Using AutoML to Conduct Propensity Score Matching

Mark Zimmerman Faculty Advisor: Dr. Wenjun Zhou

Background

Researchers are often faced with the need to estimate causal effect from observational data. Propensity score matching (PSM) has been one of the most commonly used techniques for causal analysis in observational studies.

A main challenge in making causal inference with observational data is that the treatments are not randomly assigned. PSM constructs a pseudo-experiment by selecting matched pairs of (self-selected) treatment and control samples with similar propensity of being treated at the baseline.

These propensity scores can be produced with predictive modeling using baseline characteristics as predictors. Logistic regression is among the most widely used predictive model for generating propensity scores.

Study Context

Will updates help improve user engagement with mobile apps? Researchers have tracked a large number of mobile apps listed in App Stores from Time 1 (T1) to Time 2 (T2), separated by 3 months. Some of these apps had at least one update between the two time points. The research team wondered if a rolled-out update actually helped obtain more ratings and/or higher average ratings. This information is valuable to app developers in planning.

Since app developers self-selected whether or not to update, the research team wanted to use PSM to create a balanced sample of matched pairs. We used logistic regression in each segment to predict if an app would update. This is a binary classification problem.

Predictors are: (1) static app-level information such as if the app is free, and the category and subcategory of the app, (2) prior update history, and (3) prior ratings. (1) are constant variables across all partitions, (2) and (3) are important predictors but are only available in certain partitions.

Since prior update and ratings (at baseline) are important predictors, we divided the samples into the four partitions:

			Sample Size	
Data Subsets	Updated	Rated	n	%
AE (Active & Engaged)	Y	Y	34652	17.9%
AU (Active & Unengaged)	Y	N	90344	46.7%
IE (Inactive & Engaged)	N	Y	6497	3.4%
IU (Inactive & Unengaged)	N	N	61888	32.0%
			193381	

AE = Updated and Rated before T1
AU = Updated but Not Rated before T1
IE = Not Updated but Rated before T1
IU = Not Updated and Not Rated before T1

Research Question

In our context of predicting if an app would have an update before T2, using only T1 information as predictors, will an Auto ML model, a Support Vector Machine model, a Recursive Partitioning and Regression Trees model, or an Artificial Neural Network model outperform logistic regression?

Methods

(1) Automated Machine Learning (AutoML) – automates with machine learning to provide faster and more accurate results than manually-coded algorithms. The Auto ML model is a newer model that makes machine learning available for non-machine learning experts (Freiburg-Hannover 2022).

In this project, we used H2O Auto ML which is designed to have as few parameters as possible to simplify the process for the user. H2O Auto ML performs a large number of modeling-related tasks that would usually require a lengthy amount of code (Freiburg-Hannover 2022). Using this method was beneficial for the speed of the project and can easily be duplicated with other datasets. Here is a snippet of code showing the process of making predictions using H2O Auto ML:

```
# start h2o cluster
invisible(h2o.init())

# convert data as h2o type
train_h = as.h2o(train)
test_h = as.h2o(test)

# Run AutoML for 20 base models
aml = h2o.automl(x = pred, y = 'Update', training_frame = train_h, max_models = 20,
                 seed = 1, max_runtime_secs = 20)

# prediction result on test data
test.prediction = h2o.predict(aml@leader, test_h) %>% as.data.frame()

# close h2o connection
h2o.shutdown(prompt = FALSE)
```

Other machine learning models under consideration include:
(2) Support Vector Machine (SVM) – finds a hyperplane that best divides the data into two classes (“Support Vector Machines”).
(3) Recursive Partitioning and Regression Trees (Rpart) – recursive partitioning to improve purity of class distribution in sub nodes (Schmuller 2018).
(4) Artificial Neural Network – connects nodes with weights estimated to predict the output (Chandra 2018).

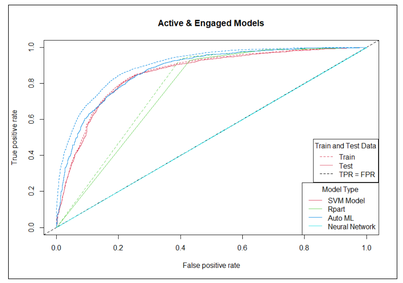
Handling Class Imbalance: SVM, Rpart, and Neural Network models used a weighted version to even class imbalance. There were many more apps that were not updated between T1 and T2, so these models were balanced to account for that difference in classes.

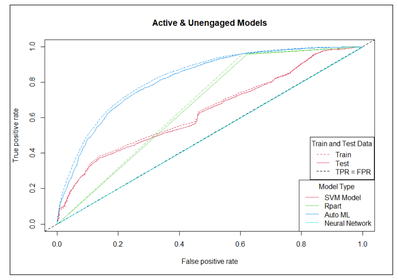
Preventing Overfit: Each partition was split into training and testing data to minimize the effects of data discrepancies and understand the characteristics of the model better.

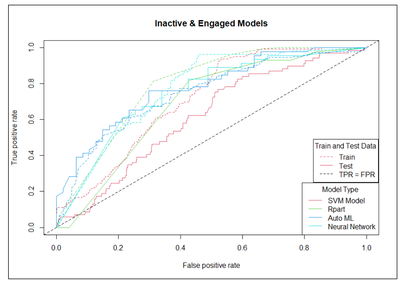
Performance Metrics:
Numeric: accuracy, precision, recall, and F1 score.
Graphic: Receiver Operating Characteristics (ROC) curves.

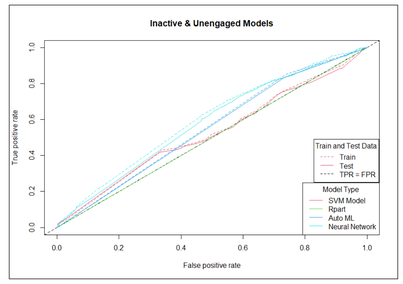
Results

ROC Curves









Each Models Percent Improvement Against Logistic Regression

Final result - % Improvement of Each Model Over the Baseline				
	% Improvement Against Logistic Regression			
	Accuracy	Precision	Recall	F1
Auto ML	16.97%	197.91%	-5.46%	89.71%
SVM	36.36%	31.67%	-85.09%	-61.24%
Rpart	-8.06%	103.81%	23.54%	73.56%
Neural Network	-12.52%	97.40%	15.01%	65.10%

Conclusion

Looking at the results, we concluded that the Auto ML model was the best model to conduct the PSM Estimation. This model gave us the most consistent results across the partitions. Instead of using a model that is common in research, we wanted to go with a newer model. Widening our model usage to less-used models allowed for us to find better results.

Using the predictions from the Auto ML model, the PSM process can continue with the most accurate predictions available.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

References

AutoML, Freiburg-Hannover, 2022, https://www.automl.org/automl/.

Chandra, Reetesh. “Neural Networks: A Comprehensive Guide.” *UpGrad Blog*, 6 Feb. 2018, https://www.upgrad.com/blog/neural-networks-for-dummies-a-comprehensive-guide/#:~:text=A%20neural%20network%20is%20built,idea)%20is%20connected%20via%20synapses.

Schmuller, Joseph. “Decision Trees in R.” *R Projects*, 9 Apr. 2018, https://www.dummies.com/article/technology/programming-web-design/r/decision-trees-r-251602/.

“Support Vector Machines.” *Support Vector Machines for Dummies; A Simple Explanation - AYLIEN News API*, Aylien, https://aylien.com/blog/support-vector-machines-for-dummies-a-simple-explanation#:~:text=Support%20Vector%20Machines%20%2D%20What%20are,focus%20on%20in%20this%20post.