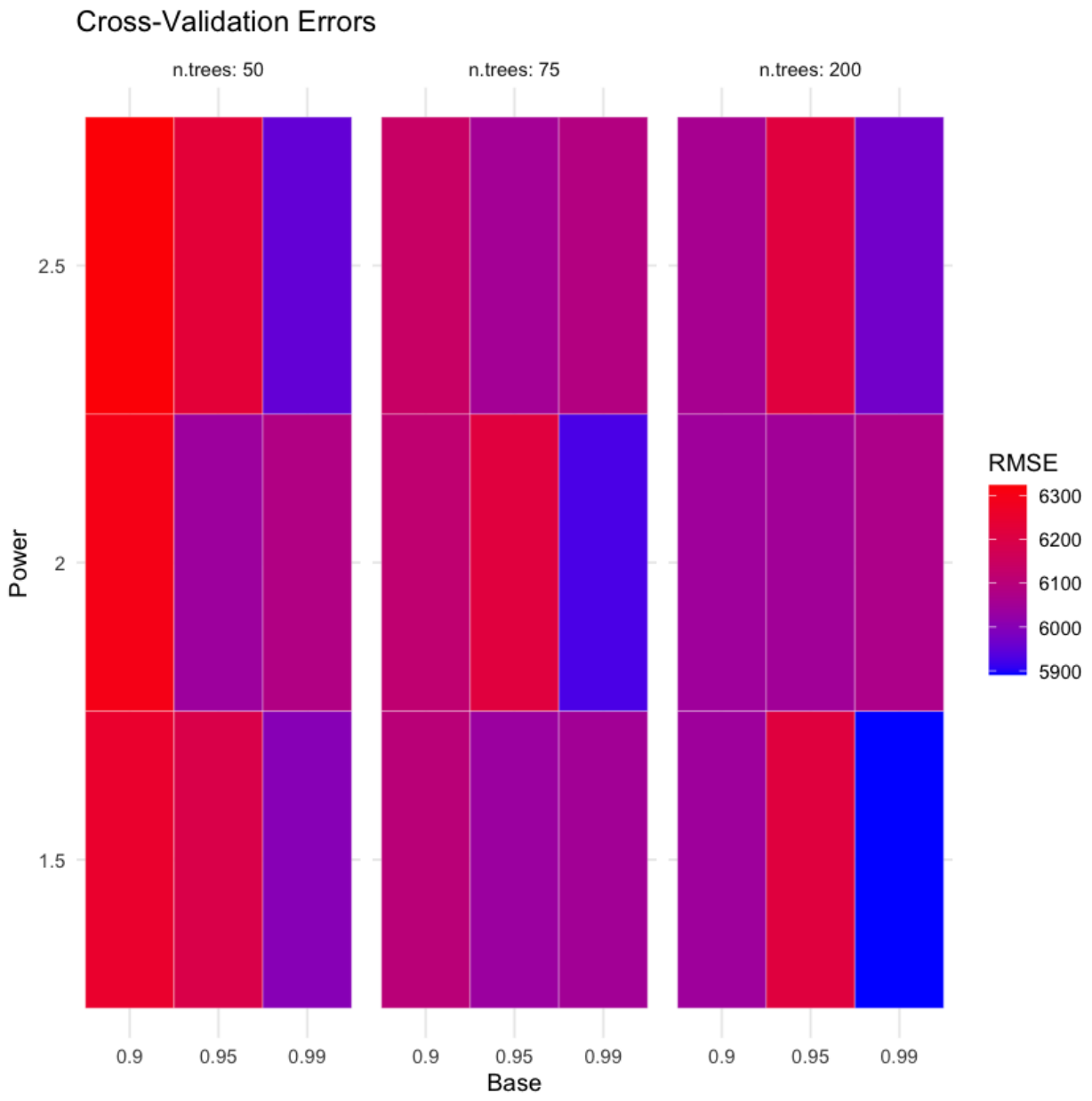

Figure 2: Cross-Validation Results

More generally, we see a trend on the diagonal of the heatmap. It appears that 0.99 is consistently the best value of α, and the optimal value of β decreases as the number of trees m increases. This suggests that, for this particular dataset, a fairly complex model with many deep trees is more appropriate than a smaller, simpler model.

Table 2 shows the two BART models and their hyperparameter values.

	BART	**XBART**
m	75	200
α	0.95	0.99
β	2.0	1.5
k	2	2
(ν, q)	(3.0, 0.90)	(3.0, 0.90)
K	1000	1000

Table 2: Model Hyperparameters

3.4 Prediction Task

The prediction task is straightforward. We train the BART, XBART, and linear regression models on the training set to predict the price of a car. As mentioned before, the BART models' predictions use the posterior mean. We then evaluate RMSE, MAE, and R^2 scores on the test set and compute metrics for feature importance that are described in the Results section.

3.5 Counterfactual Model

The goal of the counterfactual model is to determine whether driving a car for an additional 10,000 miles causes its value to drop, and by how much if so. We create a counterfactual test set by adding 10,000 miles to the odometer value of each observation in the original test set. Then we use the trained BART and XBART models from the prediction task to predict the prices of the counterfactual test set.

For each observation in the test set, we now have two predictions: the prediction for the original data that was computed in the prediction task, and the prediction for the new counterfactual data. The causal effect for each observation is the difference between the new prediction and the old prediction. The average causal effect is the mean across the entire test set.

Using this method, we obtain average causal effect values from the BART and XBART models. These values can be used to answer the original question; for example, an average causal effect of 500 implies that driving a car for an additional 10,000 miles will cause its price to decrease by $500 on average. We compare these values to the coefficient of the odometer variable in the linear regression model, which also provides an estimate of causal effect.

3.6 Binary Treatment Model

The usage of BART for binary treatment models is described by Hill, 2011. Here we create three subsets of the data, with the goal of estimating treatment effects on groups of similar observations. The three subsets are Fords sold in California, Fords sold in Florida, and Fords sold in Texas. By restricting the data to the most common car manufacturer and three most common states, we aim to estimate more accurate treatment effects without the inherent noise of dissimilar observations.

Since this method requires a binary treatment variable, we create a binary feature that is equal to 1 if the odometer is over 100,000 and 0 otherwise. The question to be answered here is whether an odometer over 100,000 causes the price of a used car to drop and by how much. We use the default hyperparameter values for `bartCause` and train the model to estimate the average treatment effect (ATE) as explained by Hill. Note that we are using ATE rather than average causal effect because we are now working with a binary treatment.

We train this model on each of the three subsets explained above. For each, we obtain an estimate of the average treatment effect, as well as 95% credible intervals for the estimate. Similar to the counterfactual model, the average treatment effect has a clear interpretation; it is the average change in price caused by an odometer over 100,000 miles.

4 Results

4.1 Prediction Results

Table 3 shows the RMSE, MAE, and R^2 scores for the three models evaluated on the test set. We see that BART offers significant improvements over linear regression, likely because the non-parametric structure of the BART model is able to better adapt to the data than a linear model. Furthermore, the XBART model performs slightly better than the standard BART model and is the best model overall.

	RMSE	**MAE**	**R^2**
Linear	7120	5456	0.67
BART	6090	4452	0.76
XBART	5952	4322	0.77

Table 3: Prediction Results

We also check the distributions of the residuals, shown in Figure 3. All models have normal shaped residual distributions centered at 0. The distributions for BART are tighter than the distribution for linear regression, indicating better overall performance.

For a linear regression model, we can determine feature importances by examining the coefficients of the variables. BART is similarly interpretable, as we can compute the average number of times each variable is used in a tree's decision node. The average is taken over all posterior draws; in other words, an average value of 100 means that the variable was used in an

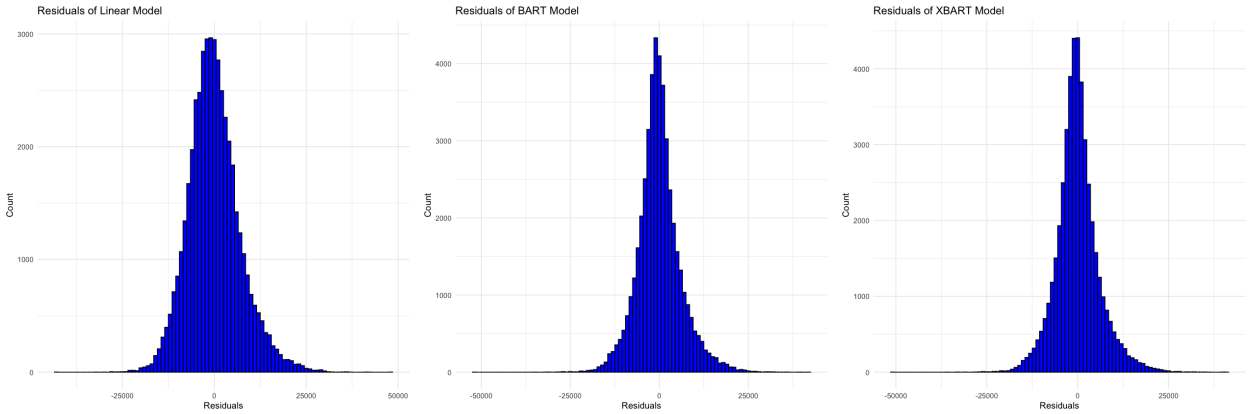

Figure 3: Distributions of Residuals

average of 100 decision nodes in each posterior draw of m trees. The top ten results for BART and XBART are shown in Table 4. Note that the XBART averages are greater than the BART averages because the XBART model has more trees in each posterior draw.

Feature	BART Average	XBART Average
Year	127.432	222.290
Odometer	115.308	209.748
Fuel.Diesel	74.220	138.850
Latitude	69.576	145.246
Longitude	68.490	133.926
Transmission.Automatic	67.864	120.624
Transmission.Other	66.322	128.276
Fuel.Gas	62.214	101.058
Manufacturer.Chevrolet	54.992	106.256
Manufacturer.Ford	51.746	103.126

Table 4: Feature Importances

The feature importances are similar for the BART and XBART models. In fact, the top ten are the same aside from minor differences in ordering. Year and odometer, which are intuitively the most important predictors of price, are weighted at the top across all models. The linear regression model also weights the features in Table 4 highly. Based on a t-test, almost all of the top ten features for the BART models have p-values less than 0.01 for their coefficients in the linear model. The one exception is longitude, which interestingly has $p = 0.22$ implying that it is not a significant feature for the linear model.

4.2 Counterfactual Results

The average causal effects of odometer on price obtained from both BART models are shown in Table 5, along with the linear regression coefficient for odometer for comparison purposes. BART and XBART give very similar causal effects of around -550, indicating that an additional

10,000 miles causes the price of a car to drop by about $550. This is lower than the linear regression coefficient of -695. Interestingly, the (absolute value of) the causal effects decreases in the same order that the models' accuracy increases. This indicates that worse performing models may overestimate the causal effect for this dataset.

	BART	**XBART**	**Linear**
Average Causal Effect	-558.58	-549.47	-695.41

Table 5: Average Causal Effect of Odometer on Price

We also examine the distribution of causal effects for the two BART models. Figure 4 shows the causal effects computed on each observation in the test set. Both models' distributions are predominantly negative and skewed toward zero, with a long tail in the negative direction and a much shorter tail in the positive direction. The distribution for XBART is slightly smoother than the distribution for standard BART, and it has more clearly defined peaks. This could indicate better accuracy or certainty about the overall causal effect.

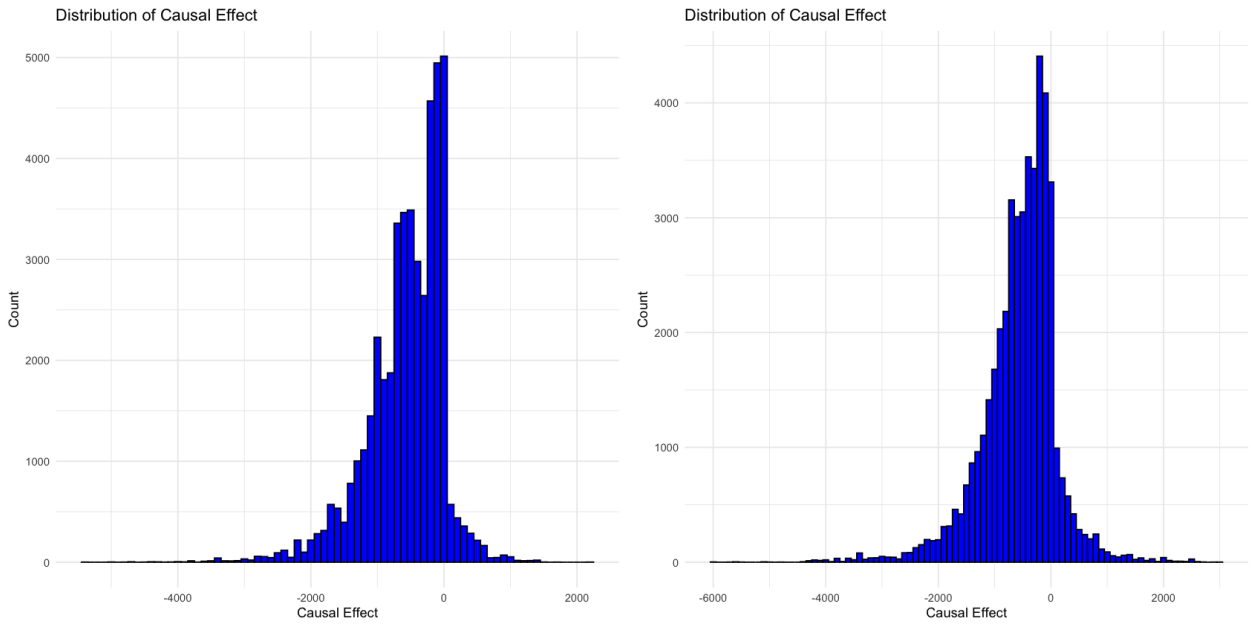

Figure 4: Causal Effect Distributions for BART (left) and XBART (right)

4.3 Binary Treatment Results

For the binary treatment model, we report the estimated average treatment effect for each subset of the data as well as 95% credible intervals for the estimate. The results are shown in Table 6.

Here, an average treatment effect of -5714 means that going over 100,000 miles on a car's odometer causes the car's price to decrease by $5714. While the estimates are different between states, the lengths of the credible intervals are similar, at around 2000. The difference in

	Average Treatment Effect	Credible Interval
Fords in CA	-5714	$(-6642, -4735)$
Fords in FL	-4873	$(-6086, -3661)$
Fords in TX	-2924	$(-4227, -1621)$

Table 6: Average Treatment Effects

average treatment effects could indicate differences between used car economies in each state. For example, California has the greatest (in magnitude) average treatment effect. This could be a result of the state's high prices and the value placed on newer cars, causing cars with more mileage to fall off quicker.

5 Conclusions

Our results confirm the advantages and potential of using BART for prediction and causal inference. One advantage is its ease of use and robustness; we were able to obtain useful results without too much hyperparameter tweaking or data transformation. Even the standard BART model performs only slightly worse than the cross-validated XBART model. These conclusions reflect those of Chipman et al, who note that "BART's performance was seen to be remarkably robust to hyper-parameter specification, and remained effective when the regression function was buried in ever higher dimensional spaces" (Chipman et al, 2010). Another advantage is the interpretability of BART, as demonstrated by the average variable usage for the prediction task.

As for future directions, there are extensions and generalizations of BART that could potentially improve on the standard model's performance (Tan and Roy, 2019). There are also similar Bayesian ensemble models for causal inference, such as causal forests (Hahn et al, 2020), that would provide a useful point of comparison. More specific to this project, it would be interesting to improve on the causal models through the use of deeper topics in causal inference, such as propensity scores, as well as more robust metrics for measuring the performance of causal inference models.

Overall, we successfully used BART for prediction and causal inference and derived useful insights from the data. BART is a worthwhile model for both tasks and a promising alternative to more standard methods.

6 Appendix

The dataset used for this project can be found at:
https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data

The code is available at in the GitHub repository:
https://github.com/jeremy-rl/bart_final_project

7 Bibliography

- Hill, Jennifer L. "Bayesian nonparametric modeling for causal inference." *Journal of Computational and Graphical Statistics* 20, no. 1 (2011): 217-240. https://doi.org/10.1198/jcgs.2010.08162
- Chipman, Hugh, Edward George, and Robert McCulloch. "Bayesian ensemble learning." *Advances in neural information processing systems* 19 (2006). https://proceedings.neurips.cc/paper/2006/hash/1706f191d760c78dfcec5012e43b6714-Abstract.html
- Chipman, Hugh A., Edward I. George, and Robert E. McCulloch. "BART: Bayesian additive regression trees." *Annals of Applied Statistics* 4, no. 1 (2010): 266-298. https://doi.org/10.1214/09-AOAS285
- Tan, Yaoyuan Vincent, and Jason Roy. "Bayesian additive regression trees and the General BART model." *Statistics in medicine* 38, no. 25 (2019): 5048-5069. https://doi.org/10.1002/sim.8347
- Hahn, P. Richard, Jared S. Murray, and Carlos M. Carvalho. "Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects (with discussion)." *Bayesian Analysis* 15, no. 3 (2020): 965-1056. https://doi.org/10.1214/19-BA1195
- Hill, Jennifer, Antonio Linero, and Jared Murray. "Bayesian additive regression trees: A review and look forward." *Annual Review of Statistics and Its Application* 7, no. 1 (2020): 251-278. https://doi.org/10.1146/annurev-statistics-031219-041110
- Dorie V (2025). *dbarts: Discrete Bayesian Additive Regression Trees Sampler*. R package version 0.9-32, https://CRAN.R-project.org/package=dbarts.