Here is some code and ideas to get a score of around 0.59347. It uses Python and Vowpal Wabbit, but the idea should work with other languages and other algorithms.

**Data reduction**

Check out the [**data reduction thread**](http://www.kaggle.com/c/acquire-valued-shoppers-challenge/forums/t/7666/getting-started-data-reduction) to understand what this is about. For this benchmark we create the category and company union.

**Feature engineering**

Feature engineering will be important in this competition, no matter the language or algorithms used. We can check out the benchmarks on the leaderboard to get a feel for features to use.

We will generate the following features:

* has\_bought\_company: the number of times a shopper has bought from the company on offer
* has\_bought\_company\_a: the total amount the shopper has bought from the company on offer
* has\_bought\_company\_q: the quantity of items bought from the company on offer.
* has\_bought\_company\_30: the number of times a shopper has bought from the company on offer in the 30 days before the date the coupon was offered.
* has\_bought\_company\_60: the number of times a shopper has bought from the company on offer in the 60 days before the date the coupon was offered.
* ...
* has\_bought\_company\_180: 180 days before
* has\_never\_bought\_company: a negative feature for when the shopper has never bought from the company on offer before.

These same features for:

* has\_bought\_category: the number of times a shopper has bought from the category on offer
* has\_bought\_brand: the number of times a shopper has bought from the brand on offer

Combinations of these:

* has\_bought\_company\_brand\_category: if this feature is present the shopper has bought from the company, brand, and category on offer.
* has\_never\_bought\_company\_brand: negative feature for the combination of brand and company purchase history.

Offer-related:

* offer\_value: The value of the coupon offer
* offer\_quantity: The number of products to redeem with the coupon

Total shopper spend:

* total\_shopper\_spend: We take the total amount spend by the shopper in the reduced dataset.

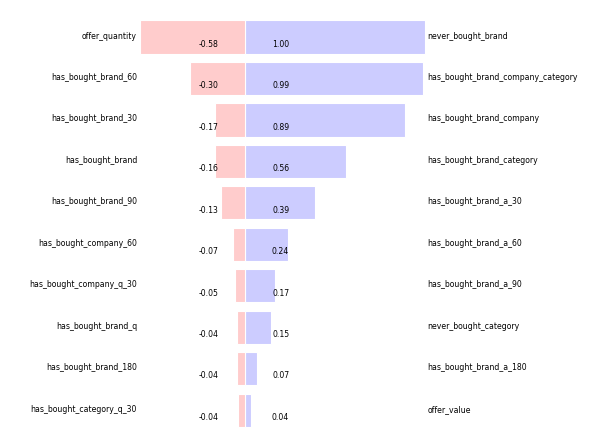
Can you name some other possibly interesting features to generate?

**Vowpal Wabbit**

We train Vowpal Wabbit using quantile regression, 40 passes and a learning rate of 0.85. We turn the predictions into Kaggle's submission format.

**Feature visualisation**

Using the output from Vowpal Wabbit's wrapper vw-varinfo we generate the feature relevance plot below (code to generate this included):



**Improvements**

If you use this code or feature ideas to improve the score of 0.59347 I'd really appreciate it if you let me know these improvements. It's very possible that another algorithm may do better with these exact features.

If you do improve the score I'd love to team up. I am still working on a few new features and we could combine our approaches for an ever better score.

**Code & Tutorial**

For latest code see the Github repo: [**https://github.com/MLWave/kaggle\_acquire-valued-shoppers-challenge**](https://github.com/MLWave/kaggle_acquire-valued-shoppers-challenge), for the tutorial/description read: [**http://mlwave.com/predicting-repeat-buyers-vowpal-wabbit/**](http://mlwave.com/predicting-repeat-buyers-vowpal-wabbit/)

All of this runs in about 15 minutes for me, from raw data to submission. Takes less than 1GB of memory.

**Todo**

Generate more features. Set up local validation. Tweak algorithms. Try other algorithms than quantile regression (sklearn has plenty).