APPENDIXES

THE RESPONSE OF CONSUMER SPENDING TO CHANGES IN GASOLINE PRICES

Michael Gelman, Yuriy Gorodnichenko, Shachar Kariv, Dmitri Koustas,

Matthew D. Shapiro, Dan Silverman, and Steven Tadelis

August 8, 2016

Appendix A: Construction of spending in the app data

This appendix discusses the data and provides details on how we prepare the data for statistical analyses.

We received anonymized data directly from the personal financial management service provider (app). The process by which the company acquires the data can differ across users, account providers (e.g., Bank of America, Wells Fargo) and time. For some account providers, the data are scraped from the website of an account provider, and in other cases a direct feed is received from the account provider. All account numbers and other personal identifying information is removed by the app company before we receive the data. Otherwise, we receive the data exactly as it is received by the app. The table below summarizes the key variables in the data that are used in our analysis:

User_id	-	Anonymous identifier constructed by the personal financial management service
Posted_date	-	Date a transaction was recorded
Account_Provider_Id	-	An identifier for a specific account provider (e.g. Bank of America)
Account_Type	-	An indicator for whether an account is a checking account, savings account, credit card, or other account.
Transaction_Amount	-	The amount of the transaction
Is_Credit	-	Whether the transaction was a credit or a debit
Transaction_Description	-	A string variable describing the transaction.

Our cleaning processes proceeds in steps outlined below:

I. Remove likely duplicates +/- 3 days

Because the data may include pending transactions, a given spending may show up multiple times in different transactions. For instance, if a transaction was pending on one day, and posted the next day, we could see a duplicate recording of the same transaction in the data, which would not reflect actual spending.

Some account providers indicate whether a transaction is pending or posted, and we first remove all transactions that are flagged as pending, or contain the word “pending” in the transaction string. Since many account providers do not indicate whether a transaction is pending, and since this information also varies across time, we deal with this problem by

removing transactions that are duplicates on the dimensions of {User_Id, Account_Provider_Id, Account_Type, Is_Credit, Transaction_Amount, Transaction_Description} over a 3 day window. This removes approximately 5% of transactions. Some of these transactions could be non-duplicates (for instance, if someone buys the exact same item every day), and so these transactions will also be removed by this procedure. Using the data with likely duplicates removed, we next proceed to calculate total spending and total income, which we aggregate to the weekly level.

II. Construct variables used in analysis and aggregate to weekly level

The transactions contain every single inflow and outflow from a household's account, some of which are not "consumption." Two problematic types of transactions are transfers across accounts and credit card payments. In most cases, transfers across accounts can be identified from the transaction strings, since they are commonly flagged as "transfer," "xfer," "tfr," "xfr" or "trnsfr." We remove all transactions with these words appearing in their description.

Credit card payments reflect lagged spending that we have already included in our measure of total spending, since we can see the individual purchases that make up the credit card payment on the credit card. Therefore, we wish to identify and remove these payments. We identify credit card payments as debits appearing on a non-credit-card account that also appears as a credit to a credit card, and remove these.

We also remove the largest transaction greater than $1,000 in a weekly window, since these transactions appear to be predominantly credit card payments and transfers missed by our procedure. As a caveat, this likely also removes mortgage payments (committed spending), extremely large durables purchases (such as a down payment on a car, although we would still see car payments), and payments on tax liabilities. To summarize, our measure of "total spending" used in this paper is defined as:

{Total spending} = {Total Account Debits} – {Flagged Duplicates} – {Transfers} – {Credit Card Payments} – {Largest Transaction > $1,000 (if any)}.

Appendix B: Machine Learning Classification of Transactions

As discussed in Appendix 1, the data we receive contain raw transaction strings. These transaction strings differ across account providers in their context. We wish to identify spending that comes from gasoline. Identifying to which of a set of categories an observation belongs, based on information in the transaction descriptions, is a classic "classification" problem in machine learning.

We seek a simple machine learning (ML) model to identify gasoline spending in the data. For this to work, we require a "training" set of data containing observations whose category membership is known. Fortunately, two account providers in our data categorize the transactions into merchant category codes (MCCs) directly in the transaction strings. These two cards represent about 3% of all transactions. As discussed in the text, it is virtually impossible to separate out our main MCC of interest, 5541, "Automated Fuel Dispensers" from MCC code 5542, "Service Stations," which in practice covers gasoline stations with convenience stores.[1] Because distinguishing gasoline purchases classified as 5542 or 5541 is nearly impossible with the information in transaction descriptions,[2] we group transactions with these two codes together.

Before proceeding with the details of the machine learning model, it is useful to discuss an alternative approach that identifies all gasoline stations in the data through string matching techniques. To see why this is infeasible, consider that the 100 most popular gasoline station strings cover approximately 50% of the total market share in the transactions where we know the MCC codes. Scaling up is costly: to get 90 percent of the market share, we would need to search for over 30,000 strings (Appendix Figure 1). Moreover, since other spending can often have similar transaction descriptions, it is hard to know what strings minimize noise while maximizing predictive power. The machine learning algorithm thus helps discipline the approach of what transaction strings contain the most useful information. The machine learning procedure proceeds in 3 steps: training, testing, and application.

Machine learning requires both a "training" data set—data actually used to fit a classification model-—and a "testing" data set to evaluate the out of sample performance of the model. In the training step, we build a prediction model using data with the MCC codes (i.e. data where classification is known). We use the larger of the two account providers as the training data

[1] To be clear, "Service Stations" do not include services such as auto repairs, motor oil change, etc.
[2] E.g., a transaction string with word "Chevron" or "Exxon" could be classified as either MCC 5541 or MCC 5542.

set, and test the performance of the model on the smaller account. We explicitly set aside the second card as the training data set because transaction strings, which we will feed into the model to classify the data, can differ across account providers. Therefore, if we train on data from the two accounts, we may fit our two cards extremely well, but we may have a poor "out of sample" fit of our model.

The classification algorithm we use is known as a random forest classifier, which fits a number of separate decision trees. A decision tree is a series of classification rules that ultimately lead to a classification of a purchase as gasoline or not. The rules, determined by the algorithm, minimizes the decrease in accuracy when a particular model "feature" is removed. The features we use are the transaction values and individual words in the transaction strings (this approach is known as "bag of words"), after some basic string cleaning. We limit the number of features to 20,000 words, and transaction amounts rounded to the nearest 50 cents. An example decision tree is shown in Appendix Figure 2.

In this example, the most important single word is "oil." If a transaction string contains the word oil, the classification rule is to move to the right, otherwise the rule is to move to the left. If the string does not contain the word oil, the next most important single word is "exxonmobil." The tree keeps going until all the data are classified.

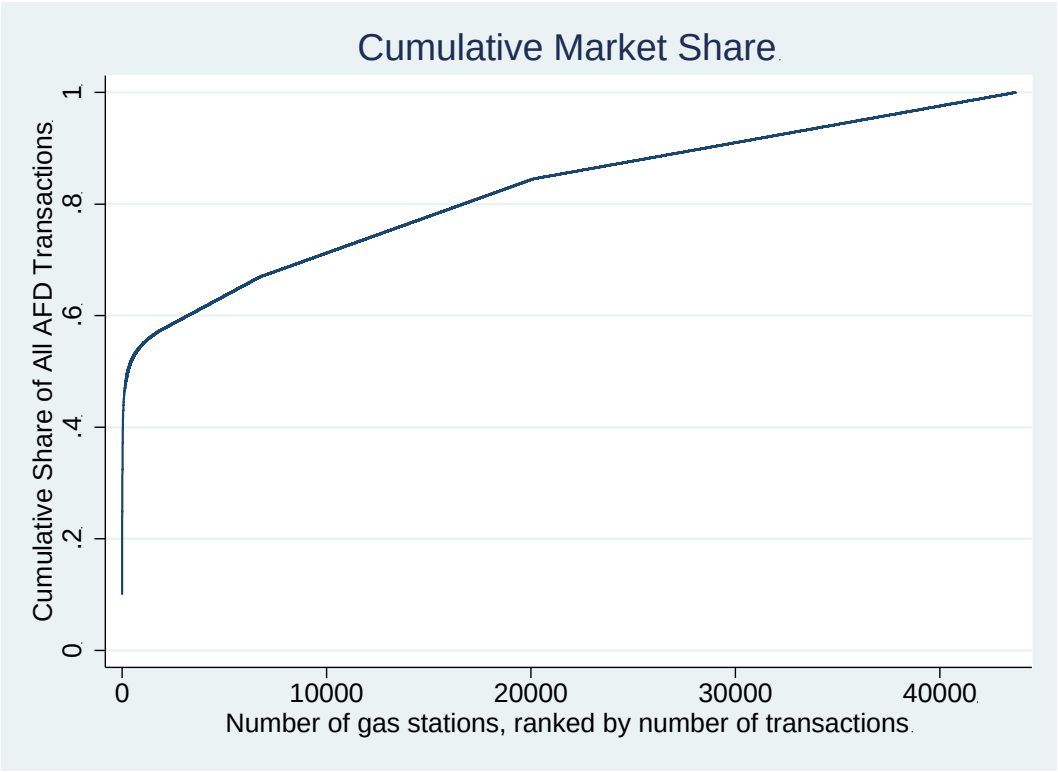
Whether a transaction is classified as gasoline spending or not is simply the majority vote over a number of decision trees. This is known as a "white box algorithm" because the model determines optimal decision rules that we can see. We use prebuilt packages from the python machine learning toolkit.[3]

The results of the model are shown in Appendix Table 1. The model predicts 292,997/(292,997+26,553) = 92% of automated fuel dispenser and service station transactions. The ratio of mis-classfications to correct classifications is (30,080+26,553)/292,997=19%.

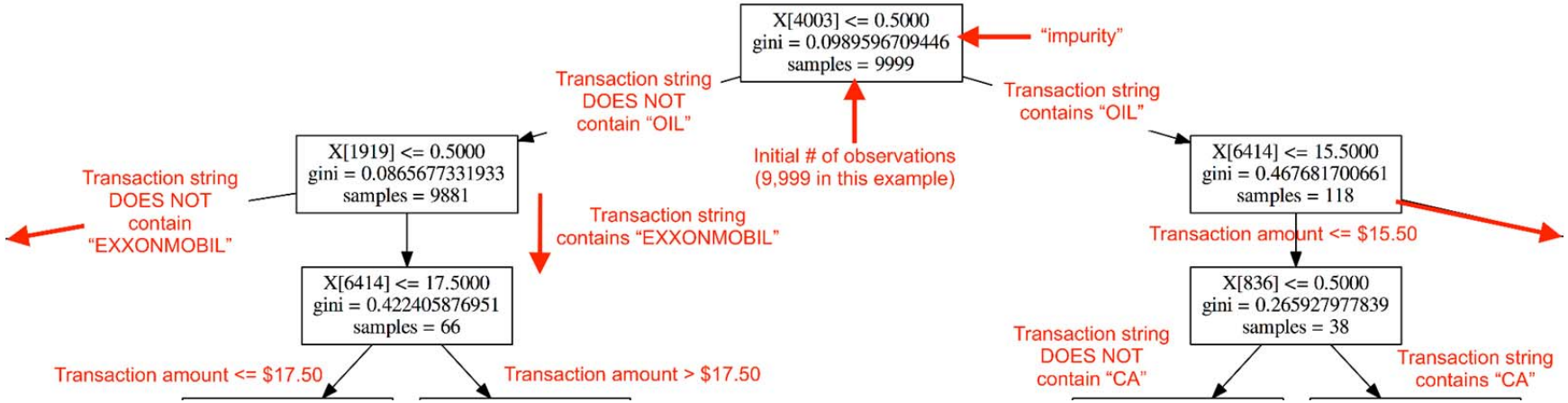
In summary, the ML approach is able to correctly classify over 90% of gasoline spending in the test data. If a human were to do this, she would need to identify over 30,000 strings. In addition, the model correctly classifies over 99.5% of the gasoline stations that would have been captured in an alternative approach of identifying the 100 largest gasoline stations by market share.

[3] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Appendix Figure 1



Appendix Figure 2. An example machine learning decision tree



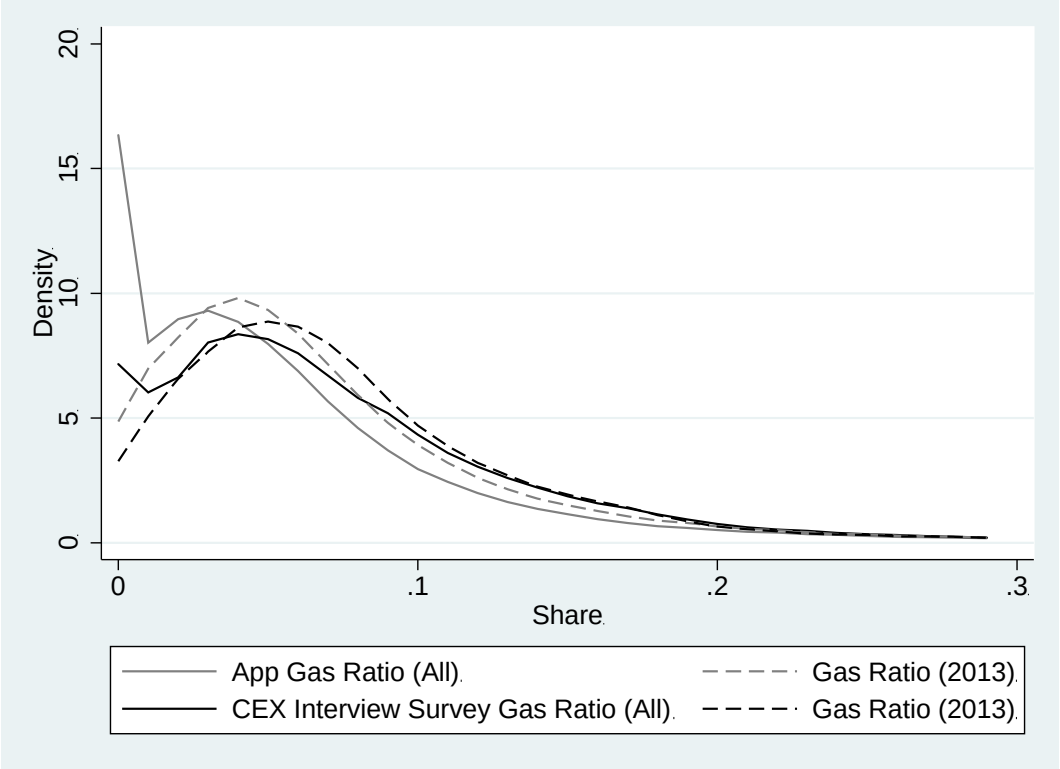
Appendix Table 1. Confusion Matrix

		Actual gasoline spending	
		No	Yes
Predicted gasoline spending	No	2,741,524	26,553
	Yes	30,080	292,997

Notes: Table shows the four possible outcomes for our testing data set which is not used in any way to train the model, as described in the text. The rows "Predicted gasoline spending" refer to the binary prediction from the model as not gasoline, "no," or gasoline, "yes". Actual gasoline refers to the "truth," which is known for the case of our testing dataset.

Appendix 3. Additional Tables and Figures

Appendix Figure 3. Distribution of Ratio of Gasoline to Non-Gasoline Spending, 2013Q1-2014Q4



Note: the figure shows the quarterly gasoline to non-gasoline spending distribution in the app data and the CEX interview survey (solid lines), and the same ratio calculated over all of 2013 (dashed lines).

Appendix Table 2. Demographic composition.

	App	American Community Survey
Sex		
Male	59.93	48.59
Female	40.07	51.41
Age		
18-20	0.59	5.72
21-24	5.26	7.36
25-34	37.85	17.48
35-44	30.06	17.03
45-54	15.00	18.39
55-64	7.76	16.06
65+	3.48	17.95
Highest degree		
Less than college	69.95	62.86
College	24.07	26.22
Graduate school	5.98	10.92
Census Bureau region		
Northeast	20.61	17.77
Midwest	14.62	21.45
South	36.66	37.36
West	28.11	23.43

Source: Gelman et al. (2014).