Allocation prediction system

Richard Lawton

08/02/2020

First created on 2020-02-08. Updated on 2020-02-09

INTRODUCTION

Reconciliation of statements from financial institutions with internal records and appropriate allocation of expense/income categories is a routine part of financial management whether for personal or business use. This project will focus on the automation of allocation to these categories by means of machine learning. It will be developed in the context of personal finance, but it is anticipated that an extension to business would be straightforward.

The acquisition of statement lines from a financial institution is largely automated, but the work of allocating to expense/income categories is a human task. Some packages allow for rules-based allocation, but that does depend on human intervention to devise appropriate rules and update them.

To what extent is it possible or desirable to automate this process? Some observations:

1.) Any schema for allocation is to an extent an arbitrary device expressing the particular intention of the creator. Inevitably this expression will not have considered many contingencies, with the result that, even for the original creator, the allocation process may not be consistent. For example does a meal eaten at a fast food restaurant while on holiday count as "food", "entertainment" or "holiday"? The resolution of such questions is often a gradual process of developing possibly unspoken rules. There will always therefore need to be human review of machine-based allocations to answer the question: "Is this working out the way I planned?"

2.) Some categorisations will never be possible from a statement line. Obviously new types of transactions will have no history on which to base an allocation. But more profoundly statement lines are limited in their information. For example, a credit card purchase at a supermarket may be "food" or "toiletries" or "household cleaning", or even a combination of these categories. Humanly it is not possible to allocate such an expense from the statement line alone. It will be necessary to see in addition a detailed invoice of what was bought. Typically this will not be available on line, and certainly not from the credit card company.

Is a bad guess better than not trying at all? Probably not. An allocation fail will automatically demand human intervention, whereas a mis-allocation may slip past human scrutiny and cause problems at a future date when the details of the transaction are no longer fresh in anyone's mind.

In such a case the desired response of an automated system should be "insufficient information" rather than a hopeful guess. So a key requirement is to allow our system to fail to make an allocation.

3.) Beyond these two considerations there is undoubtedly a wide range of transactions which can be securely allocated on the basis of machine recognised patterns which would alleviate the human effort of this work.

METHODOLOGY

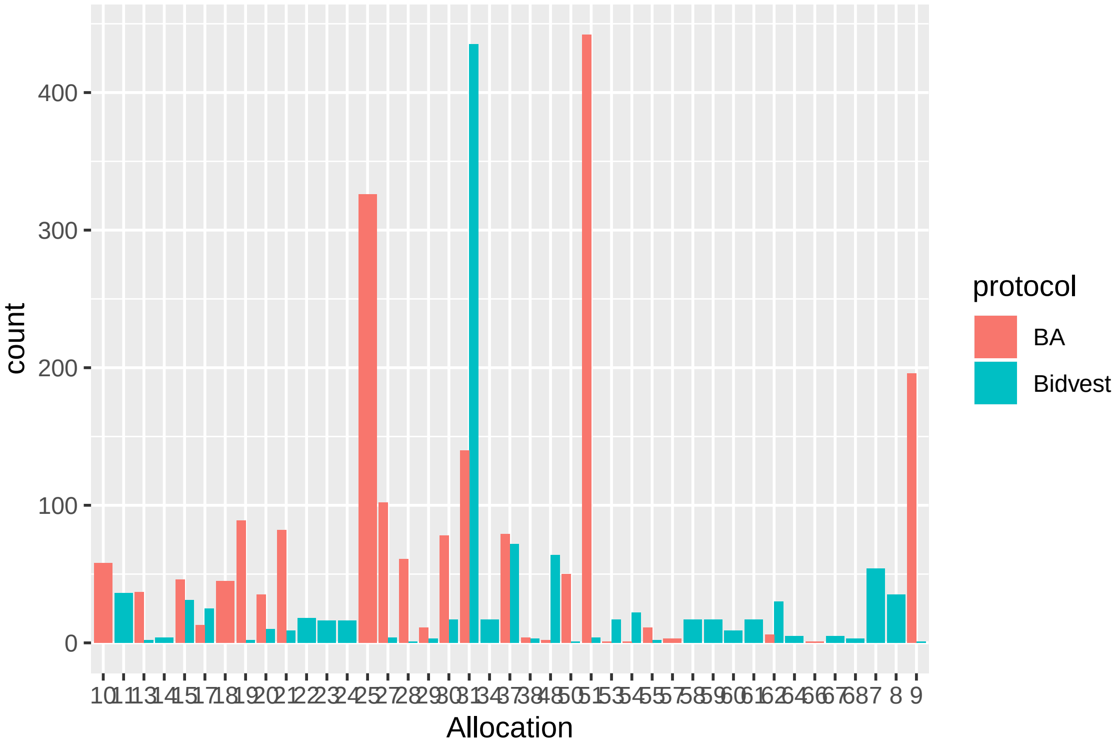
Dataset

The data set (taken from my personal financial records stored in a MySQL database) consists of 6770 records involving multiple current accounts, credit cards and savings accounts over 5 years. Cash transactions were ignored as they can never feature in a statement. The data set has the following columns:

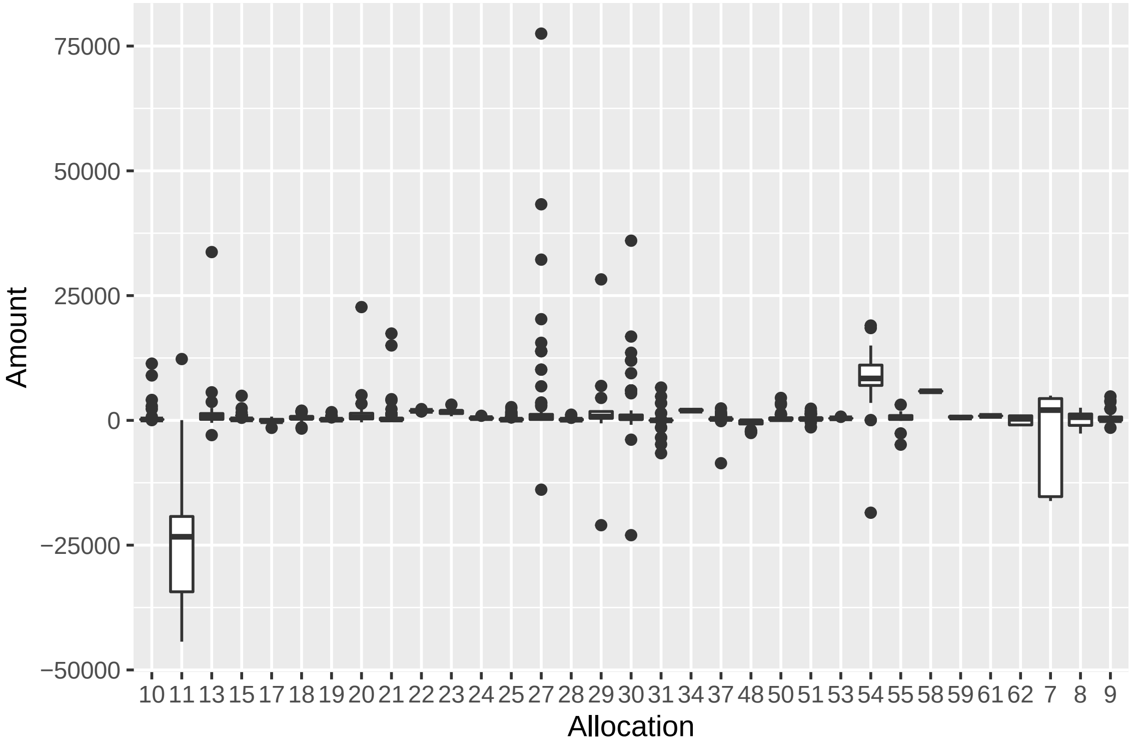
- Transaction id
- Date
- Amount (positive as expense, negative as income)
- Reference for card, bank account
- Reference for allocation
- Commentary taken from the statement line

An examination of statements reveals at once that the style of comment lines varies significantly between institutions. The dataset was therefore reduced to the two institutions I currently deal with, and divided between them, yielding 1919 records for a BA branded MasterCard and 1024 records for accounts held with Bidvest Bank. We will build separate models for each institution.

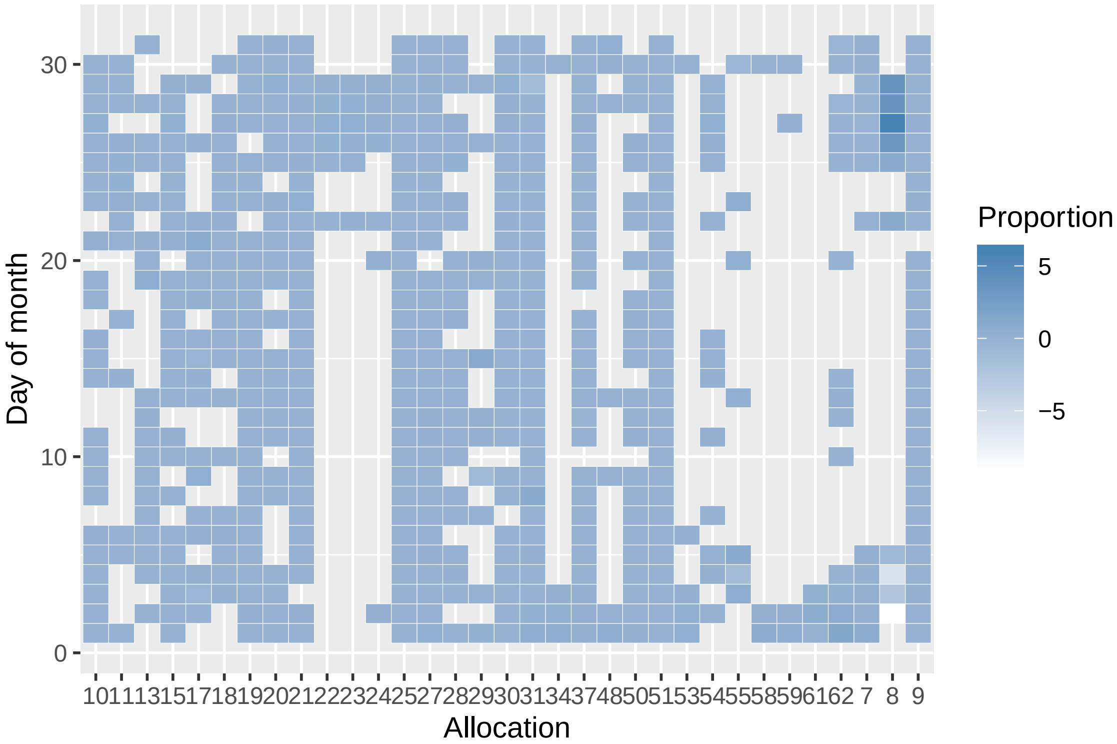
Distribution of transactions between these allocation varies enormously:



The range of amounts allocated also varies:



Transactions with different allocations may also occur at distinct times of the month:



Choice of predictors

1.) Day of month is clearly predictive of the appropriate allocation for the transaction. Certain payments occur at month end only for example.

2.) Amount can also be a clear predictor of category. Filling station expenses for the car would typically be in excess of R600, while for the motorbike (with a much smaller tank capacity), the expense would be under R300.

3.) It is expect that the commentary field will provide most information, however. The text may include:

- a descriptor such as "fee"
- an identification of a 3rd party (merchant or payee)
- a personal annotation that I have added in setting up an EFT with the bank
- a geographic indicator
- various other letters, symbols and numbers

It is rare that a commentary field will ever be reproduced exactly – for this reason the field as it stands is not going to be a useful predictor. We need instead to extract components. The procedure for doing this was as follows:

a) Break up the field into fragments divided by spaces (multiple), '*' or '#'
b) Gather the dataset into tidy format, with a row for each fragment-transaction combination
c) Eliminate all single character fragments
d) Eliminate all fragments that are rare, and are therefore unhelpful predictors
e) Eliminate all fragments that appear in multiple (>2) allocations and are also unhelpful predictors
f) Eliminate all remaining duplicate entries (where the same fragment occurs twice in the same transaction)
g) Spread back into a wide format with each fragment represented by its own column, with value 1 for present or 0 for absent in that particular transaction.

At this point we have 133 fragment columns remaining (BA) or 43 (Bidvest). Can we reduce these further. It is evident that some fragments are highly correlated. These were identified ($r > 0.9$) and one of each pair eliminated, leaving 93 fragment columns (BA) and 28 for Bidvest.

Modelling strategy

Certain features direct us towards a particular modelling strategy:

1.) We are left with high dimensionality, which indicates some kind of decision tree model, rather than smoothing.

2.) The problem is one of classification, BUT we wish to apply some probability cut off to answers we will accept. For this reason we will use a regression framework, which gives us direct access to probabilities in a decision tree, rather than a simple majority vote.

3.) We want a model that can give us a "none of the above" result. This is not simply one more category, but a failure to match any of the existing categories.

Our innovative strategy will be as follows then:

1.) We will construct a decision tree for each category based on the predictors we have identified.

2.) The decision will be a binary question: this particular category or not?

3.) We will record all outcomes as “No” unless the regression value for “Yes” exceeds our threshold probability (set at 0.7)

4.) If the outcome is “Yes” this category will be recorded as the predicted allocation. If “No”, the model will chain into the next category.

5.) If none of the categories are found to match, then the predicted allocation is “I don’t know”.

This chaining of decision trees is not a regular feature of the *rpart* package. Furthermore we will wish to export our model back to a SQL environment, which understands only tables, not trees. It will therefore be necessary to extract from the individual *rpart* tree models, sufficient information to construct an equivalent table version of the decision tree and bind these tables together into one overall model.

Evaluation

A conventional train/test split was performed, removing 20% of records for testing. How do we run a prediction modelled on a restructured training set where one text column has been transformed into 93 fragment columns? It is inadequate to ask simply “is this fragment in the test commentary field” simply because our fragments are delimited by space(s), "*" or “#”. The same character string occurring in the middle of a word will not count. We must therefore break the test set commentary field into similar fragments, but we do not have to spin out to 93 fields. Instead we can ask simply “is this fragment from the model found among the fragments of this commentary field?”

A second means of evaluation is available to us. I have previously pursued the “set of rules” approach to allocation. We can test these rules also on our test set to see how they perform.

RESULTS

The models

Decision trees were trained for optimum complexity parameter for all relevant allocations. Some trees returned only the root node and were discarded. The final model for BA consists of 30 decision nodes, leading to 11 possible allocations and the twelfth “unknown”. For Bidvest the model consists of 21 decision nodes leading to 14 possible allocations plus “unknown”. Some splits are self-evident: fragment “Gym” -> Health: Gym, vitamins, non-script, while others are more subtle: amount > 410.325 AND fragment “ABC-MOMENTUM” -> Health: Gym, vitamins, non-script. These models contrast with a table of 451 rules manually devised for BA and 109 for Bidvest.

Performance

BA

Outcome	chained trees	devised rules
Correct	0.275	0.257

Outcome	chained trees	devised rules
Unguessed	0.700	0.003
Wrong	0.025	0.740

Our new model correctly allocates only 27.5% of the test set. (But the devised rules only manage slightly less at 25.7%) The major difference is that our new model declines to offer a prediction on 70% of cases, and gets only 2.5% wrong. The devised rules offers a prediction on almost all cases and gets a staggering 74% wrong. Clearly we have a significant improvement here, especially since the new model requires only 30 steps as opposed to 451 rules.

Bidvest

Outcome	chained trees	devised rules
Correct	0.691	0.741
Unguessed	0.255	0.022
Wrong	0.055	0.236

Our new model does much better with Bidvest, correctly allocating 69.1% of the test set (but the devised rules do even better at 74.1%). This improvement is not surprising since bank account transactions tend to be less varied than credit card purchases. But again our new model excels in declining to offer a prediction on 25.5% of cases, and gets only 5.5% wrong. The devised rules model again offers a prediction on almost all cases and gets a significant 23.6% wrong. Again we have a significant improvement here. The new model requires only 21 steps as opposed to 109 rules.

CONCLUSION

Our modelling approach has achieved two significant advances over the previous manual rules procedure. First we have been able to virtually eliminate erroneous allocations, while improving (or maintaining) correct predictions. Second we have been able to automate the process of machine learning. The major difficulty with the rules-based approach is that the rules become easily outdated, and maintaining a set of 451 rules is a major task. The modelling update took less than 1 minute to run on a small laptop.

APPENDIX A - Allocation schema

id	name
7	Queensbridge
8	Queensbridge electricity
9	Car
10	Motorcycle
11	Income: Hillside
13	Expenses: Hillside
14	Gifts received
15	Generosity
17	Publishing
18	Clothes
19	Hair, beauty, toiletries
20	Medical script, consult
21	Maintenance: house & garden
22	Rates
23	Electricity
24	Water
25	Entertainment
27	Holidays
28	Fun
29	Furnishings & appliances
30	Family gifts
31	Bank charges
34	Insurance
37	Health: Gym, vitamins, non-script
38	SARS
48	Interest
50	Cottage rental
51	Food
53	Security
54	Tithe
55	Travel - ministry
57	Croindene
58	Queensbridge bond
59	Life policies
60	Capital investment
61	DSTV
62	Internet
64	Wedding
66	Business R
67	Cellphones
68	Medical aid

APPENDIX B - BA model

row_names	line	field	split	left	right	frag	value
1	1	f110	0.500	2	5	SHELL	
2	2	f31	0.500	3	5	BP	
3	3	f105	0.500	5	4	SANDOWN	
4	4	ac_amount	247.965	0	5		10
5	5	f117	0.500	6	0	SUPERBALIST	18
6	6	f33	0.500	7	0	BUILD	21
7	7	f121	0.500	8	0	TIMES	25
8	8	f87	0.500	9	0	NETFLIX.COM	25
9	9	f116	0.500	10	0	StrettaCafe	25
10	10	f15	0.500	11	0	852429673	25
11	11	f13	0.500	12	0	752638892	25
12	12	f54	0.500	13	0	FR	27
13	13	f12	0.500	14	0	555044372	27
14	14	f65	0.500	15	0	Kindle	28
15	15	f47	0.500	16	0	EU-UK	30
16	16	f50	0.500	17	0	FEE	31
17	17	f56	0.500	18	0	Health	37
18	18	f79	0.500	19	0	MANOLI'S	51
19	19	f71	0.500	20	0	LineageCoffeeHil	51
20	20	f51	0.500	21	0	FIN	51
21	21	f72	0.500	22	0	LineageCoffeeHillc	51
22	22	f96	0.500	23	0	PLAZA	9
23	23	ac_amount	14.250	29	24		
24	24	ac_amount	582.500	31	25		
25	25	ac_amount	265.950	0	26		9
26	26	f110	0.500	27	0	SHELL	9
27	27	ac_amount	718.360	28	31		
28	28	ac_amount	678.930	31	0		9
29	29	ac_amount	1.105	31	30		
30	30	day	12.500	0	0		9

APPENDIX C - Bidvest model

row_names	line	field	split	left	right	frag	value
1	1	f25	0.500	2	0	For	15
2	2	f40	0.500	3	0	Water	24
3	3	ac_amount	110.670	4	5		
4	4	ac_amount	-0.390	5	0		31
5	5	f11	0.500	6	0	ACB-SANTAM	34
6	6	f26	0.500	7	0	Gym	37
7	7	f12	0.500	8	0	ACB-STRATUM	37
8	8	f5	0.500	9	0	ACB-LEGENDS	37
9	9	f7	0.500	11	10	ACB-MOMENTUM	
10	10	ac_amount	410.325	0	11		37
11	11	f28	0.500	12	0	Interest	48
12	12	f8	0.500	13	0	ACB-MULTID	53
13	13	f39	0.500	14	0	Tithe	54
14	14	f34	0.500	15	0	Naedo	58
15	15	f1	0.500	16	0	212389351	59
16	16	f6	0.500	17	0	ACB-M-CHOICE	61
17	17	f4	0.500	18	0	ACB-AFRIHOST	62
18	18	f13	0.500	19	0	CHURCH/AFRIHOST	62
19	19	f35	0.500	20	0	QB	7
20	20	ac_amount	-8949.675	0	21		7
21	21	f9	0.500	22	0	ACB-NETVENDOR	8