International Journal Research of Leading Publication (IJLRP)

E-ISSN: 2582-8010 ● Website: www.ijlrp.com ● Email: editor@ijlrp.com

Estimating Customer Potential in Cloud Computing Using Semi-Supervised Learning

Pavan Nithin Mullapudi

Pavannithin123@gmail.com
Data Scientist
Amazon Web Services, Seattle, WA

Abstract

Estimating customer potential is crucial for organizations, especially in the cloud computing domain. Traditional methods often rely on total addressable spend (TAS) estimates, which can be inaccurate or incomplete. This paper explores the use of semi-supervised learning techniques [1] to identify customers with untapped potential on the cloud. By leveraging external datasets and positive-unlabeled (PU) learning algorithms, we aim to improve the accuracy of customer potential estimation and provide a more effective approach for finance organizations in the cloud space. Our results demonstrate that PU learning models, particularly those utilizing technographic embeddings and bagging techniques, can significantly outperform traditional TAS-based methods.

Keywords: Machine Learning, Cloud Computing, Semi-Supervised Learning, Revenue Prediction

1. Introduction

Total addressable spend (TAS) is a vital metric for sales and financial planning in many organizations. Accurate TAS estimation allows businesses to identify opportunities, allocate resources effectively, and optimize revenue generation. However, traditional methods for estimating TAS can be unreliable, leading to discrepancies and suboptimal decision-making.

In the context of cloud computing, accurately assessing customer potential is particularly challenging. Traditional TAS estimates may fail to capture the dynamic nature of cloud adoption and the diverse needs of customers. In our analysis of a large public cloud data set's spend, we identified significant inconsistencies in existing TAS estimates. Specifically, we found that a substantial percentage of TAS estimates for large customers were below their actual cloud revenue, indicating a significant underestimation of their potential. Our analysis found that 37% of existing TAS estimates for large customers are below the large public cloud data set's spend revenue. A key challenge of estimating the quality of a TAS estimate is that we do not know the true TAS of a customer. We consider a customer's TAS plausible if it meets three criteria – TAS should be greater than customer's cloud revenue, TAS should be less than customer's IT budget, and TAS should be less than the company revenue of the customer.

A key challenge in evaluating the quality of TAS estimates is the absence of ground truth data. We lack definitive knowledge of a customer's true TAS, making it difficult to assess the accuracy of existing estimates. To address this challenge, we propose a semi-supervised learning approach that leverages external datasets and machine learning techniques to identify customers with untapped potential for a large public cloud data set's spend.

Our approach focuses on identifying medium-sized customers who share similar external data attributes with existing large customers. We hypothesize that customers with similar technology stacks have similar cloud computing needs. By applying positive-unlabeled (PU) learning algorithms, we can learn from the available data and predict which customers have the potential to become large customers if they fully utilize cloud services.

The goal of this research is to develop a model that outperforms existing TAS estimates in identifying customers with high growth potential for cloud customers. By leveraging external datasets and advanced machine learning techniques, we aim to provide a more accurate and effective approach for customer potential estimation in the cloud computing domain.

2. Related Work

Semi-supervised learning has emerged as a powerful approach for addressing classification problems with limited labeled data. PU learning, in particular, has gained traction in various domains where only positive and unlabeled data are available. This section reviews relevant literature on semi-supervised learning, PU learning, and their applications in customer relationship management and sales potential estimation.

2.1 Semi-Supervised Learning

Semi-supervised learning leverages both labeled and unlabeled data to improve the performance of machine learning models. Several techniques have been developed, including self-training, co-training, and label propagation. These methods aim to exploit the underlying structure of the data to infer labels for the unlabeled instances. Self-training involves iteratively training a classifier on labeled data, predicting labels for unlabeled data, and adding the most confidently predicted unlabeled instances to the labeled set [3]. Co-training uses multiple classifiers trained on different feature sets to iteratively label unlabeled data [11]. Label propagation aims to propagate labels from labeled instances to unlabeled instances based on the similarity between them [9].

More recently, graph-based semi-supervised learning methods have gained popularity. These methods represent data points as nodes in a graph, with edges connecting similar instances. Labels are then propagated through the graph, allowing unlabeled instances to infer their labels from their neighbors[3]. Another line of research has focused on using generative models for semi-supervised learning. These models learn the underlying data distribution and use it to generate labels for unlabeled instances [12].

2.2 Positive-Unlabeled Learning

PU learning is a specific type of semi-supervised learning that deals with datasets containing only positive and unlabeled examples. This scenario arises in various applications, such as identifying potential customers, detecting fraudulent transactions, and discovering new drug targets. PU-learning is a natural fit in medical applications, where a very small set of genes are known to cause or influence specific diseases, and very little is known about other genes. Another typical example is email spam identification where users identify some emails as spam, which make up the positive class, and the rest are considered unknown.

Various algorithms have been proposed for PU learning, including:

- Naive PU learning: This is the most obvious application of PU learning where we treat all unlabeled examples as negatives and train a traditional classifier that, in effect, predicts Pr(s=1|x). The classifier assigns a probability to each of the points (both positive and unlabeled), and among the unlabeled data points, the ones with the highest score are likely to be positives. This approach is simple but can be biased due to the incorrect assumption that all unlabeled examples are negative[4].
- Two-step approach: Identifying reliable negative examples from the unlabeled set and then training a classifier using both positive and negative examples. This method aims to improve the accuracy of the classifier by explicitly identifying negative examples. The performance of this approach depends on the accuracy of the reliable negative identification step [5].
- PU bagging: This algorithm makes use of bagging. It involves splitting the unlabeled data randomly into two sets. We build a naïve classifier with one of the sets (“bootstrap set”) and predict on the other set (out-of-bag set or OOB). This process is repeated a fixed number of times, and a series of binary classifiers are fitted until each of the unlabeled observations has a set of OOB scores. The OOB scores are then averaged to arrive at the final score for all the unlabeled data points. [10].
- Elkanot classifier: Adjusts probabilities to account for the class imbalance inherent in PU learning [4]. This method modifies the output probabilities of the classifier to account for the fact that the proportion of positive examples in the unlabeled set is unknown. By adjusting the probabilities, the classifier can provide more accurate predictions [2].
- Cost-sensitive PU learning: Incorporating costs to penalize misclassification of positive and unlabeled examples. This approach aims to address the class imbalance problem in PU learning by assigning different costs to misclassifying positive and unlabeled examples. By minimizing the cost, the classifier can achieve better performance [Joshi et al., 2001].

2.3 Applications in Customer Relationship Management

Semi-supervised learning techniques have been applied in customer relationship management to address various tasks, such as customer segmentation, churn prediction, and sales lead generation [8]. These methods can leverage both customer data and external datasets to improve the accuracy of predictions and provide valuable insights for businesses.

[6] proposed a PU learning approach for classifying text documents and showed its effectiveness in identifying relevant documents from a large collection of unlabeled documents. [7] applied one-class SVMs for document classification, which is another technique suitable for PU learning scenarios.

3. Model Overview

We propose a ML model as a TAS alternative to identify medium size customers who share external data attributes of existing large customers. Our model leverages machine learning techniques to identify medium-sized customers who share external data attributes with existing large customers of a large public cloud data set's spend. The model predicts whether these customers would at least be large (potentially large) if they chose Cloud for their current needs. Our rationale for focusing on external data sets, specifically technographics, is the underlying hypothesis that customers who have similar tech stacks share similar cloud computing needs all else equal. The model's goal is to achieve better performance versus existing TAS estimates.

4. Data

Our main source of data is third-party data that includes customer-product install pairs (technographic data) across 14k unique products along with top line data such as revenue and IT budget. Our primary data source is third-party data that includes customer-product install pairs (technographic data) across a wide range of products, along with top-line data such as revenue and IT budget.

We use principal component analysis (PCA) to derive input features from the technographic data, and also incorporate collaborative-filtering (CF) based embeddings from the same data set learnt as part of a different project. We use principal component analysis (PCA) to derive input features from the technographic data and incorporate collaborative-filtering (CF) based embeddings from the same dataset.

5. Methodology

Given we know the true large customers, and the goal is to identify potential large customers from the unlabeled data set of medium customers, standard supervised learning techniques cannot be applied because of the lack of negative labels. Given that we know the true large customers, and the goal is to identify potential large customers from the unlabeled dataset of medium-sized customers, standard supervised learning techniques cannot be applied because of the lack of negative labels. We do not know which of the medium customers are potential large customers and which are true medium customers. We apply PU-learning algorithms to learn from the positive and unlabeled datasets. We treat all large customers on a large public cloud data set's spend as true positives and all medium customers as unlabeled.

Within the PU-learning framework, we represent the input as (x,y,s) where x is the input feature data, y is the true label (potential large customer) and s indicates if the observation is labeled or not (true large customer). Within the PU-learning framework, we represent the input as (x,y,s) where x is the input feature data, y is the true label (potential large customer) and s indicates if the observation is labeled or not (true large customer). The fact that only positive observations are labeled can be expressed as

Pr(s=1|x,y=0)=0. The fact that only positive observations are labeled can be expressed as Pr(s=1|x,y=0)=0. Our goal is to learn a classifier that approximates a function f(x)which is as close to Pr(y=1|x) as possible. Our goal is to learn a classifier that approximates a function f(x) which is as close to Pr(y=1|x) as possible. However, this can only be done with a certain set of assumptions about which of the positive examples are labeled. These assumptions are related to the labeling mechanism and how the unlabeled examples are treated. These assumptions are related to the labeling mechanism and how the unlabeled examples are treated.

We implemented three different PU techniques to CUSP with a Random Forest classifier as the base classifier:

1. Naive PU classifier - This is the most obvious application of PU learning where we treat all unlabeled examples as negatives and train a traditional classifier that in effect predicts Pr(s=1|x). The classifier will assign a probability to each of the points (both positive and unlabeled) and among the unlabeled data points, the ones with the highest score are likely to be positives.
2. PU bagging technique - This is a more sophisticated algorithm that makes use of bagging. It involves splitting the unlabeled data randomly into two sets. We build a naïve classifier with one of the sets ("bootstrap set") and predict on the other set (out-of-bag set or OOB). This process is repeated a fixed number of times and a series of binary classifiers are fitted until each of the unlabeled observations has a set of OOB scores. The OOB scores are then averaged to arrive at the final score for all the unlabeled data points.
3. Label propagation techniques - This algorithm starts off by training with the known positives and a "reliable" set of negatives from the unlabeled dataset. These reliable sets of negatives can either be based on domain knowledge or an output of the Naive classifier. The algorithm then iteratively labels the unlabeled observations based on rules around predicted probabilities. If the predicted probabilities for any of the unlabeled are above or below the range of known positives, then label them as negatives and positives respectively. This process is repeated until a stopping criterion is met.

6. Model Evaluation

To evaluate the performance of the model, we first split the large dataset (2416 observations) in half and "mask/hide" 1208 observations with medium customers to achieve a 1:40 ratio of hidden large customers to medium customers in the target dataset. To evaluate the model's performance, we first split the large customer dataset in half and "mask/hide" observations with medium-sized customers to achieve a specific ratio of hidden large customers to medium-sized customers in the target dataset. The 1:40 ratio is an assumption of the prior distribution of medium customers vs large customers. This ratio is an assumption of the prior distribution of medium-sized vs. large customers. The algorithms are evaluated based on how well the hidden large customers are retrieved as potential large customers in the target set. The algorithms are evaluated based on how well the hidden large customers are retrieved as potential large customers in the target set. The algorithm is bootstrapped multiple times drawing a different set of hidden large customers and the results are averaged to arrive at performance metrics.

We compare the performance of the model versus baseline models that use TAS and IT budget. We compare the performance of the model versus baseline models that use TAS and IT budget. The key

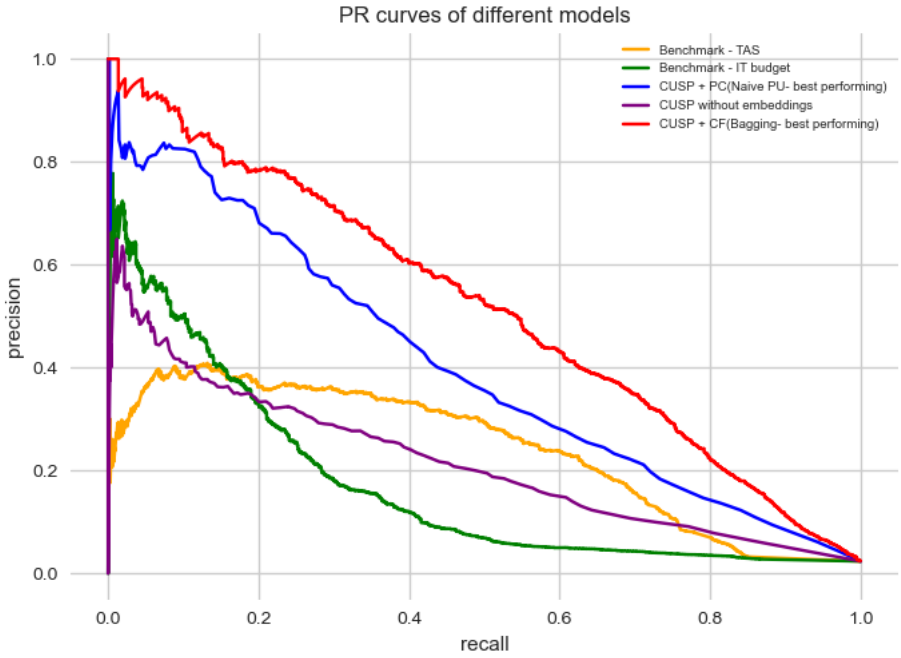
model comparison metric we use is average precision and precision for specific levels of recall. The key model comparison metric we use is average precision and precision for specific levels of recall. Precision is defined as the percentage of potential large customers that are true large customers. Recall is defined as the proportion of the complete set of hidden large customers that are identified as potential large customers. Precision and recall are dependent on the probability threshold chosen to classify customers as potential large customers. Average precision, a measure of the precision-recall curve, is the weighted mean of precision at each probability threshold, with the increase in recall from the previous threshold. Depending on the precision/recall requirement, we will choose a different probability. We only show here the results from naïve and bagging PU techniques because we encountered convergence issues with label propagation. We only show here the results from naïve and bagging PU techniques because we encountered convergence issues with label propagation.

Table 1 summarizes model performance. Our model with bagging PU trained with CF based technographic embeddings achieves an average precision of 50%, 27pp higher than a baseline that uses existing TAS. If we use PCA embeddings, the average precision is at 42%, still a significant lift of 19 pp over the baseline. Table 1 summarizes model performance. The model with bagging PU trained with CF-based technographic embeddings achieves a high average precision, significantly higher than a baseline that uses existing TAS. Bagging technique improves over the naïve technique (50% vs 46% average precision). If we use PCA embeddings, the average precision is at 42%, still a significant lift of 19 pp over the baseline. Using PCA embeddings also yields a substantial improvement over the baseline. However, the bagging technique does not improve performance when trained with PCA embeddings. We also see that not incorporating technographic data (but using high level data such as revenue/IT budget) does not improve over the baseline, and leads to an average precision of 20%, which is 3pp lower than the benchmark TAS model. The bagging technique improves over the naïve technique. We also see that not incorporating technographic data (but using high level data such as revenue/IT budget) does not improve over the baseline, and leads to an average precision of 20%, which is 3pp lower than the benchmark TAS model. However, the bagging technique does not improve performance when trained with PCA embeddings. Not incorporating technographic data (but using high-level data such as revenue/IT budget) does not improve over the baseline.

Table 1 Model performance versus baseline

Model	PU technique	Average precision
CUSP with Collaborative Filtering embeddings	Bagging	0.5
CUSP with Collaborative Filtering embeddings	Naïve	0.46
CUSP with PCA embeddings	Bagging	0.38
CUSP with PCA embeddings	Naïve	0.39
CUSP without embeddings	Naïve	0.2
Benchmark - TAS		0.23
Benchmark - IT Budget		0.17

Figure 1 Precision/Recall curves



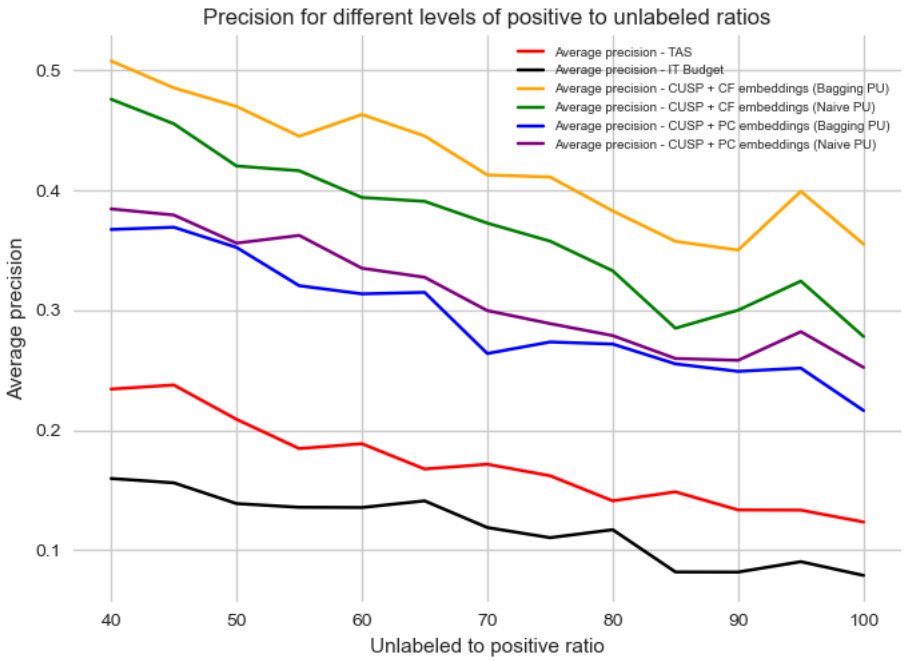
To evaluate the robustness of the performance measure, we vary the ratio of positives to unlabeled, bootstrap the algorithm multiple times to arrive at average precisions for the chosen ratios. To evaluate the robustness of the performance measure, we vary the ratio of positives to unlabeled, bootstrap the algorithm multiple times to arrive at average precisions for the chosen ratios. For this sensitivity analysis, we keep the ratio of P/U and hidden ratio (unmasked to masked positives) the same. For this sensitivity analysis, we keep the ratio of P/U and hidden ratio (unmasked to masked positives) the same. We see that the pattern we observed with a 1:40 ratio holds up for different ratios in figure 2. We see that the pattern we observed holds up for different ratios in figure 2. CUSP model that uses CF embeddings and bagging PU techniques outperforms all the other models. The CUSP model that uses CF embeddings and bagging PU techniques outperforms all the other models. The bagging technique is able to outperform the naïve technique by 5% average precision on average when using CF embeddings. The bagging technique can outperform the naïve technique by a significant margin average precision on average when using CF embeddings.

Table 2 Precision vs. unlabeled/positive ratio

Unlabeled/Positive ratio	Average precision - TAS	Average precision - IT Budget	Average precision - CUSP + PCA + Bagging PU	Average precision - CUSP + PCA + Naive PU	Average precision - CUSP + CF + Bagging PU	Average precision - CUSP + CF + Naive PU
40	0.23	0.16	0.37	0.38	0.51	0.48
45	0.24	0.16	0.37	0.38	0.49	0.46

50	0.21	0.14	0.35	0.36	0.47	0.42
55	0.18	0.14	0.32	0.36	0.45	0.42
60	0.19	0.14	0.31	0.34	0.46	0.39
65	0.17	0.14	0.31	0.33	0.45	0.39
70	0.17	0.12	0.26	0.30	0.41	0.37
75	0.16	0.11	0.27	0.29	0.41	0.36
80	0.14	0.12	0.27	0.28	0.38	0.33
85	0.15	0.08	0.26	0.26	0.36	0.29
90	0.13	0.08	0.25	0.26	0.35	0.30
95	0.13	0.09	0.25	0.28	0.40	0.32
100	0.12	0.08	0.22	0.25	0.36	0.28

Figure 2 Precision for different unlabeled to positive ratios



7. Conclusion and Future Work

While the current model shows a significant improvement over using TAS or IT budget, there is room for improvement in the areas of customer data embeddings, PU learning algorithms and evaluation. The current model shows a significant improvement over using TAS or IT budget alone. There remains room for improvement in the areas of customer data embeddings, PU learning algorithms, and evaluation. Customer embeddings can be enhanced by augmenting external datasets with internal data sets that represent cloud behavior of customers already on a large public cloud data set's spend. Customer embeddings can be enhanced by augmenting external datasets with internal datasets that represent the cloud behavior of customers already on a large public cloud data set's spend. The current model uses a random forest classifier as the base classifier for the PU-learning methods. The current model uses a random forest classifier as the base classifier for the PU-learning methods. We are evaluating alternate

classifiers. We are evaluating alternate classifiers. The algorithm evaluation can be made more robust by evaluating on reliable negatives in addition to the positives. The algorithm evaluation can be made more robust by evaluating reliable negatives in addition to the positives. These reliable negatives can be gathered from domain experts. These reliable negatives can be gathered from domain experts. As mentioned earlier in the evaluation stage, the sensitivity analysis was done by keeping P/U and the hidden ratio the same. As mentioned earlier in the evaluation stage, the sensitivity analysis was done by keeping P/U and the hidden ratio the same. However, these can be different and the model needs to be evaluated for the scenarios where these ratios are different. However, these can be different, and the model needs to be evaluated for the scenarios where these ratios are different. We have approached this problem from a PU perspective but with the help of a reliable set of negatives, PNU (positive-negative-unlabeled) techniques can also be applied and compared against the current model's performance. We have approached this problem from a PU perspective, but with the help of a reliable set of negatives, PNU (positive-negative-unlabeled) techniques can also be applied and compared against the current model's performance.

Our framework to apply PU-learning to customer potential can be extended to other areas of interest. Our framework to apply PU-learning to customer potential can be extended to other areas of interest. We have multiple use-cases where we know a small set of customers who meet success criteria that may include early service adoption, time-series usage patterns that show a definite trend, and workload adoption and growth. We have multiple use-cases where we know a small set of customers who meet success criteria that may include early service adoption, time-series usage patterns that show a definite trend, and workload adoption and growth.

References

1. Chapelle, O., Scholkopf, B., & Zien, A. (2006). Semi-Supervised Learning. *IEEE Transactions on Neural Networks*, *17*(3), 683-697.
2. Bekker, J., & Davis, J. (2020). Learning from Positive and Unlabeled Data: A Survey. *Data Mining and Knowledge Discovery*, *34*(3), 719-760.
3. Zhu, X. (2005). Semi-Supervised Learning Literature Survey. University of Wisconsin-Madison, Department of Computer Sciences.
4. Elkan, C., & Noto, K. (2008). Learning Classifiers from Only Positive and Unlabeled Data. *Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining*, 213-220.
5. Liu, B., Lee, W. S., Yu, P. S., & Li, X. (2003). Partially Supervised Classification of Text Documents. *Proceedings of the 20th International Conference on Machine Learning*, 488-495.
6. Li, X., & Liu, B. (2003). Learning to Classify Texts Using Positive and Unlabeled Data. *Proceedings of the 19th International Conference on Computational Linguistics*, 426-432.
7. Manevitz, L. M., & Yousef, M. (2001). One-Class SVMs for Document Classification. *Journal of Machine Learning Research*, *2*, 139-154.
8. Larose, D. T., & Larose, C. D. (2014). *Discovering Knowledge in Data: An Introduction to Data Mining*. John Wiley & Sons.
9. Zhou, D., Bousquet, O., Lal, T. N., Weston, J., &Schölkopf, B. (2004). Learning with Local and Global Consistency. *Advances in Neural Information Processing Systems*, *16*, 321-328.

10. Aggarwal, C. C. (2015). Data Mining: The Textbook. Springer.
11. Blum, A., & Mitchell, T. (1998). Combining Labeled and Unlabeled Data with Co-Training. *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 92-100.
12. Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems* (pp. 3581-3589).