Applied Uplift Modeling with Meta-Learners

By

Shelby Temple

A Capstone Project Submitted in Partial Fulfilment of the

Requirements for the Degree of

Master of Science

In

Data Science

University of Wisconsin – Oshkosh

Oshkosh, Wisconsin

May 2020

**Abstract**

The project explores cutting-edge uplift modeling techniques for a marketing client.Uplift modeling is a set of data science techniques that leverage both causal inference and machine learning to predict the effect of a treatment (Gutierrez, 2016). It can be used to understand if a treatment (such as a marketing advertisement) caused the desired outcome (such as a purchase).

The project contains both research and application. In result, the project is presented as two components:

1. An in-depth review of uplift modeling and its applications
2. A deep-dive on a special set of uplift methods available in Python called meta-learners

First, it was found that the original package of interest Pylift was largely surpassed by another package causalML. Pylift has less support and it also only implements 1 meta-learner. In comparison causalML has much more support and currently implements 4 meta-learners and 4 direct methods for a total of 8 supported models.

It is ultimately found that just like any machine learning project, there is no free lunch for uplift modeling as well. There is no best algorithm that wins for all uplift modeling projects. Strong performance is largely dependent on the dataset and project constraints.

Across both the packages and techniques, there are tradeoffs between simplicity, speed, and accuracy. Also, each algorithm has situations where they might perform better or worse. For example, one algorithm might do well with large balanced datasets and another might tend to do well with small imbalanced datasets.

Despite there not being a silver bullet solution, there is still a guide framework that can guide a data scientist to at least focus on a recommended approach first.

The writing approach leans into plain and practical explanations over scientifically precise explanations.

In addition, there are uplift modeling applications for any industry and business unit, but this capstone will focus on marketing applications. I prefer this approach as the generalizations can be a little harder to follow than a concrete example.

**Table of Contents**

Table of Contentsiii

List of Figuresiv

List of Tablesv

**Introduction1**

Author Summary**1**

Uplift Modeling Introduction**2**

Three Examples in the Field**4**

How Uplift Modeling Works**5**

Other Important Uplift Concepts**8**

Current Challenges in the Field**9**

**Summary of Popular Approaches11**

Traditional Approaches**11**

Direct Modeling**14**

Standard Meta-learners**14**

Transformed Outcome Meta-learners**17**

**Capstone Details18**

Goal of Capstone**18**

Synthetic Data Generation**20**

Feature Importance Assessment**25**

Two-Model Approach**29**

Pylift**32**

CausalML**35**

**Final Recommendations40**

**Suggestions for Further Research42**

**References44**

**List of Figures**

The Classic Uplift Segments 6

Advertisement Impact to Individuals24

Differences in Segments Across Synthetic Datasets24

Basic Dataset Correlation Plot26

Assessing Features Examples26

Complex Dataset Univariate CV5 Importance by AUC28

Complex Dataset Random Forest Importance Ranking28

Basic - Two Model Results31

Complex - Two Model Results32

Basic - Pylift Results 34

Complex - Pylift Results35

Basic - causalML Results38

Complex - causalML Results40

Package and Technique Comparison41

Uplift Model Framework42

**List of Tables**

Basic causalML AUQC37

Complex causalML AUQC39

**Introduction**

**Author Summary**

Shelby Temple is an experienced marketing data science & analytics consultant with a history of success solving complex problems for large corporations.

For undergraduate studies, he completed Bachelor’s in Science with a double major in mathematics and economics from Illinois State University (2012.) More recently, this spring he will complete a master’s in data science from University Wisconsin Oshkosh (pending the success of this project.)

His career in data science & analytics began at Uline, a large industrial supply company. He started in customer service but eventually found his way to the database and marketing team. Most recently, he has spent the last four years with Allstate Insurance Company working on the marketing data science & analytics team.

At Allstate he specializes in digital marketing projects such as bidding optimization for Google search engine text ads and marketing attribution modeling that helps track marketing effectiveness even when the customer completes the purchase offline.

Shelby enjoys working on end-to-end machine learning projects and building data visualizations to communicate insight.

**Uplift Modeling Introduction**

An insurance company is allocating new leads to insurance agents from the prior day based on the order they were gathered. Combining both an email and phone call agents can convert 5% of worked leads to sales. The average new customer generates profit of $1,000. Not accounting for costs such as staffing, commission, or lead acquisition, 100 leads worked results in 5 sales or $5,000 profit.

As the company has grown they have run into a dilemma. They generate more leads in a day than agents can work. There is also a sense that a lot of the leads are a waste of time.

With a desire to be more data-driven, the company’s data science team wants to optimize the order in which leads are worked. Using the lead data collected from the website, they built a model scoring the probability each lead was likely to convert to a sale. This way an agent can start each day paying close attention to leads that have the best chance to convert to a sale.

Once the model was put into production the company observed that agents were now converting 10% of worked leads into sales. The project was deemed a success by all stakeholders since the company can now expect a $10,000 profit for every 100 leads worked leveraging the new propensity model built by the data science team.

Is it possible that the company could be worse off profit-wise than assigning the leads randomly or by the order in which they were received? The answer is yes.

All the propensity model did was gather the customers that are most likely going to convert to sales. It did not separate the customers who are going to purchase no matter what from the customers who needed additional persuasion. In a sense, the project is just cherry picking.

It could be possible that all the leads that are now being worked were already going to become customers without the lead program. In other words, these customers did not need to be persuaded, since they were going to reach out to close the sale themselves. It may seem that they are generating more profit with the lead program, but it could be the opposite. There is a chance that the lead program is just measuring more customers who were already going to purchase regardless than before.

The best approach would be to find the leads that were not going to purchase, but with the extra nudge from the lead program will purchase. In other words, find the leads where the lead program would directly cause the customer to convert.

Welcome to the world of uplift modeling—a bridge between causal inference and machine learning. What the synthetic case study above forgot to consider is persuasion, incrementality, or the true effect of the outbound lead campaign. Uplift modeling looks to see if the outbound lead campaign caused the sale.

More broadly, uplift modeling is a set of data science techniques that leverage both causal inference and machine learning to predict the effect of a treatment (Gutierrez, 2016). In the case study above, the treatment was to send emails and make phone calls to customers. The applications have a much wider scope than lead optimization. In marketing, uplift modeling can improve the return-on-investment (ROI) for any type of campaign where the effect of an advertisement can be estimated to groups or individuals. Other applications include treatments in healthcare, customer retention, price changes, and user experience. Anytime a company wants to interact with its clients to improve an outcome, uplift modeling should be involved.

**Three Examples in the Field**

To start with the most famous example, uplift modeling is often citied as a key component to Barack Obama winning his 2012 election (Stedman, 2013). The campaign used predictive analytics to target the individuals who were most likely to be persuaded by ads, mailings, phone calls, and other outreach efforts. The idea was to find the people who were on the fence, could vote either way, and to make sure they vote blue. Also, to avoid people who were dead set and could not be persuaded to change their vote. It’s been reported that the uplift model impacted millions of people to vote differently than they would have (Siegel, 2013).

Wayfair, an e-commerce tech company known best for their home furniture offerings, leverage uplift modeling as well. Wayfair implements uplift modeling in the display banner advertisement marketing channel (Stichnoth 2018.) Display banner ads are a purchased in a real-time-auction. Where many companies can bid for their advertisement to appear on a website and only the winner appears. In this space the inventory is super cheap. In today’s display market 1,000 impressions can cost anywhere from $3-$15. To put this into perspective in Google’s search engine high end keywords can cost $50-100 per click. Wayfair uses uplift modeling to creates a prediction of incremental revenue for every customer. This prediction along with a predicted click through rate work together to create the final bid amount to have their banner ad appear.

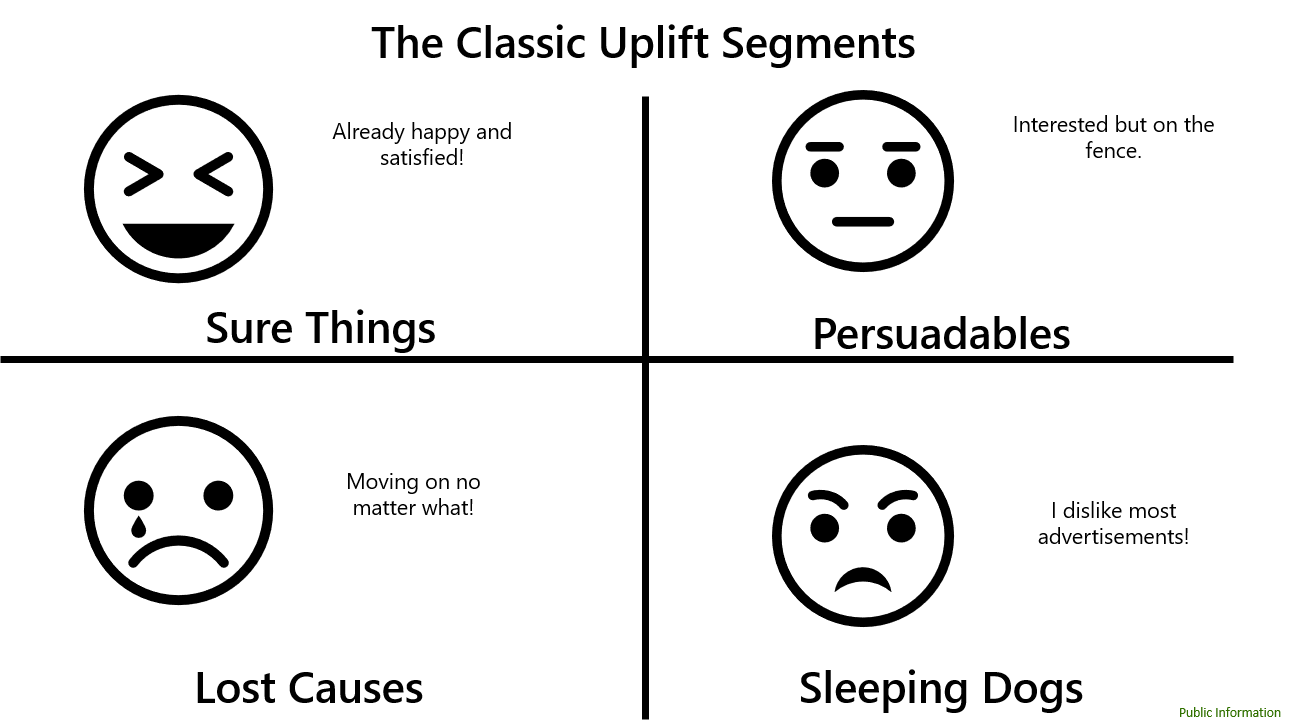
More recently Uber, has opened up about how they are using causal inference heavily to improve their marketing and user experience (Harinen 2019.) They take uplift modeling to the next level by putting into production multiple treatment group tests. For example, they might test different communication channels and promotion types simultaneously. Most other uplift model projects only change one lever at a time. Uber has even incubated and open sourced a python package, causalML, to let the entire data science community leverage cutting-edge techniques (Chen 2020.)

Although there are uplift modeling applications across any industry and business unit this capstone will mostly focus on the marketing application. I prefer this approach as the generalizations can be a little harder to follow than a concrete example.

**How Uplift Modeling Works**

The goal behind uplift modeling is to predict the impact of a treatment on an individual to see if it caused the desired outcome. In marketing, the treatment can be an advertisement and the desired outcome can be many different things such as an app download, website visit, purchase of a product, or enrollment into a program. In the classic marketing problem set up there are four segments to consider (Siegel 2011):

1. People who **will** purchase no matter what (sure things)
2. People who **will** purchase only if they are exposed to an advertisement (persuadables)
3. People who **will not** purchase no matter what (lost causes)
4. People who **will not** purchase if they are exposed to an advertisement (sleeping dogs)



Uplift modeling optimizes the situation by showing us who we should advertise to (persuadables) and how to limit advertising to the other groups (sure things, lost causes, and sleeping dogs). Why advertise to someone who will purchase no matter what? Also, it is not a good idea to advertise to someone who will be negatively impacted by the advertisement. As previously stated, uplift modeling bridges together causal inference and machine learning to solve the problem.

It is a causal inference problem in that we need to understand two realities that are exclusive to each other. We need to understand both what would happen in *Reality A* where a customer saw an advertisement and what would happen in *Reality B* when a customer didn’t see an advertisement. However, it is impossible for a customer to both see and not see an advertisement. Causal inference techniques can help solve for estimating the counterfactual. In other words, it can help us simulate the scenario that does not exist.

In addition to using causal inference, uplift modeling also mirrors a lot of the typical tasks in an end-to-end machine learning project. It includes tasks such as:

* Data engineering a raw dataset
* Feature engineering
* Exploratory data analysis
* Pre-processing categorical and missing values
* Feature selection techniques
* Splitting the data into a test and validation set
* Selecting a training metric such as accuracy or Area-Under-Curve
* Selecting the best model and hyperparameters by searching tons of combinations
* Taking the winning model and put it into a production environment

Another way to put it is that in machine learning you usually just have features (inputs to help with a prediction) and the ground truth labels (the target to predict). In uplift modeling, the ground truth is never available because an individual cannot both see and not see the advertisement. Causal inference simulates the scenario that didn’t happen so then it is possible to estimate the ground truth labels and treat it as a machine learning problem.

The output an uplift model creates is usually referred to as the conditional average treatment effect (CATE). It is the expected causal effect of the treatment for a subgroup in the population conditioned on the features used in the prediction (Haupt 2019.) For example, if we knew a person’s age, state, gender, and income an uplift model can look at those inputs and make a prediction on if they will purchase both if we show an advertisement to them and if we don’t show an advertisement to them. From there it will assess the difference between these two predictions to understand the incremental effect, or contribution an advertisement would make in getting a customer to purchase.

Since the input variables will change across each potential prospect the uplift score will as well. The final practice in marketing when applying the uplift model is to use the uplift scores more like a content recommender. You start by targeting the customers with the most uplift and move down the list until you are targeting the customers with the least uplift.

**Other Important Uplift Concepts**

Another thing to consider is how the uplift modeling data is created. Ideally the treatment and control groups are generated via random assignment. In practice a 50/50 randomly assigned experiment can be unfeasible and/or uneconomical. In marketing, a random assignment will usually be in the neighborhood of 80% treatment and 20% control.

In the field of uplift modeling you will often come across the phrase *heterogenous treatment effects*. This is just a fancy way to say that an advertisement will affect people (or groups of people) in different ways and at different rates. Which is intuitive in that if you showed an advertisement to a hundred people you wouldn’t expect everyone’s probability to purchase to increase by exactly 2%. Some people might be persuaded much more, and some people may have even disliked the advertisement (their probability to purchase decreased.)

Another concept is that customers are at different starting spots in their journey towards the outcome. Even if I could run a marketing campaign that has a large positive impact on every customer I can’t expect them all to reach the threshold that makes them convert. Some people are not in market for a product while others are actively shopping. Some people are aware and interested in a brand and others have never heard of the brand. As a result, the optimal solution is usually not just to find the people who have the highest positive impact when treated, but to also account for the people who are on the fence and just need a nudge to arrive to the desired outcome. In marketing this concept is often describe as a stage in the funnel.

**Current Challenges in the Field**

When I first started exploring uplift modeling I immediately noticed a ton of inconsistency across the research, tutorials, and documentation. Many techniques are brand new and/or constantly evolving. Due to the rapid pace of change, there seems to be no consensus on a championed framework for building uplift models.

First, there is inconsistency in the names of things across different sources. Even the name uplift modeling can change depending on where you look. Five other names I saw for uplift modeling were:

1. Estimating Heterogeneous Treatment Effects
2. Incremental Modeling
3. Net Scoring
4. Average Causal Effect Modeling
5. True Response Modeling

It is also difficult to understand how many different uplift modeling techniques and algorithms there are because different sources will reference the same technique but by different names. For example, there is a technique where you transform the dependent variable from a binary classification target to a numerical regression target. I’ve seen the technique referred to as: class transformation method, uplift increaser modeling, meta-learners, the modified outcome method, and the transformed outcome method.

The final major inconsistency to highlight is the metrics, validation, and visualization. Some of the most popular metrics Area Under Gains Curve and Area Under Qini Curve are difficult to explain and have limitations. Each package I looked at had a different set of answers for this issue. Unless the dataset is simulated, the ground truth effect from the treatment is never known. We cannot make someone both see and not see the advertisement.

Besides inconsistency another challenge is that many of the highly optimized (for speed) machine learning packages like scikit-learn and TensorFlow cannot simply be used to model uplift directly. As previously discussed in a typical machine learning project you use the features, X, in a dataset as inputs to train a model to predict the target variable, Y. In uplift modeling you are trying to understand the impact of treatment, T, conditioned on the features, X, to predict a target, Y. Scikit-learn cannot directly solve for this question. There are direct modeling techniques, but they are regrettably slow even on medium sized datasets.

There are techniques such as using meta-learners that allow you use a package like scikit-learn directly. However, for some problems even scikit-learn is slow. For uplift modeling there isn’t anything I found that allows for easy integrations to allow for executing uplift modeling on big data. One of the “quick” methods I tested was going to take 24 hours to train on a dataset with 400,000 rows and 130 features.

**Summary of Popular Approaches**

As mentioned before it is unclear what a comprehensive list of uplift modeling techniques should look like. The scope of the capstone will be to explore meta-learners available in python. Disclaimers aside, it is still worthwhile to cover the popular approaches I found.

In marketing, it is understood that it does not make sense to advertise to everyone. There are economical constraints such as the cost of advertising and customer constraints such as their affinity towards a brand. In terms of solving for this business problem there are three major approaches. Each approach has many methods. The first approach is propensity modeling. As previously demonstrated, propensity modeling is not uplift modeling—there is no causal inference. However, uplift modeling is essentially a more advanced extension of this traditional approach, so its coverage is relevant. After propensity modeling, the remaining methods can be split into two approaches. Modeling uplift directly and indirectly through meta-learners (Gutierrez & Gerardy 2016).

**The Traditional Approach**

Propensity modeling is often referred to as the traditional (or classical) scoring method. The book, *Targeting Uplift: An Introduction to Net Scores*, labels the approach as gross scoring (Michel 2019). Propensity modeling uses machine learning to assess the probability of an outcome in the future. For example, a loan company might want to understand if they approve an individual for a loan what is the probability it will be paid back. The target is often binary with 1 representing the presence of the outcome of interest and 0 representing the non-presence.

I have seen scenarios where the data is filtered first to only on the customers who have received the treatment. This seems like a logical step in that it is more about exploring the how other features play into the response given the treatment was provided. However, in many cases it is enough in the eyes of the stakeholder to either build the model with having the treatment history as a feature or having treatment history absent altogether.

Some advantages of propensity modeling include:

* Will likely do better than random by eliminating most of the lost causes (people who have a very low probability to purchase with or without advertisements)
* There is consistency in research, documentation, and validation with the approach because the project is set up and executed just like any other typical machine learning project
* Highly optimized software is available again resulting from the project set up
* Can usually leverage observational data that is likely already being collected by the company—no experiment is required to create the modeling data

Some disadvantages of propensity modeling are:

* It is often not answering the real question of interest which would be: “If I advertise to a person or group what will be the incremental outcome and return on investment compared to never advertising.”
* There is a possibility that the model ends up recommending tons of sure things (people who would have reached the outcome without the treatment) and due to this the outcome could be worse than random
* It does not carefully consider negative effects from the treatment—for example a if a customer starts with a high propensity a propensity model might suggest targeting them even if the impact of the advertisement is negative

**Modeling Uplift Directly**

Modeling uplift directly is utilizing custom machine learning algorithms that can directly maximize the understanding of treatment effects. An example of a direct model is the uplift forests. The uplift forest mostly functions like a random forest, but it has special tweaks that allow it to optimize and output the uplift predictions. One example of a tweak is that it will select the feature and split to two new nodes based on the feature that shows the biggest difference in outcome values across treatment and control. Where random forest only considers splitting to separate classes on the outcome values.

Some advantages of direct modeling include

* Will likely give the best and most accurate results since the models are designed for uplift modeling
* Often inherit concepts from popular machine learning techniques so to gain an understanding you really only need to understand the tweaks of the algorithm

Some disadvantages of direct modeling include

* The available algorithms are very slow, making it extremely difficult to use them with moderate to large size datasets often encountered in the field

**Standard Meta-Learners**

In the realm of uplift models meta-learning can be described as tricks that allow you to indirectly estimate uplift. It can get a little confusing across the literature, but I have arrived at the conclusion that one can split meta-learning into two categories. Standard meta-learning and transformed outcome meta-learning.

Standard meta-learning are methods that take creative shortcuts in order to allow for the use of standard classifier machine learning algorithms (i.e. logistic regression or xgb classifier) to estimate uplift. Some popular examples include the single model approach, two-model approach, and the four-quadrant approach.

The key component of the single model method is using the treatment flag as a feature in the model. First you train one model using a training dataset with both treatment and control observations. For new observations the uplift prediction is the difference between the predicted values when the treatment indicator is changed from control (0) to treatment (1) (Kunzel 2019).

The two-model approach is one of the most popular uplift techniques. As the name suggests the dataset is split into two and two independent models are trained (Lee 2019). One dataset and model only focuses on the subjects that received treatment and the second model and dataset only contains subjects in the control dataset. Once the two models are built they can both be applied to a new observation. The effect is that there is a treatment prediction and a control prediction. Subtracting these two values provides an estimate of uplift. For example, if the treatment model gives me a high prediction and the control model gives me a low prediction. It can be concluded that the treatment had a big impact. If the treatment model gives me a high prediction and the control model gives me a low prediction I can conclude that that person was likely to purchase with or without treatment.

Finally, in the four-quadrant approach the treatment binary flag (1 means treatment group 0 means control group) is combined with the outcome binary (1 means purchase 0 means didn’t purchase) to create four groups to predict. The problem is essentially transformed into a multi-classification problem and a model like xgb will output a probability that a new observation belongs to any of the four groups. The final uplift score is:

*PTR (probability of treated and response) + PCN (probability control no response) – PTN (probability treated no response) – PCR (probability control and response)*

The intuition is that PTR is definitely good, PCN could be good, PTN is bad, and PCR is bad. By adding and subtracting the predicted probabilities you end up with a proxy for uplift (Lee 2019).

Some advantages of standard meta-learning include:

* Out of the meta-learning options the standard ones are easier to understand and explain
* Can use any optimized machine learning model such as XGB as is

Drawbacks of standard meta-learning include:

* The feature selection, model selection, and tuning are not optimized for maximum uplift
* Each model has its owns set of potential flaws created by the shortcuts or assumptions it makes
* The models have been shown to perform worse than both direct modeling approaches and transformed outcome meta-learners

**Transformed Outcome Meta-Learners**

Transformed outcome meta-learning will take the information about treatment, treatment assignment probability, and the target outcome variable and transform it into a new numerical target variable. Predicting this new target is also estimating uplift. With the transformed outcome uplift modeling has been decomposed into one or many standard regression problems. At the time of this writing, there are four well established transformed outcome meta-learners: S-learner, T-learner, X-learner, and R-learner (Kunzel 2019).

The S-learner inherits its approach from the single model approach. Similarly, the T-learner inherits its approach from the two-model approach.

The X-learner is an extension of T-learner with some extra tweaks. It leverages the observed outcomes to impute the unobserved scenario. Every observation that was in the treatment group has its control outcome imputed and every observation in the control group has its treatment outcome imputed. The X-learner is then able to estimate the uplift in a second step as if both scenarios (both treated and not treated) were observed. The advantage over two-model is that the model can still use information from the control group to derive better estimators for the treatment group and vice-versa.

Finally, the last meta-learner is the R-Learner which uses cross-validation estimates of outcomes and propensity scores to fit machine learning models. The R-learner get’s its name by the special optimization metric, R-Loss, it tries to minimize when running cross-validation.

Some advantages of transformed outcome meta-learning include:

* The models tend to be faster to implement than other techniques
* In many cases the models also tend to perform just as well as direct uplift models
* The transformed outcome meta-learning has been generalized for imbalanced treatment groups

Drawbacks of transformed outcome meta-learning include:

* Out of the meta-learning options these models can be more difficult to explain
* Each model has its owns set of potential flaws created by the shortcuts or assumptions it makes

To summarize, uplift modeling is a smarter way than traditional propensity models to priority score an audience for treatment. However, with uplift modeling it seems that there is a pick your poison scenario. Direct uplift modeling should provide the best accuracy results in most scenarios, but at the time of writing the current open source implementations have speeds that are hard to overlook. Meta-learners use shortcuts that allow for speed but might not perform as accurately as direct models. It is unclear what those tradeoffs might look like in practice. This is a big part of the motivation behind this capstone.

**Capstone Details**

**Goal of the Capstone**

The primary goal of the capstone is to focus on exploring cutting-edge uplift modeling techniques for a marketing client. The stakeholders for this project are a marketing data science team for a top auto insurance provider. Currently the team has traditional propensity models in flight that inform targeting for addressable marketing tactics such as direct mail and cross-sell campaigns. As previously mentioned propensity modeling can be a useful solution in practice, but it is not optimizing incremental purchases like uplift modeling would. It had already been determined by the client that direct uplift models are an infeasible solution due to their slow speed.

However, there is also concern with the transformed outcome meta-learner algorithms because they are largely unproven in practice. It is unclear how much accuracy these algorithms might trade for speed. The existing tests and validations of the algorithms tend to use datasets that are not structured like the ones the client works with. The existing tests and validations tend to use small to moderate sized datasets (5k-50k rows and 3-15 features) and a balanced treatment and control group. The client typically works with datasets that have at least 1 million rows, over 150 features, and imbalanced treatment groups (80% treatment, 20% control).

The client was interested in, Pylift a package that implements a transformed outcome variable meta-learner approach. However, it made sense to take a step back and evaluate all current python packages that handle uplift modeling. After completing a survey, it seemed like there could be more viable solutions than just Pylift. In result, I incorporated some of those packages as well.

The work will result in a few different recommendations. An explanation on which Python packages will tend to be most useful for the client. An explanation on which uplift techniques will tend to be most useful for the client. Finally, a new framework is proposed in order to deal with the issue that no one algorithm will be the best solution for every scenario and dataset size.

**Synthetic Data Generation**

It was agreed upon that the data used in this project would be synthetically generated. The first advantage would be that the capstone could be shared with no legal or proprietary concerns. There is also a need to test multiple different datasets to understand how the algorithms might respond to different scenarios. Getting multiple real-world uplift datasets that very in dataset size, structural complexity, and relationships between variables would be challenging. Also, in practice real world datasets don’t have the ground truth labels. As mentioned before you can’t have someone both see and not see an advertisement. Synthetic datasets can simulate both events. In result, it can be understood how close the model is doing to predict ground truth. It opens additional visualization and validation checks that wouldn’t be available with real world datasets.

Although synthetic data generation was a good idea with plenty of advantages, it was still a very challenging roadblock to overcome. Typical machine learning data generators such as scikit-learn’s datasets module would not be satisfactory as they will only generate features and a target. They aren’t designed to allow for the implementation of a binary treatment flag with a heterogeneous treatment effect that is needed for uplift modeling. They also don’t allow for the data to be simulated as a random experiment or as observational data which uplift techniques require.

Another choice was to look at data generation models that are apart of existing uplift packages. Although they were directly designed for uplift they were very limited in allowing for big changes in complexity, treatment and control percentages, and allowing for a large and diverse set of features.

The client also had their own in-house synthetic dataset generator. However, it seemed like it was going to take a decent amount of time to learn and even more time to build very large and wide datasets. This was decided as plan B if I couldn’t find a more end user friendly interface.

Through many searches I found a package called Opossum that aligned nearly perfect to the capstone’s use case. The package was original released in August 17, 2019. Some of the key features included:

* A user-friendly interface that allows for quick adjustment of data generation
* Solid documentation on how to understand and use the package
* Parameters that let you control the number and variety of features
* It lets you choose between a binary or continuous target variable
* You can control if the groups are randomly assigned and if the groups are imbalanced
* The intensity and properties of the treatments effect
* The relationship between the features and target and if there are interactions across the features

One option the Opossum package seemed to be missing is allowing for custom response rates in the case of binary. Instead I just used a flat percentile transformation on the continuous response variable to first select the top 50% for basic dataset and top 3% for the complex dataset.

After selecting Opossum for synthetic data generation, the next idea was to generate two datasets: a basic dataset and a complex dataset. The basic dataset would help establish a baseline for the different packages and techniques I tried while the complex dataset would help understand if the package and technique will translate well to the client’s typical use case.

The basic dataset was generated with the following properties:

* 50,000 rows
* 10 features all intended to be useful in predicting
* Feature types included 6 numerical continuous and 4 binary categorical features
* The intensity of the lift was set to high (10 on a scale of 10)
* Probability an observation was exposed to an advertisement was 50%
* The relationship between features and target was linear with no interactions
* The base propensity to convert in the future with no advertisement is 50%
* 90% of treatment effects were positive, with 6.6% of treatments having no impact, and 3.3% of treatments having a negative impact

The complex dataset was generated with the following properties:

* 2 million rows
* 200 features with only 60 (30%) of then being useful the rest as noise
* 40 features are numerical continuous and 20 are categorical with a mix of levels ranging from two to fifty
* The intensity of the lift was set to high (10 on a scale of 10)
* Probability an observation was exposed to an advertisement was 75%
* The relationship between features and target was linear with no interactions
* The base propensity to convert in the future with no advertisement is 3%
* 90% of treatment effects were positive, with 6.6% of treatments having no impact, and 3.3% of treatments having a negative impact

As previously mentioned an advantage of synthetic data generation is that you can explore the ground truth labels in the dataset. For each observation I can see both what would happen when showing a customer an ad and not showing a customer an ad.

Figure 2 shows the true impact of the advertisement. This is consistent across the basic and complex datasets since I selected an intensity of 10 and the same distributions of positive and negative impacts.

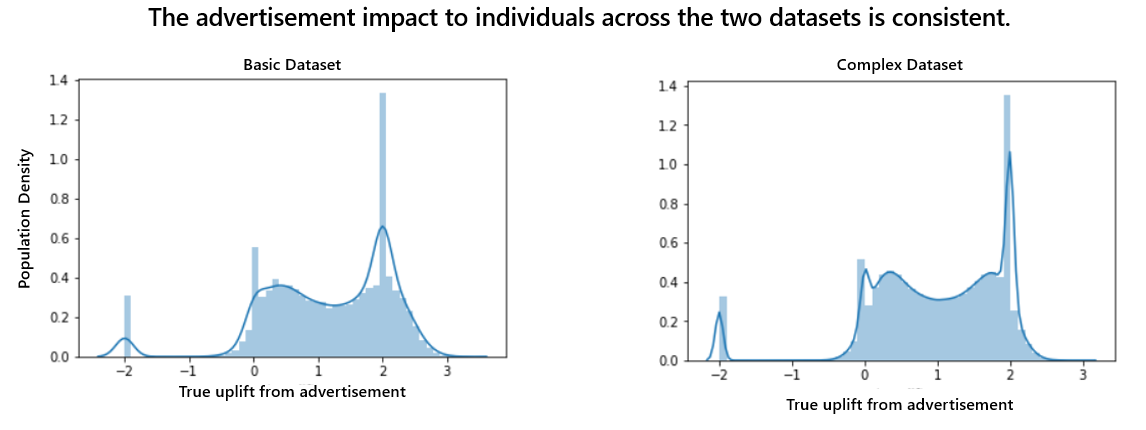
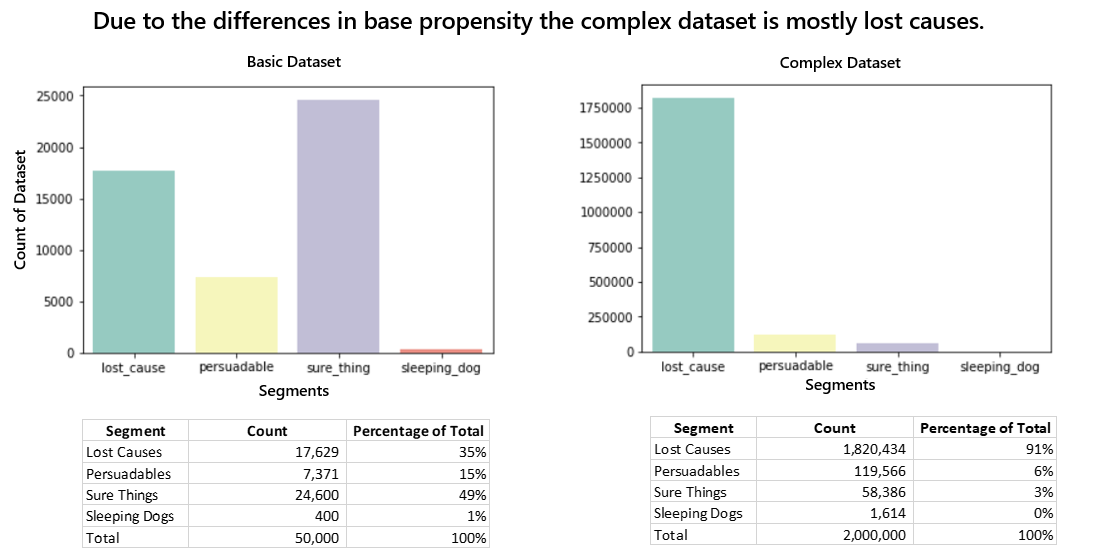


Figure 3 shows the true segments each observation belongs to in the dataset. In practice the true segments are not known, they can only be estimated with causal inference. There are notable differences across the two datasets is due to the difference in base propensity. Having an extremely low response rate will result in many more lost causes with everything else fixed.

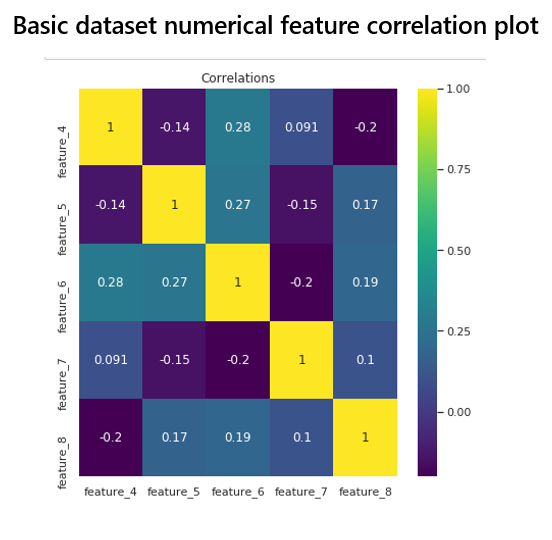


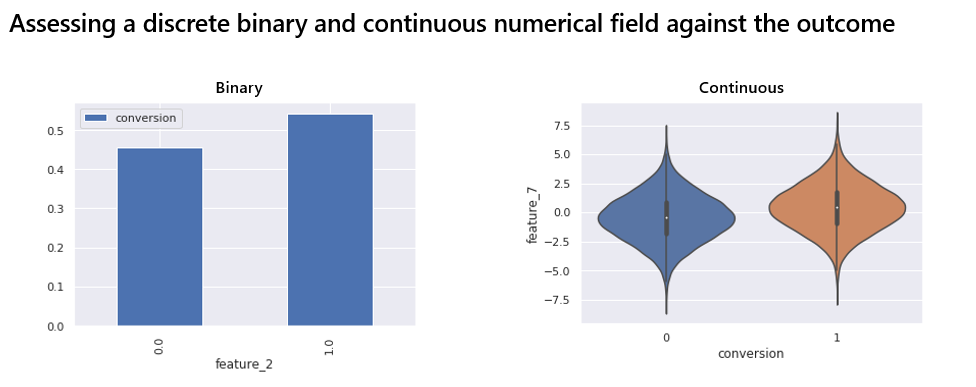
In the client’s industry (property and casualty insurance) the complex dataset is much more reflective of the situation in that a large segment of the population rarely switches between different insurance carriers. In result, a very large percentage of the population is in the lost cause segment—regardless of showing an advertisement the customer is not going to purchase in the short term. Relative to other dataset structures the solution to finding as many persuadables as possible becomes much more similar to finding a needle a haystack.

**Feature Importance Assessment**

Although we know a lot about the two datasets already, I wanted to go ahead and take an approach that included everything that is typically completed in an end-to-end data science project. It is a best practice in data science to do some form of exploratory data analysis (EDA). Then, the next step is to complete additional feature engineering and pre-processing of data, so it is ready to be shipped to machine learning algorithms. Finally, especially with large datasets where model selection and tuning can be very expensive, it is necessary to look at feature selection and dimensionality reduction techniques.

For the basic dataset visualizations and describe feature functions are much more manageable. It is easy to look at an exhaustive list of correlations (figure 4) or plots that show if it tends to have a positive or negative impact to the outcome variable (figure 5). It was found in figure 4 that there was no multi-collinearity or redundant numeric variables. In terms of feature selection one feature was found to not be useful in two standard machine learning approaches univariate importance and random forest importance. For uplift there is a special importance assessment net information value which can help with feature selection specifically for uplift modeling. The same feature also performed poorly in that assessment as well. Knowing that including the variable could hurt model accuracy performance I made the call to drop it.

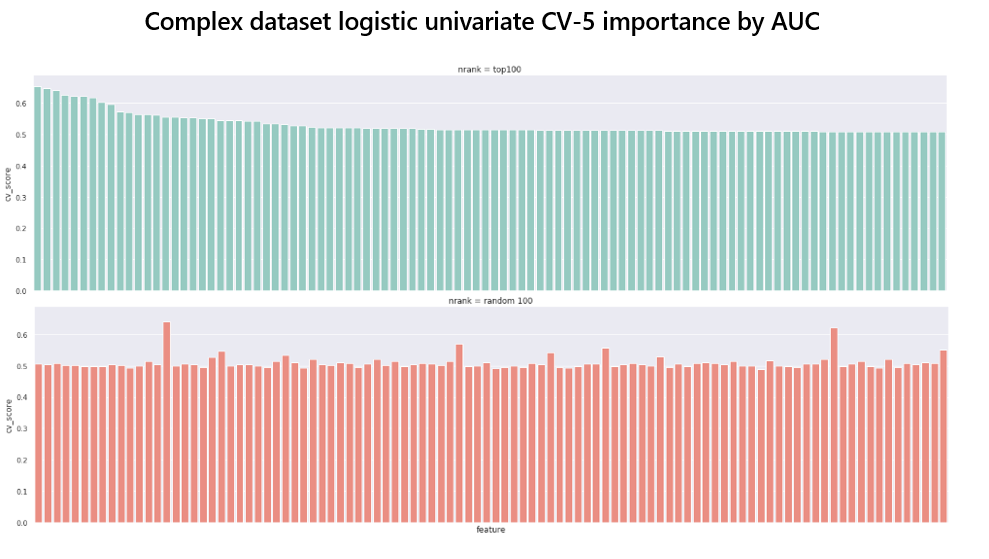


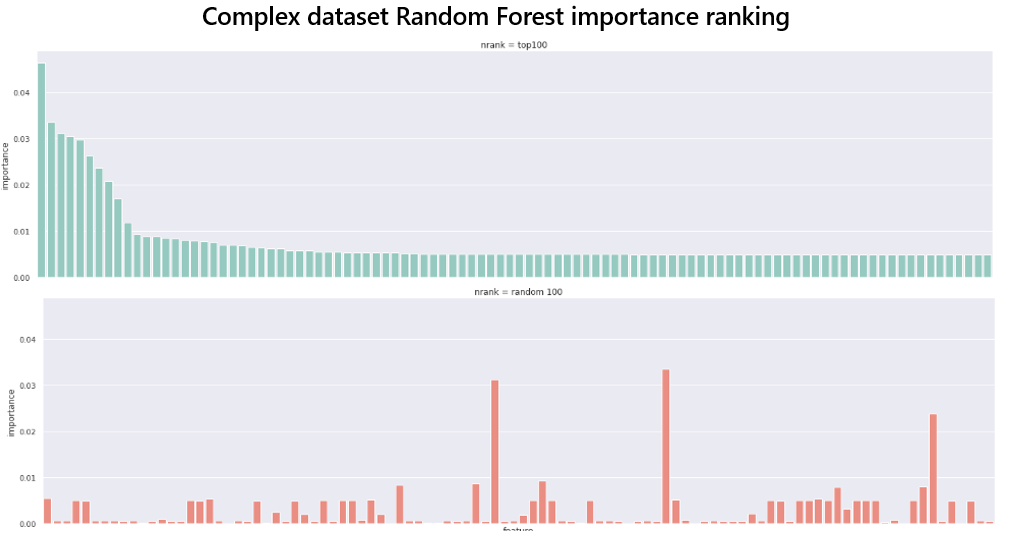


Unlike the basic dataset, the complex dataset has categorical variables with more than two levels. Also, by design it has approximately 70% of variables that are useless, noise when predicting. In order to prepare for standard machine learning models one-hot-encoding was used to transform 3+ level categorical columns into a series of binary levels. Doing this expanded the width of the dataset from 200 features to 322 features. At this point the dataset is going to take a ton of time to work with even using highly optimized machine learning solutions such as scikit-learn. Variable reduction can greatly improve both model accuracy performance and performance in speed to select, tune, and predict.

Unfortunately, it is mostly unfeasible to use any of the visualizations above with such a wide dataset. Besides the challenge of inspecting many features, functions like the net information value in Pylift couldn’t handle 322 features in a reasonable amount of time. I was able to inspect both the univariate and random forest feature importance assessment conditioned on the outcome variable. This is flawed in that variable reduction should be an assessment on the power of x to predict the impact of t on y (standard machine learning feature importance procedures just assess the power of x to predict y.) Having a perfect solution at this time was not practical.

In order to make processing time more manageable I decided to use the intersection of univariate logistic AUC scores greater than .5 (which is basically a random model) intersected with the top 200 models ranked by the random forest importance procedure. This resulted in cutting down the number of features to 149 for the complex dataset. The example figures (6 and 7) show the univariate AUC and random forest outcomes for the top 100 features then a randomly selected 100 features.





At this point I felt that the two datasets had been explored, engineered, and trimmed to the appropriate features/dimensions. The final step was to explore performance of different packages and techniques.

**Two-Model Classifier Approach**

With two datasets generated, explored, and pre-processed the next step is to build the models for each dataset and assess their performance. The first approach I took was building a standard meta-learner without utilizing any special uplift modeling packages.

As mentioned above the two-model approach splits the dataset and modeling into two independent groups (treatment group and control group). Once the two classifier models are built you have it look at a new observation and make two predictions—the probability of response if exposed (from the treatment model) and probability of response if not exposed (from the control model). The final step is to subtract these two predictions to get a proxy for uplift.

For the basic dataset 5,000 observations were put aside for holdout validation. The remaining 45,000 were used in model selection and hyperparameter tuning. Next, unique to the two-model approach is to split the 45,000 observations into two datasets based on if they were exposed to an advertisement or not. 18,046 made it into the treatment dataset and 17,953 made it into the control dataset. In the treatment dataset the target variable class (purchase vs no purchase) was imbalanced with 77% purchasing. In the control dataset the class (purchase vs no purchase) was balanced 50/50 which makes sense because that was the base propensity setting. It seems like the overall lift from running the campaign was 27%.

Basic dataset model selection and hyperparameter tuning was really quick only taking a few minutes to run. The winning model for both datasets was the XGBoost classifier. The hyperparameters needed to have high regularization values to avoid overfitting.

For the complex dataset 100,000 observations were put aside for holdout validation. The remaining 1.9 million were put in a development set that would be split and under-sampled for class balance used in model selection and hyperparameter tuning. Under-sampling had two benefits in that it reduced the size of the training data substantially for treatment and control datasets. The drawback is that it might be removing observations that could be helpful for accuracy. The balanced treatment dataset was still 247,000 while the control dataset was 23,000. The differences between the two datasets can be partially explained by the difference in the experiment treating 75% of the observations and leaving a control group of 25%.

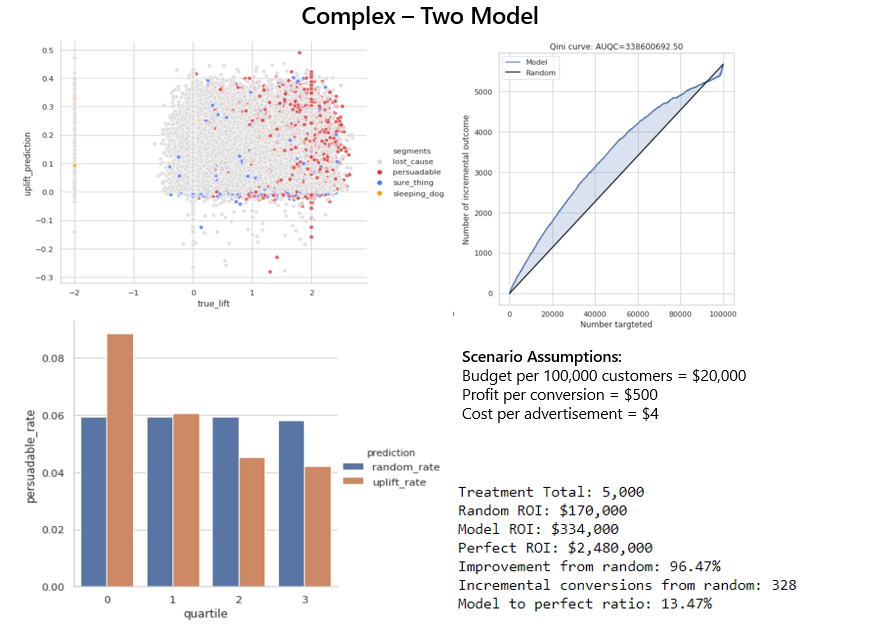
Complex dataset model selection and hyperparameter tuning was extremely slow for the treatment taking five hours to run. The winning model for both datasets was the logistic regression classifier.

For the basic dataset, the two-model approach improved moderately over random targeting. In the hypothetical scenario, it demonstrated a 70% improvement to ROI and in the first quartile of targeting it was finding persuadables at a 22% rate (7% greater than the overall distribution of 15%).



To explain the visuals in the top left you want to see the red persuadables float to the top of the chart. Since True lift is positively correlated a cluster in the top right corner would also be good. Top right chart is the area under qini curve. If the targeted audience was shuffled in order of highest predictions to lowest the curve shows you how many incremental purchases you have over the a random model, the black line. The higher the AUQC the better. The bottom left bar charts compare how often you are targetining a persuadable with the model vs random across four quartiles. The 1st quartile shows what the model thinks were the best targets. Finally, bottom right contains a trivial scenario that is somewhat realistic.

For the complex dataset, the two-model approach also improved moderately over random targeting. In the hypothetical scenario it had a 96% improvement to ROI and in the first quartile of targeting it was finding persuadables at an 8.5% rate (2.5% greater than the overall distribution 6%).



**Pylift**

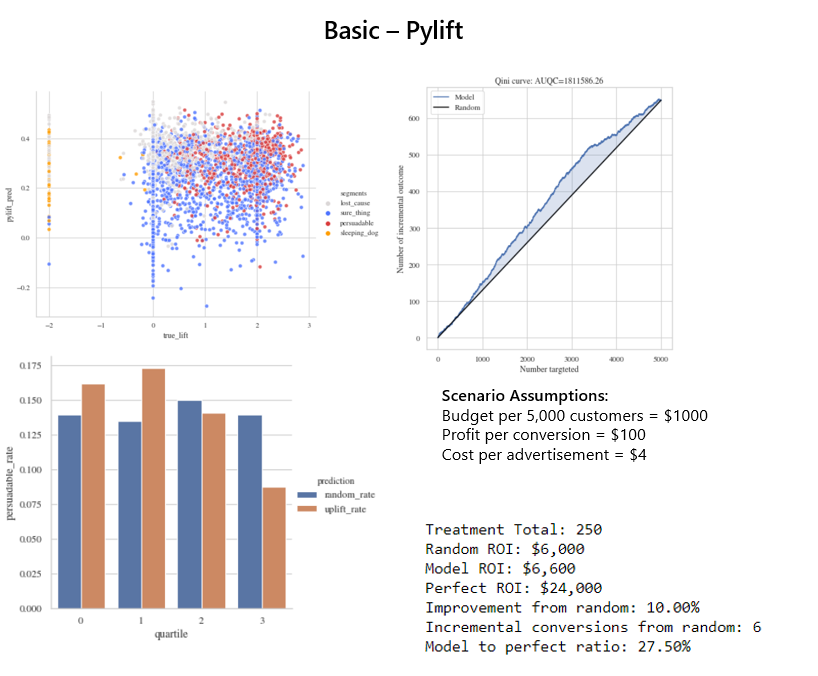
Pylift is a package developed by Wayfair’s data scientists. It uses a transformed outcome meta-leaner to decompose the problem into a set of regression problems. Based on the documentation it looks like it aligns to my previous description of a transformed outcome T-learner although the documentation never explicitly labels itself as a T-learner. The package also has its own set of metrics and visualizations that are not present in any other uplift package.

Unfortunately, it was also unclear if it is possible to make predictions on how to make brand new predictions on a dataset in production. This was likely just me not being able to figure it out, but this caused a little bit of inconsistency with the holdout validation across the other two methods. The other datasets got to tune on a development set of test examples then were validated against a separate holdout that was never tuned against. Pylift got to use its development set of test observations for both tuning and final validation.

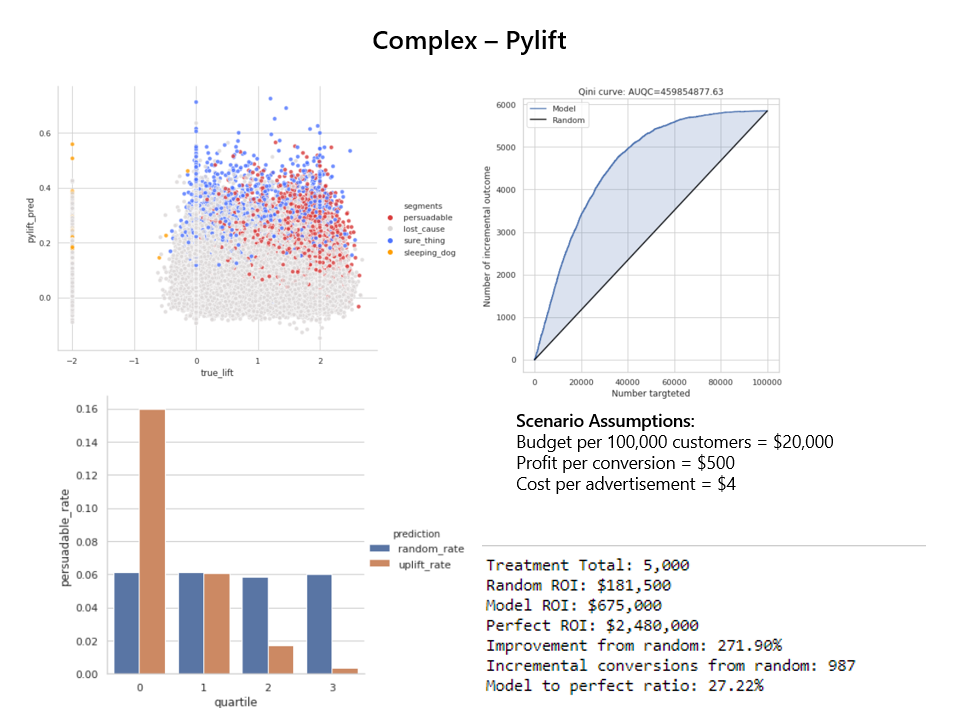
For the basic dataset, I set the training data to 40,000 observations and the test set to 10,000. Pylift allows for hyperparameter tuning. I ran 150 variations of XGBoost with five cross validation folds in about 20 minutes.

For the complex dataset, I set the training data to 400,000 observations and the test set to 100,000. I ran 10 fits with two cross validation folds in 40 minutes. The hyperparameter tuning times can get out of hand quickly. 120 fits with five folds was going to run about 24 hours.

Even with the advantage of validation on the development test data Pylift performed much worse that the two-model approach for the basic dataset. With only a 10% improvement to ROI from random targeting and just a half percentage point (15.5% to 15% baseline) in first quartile improvement to persuadable rate.



On the other hand, Pylift did better than the two-model approach with the complex dataset. With a 156% improvement to ROI from random targeting and in the first quartile of targeting it was finding persuadables at a 16% rate (10% greater than the overall distribution 6%). At first, I was really excited with these results, until I realized from the scatter plot the model had simply figured out which observations were going to convert and not prioritizing persuadables over sure things. The perfect uplift model should favor persuadables over sure things. This model seems to favor sure things over persuadables which is not ideal. In theory, one could essentially get the same result with a strong traditional propensity model. It is unclear if this result would provide much improvement to the client who already has a traditional propensity model in production.

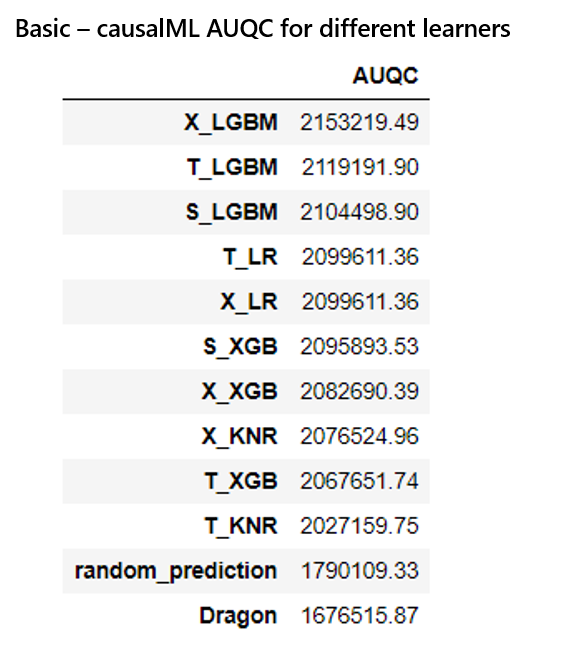


**causalML**

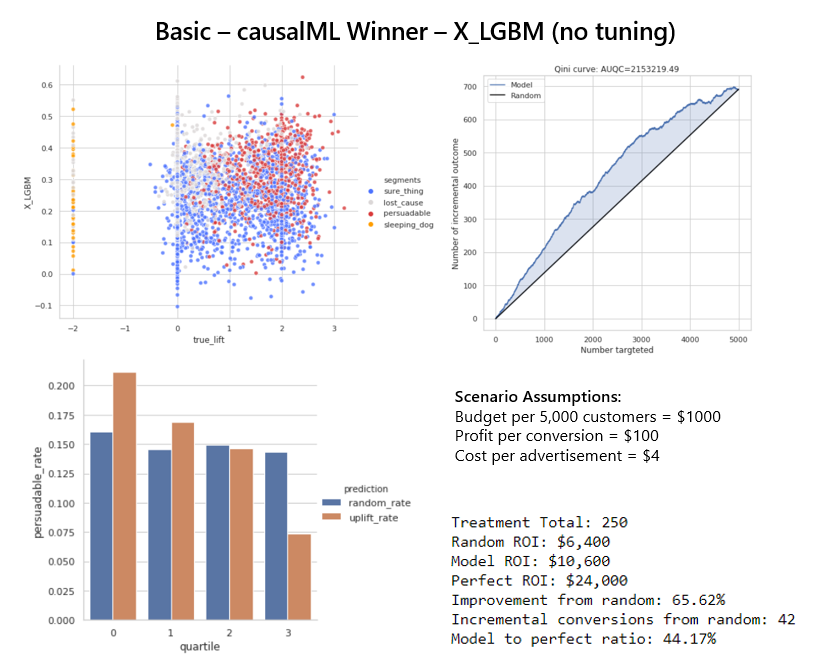
causalML is a packaged developed and incubated by Uber’s data science team. They have created the package as the premier one-stop-shop for uplift modeling currently implementing 8 uplift models (4 direct uplift trees and 4 meta-learners). For the project I focused on the 4 meta-learners. I also discovered a bug with the R-learner and at the time of writing and they won’t have it corrected for a few more days. Due to the issue I was limited to the S-learner, T-learner, and X-learner.

Also, at the time of writing causalML does not have any built-in feature for hyperparameter tuning to maximize uplift for the base learner models. With that said, it is possible to improve the result by evaluating different regression models by the different learners. For example, I can compare S-learner with linear regression to X-learner with linear regression. I can also compare S-learner with linear regression to S-learner with XGBoost. However, I cannot easily run through 100 different S-learners with 100 different XGBoost parameter settings to maximize uplift.

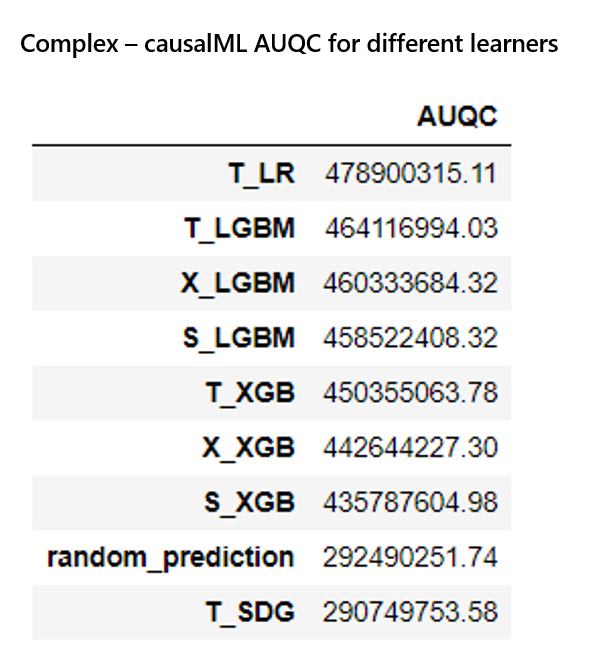
For the basic dataset 5,000 observations were put aside for holdout validation. The remaining 45,000 were used in model selection. For optimizing uplift, 11 different learner model combos were evaluated. All models were able to run in under 30 seconds except the neural net dragon learner (an experimental learner they recently added). I picked the winning model based on the metric area under qini curve (AUQC). The winner based on the holdout AUQC was the X-learner with a lightGBM regressor.



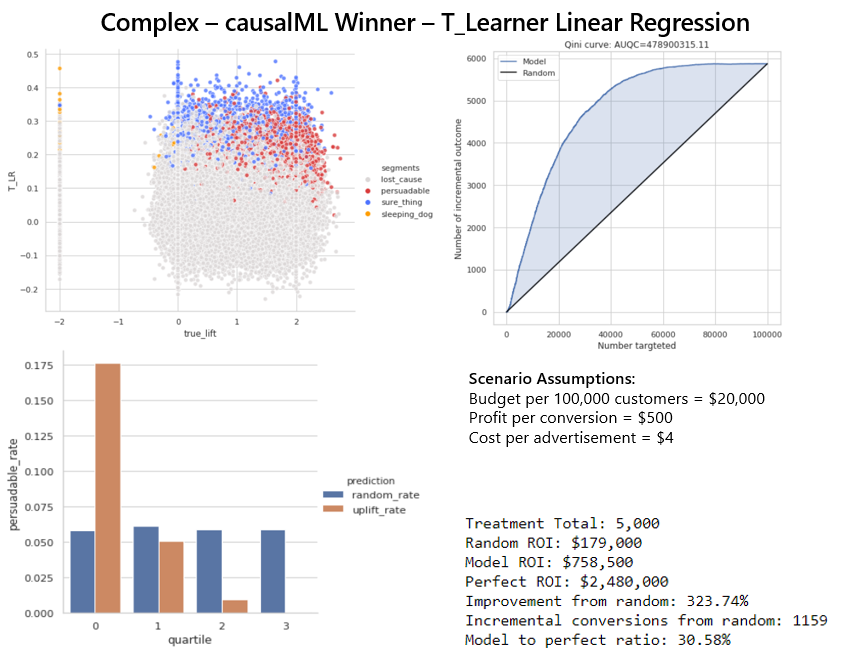
The basic dataset with causalML performed comparable to the Two-Model approach. In the hypothetical scenario it had a 66% improvement to ROI and in the first quartile of targeting it was finding persuadables at a 21% rate (6% greater than the overall distribution 15%).



For the complex dataset 100,000 observations were put aside for holdout validation. The remaining 1.9 million were making the X-learners run indefinitely. To mitigate this issue, I reduced the training set to 400,000 observations to use in model selection and model training. For optimizing uplift, 8 different learner model combos were evaluated. The X-learner lightGBM and X-learner XGB took 6 and 9 hours to fit and predict respectively. I picked the winning model based on the area under qini curve (AUQC). The winner based on the holdout AUQC was the T-learner with linear regression base learners. This model ran in six seconds.

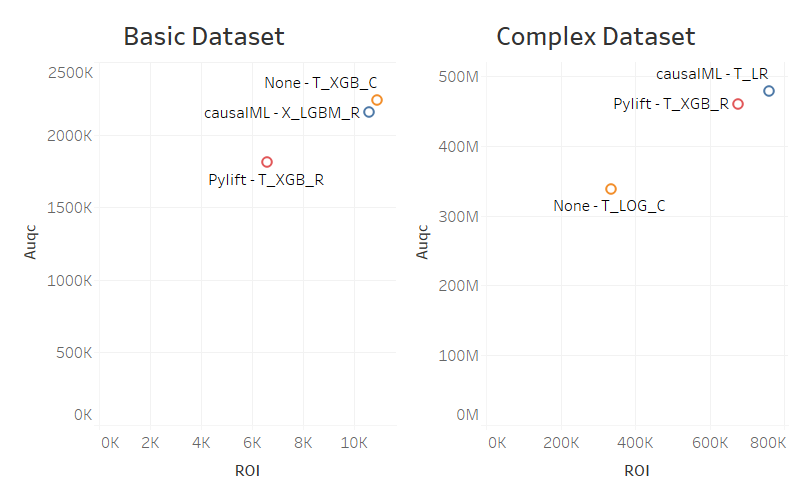


For the complex dataset, the T-learner with linear regression approach provided the best results yet. In the hypothetical scenario it had a 323% improvement to ROI and in the first quartile of targeting it was finding persuadables at an 17.5% rate (11.5% greater than the overall distribution 6%). Similar to the Pylift performance on complex dataset, this model tends to prefer sure things over persuadables. Where the ideal model would prefer persuadables to sure things. It could be that it is overfitting to signals that explain which observations will convert.



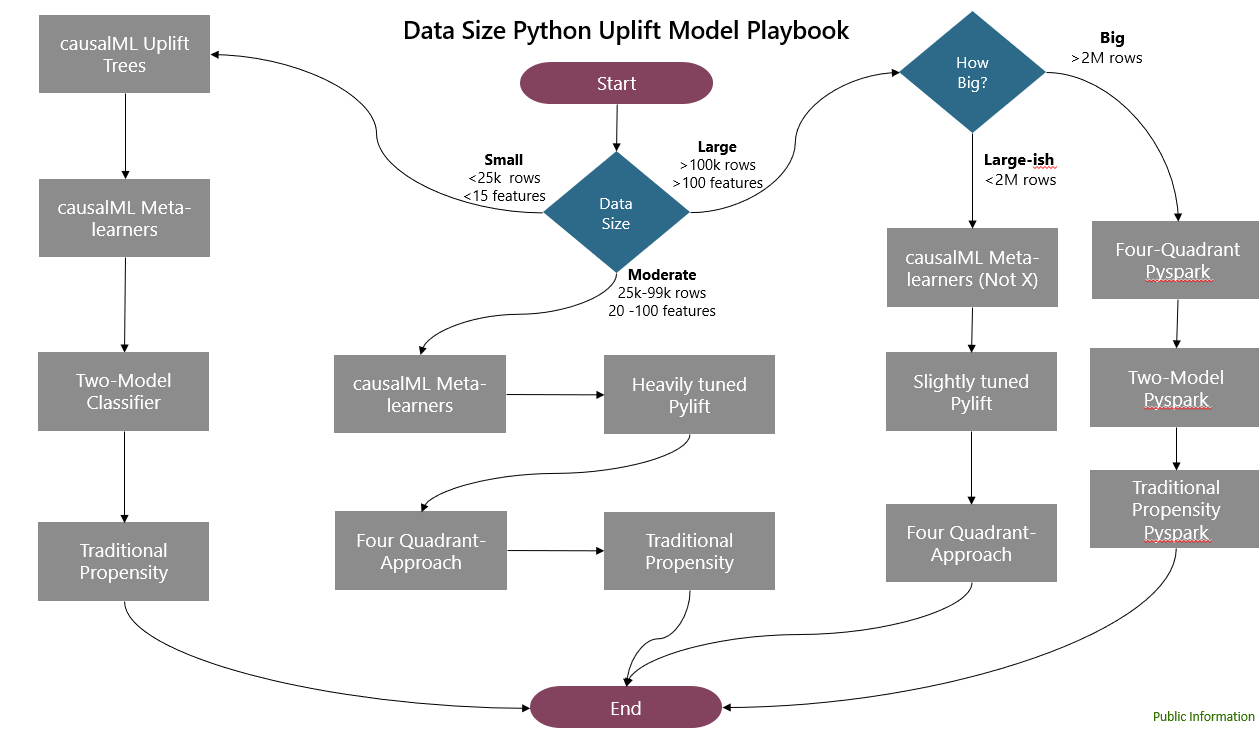
**Final Recommendation**

After reviewing three different package approaches along with five different meta-learner techniques, and training thousands of models across two datasets I have come to a consensus that causalML is the best package and T-learner tended to be the best meta-learner for the two datasets considering accuracy, speed, and difficulty to implement.



Although T-learner gave me the best all-around results that does not mean it is the best learner overall. Although the two datasets I created are very different from each other, they do not represent the unlimited combination of scenarios someone might encounter in practice. For example, this capstone did not consider a dataset that is highly imbalanced with less than 5,000 observations but 300 useful features.

Based on the creators of the X-learner, they say that the best learner can change by dataset (Künzel 2019). For example, they mention that the X-learner tends to do best when the dataset is imbalanced and small to moderate (just like the example I listen in the previous paragraph). Understanding that uplift modeling is just like any other machine learning project, where success is dataset dependent, I created a final framework for how I would go about building an uplift model based on dataset size. This is intended to be an inexact guide to demonstrate which approaches might work best.



**Suggestions for Further Research**

Even with a comprehensive deep-dive on meta-learners I still feel like I left a lot of stones unturned. Here is a list of things I think would be worthwhile to investigate further:

* An assessment using even more variety in synthetic datasets
* The performance of the causalML R-learner on these datasets
* The performance of a standard four quadrant meta-learner
* Completing uplift modeling in the Pyspark framework
* Building out the possibility to hyperparameter tune the base regressors in causalML
* Comparing the speed and accuracy of the meta-learners covered with a traditional propensity model
* Comparing the speed and accuracy of the meta-learns with uplift trees using the causalML model
* Research that improves the inconsistency across documentation, research, and blogs
* Research that improves the credibility and standard for uplift model performance metrics and visualizations

**References**

Chen, H., Harinen, T., Lee, J., … Zhenyu. (2020, March 2). CausalML: Python Package for Causal Machine Learning. Retrieved from <https://arxiv.org/abs/2002.11631>

Fei, G. (2019, October 29). Modeling Uplift Directly: Uplift Decision Tree with KL Divergence and Euclidean Distance as Splitting Criteria. Retrieved from https://tech.wayfair.com/data-science/2019/10/modeling-uplift-directly-uplift-decision-tree-with-kl-divergence-and-euclidean-distance-as-splitting-criteria/

Gutierrez, P. & Gerardy, J. (2016). Causal Inference and Uplift Modeling A review of the literature. *JMLR: Workshop and Conference Proceedings 67:1–13*

Haupt, J. (2019, November 17). Causal Machine Learning: Individualized Treatment Effects and Uplift Modeling. Retrieved from https://johaupt.github.io/causal machine learning/Uplift\_ITE\_summary.html

Harinen, T. (2019, June 20). Using Causal Inference to Improve the Uber User Experience. Retrieved from https://eng.uber.com/causal-inference-at-uber/

Karaman. (2019, September 26). Uplift Modeling. Retrieved from https://towardsdatascience.com/uplift-modeling-e38f96b1ef60

Lee, J. X. J. (2018, November 6). Simple Machine Learning Techniques To Improve Your Marketing Strategy: Demystifying Uplift Models. Retrieved from https://medium.com/datadriveninvestor/simple-machine-learning-techniques-to-improve-your-marketing-strategy-demystifying-uplift-models-dc4fb3f927a2

Michel René, Schnakenburg, I., & Martens, T. von. (2019). *Targeting Uplift: An Introduction to Net Scores*. Cham: Springer International Publishing.

Shevchenko, M. (2020). The overview of the basic approaches to solving the uplift modeling problem. Retrieved from https://nbviewer.jupyter.org/github/maks-sh/scikit-uplift/blob/master/notebooks/RetailHero\_EN.ipynb

Siegel, E. (2011, October 19). Persuasion by the Numbers: Optimize Marketing Influence by Predicting It. Retrieved from https://www.predictiveanalyticsworld.com/newyork/2011/presentations/pdf/DAY1\_0900\_siegel-web.pdf

Siegel, E. (2013, January 21). The Real Story Behind Obama's Election Victory. Retrieved from http://www.thefiscaltimes.com/Articles/2013/01/21/The-Real-Story-Behind-Obamas-Election-Victory

Künzel S., Sekhon J., Bickel P., and Yu B. (2019) Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165

Stedman, C. (2013, November 11). How uplift modeling helped Obama's campaign - and can aid marketers - Machine Learning Times - machine learning & data science news. Retrieved from https://www.predictiveanalyticsworld.com/machinelearningtimes/how-uplift-modeling-helped-obamas-campaign-and-can-aid-marketers/2613/

Stichnoth, P. (2018, May 2). Wayfair's Data Science Team and Case Study: Uplift Modeling. Retrieved from https://www.slideshare.net/PatriciaStichnoth/wayfairs-data-science-team-and-case-study-uplift-modeling

Yi, R., & Frost, W. (2019, February 7). Pylift: A Fast Python Package for Uplift Modeling. Retrieved from https://tech.wayfair.com/data-science/2018/10/pylift-a-fast-python-package-for-uplift-modeling/