Our 2015 Data Mining Cup Solution

Iowa State University Team 1

or

How I Learned to Stop Worrying and Love the Grind

Our github page:
github.com/imouzon/dmc2015

Iowa

Iowa State University

and

Our Team

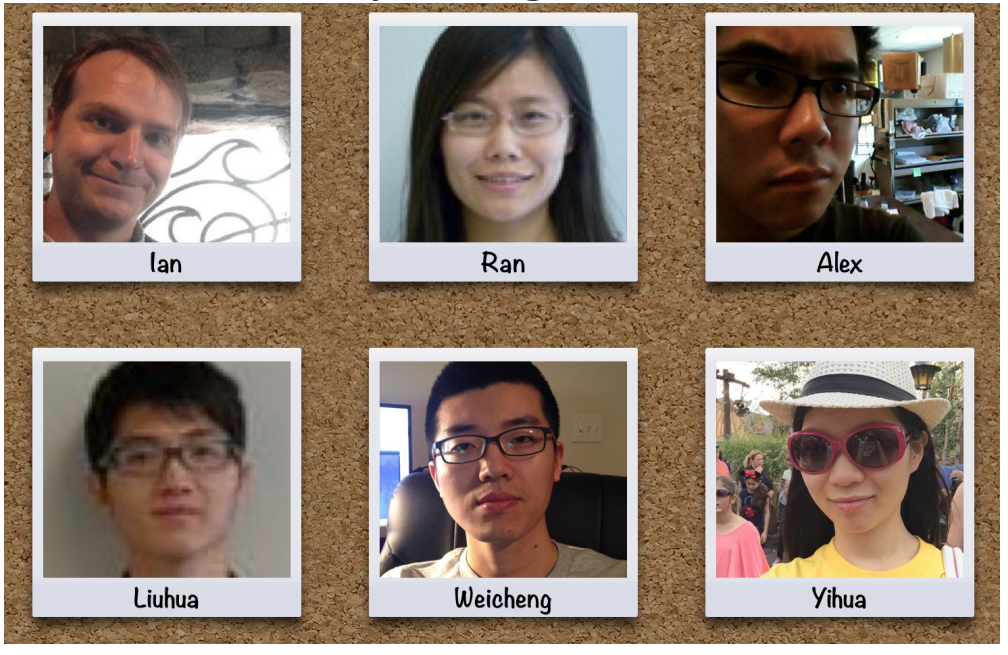
Iowa

About Iowa

Iowa is in America's Midwest

- The largest city, Des Moines, only has a population of about 600,000
- The largest producer of corn, ethanol, and soybeans in the United States
- There are more pigs than people in Iowa
- It is named after the Iowa river which itself is named after the Native American "Ioway" tribe.
- It is the birth place of the actor John Wayne (The Searchers, Red River, ...)

We were able to put together a terrific team



- A result can only be as good as the team that makes it
- We had a lot of diverse talent
- We met twice weekly to share results
- Primary tools: GitHub, R

Our Approach

The First 10 Days

The Data Didn't Just Appear Randomly

- There must be some **structure** underlying it
- There must be some **unit** generating it in the context of the structure

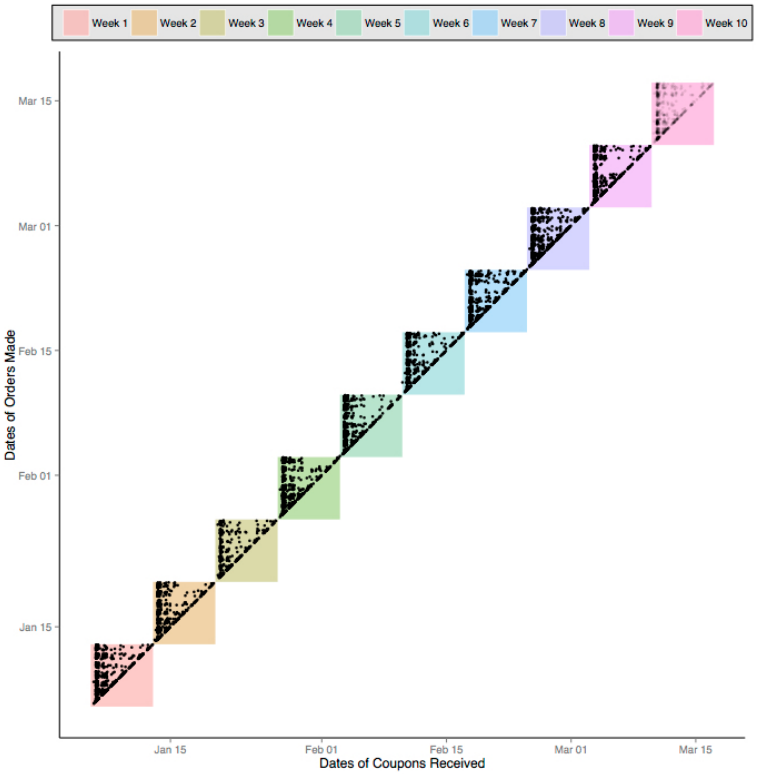
Here's an analogy:

- The tracks the structure
- The train is the unit, whose future position is uncertain but constrained by the tracks

First 10 Days

Find Structure

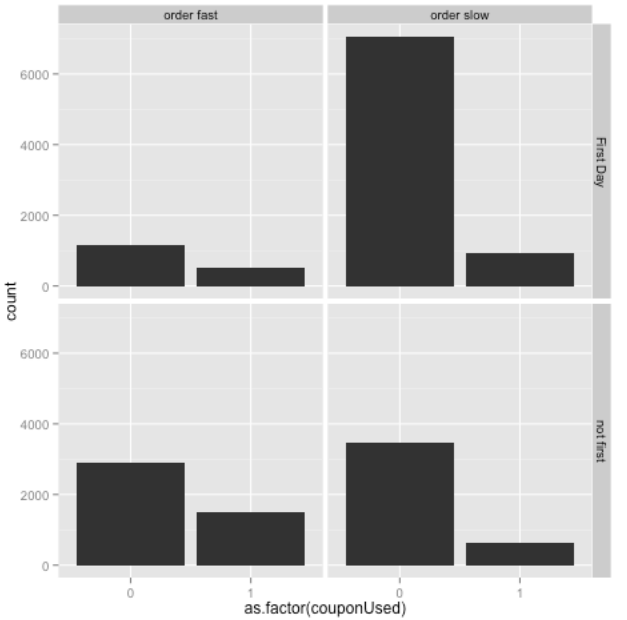
Coupons Expire



- Batches of coupons are sent every Monday at 11:59 p.m.
- They keep the same coupons for one week
- After that week has passed, they can not use the coupons anymore and instead have three new

The Structure Matters

- We can examine how the customers (the units) behave in this structure of coupon delivery and expiration



- This structure is important - customers who receive coupons early and shop quickly are more likely to use them.

We were off to a good start

We had an good grasp of the data structure

We had an intuitive feel for the loss function

We had a lot of time left to find the important features

We Just Needed Was a Simple Set of Features

Nothing Else We Did Worked

We had nothing working for us at all

- Attempts to uncover the base value of the coupons: **failed**
- Attempts to fit a statistical model to user behavior: **failed**
- Attempts to exploit data "oddities": **failed**

There was just no clever way to unlock the data

We just had to work a little harder

No small set of clever features exists?

Make a BIG Set of Features and Reduce!

A Feature Matrix Can Get Too Big

And The Features Still Need to Be Useful

We needed good general methods

- Give up trying to find clever features use best tools to chip away
- Use structure and meaning to find aspects of the data structure
- **LLR**: likelihood a coupon is used examined through different filters
- **tf-idf**: measure similarity of a trio of coupons

Lots of Features Means Lots of Rules

- Historical data sets used to provide background data
- Train on one set, validate all methods on common set

The new challenge is balancing this enormous set in a way that we still get reliable predictions

Reduce features by selection

Once we had applied these general methods we had another problem: almost 1000 columns in feature matrix.

- Hundreds of features based on LLRs and tf-idf
- Most features are only weakly related to response
- Needed an easy way to select strongest features.

Reducing the Features

Some ML Methods Can Identify Important Features

Many machine techniques have importance criteria to help identify usefule features. Iteratively run on subsets of data, these can help us reduce the size of our feature matrix:

- Random forest found 305 important features
- C5.0 found 83
- adaboost found 105
- Lasso (**l**east **a**bsolute **s**hrinkage and **s**election **o**perator) chose 78

These featues made up our final feature matrix

Safety Checks

These methods

Some ML Methods Can Identify Important Features

Many machine techniques have importance criteria to help identify usefule features. Iteratively run on subsets of data, these can help us reduce the size of our feature matrix:

- Random forest found 305 important features
- C5.0 found 83
- adaboost found 105
- Lasso (**l**east **a**bsolute **s**hrinkage and **s**election **o**perator) chose 78

These featues made up our final feature matrix

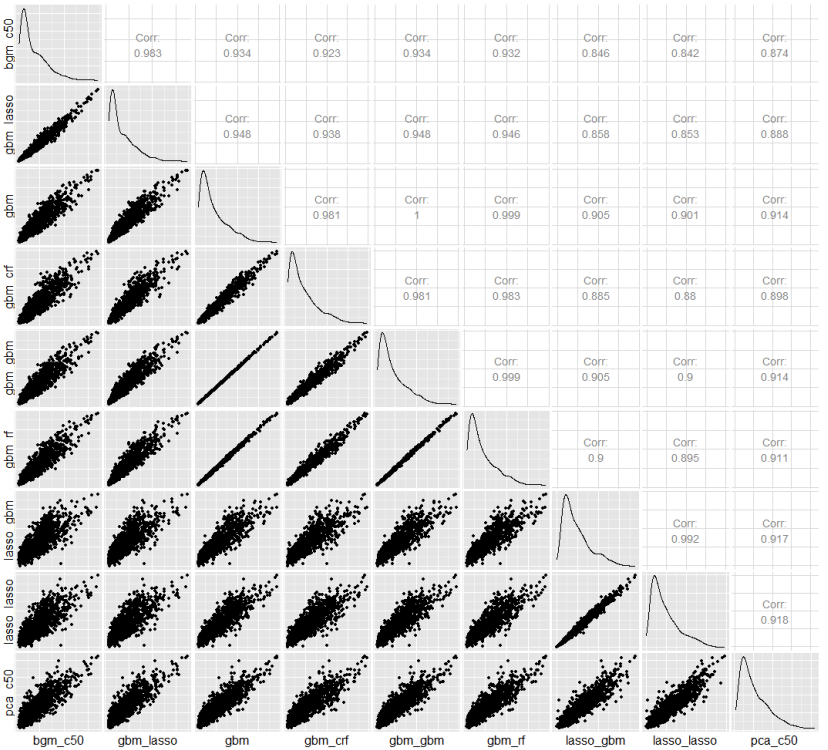
Final Predictions

Creating a Final Prediction

Multiple Methods

Aviod Correlation

Improve By Combining

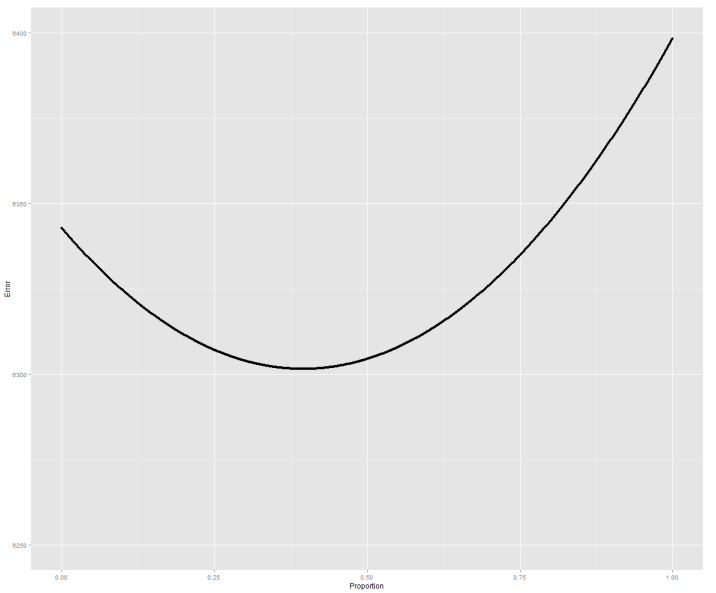


Putting Predictor Together

Combining predictors leads to the best final prediction

The best solution involved a 45/55 part combination of

1. Predictions fit by lasso based on variables selected by Gradiant Boosting Machines
2. Predictions fit by Gradient Boosting Methods based on variables selected by C5.0



We are Thankful for This Opportunity to Represent

Our Team, Our Department, and Our University

Thank You For Your Attention