Causal Inference - Final Project

Ofek Glick

October 3, 2022

Abstract

Stackoverflow is an online Q&A platform for professional and enthusiast programmers, that currently has approximately 14 million registered users, 23 million questions and 31 million answers. Through membership and active participation, users may "upvote", "downvote" or edit questions and answers in a fashion similar to a wiki platform. In this project we will discuss the causal question: **Does adding a code sample to a question improve its chances of receiving an answer?**

Contents

Data

Our data consists of 50,000 questions from Stackoverflow asked between the dates 22.7.22 and 6.8.22, that were collected from Stackoverflow's "Data Explorer" platform, with the query save in file "Data Extraction Query".
In addition to querying questions, we queried all existing tags and ordered them by popularity.

Causal Graph Representation

We created the following causal graph which we believe represents the DGP in a satisfactory manner based on domain knowledge and frequent usage of the site (Of course we could never model the "real" causal graph but we expect this graph to be close enough for our goals).

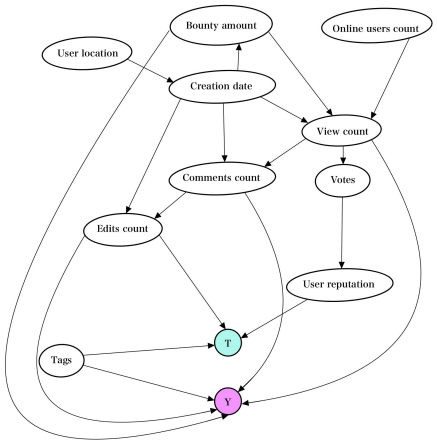


Figure 1: Causal Graph representing the DGP

- User Location - The users self-reported city and state of residence.
- Bounty Amount - Users can offer a "bounty" of points to reward other users to answer their questions, this variable represents the bounty set for a given question.
- Online User Count - The amount of people currently logged in to Stackoverflow.
- Creation Date - Date and time of the creation of the question (UTC time)
- View Count - How many different people have viewed the question. (different = different IP's)
- Votes - Users on stack overflow can "Up Vote" or "Down Vote" questions based on their assessment of the quality of the question, this variable is the sum of all Up Votes (+1) and Down Votes (-1).
- Comment Count - The amount of comments a certain post has received.
- Edit Count - The amount of times a certain post has been edited.
- Tags - Users can "tag" their questions with the topics of the question, those tags make it easier for people with the relevant expertise to find and answer their questions. This variable stores all the tags a certain question has.
- User Reputation - A rough measurement of how much the community of Stackoverflow trusts the user. Users gain (or lose) reputation mostly when their questions and answers are voted up (or down).

- **T (Treatment)** - Whether the question contains a code sample in it. All stackoverflow questions are formatted using HTML, and when a code sample is included it is contained in "<code> sample code </code>" tags. We created a Boolean feature that represents whether the body of the question contains those code tags.
- **Y (Outcome)** - Whether a question has received an answer or not. Any question can receive multiple answers, we created a Boolean variable to represent whether a question has received 0 answers or at least 1 answer.

Data Preprocessing

- Initial cleaning - dropping rows with null values
- For each continuous variable, we defined the IQR to be $IQR = q_{0.8} - q_{0.2}$ and removed all samples who's variables are not within the range $[q_{0.2} - IQR, q_{0.8} + IQR]$.
  The Variable "Edit Count" wasn't filtered this way in since filtering it would cause us to remove all edited questions.
- Transferred the "Tags" column to a k-hot representation, in order to not deal with a large amount of columns due to the large amount of different tags, we filtered all questions who do not use at least 1 of the 50 most popular tags.
- Normalized all continuous variables with the standardization normalization: $\frac{x_i - mean(x)}{std(x)}$

Back-door criterion

We used the back-door criteria in our causal graph in order to isolate the direct effect of T over Y from any other spurious correlations in the graph by discarding unnecessary confounders. Reducing the number of confounders will help us later to generalize better in our propensity score predictions and in the confounder matching. Additionally, we addressed in our causal graph to confounders that are not in our dataset. By using the back-door we will prove that those confounders are not a part of a direct causal path from T to Y. According to backdoor criteria, a group of confounders that blocks all paths from Y to T (with arrow into T) and that doesn't contain any successor of T maintains the back-door property. We will select the minimal group of confounders that maintains the criteria.
All possible paths from Y to T which contain arrow into T:

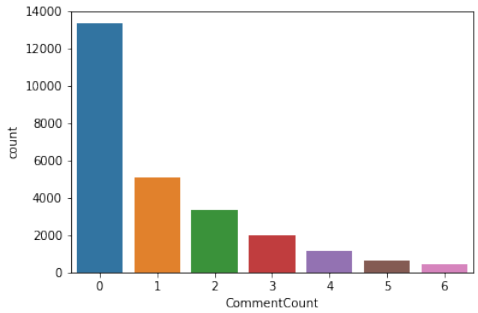
1. Y←**Tags** →T
2. Y←**Edits count**→T
3. Y←Views count→Comments count→**Edits count**→T
4. Y←Views count→Votes→**User reputation**→T
5. Y←Bounty amount→Views count→**Comments count**→**Edits count**→T
6. Y←Bounty amount→Views count→**Comments count**→T
7. Y←**User reputation**→T

The minimal group that satisfies the criterion consists of the following confounders : **Tags, Edits count, User reputation**.

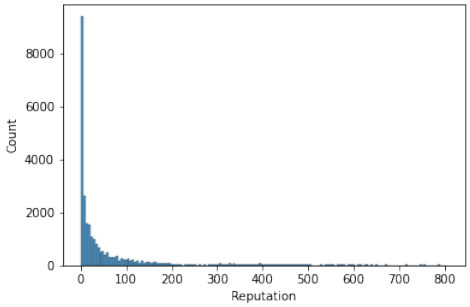
Challenges with the Data

We face to potential problems with our data.

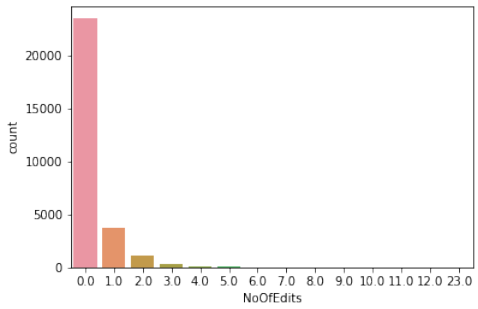
1. "Long-Tail" continuous variables - all of our continuous variables have "Long-Tail" distributions so we might encounter trouble trying to predict either treatment/outcome to underrepresented observations.



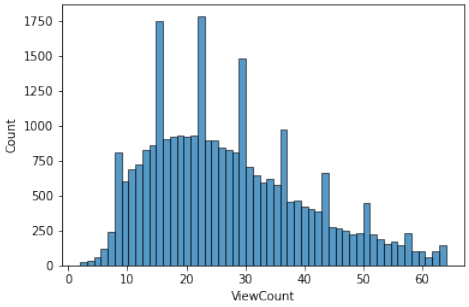
(a) Comment Count histogram



(b) User Reputation histogram



(c) Question Edit Count histogram

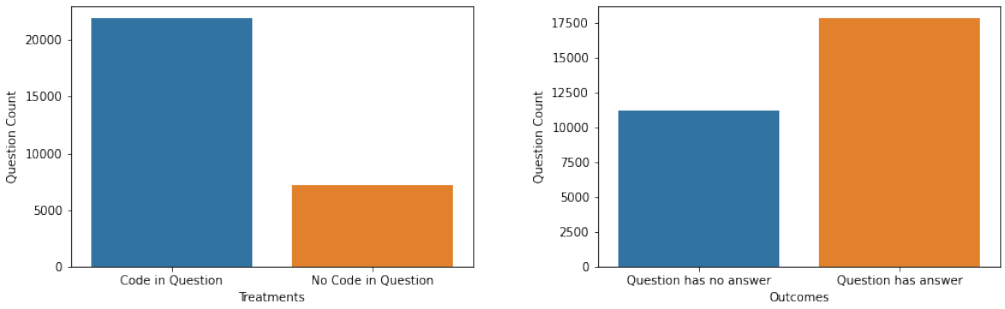


(d) Question View Count histogram

Figure 2: Examples of Continuous Variables Distributions

2. Many different tags - since we transferred our "Tags" column to a k-hot encoding representation, we face many binary columns that could cause problems when using the features for classification tasks.

3. Unbalanced treatment and outcome observations.



(a) Treatment Distribution

(b) Outcome Distribution

Figure 3: Binary Variables Distributions

Identification

Ignorability

We assume that most of the confounders are in the data, those that are not probably have a negligible effect.

Common Support

To determine the validity of the Common Support assumption, we trained multiple models to predict the probability of treatment assignment among observations.
We shuffled and split our data into a train and test set with a ratio of 4:1, and we tested the following models:

1. Weighted Logistic Regression with L1 regularization (also a CalibratedCV model, with a base estimator of Logistic regression)
2. Random Forest w. depth=15
3. Gradient Boosting Classifier
4. Naive Bayes Classifier

To compare our models and see which one performs the best, we created a AUROC graph:

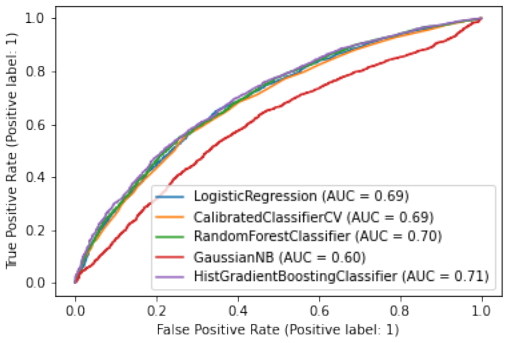


Figure 4: AUROC graph of all of our models, it can be seen that all of them perform more or less the same.

Additionally, we check additional metrics to determine which model performed the best.

Model Name	Accuracy	Precision	Recall	F1	Brier Score
Weighted Logistic Regression	~0.76	~0.76	~0.99	~0.86	~0.17
CalibratedCV(LR)	~0.76	~0.76	~0.99	~0.86	~0.17
Random Forest w. Depth=15	~0.76	~0.77	~0.97	~0.86	~0.16
Naive Bayes Classifier	~0.61	~0.79	~0.64	~0.71	~0.26
GradientBoostingClassifier	~0.76	~0.78	~0.96	~0.86	~0.16

Table 1: Score metrics showing how well each model performed, same as Figure 4, it seems most models perform more or less the same

Apart from the NB model which performed poorly, all models performed more or less the same, so we chose to use the "Weighted Logistic Regression" for propensity estimation because of it's simplicity.
Next, we wanted to check for propensity score overlap.

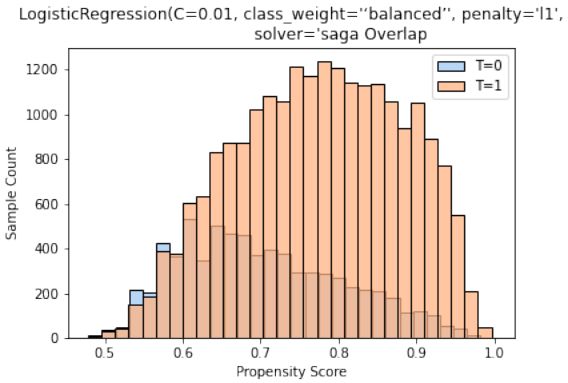


Figure 5: Sample frequency per propensity score

We can see that there is a clear overlap in the range [0.5,0.95] so we will trim our dataset to consist only of samples belonging to that range.

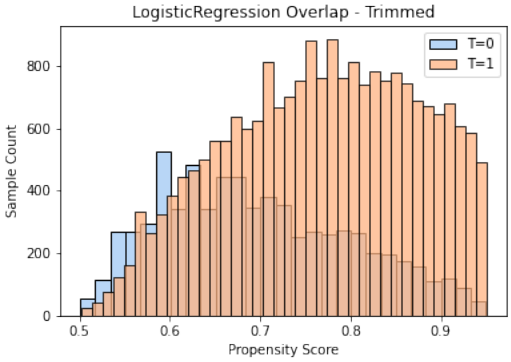


Figure 6: Sample frequency per propensity score, after trimming between the range [0.5,0.95]

Consistency

For every question asked, we see the real outcome of whether it received an answer or not for the correct given treatment. To increase the validity of this assumptions we filtered our dataset to contain only questions that the last edit on the question occurred **after** the first answer was posted, thus discarding questions that could have contained/not contained code, received an answer and were later edited in a manner that could alter our treatment (that is, a question could have had code, received an answer, and then edited without code - examples like this would not be in our dataset).
Additionally, according to an analysis conducted on Stackoverflow's Meta Site, the median answer time to a question is roughly 20 minutes. To avoid seeing wrongful outcomes to questions, we filtered our questions to be from at least 1 day from the last time the dataset was updated, that way we know that all questions had enough time to be seen and possibly be answered.
More formally $Y = T \cdot Y_1 + (1 - T) \cdot Y_0$

SUTVA

- We assume that the potential outcomes for any question does not vary with the treatments for other questions. That is, we assume that adding code to a certain question has no effect on whether code is added to a different question.

- To consider different types of treatment, we would have to think of adding code examples to a question as whether the code contributed to the understanding of the question and therefore contributed to an answer being posted. Stackoverflows user base holds users accountable to their questions, and questions of poor quality or not enough focus are removed from the site. We chose to work on questions who were not flagged or removed and therefor we can assume that if a question has a code example, it is relevant to the question and is clear and understandable.
  After making these assumptions, we arrive at the conclusion that if a question contains code, there are no different levels of treatment, since different levels of treatment would have been filtered out by the websites community.

Estimation Methods

In the following sections we will discuss the methods we used to calculate the ATE (or CATE). Using each method, we calculated a bootstrap confidence interval with 100 iterations and $\alpha = 0.05$.

IPW

We predicted the ATE using the propensity scores calculated in the previous step:

$$\hat{ATE} = \frac{1}{n}\sum_{i=1}^{n}\frac{t_i \cdot y_i}{\hat{e(x_i)}} - \frac{1}{n}\sum_{i=1}^{n}\frac{(1-t_i)\cdot y_i}{1-\hat{e(x_i)}}$$

Stabilized IPW

Since we have seen that IPW sometimes breaks down when we have many samples who's propensity scores are close to 1 (just like in our case) we decided to validate our results by using a stabilized IPW approach:

$$\hat{ATE} = (\sum_{i=1}^{n}\frac{t_i}{\hat{e(x_i)}})^{-1}\cdot\sum_{i=1}^{n}\frac{t_i\cdot y_i}{\hat{e(x_i)}} - (\sum_{i=1}^{n}\frac{1-t_i}{1-\hat{e(x_i)}})^{-1}\cdot\sum_{i=1}^{n}\frac{(1-t_i)\cdot y_i}{1-\hat{e(x_i)}}$$

Propensity Score Matching

This method uses Nearest Neighbor matching using the propensities of the samples. For each observation with a given treatment, we matched an observation from the other treatment group according to the L1 distance of the propensity scores. The ATE is then calculated like so:

$$\hat{ATE} = \frac{1}{n}\sum_{i=1}^{n}\hat{ITE}(i)$$

When $\hat{ITE}(i) = y_i - y_{j(i)}$ for $t_i = 1$ and $\hat{ITE}(i) = y_{j(i)} - y_i$ for $t_i = 0$

Confounder Matching

Similarly to the Propensity Score Matching, we perform Confounder matching where for each observation with a given treatment, we matched an observation from the other treatment group using the following metric:
Let us denote the group of all unique tags $T = \{t_1, ..., t_m\}$, and let us denote the tags associated with an observation x_i as T_{x_i}. The categorical distance (CD) between two observations $x_i, x_j$is defined to be

$$CD(x_i, x_j) = \left(\frac{|\{t|(t \in T_{x_i}, t \in T_{x_j})or(t \notin T_{x_i}, t \notin T_{x_j})\}|}{|T|}\right)^{-1}$$

Notice that using this metric observation with similar tags have a low CDscore and observation who have less tags in common have a high CD score.
Finally, we define our distance metric to be the euclidean distance of the continuous variables between two samples multiplies by the CD distance and the ATE was calculated in the same manner as with the propensity matching.

T-Learner

This method is used to calculate an estimator for **CATE**, conditional average treatment effect, which is another measure defined as:

$$CATE = \mathbb{E}[Y(1) - Y(0)|X = x] = \mathbb{E}[Y(1)|X = x] - \mathbb{E}[Y(0)|X = x] = \mu_1(x) - \mu_0(x)$$

This method consists of two steps:
1. Calculate the estimators $\hat{\mu}_1(x), \hat{\mu}_0(x)$ using multiple classification models. We will use observations from the control group to estimate $\mu_0(x)$and we will use observations from the treatment group to estimate $\mu_1(x)$.
2. estimate the CATE by : $\hat{\mu}_1(x) - \hat{\mu_0}(x)$

S-Learner

Similarly to T-Learner, this method is also used to calculate an estimator for **CATE**, but unlike T-Learner, it treats the treatment as another confounder. In this case, instead of having two models for the potential outcomes as functions of the confounders, we will use a single model that will estimate the following function:

$$\mu(x_i, t_i) = \mathbb{E}[Y^{obs}|X = x_i, T = t_i]$$

This method consists of two steps:
1. Use all observations to estimate the potential outcome function, ,with multiple classification models.
2. Estimate the CATE by: $\hat{\mu}(x, 1) - \hat{\mu}(x, 0)$

Results

Method	Lower Bound	Upper Bound	$\hat{ATE}$
IPW	0.131	0.147	**0.137**
Stable IPW	0.047	0.057	**0.052**
Propensity Score Matching	0.191	0.258	**0.224**
Confounder Matching	-0.015	0.058	**0.021**
S-Learner Logistic Regression	0.053	0.062	**0.057**
S-Learner GB Random Forest	0.006	0.024	**0.012**
T-Learner Logistic Regression	0.040	0.051	**0.045**
T-Learner GB Random Forest	0.017	0.041	**0.028**

Table 2: Confidence intervals for the ATE for each estimation method

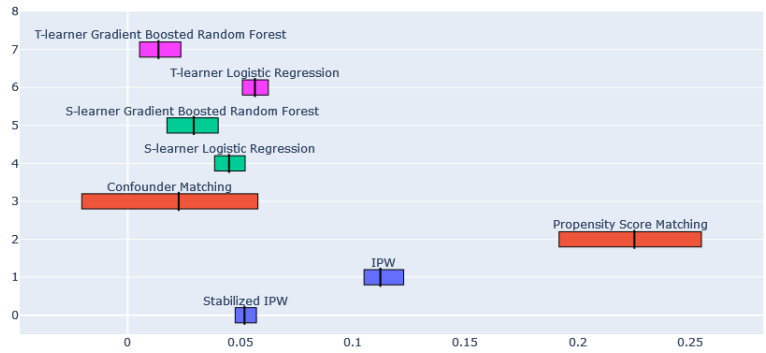


Figure 7: Visualization of the confidence intervals, methods who work in a similar fashion are colored in the same color

As can be seen in Table 2 and Figure 7, Most of our ATEs (apart from Propensity Score Matching) are fairly close to 0 which would lead us to conclude that even if there was a causal relation between T and Y, it is negligible. In addition, most of our CIs (apart from Confounder Matching) do not include 0 so this leads us to conclude that there might be **some** causal effect, but as mentioned earlier, it is of very little strength. We do notice however that both the IPW ATE and Propensity Score Matching ATE are higher than the other ATEs, but we assume that this difference is most likely due to bias introduced to the estimator when using our propensities for estimations. As discussed in previous sections, our dataset is imbalanced with respect to the treatment and as seen in the assumptions section, we have an imbalance in our propensities among the observations with respect to the treatment groups (there are significantly more treated observations with high probability that non-treated observations with high probability - this leads to bias when relaying on propensities for estimation). It is worth noting, that stabilizing the IPW estimator results in an ATE estimation that is close to other ATEs that did not rely on propensities.

Potential Weaknesses

- There is always the possibility that our domain knowledge was not enough to create a causal graph represents the DGP in an accurate manner, there could be hidden confounders that we have not taken into account, and even if we would have taken them into account, we might not have them measured.
- We could have accidentally introduced bias into our dataset with our query, the site from which we extracted our data has a limit of 50,000 records, which means we had to settle for subset of our data. We hope that 50,000 questions is enough to reduce such bias but because of the sheer volume of questions in the site we can not know for sure how well our subset represents the general dataset.
- Since questions on Stackoverflow can be answered at anytime, it is possible that our consistency assumption might not be entirely accurate. We do believe however that this assumption is relatively sound, since previous analyses have shown that most questions are answered within the first day or two and if they haven't been answered in that time, most likely they won't be answered.
- Preprocessing techniques - Tt is possible that our methods for outlier detection and normalization introduced bias to our dataset. We could have tried multiple techniques such as IsolationForest, MinMax normalization, Robust Data Standardization and many others to see if we would have received different results.
- Data Representation - We chose to represent our categorical variables as a k-hot encoding representation, that led to 50 binary columns which could have influenced all of our models which relies on the confounders to make predictions. We thought about using Target-Encoding to transform the categorical data to numerical but given the fact that this uses the target variables, we wanted to avoid using techniques like this when dealing with a causal question.
- Propensity Estimation - it is possible that we did not estimate the propensity well enough and that could have affected later estimations.
- Confounder Matching Distance Metric - We have used our own distance metric to find "close" neighbors based on our confounders, this distance metric could be switched to other metrics and pehaps receive different results.

Conclusions and Discussion

In the project we have attempted to answer the causal question "**Does adding a code sample to a question improve its chances of receiving an answer?**" by modeling the DGP and estimating the ATE and CATE. We employed multiple estimation techniques and have encountered pitfalls regarding some of them. We have seen that the IPW estimator tends to over estimate when the propensities are not balanced between the groups and that could lead to false assumptions of causality. We have found evidence that there could be a causal affect between "adding code" to "receiving an answer" but the affect is probably negligible, since questions which require code either already have it, or it is being added later with the request of the community and questions which do not require code could receive an answer regardless of whether code was supplied or not.
The minute (small) affect that we are seeing could be related to questions of low quality, which either require code and do not provide it or do no require code but too much of it is provided which makes the question less understandable and harder to answer.
For future work we would try to work on a larger dataset with more observations that could better represent the general dataset and we would try to use more advanced ATE estimation techniques that were not used in the scope of this project. Also, we would strive to obtain more datapoints that could help us overcome possible missing confounders.

References

[1] Course Lectures and Tutorials

[2] Data Explorer Schema documentation

[3] Stackoverflow Data Explorer