What is this

LendingClub is a platform for peer-to-peer lending. If you're not familiar with it, here are some key points:

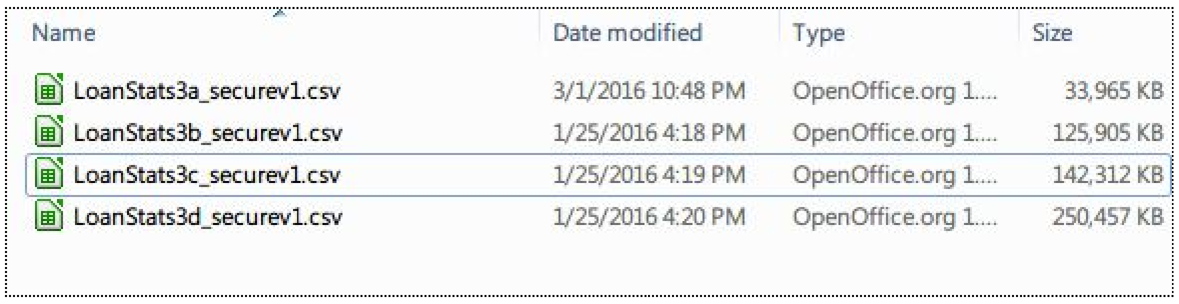
- **P2P lending** -- It's crowdfunding peoples' loans and they are expected to pay back with interest. .
- **Note** -- A piece of a loan. If somebody wanted to borrow $1000, a note could be a $25 chunk of that.
- **Secondary market** -- After a loan has been issued (the borrower got his/her crowdfunding goal), investors can sell notes that they own to other investors. They can sell at a discount for some quick cash or if they don't have faith in a note. Or they can sell at a premium.
- **Charged off** -- When a borrower stops paying his/her loan and there is no chance of recovering any money. This is bad for the investors.

I looked through existing loan data which LendingClub provides, created my own model to predict good or bad loans, and set up scripts to find and buy those the notes. My adjusted annual return is negative yet I made more in interest than I did in loans that got charged off, so I can't say with certainty that my model was a success. But the project was an excellent learning experience where I tried some new tools and learned a couple of lessons.

Step 1. Get the dataset

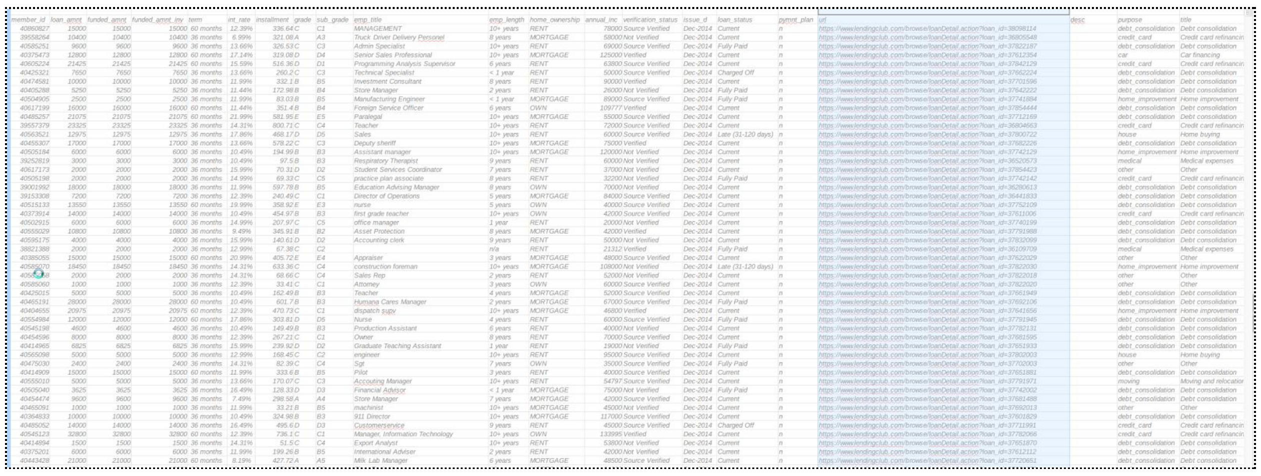
LendingClub provides a history of all the loans they've issued and what the status of those loans are, updated every quarter. It's a big dataset, and it's what sparked my *interest* in doing all of this. They provide a set of .csv files with info about each loan such as loan amount, interest rate, purpose, date issued, current status, et al.

The .csv files are fairly large with 100k+ rows per file:



Name	Date modified	Type	Size
LoanStats3a_securev1.csv	3/1/2016 10:48 PM	OpenOffice.org 1....	33,965 KB
LoanStats3b_securev1.csv	1/25/2016 4:18 PM	OpenOffice.org 1....	125,905 KB
LoanStats3c_securev1.csv	1/25/2016 4:19 PM	OpenOffice.org 1....	142,312 KB
LoanStats3d_securev1.csv	1/25/2016 4:20 PM	OpenOffice.org 1....	250,457 KB

More than enough data for LibreOffice to handle on my computer (it's actually not responding in the screenshot below)



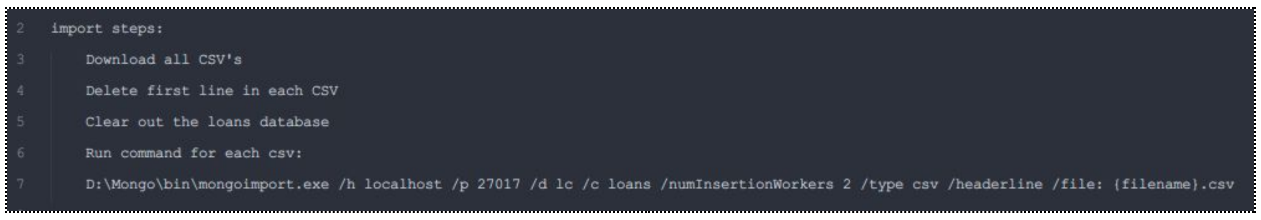
member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_title	emp_length	home_ownership	annual_inc	verification_status
40860827	15000	15000	15000	60 months	12.39%	336.64	C	C1	MANAGEMENT	10+ years	RENT	78000	Source Verified
39558264	10400	10400	10400	36 months	6.99%	321.08	A	A3	Truck Driver Delivery Personel	8 years	MORTGAGE	58000	Not Verified
40585251	9600	9600	9600	36 months	13.66%	326.53	C	C3	Admin Specialist	10+ years	RENT	69000	Source Verified
40375473	12800	12800	12800	60 months	17.14%	319.08	D	D4	Senior Sales Professional	10+ years	MORTGAGE	125000	Verified
40605224	21425	21425	21425	60 months	15.59%	516.36	D	D1	Programming Analysis Supervisor	6 years	RENT	63800	Source Verified
40425321	7650	7650	7650	36 months	13.66%	260.2	C	C3	Technical Specialist	< 1 year	RENT	50000	Source Verified
40474581	10000	10000	10000	36 months	11.99%	332.1	B	B5	Investment Consultant	8 years	RENT	90000	Verified
40405288	5250	5250	5250	36 months	11.44%	172.98	B	B4	Store Manager	2 years	RENT	26000	Not Verified
40504905	2500	2500	2500	36 months	11.99%	83.03	B	B5	Manufacturing Engineer	< 1 year	MORTGAGE	89000	Source Verified
40617199	16000	16000	16000	60 months	11.44%	351.4	B	B4	Foreign Service Officer	6 years	OWN	109777	Verified
40485257	21075	21075	21075	60 months	21.99%	581.95	E	E5	Paralegal	10+ years	MORTGAGE	55000	Source Verified
39557379	23325	23325	23325	36 months	14.31%	800.71	C	C4	Teacher	10+ years	RENT	72000	Source Verified
40563521	12975	12975	12975	36 months	17.86%	468.17	D	D5	Sales	10+ years	RENT	60000	Source Verified
40455307	17000	17000	17000	36 months	13.66%	578.22	C	C3	Deputy sheriff	10+ years	MORTGAGE	75000	Verified
40505184	6000	6000	6000	36 months	10.49%	194.99	B	B3	Assistant manager	10+ years	MORTGAGE	120000	Not Verified
39252819	3000	3000	3000	36 months	10.49%	97.5	B	B3	Respiratory Therapist	9 years	RENT	60000	Not Verified
40617173	2000	2000	2000	36 months	15.99%	70.31	D	D2	Student Services Coordinator	7 years	RENT	37000	Not Verified
40505198	2000	2000	2000	36 months	14.99%	69.33	C	C5	practice plan associate	8 years	RENT	32200	Not Verified
39001992	18000	18000	18000	36 months	11.99%	597.78	B	B5	Education Advising Manager	8 years	OWN	70000	Not Verified
39153308	7200	7200	7200	36 months	12.39%	240.49	C	C1	Director of Operations	5 years	MORTGAGE	84000	Source Verified
40515133	13550	13550	13550	60 months	19.99%	358.92	E	E3	nurse	5 years	OWN	40000	Source Verified
40373914	14000	14000	14000	36 months	10.49%	454.97	B	B3	first grade teacher	10+ years	OWN	42000	Source Verified
40502915	6000	6000	6000	36 months	14.99%	207.97	C	C5	office manager	1 year	RENT	20000	Not Verified
40555029	10800	10800	10800	36 months	9.49%	345.91	B	B2	Asset Protection	8 years	MORTGAGE	42000	Verified
40595175	4000	4000	4000	36 months	15.99%	140.61	D	D2	Accounting clerk	9 years	RENT	50000	Not Verified
38821388	2000	2000	2000	36 months	12.99%	67.38	C	C2		n/a	RENT	21312	Verified
40385055	15000	15000	15000	60 months	20.99%	405.72	E	E4	Appraiser	3 years	MORTGAGE	48000	Source Verified
40585070	18450	18450	18450	36 months	14.31%	633.36	C	C4	construction foreman	10+ years	MORTGAGE	108000	Not Verified
[illegible]	2000	2000	2000	36 months	14.31%	68.66	C	C4	Sales Rep	2 years	RENT	52000	Not Verified
40585060	1000	1000	1000	36 months	12.39%	33.41	C	C1	Attorney	3 years	OWN	60000	Source Verified
40425015	5000	5000	5000	36 months	10.49%	162.49	B	B3	Teacher	4 years	MORTGAGE	52000	Source Verified
40465191	28000	28000	28000	60 months	10.49%	601.7	B	B3	Humana Cares Manager	2 years	MORTGAGE	67000	Source Verified
40404655	20975	20975	20975	60 months	12.39%	470.73	C	C1	dispatch supv	10+ years	MORTGAGE	46800	Verified
40554984	12000	12000	12000	60 months	17.86%	303.81	D	D5	Nurse	4 years	RENT	60000	Source Verified
40545198	4600	4600	4600	36 months	10.49%	149.49	B	B3	Production Assistant	6 years	RENT	40000	Not Verified
40454596	8000	8000	8000	36 months	12.39%	267.21	C	C1	Owner	8 years	RENT	70000	Source Verified
40414965	6825	6825	6825	36 months	15.99%	239.92	D	D2	Graduate Teaching Assistant	1 year	RENT	19000	Not Verified
40565098	5000	5000	5000	36 months	12.99%	168.45	C	C2	engineer	10+ years	RENT	95000	Source Verified
40475030	2400	2400	2400	36 months	14.31%	82.39	C	C4	Sgt	7 years	OWN	35000	Source Verified
40414909	15000	15000	15000	60 months	11.99%	333.6	B	B5	Pilot	3 years	RENT	40000	Source Verified
40555010	5000	5000	5000	36 months	13.66%	170.07	C	C3	Accouting Manager	10+ years	RENT	54797	Source Verified
40505040	3625	3625	3625	36 months	16.49%	128.33	D	D3	Financial Advisor	< 1 year	MORTGAGE	75000	Not Verified
40454474	9600	9600	9600	36 months	7.49%	298.58	A	A4	Store Manager	7 years	MORTGAGE	42000	Source Verified
40465091	1000	1000	1000	36 months	11.99%	33.21	B	B5	machinist	10+ years	MORTGAGE	45000	Not Verified
40364833	10000	10000	10000	36 months	10.49%	324.98	B	B3	911 Director	10+ years	MORTGAGE	117000	Source Verified
40485052	14000	14000	14000	36 months	16.49%	495.6	D	D3	Customerservice	9 years	RENT	45000	Source Verified
40545123	32800	32800	32800	60 months	12.39%	736.1	C	C1	Manager, Information Technology	10+ years	OWN	133995	Verified
40414994	1500	1500	1500	36 months	14.31%	51.5	C	C4	Export Analyst	10+ years	RENT	53800	Not Verified
40375201	6000	6000	6000	36 months	11.99%	199.26	B	B5	International Adviser	2 years	RENT	42000	Not Verified
40443428	21000	21000	21000	60 months	8.19%	427.72	A	A5	Milk Lab Manager	6 years	MORTGAGE	48500	Source Verified

member_id	issue_d	loan_status	pymnt_plan	url	desc	purpose	title
40860827	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=38098114		debt_consolidation	Debt consolidation
39558264	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=36805548		credit_card	Credit card refinancin
40585251	Dec-2014	Fully Paid	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37822187		debt_consolidation	Debt consolidation
40375473	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37612354		car	Car financing
40605224	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37842129		credit_card	Credit card refinancin
40425321	Dec-2014	Charged Off	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37662224		debt_consolidation	Debt consolidation
40474581	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37701596		debt_consolidation	Debt consolidation
40405288	Dec-2014	Fully Paid	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37642222		debt_consolidation	Debt consolidation
40504905	Dec-2014	Fully Paid	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37741884		home_improvement	Home improvement
40617199	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37854444		debt_consolidation	Debt consolidation
40485257	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37712169		debt_consolidation	Debt consolidation
39557379	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=36804663		credit_card	Credit card refinancin
40563521	Dec-2014	Late (31-120 days)	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37800722		house	Home buying
40455307	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37682226		debt_consolidation	Debt consolidation
40505184	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37742129		home_improvement	Home improvement
39252819	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=36520573		medical	Medical expenses
40617173	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37854423		other	Other
40505198	Dec-2014	Fully Paid	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37742142		credit_card	Credit card refinancin
39001992	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=36280613		debt_consolidation	Debt consolidation
39153308	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=36441833		debt_consolidation	Debt consolidation
40515133	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37752109		debt_consolidation	Debt consolidation
40373914	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37611006		credit_card	Credit card refinancin
40502915	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37740199		debt_consolidation	Debt consolidation
40555029	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37791988		debt_consolidation	Debt consolidation
40595175	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37832099		debt_consolidation	Debt consolidation
38821388	Dec-2014	Fully Paid	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=36109709		medical	Medical expenses
40385055	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37622029		other	Other
40585070	Dec-2014	Late (31-120 days)	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37822030		home_improvement	Home improvement
[illegible]	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37822018		other	Other
40585060	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37822020		other	Other
40425015	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37661949		debt_consolidation	Debt consolidation
40465191	Dec-2014	Fully Paid	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37692106		debt_consolidation	Debt consolidation
40404655	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37641656		home_improvement	Home improvement
40554984	Dec-2014	Fully Paid	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37791945		debt_consolidation	Debt consolidation
40545198	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37782131		debt_consolidation	Debt consolidation
40454596	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37681595		debt_consolidation	Debt consolidation
40414965	Dec-2014	Fully Paid	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37651933		debt_consolidation	Debt consolidation
40565098	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37802003		house	Home buying
40475030	Dec-2014	Fully Paid	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37702003		other	Other
40414909	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37651881		debt_consolidation	Debt consolidation
40555010	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37791971		moving	Moving and relocatio
40505040	Dec-2014	Fully Paid	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37742002		debt_consolidation	Debt consolidation
40454474	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37681488		debt_consolidation	Debt consolidation
40465091	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37692013		other	Other
40364833	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37601829		debt_consolidation	Debt consolidation
40485052	Dec-2014	Charged Off	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37711991		credit_card	Credit card refinancin
40545123	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37782066		credit_card	Credit card refinancin
40414994	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37651870		debt_consolidation	Debt consolidation
40375201	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37612112		debt_consolidation	Debt consolidation
40443428	Dec-2014	Current	n	https://www.lendingclub.com/browse/loanDetail.action?loan_id=37720651		debt_consolidation	Debt consolidation

I had a little under a GB of data. A few hundred thousand rows and 50+ columns. Seemed like a pretty good fit for a relational database like MySQL with some indexing which I was used to from back in my PHP days (which weren't over, of course)

Around the time I came across the dataset, MongoDB was quite popular and I wanted a ride on the hype train. This was an excellent excuse to try it out! [Lesson to be learned here.]

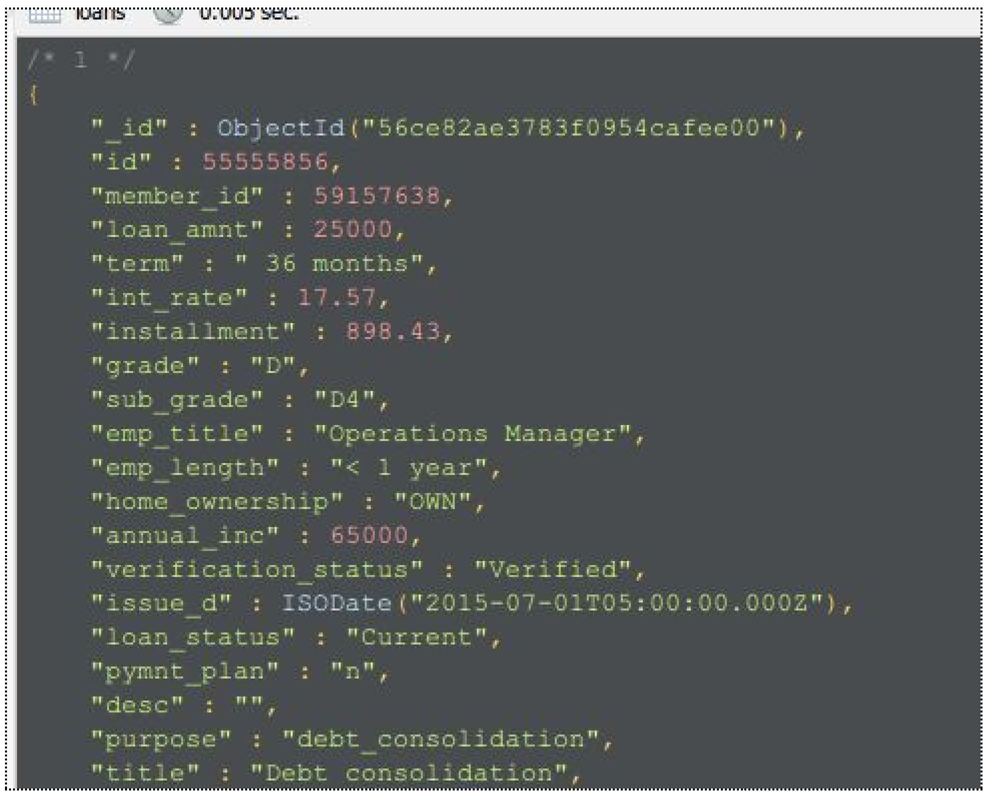
After a few hours (maybe it was days) of getting MongoDB set up on my computer and trying some queries, I was ready to import all the data from the csv files. Thankfully there was a mongoimport tool. Because LendingClub updates the csv files once a quarter, I expected the need to have to re-import the CSVs each time they did. So I was sure to write down the process so I wouldn't have to figure it out again each time.



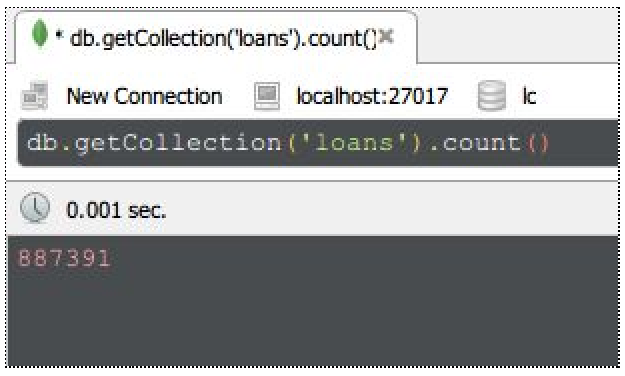
```
import steps:
    Download all CSV's
    Delete first line in each CSV
    Clear out the loans database
    Run command for each csv:
    D:\Mongo\bin\mongoimport.exe /h localhost /p 27017 /d lc /c loans /numInsertionWorkers 2 /type csv /headerline /file: {filename}.csv
```

Clean it up

Import worked and I have ~900k loans in the database.



```
loans    0.005 sec.
/* 1 */
{
    "_id" : ObjectId("56ce82ae3783f0954cafee00"),
    "id" : 55555856,
    "member_id" : 59157638,
    "loan_amnt" : 25000,
    "term" : " 36 months",
    "int_rate" : 17.57,
    "installment" : 898.43,
    "grade" : "D",
    "sub_grade" : "D4",
    "emp_title" : "Operations Manager",
    "emp_length" : "< 1 year",
    "home_ownership" : "OWN",
    "annual_inc" : 65000,
    "verification_status" : "Verified",
    "issue_d" : ISODate("2015-07-01T05:00:00.000Z"),
    "loan_status" : "Current",
    "pymnt_plan" : "n",
    "desc" : "",
    "purpose" : "debt_consolidation",
    "title" : "Debt consolidation",
```



```
db.getCollection('loans').count()
New Connection   localhost:27017   lc
db.getCollection('loans').count()
0.001 sec.
887391
```

Take a look at the issue_d field. It wasn't an ISODate when I imported it! I realized there were a number of date fields and strings like "6%" that needed to be cleaned up that I knew could save space and make it easier to sort, search, and analyze.   The term for this which I later came to learn: data munging. Here's an example of a row of data from the CSV file.

```
"1069410","1303652","21000","21000","20975"," 60 months"," 19.91%","555.33","E","E4","Costco","7
years","RENT","50000","Verified","Dec-2011","Charged
Off","n","https://www.lendingclub.com/browse/loanDetail.action?loan_id=1069410","  Borrower added on
12/20/11 > consolidation  of credit cards.<br>","debt_consolidation"," Bill pay
of","980xx","WA","21.58","0","Sep-1998","680","684","1","","","7","0","19448","97.6%","14","f","0.00","0.00
","18319.14","18297.35","8990.81","9328.33","0.0","0.0","0.0","Oct-2014","555.33","","Jan-2016","634","630"
,"0","","1","INDIVIDUAL","","","","0","","","","","","","","","","","","","","","","",""

Red = Don't need this         Blue = Convert to date        Purple = Convert to float
```

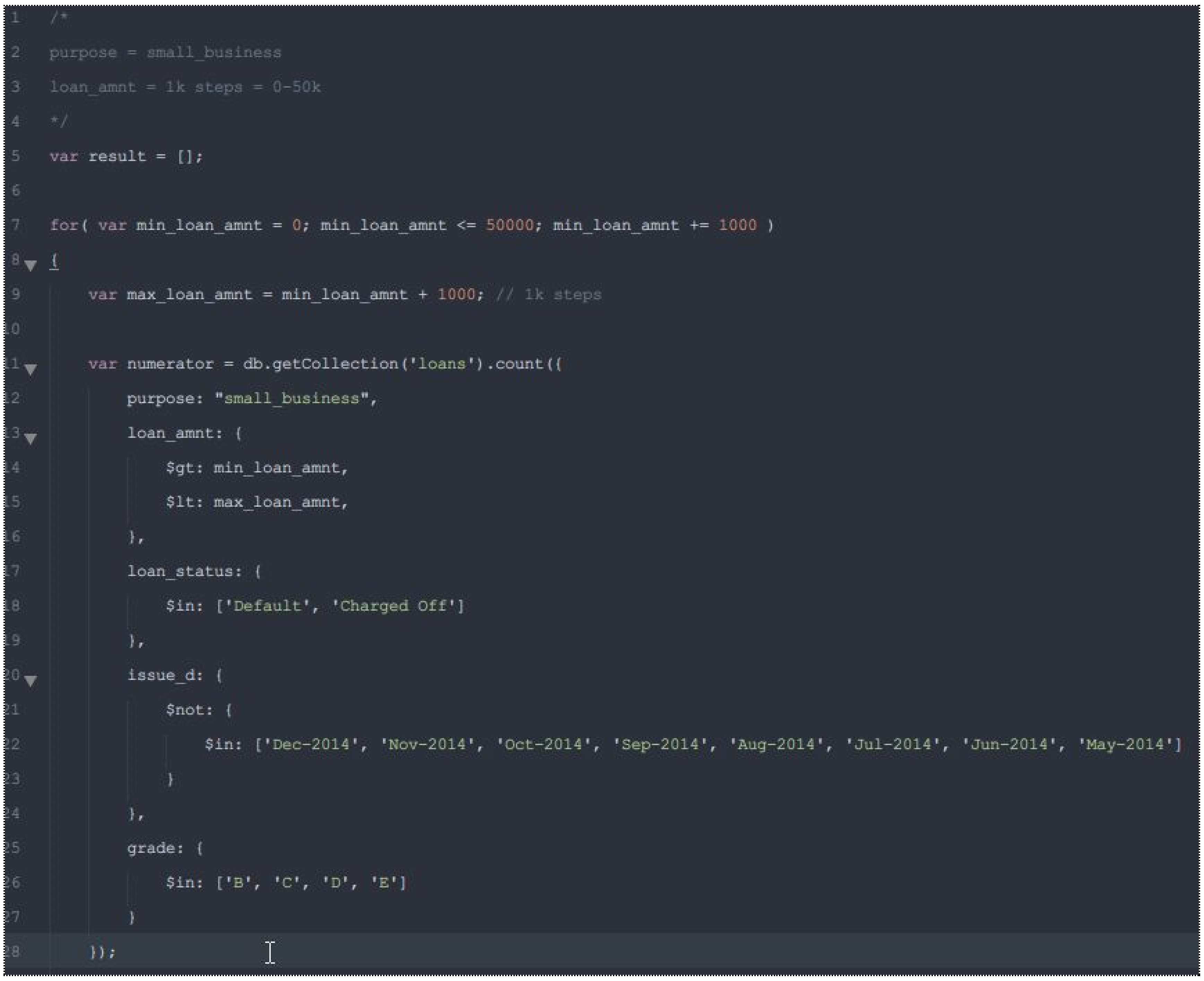
With very little Googling, I found out that I could do this directly in Mongo with Javascript! Very convenient.

```
/*
Fix dates
*/
db.loans.find({
      issue_d: {
            $type: 2
      }
}).forEach( function( elem ) {
      var newDate = elem.issue_d.replace( '-', ' 1 ' );
      elem.issue_d = new Date( newDate );
      db.loans.save( elem );
});
```

I ended up writing a bunch of these and put them into “post_processing.js”. Warning: It takes a while to run!

Step 2. “Analysis”

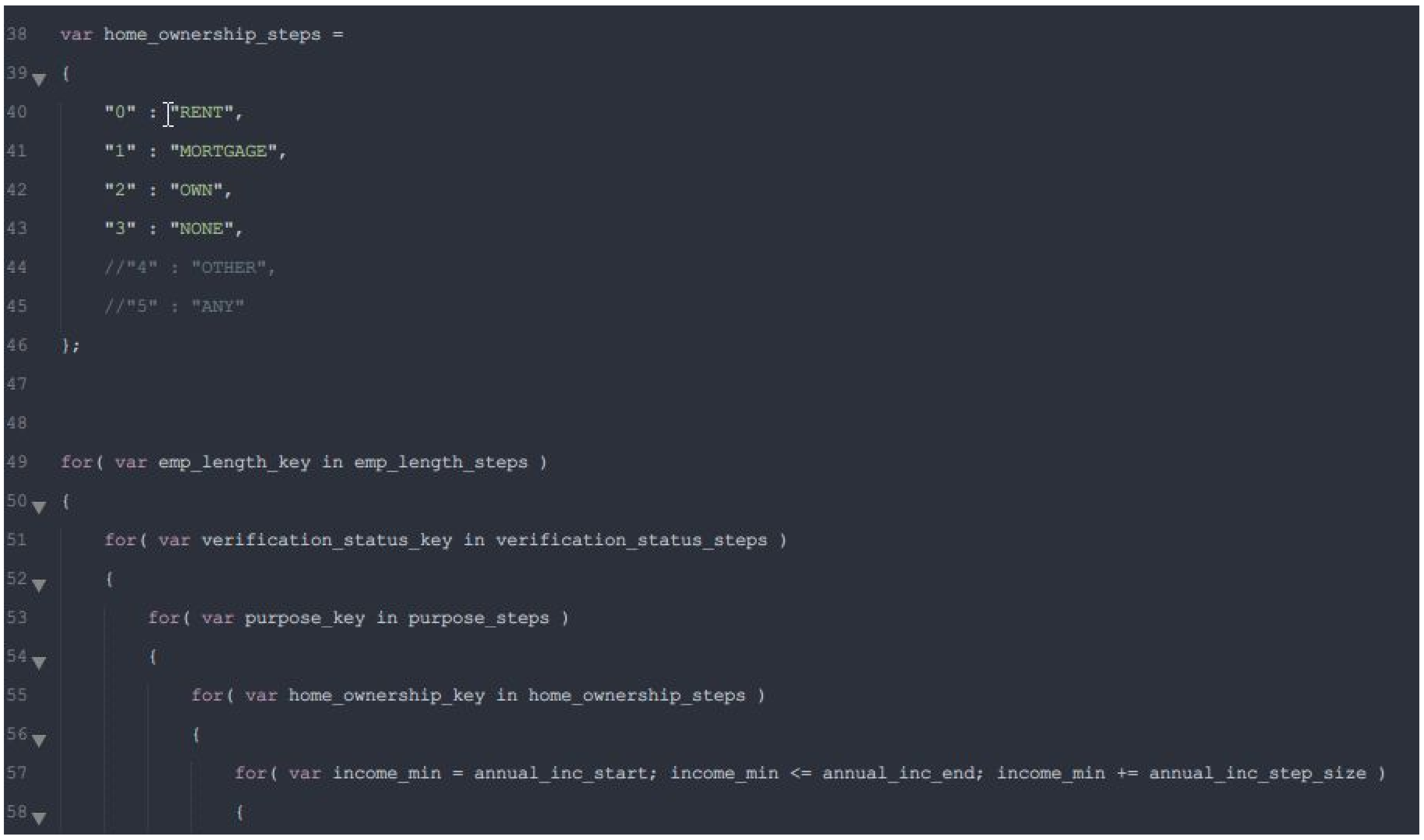
Lots of data columns (later learned the term “Features”) to investigate. I started out with one at a time. Then two, and more...



```
/*
purpose = small_business
loan_amnt = 1k steps = 0-50k
*/
var result = [];

for( var min_loan_amnt = 0; min_loan_amnt <= 50000; min_loan_amnt += 1000 )
{
    var max_loan_amnt = min_loan_amnt + 1000; // 1k steps

    var numerator = db.getCollection('loans').count({
        purpose: "small_business",
        loan_amnt: {
            $gt: min_loan_amnt,
            $lt: max_loan_amnt,
        },
        loan_status: {
            $in: ['Default', 'Charged Off']
        },
        issue_d: {
            $not: {
                $in: ['Dec-2014', 'Nov-2014', 'Oct-2014', 'Sep-2014', 'Aug-2014', 'Jul-2014', 'Jun-2014', 'May-2014']
            }
        },
        grade: {
            $in: ['B', 'C', 'D', 'E']
        }
    });
```

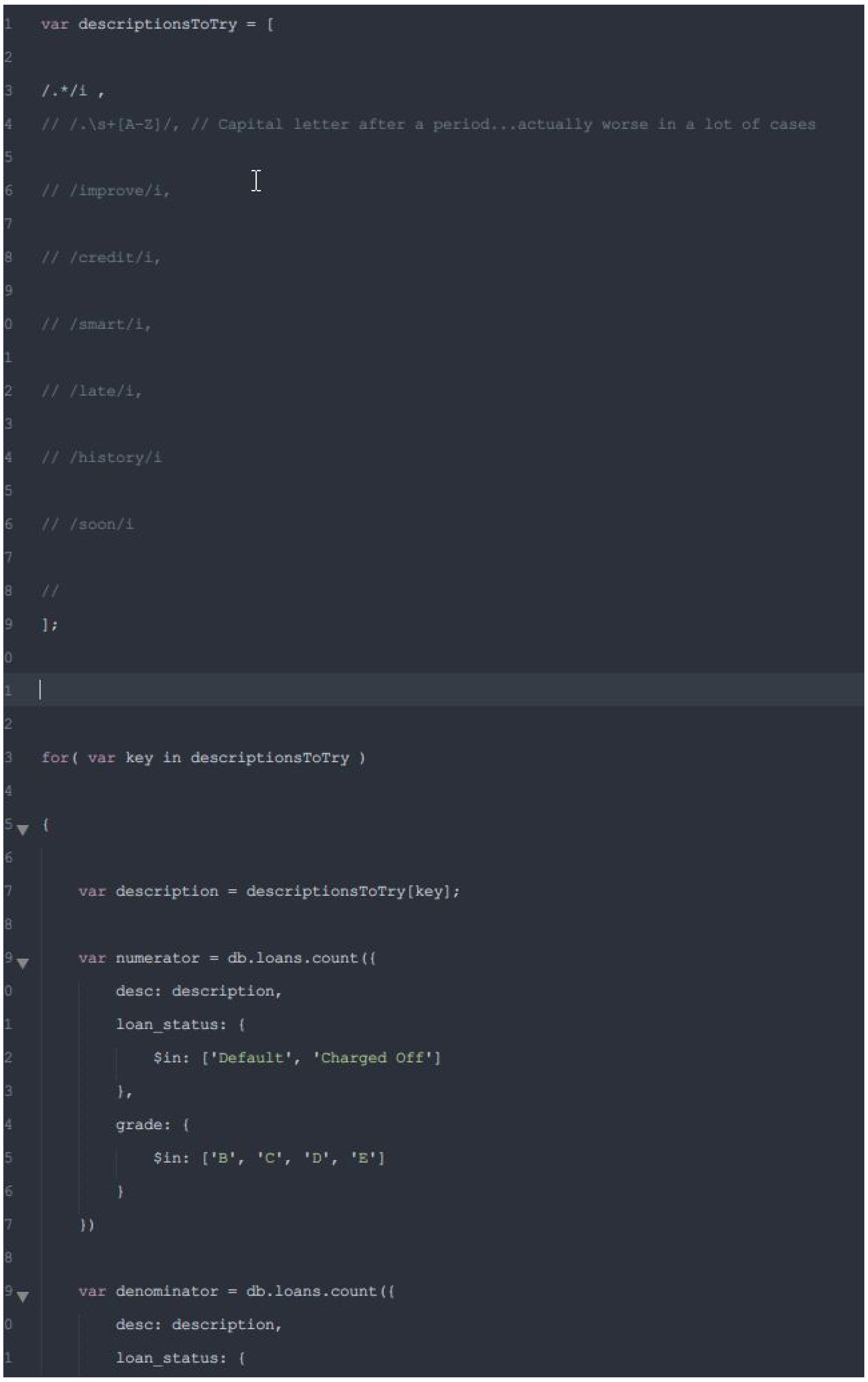
I counted the number of bad loans (charged off, default) while stepping through feature values (loan amount is depicted above). Then, I counted the total number of loans (good or bad) that fit the criteria, and used that as my denominator. The idea was to see whether there was a higher “bad” loan ratio for the given set of feature values.

Over time, I got better at it (obviously joking):



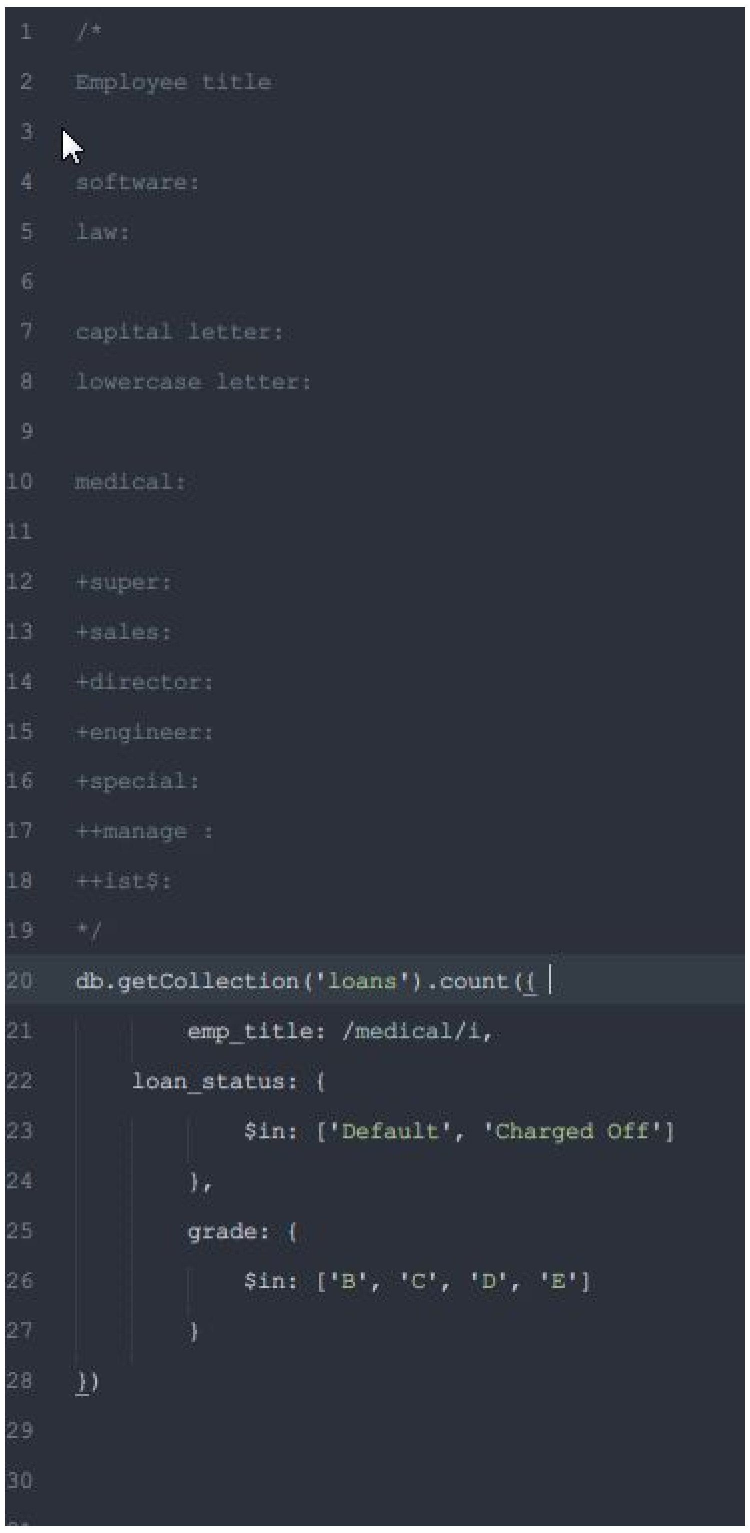
```
var home_ownership_steps =
{
    "0" : "RENT",
    "1" : "MORTGAGE",
    "2" : "OWN",
    "3" : "NONE",
    //"4" : "OTHER",
    //"5" : "ANY"
};


for( var emp_length_key in emp_length_steps )
{
    for( var verification_status_key in verification_status_steps )
    {
        for( var purpose_key in purpose_steps )
        {
            for( var home_ownership_key in home_ownership_steps )
            {
                for( var income_min = annual_inc_start; income_min <= annual_inc_end; income_min += annual_inc_step_size )
                {
```

Looking back, I’m impressed because I even tried language processing (if you’d be kind enough to allow me to call it that). In the loan description, did borrowers use lowercase/capital “i” when referring to themselves? Were there any keywords that could be red flags?



```
var descriptionsToTry = [

/.*/i ,
// /.\s+[A-Z]/, // Capital letter after a period...actually worse in a lot of cases

// /improve/i,

// /credit/i,

// /smart/i,

// /late/i,

// /history/i

// /soon/i

//
];



for( var key in descriptionsToTry )

{

    var description = descriptionsToTry[key];

    var numerator = db.loans.count({
        desc: description,
        loan_status: {
            $in: ['Default', 'Charged Off']
        },
        grade: {
            $in: ['B', 'C', 'D', 'E']
        }
    })

    var denominator = db.loans.count({
        desc: description,
        loan_status: {
```

How about what they put for their job title?



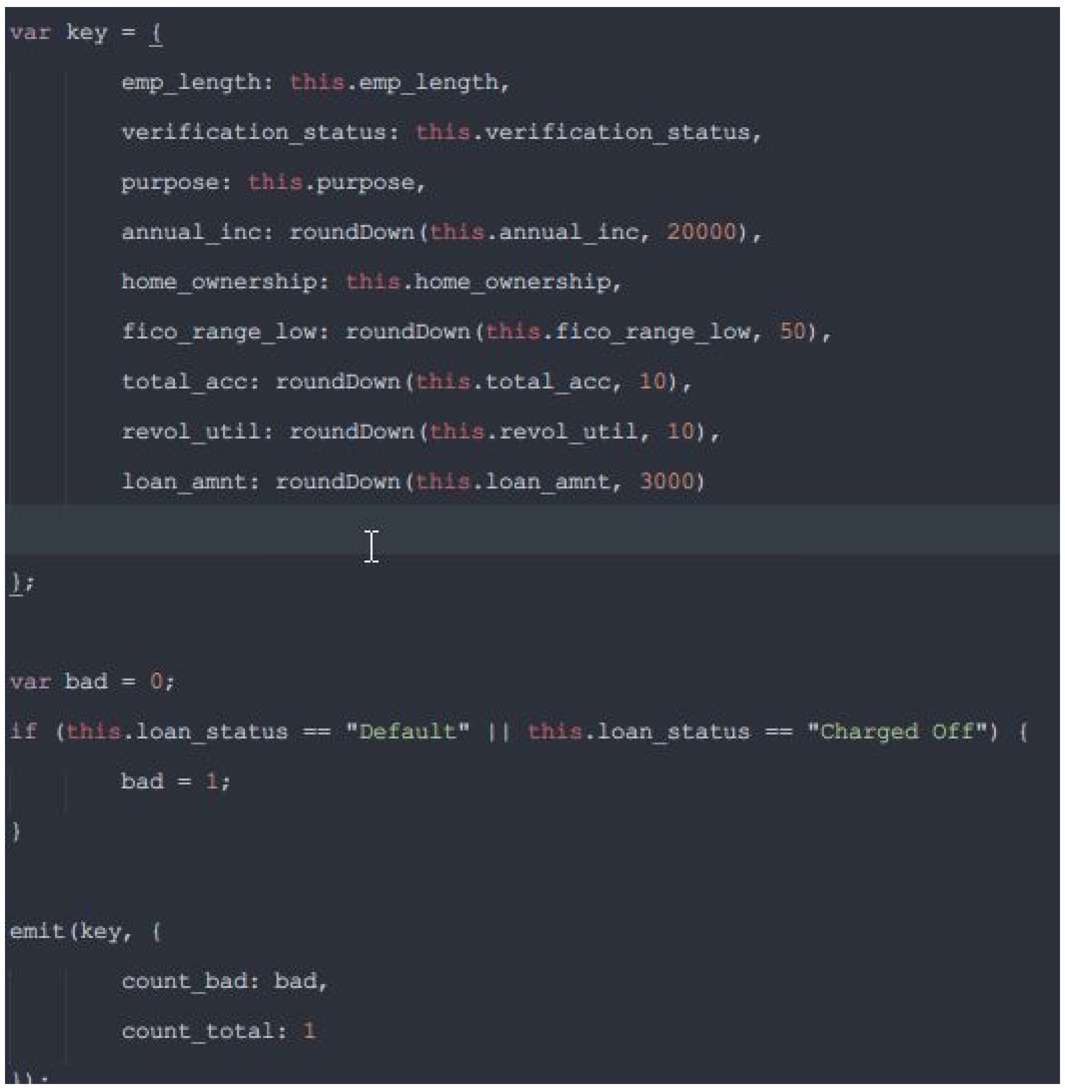
```
/*
Employee title

software:
law:

capital letter:
lowercase letter:

medical:

+super:
+sales:
+director:
+engineer:
+special:
++manage :
++ist$:
*/
db.getCollection('loans').count({
        emp_title: /medical/i,
    loan_status: {
            $in: ['Default', 'Charged Off']
        },
        grade: {
            $in: ['B', 'C', 'D', 'E']
        }
})
```

I don’t think I reached any solid conclusions from this, but it was cool to try out. I bet with more sophisticated language processing techniques, some good patterns can be found.

MapReduce

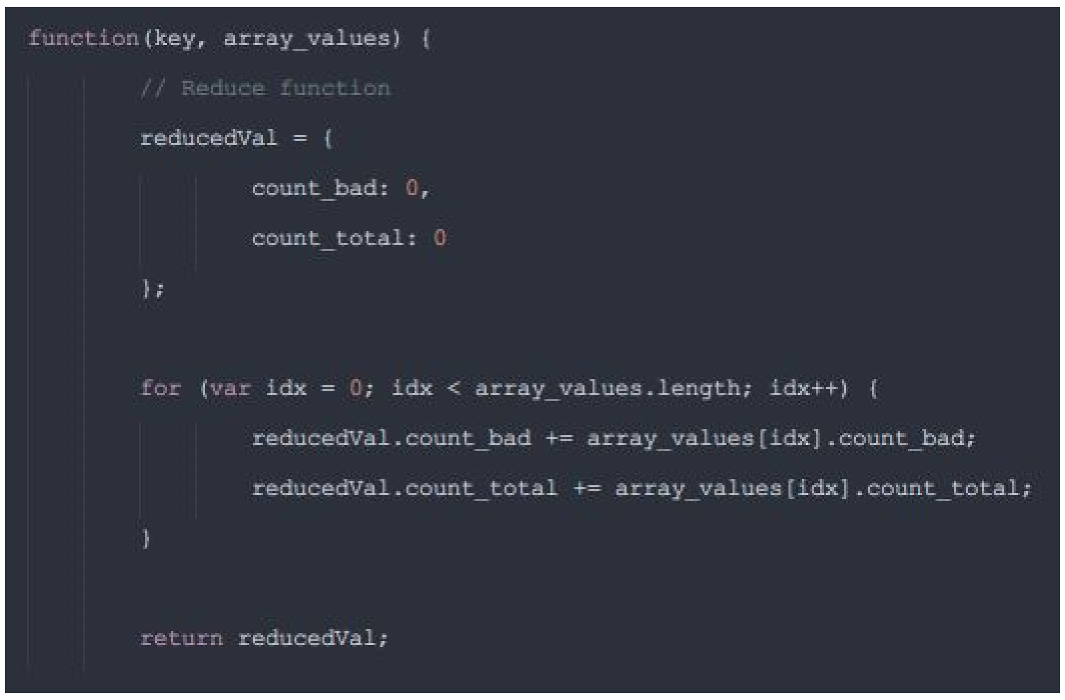
Booyah! A little more Googling and reading, I came across the notorious MapReduce. It’s a cool way to filter and process a large number of results in parallel. Although the way I used it, it ended up being a complicated GROUPBY statement.

In my map function, I emitted a key of all the features I thought would be important in guessing if a borrower was “good” or “bad.”



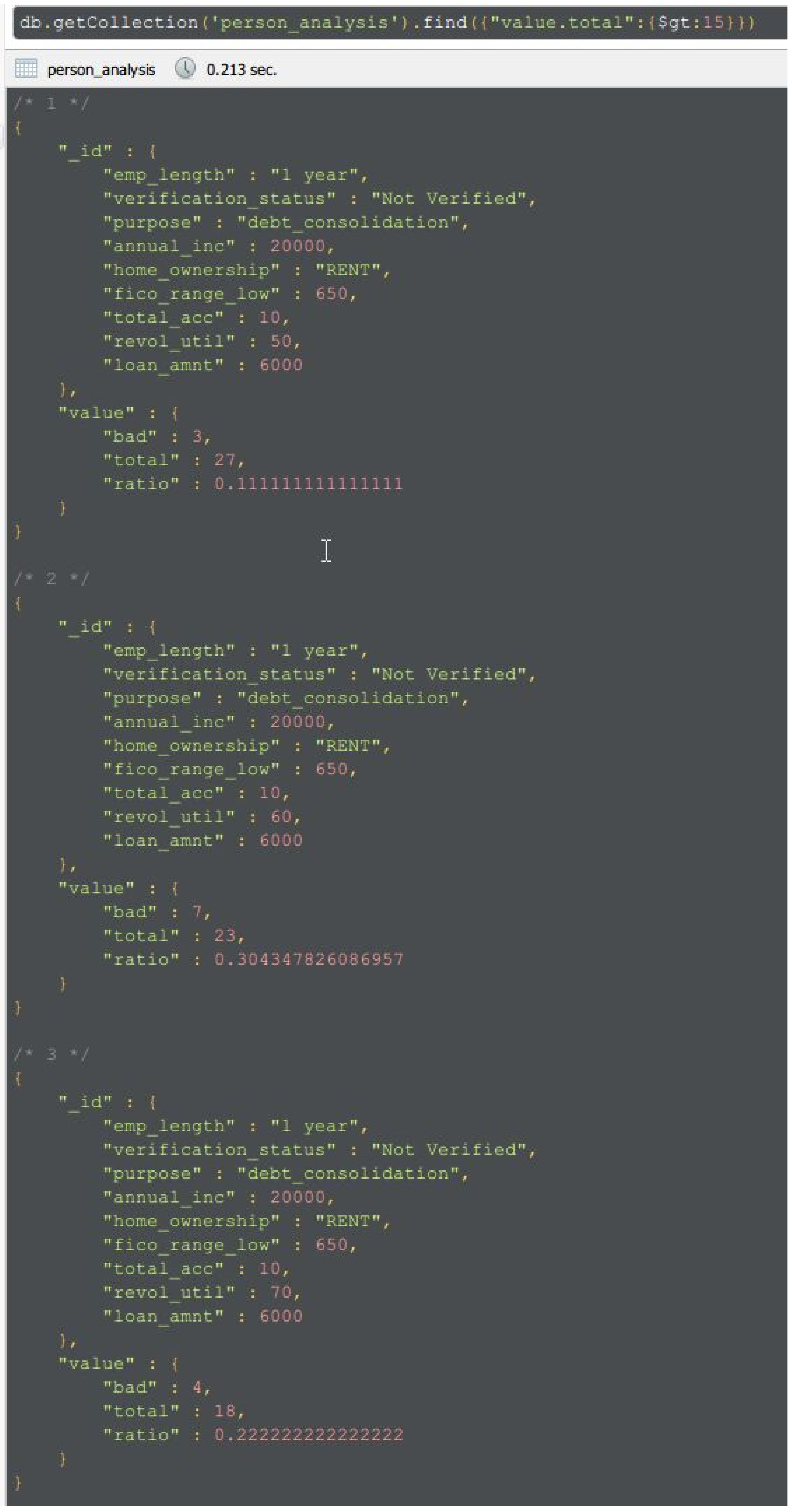
```javascript
var key = {
        emp_length: this.emp_length,
        verification_status: this.verification_status,
        purpose: this.purpose,
        annual_inc: roundDown(this.annual_inc, 20000),
        home_ownership: this.home_ownership,
        fico_range_low: roundDown(this.fico_range_low, 50),
        total_acc: roundDown(this.total_acc, 10),
        revol_util: roundDown(this.revol_util, 10),
        loan_amnt: roundDown(this.loan_amnt, 3000)

};

var bad = 0;
if (this.loan_status == "Default" || this.loan_status == "Charged Off") {
        bad = 1;
}

emit(key, {
        count_bad: bad,
        count_total: 1
});
```

Then in the reduce function, I would add up the counts:



```javascript
function(key, array_values) {
        // Reduce function
        reducedVal = {
                count_bad: 0,
                count_total: 0
        };

        for (var idx = 0; idx < array_values.length; idx++) {
                reducedVal.count_bad += array_values[idx].count_bad;
                reducedVal.count_total += array_values[idx].count_total;
        }

        return reducedVal;
```

And the count_bad/count_total was the final ratio for some set of features. A higher ratio meant more risk. I threw this in a new collection which looked like this:



```
db.getCollection('person_analysis').find({"value.total":{$gt:15}})
```

person_analysis 0.213 sec.

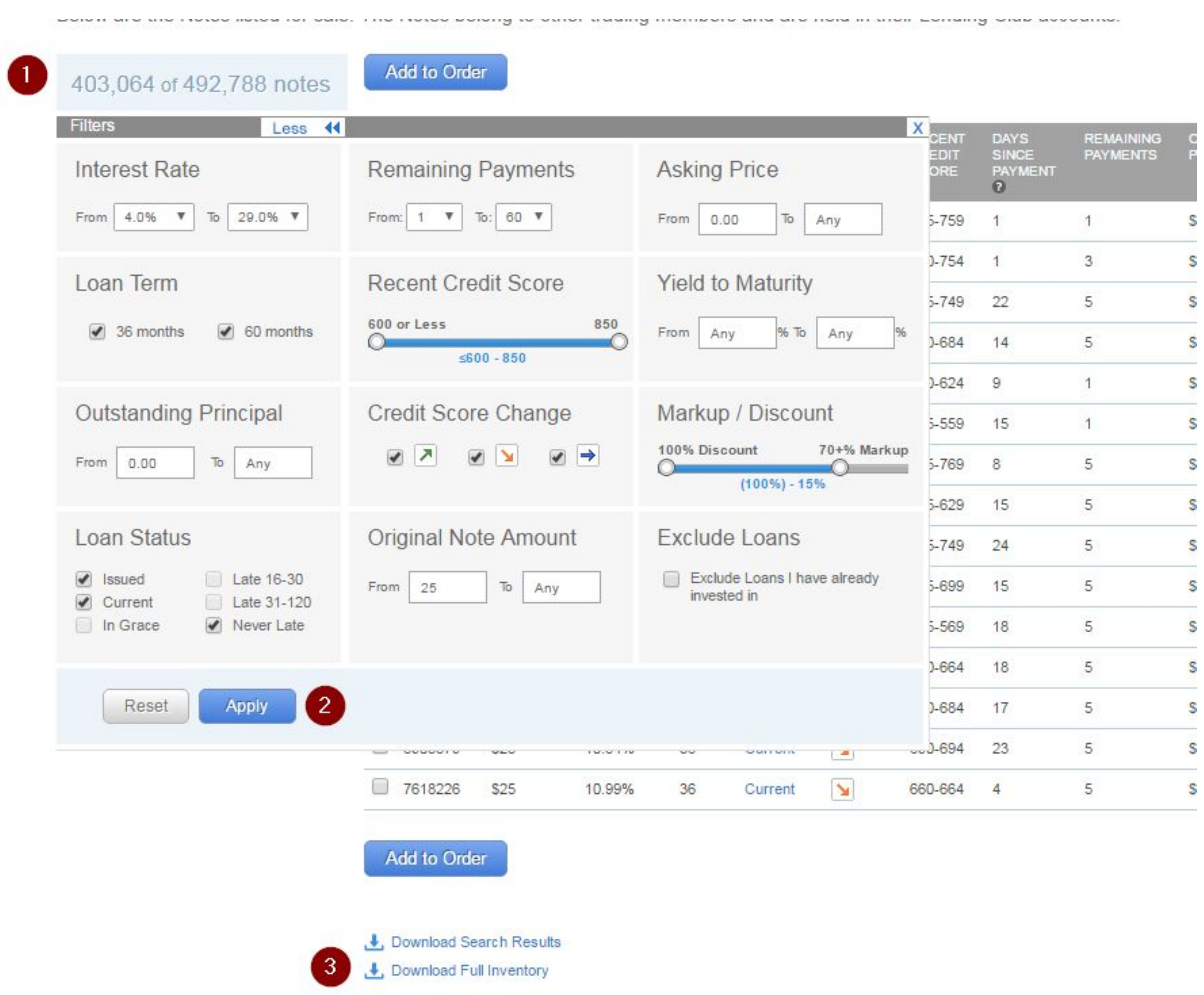
```
/* 1 */
{
    "_id" : {
        "emp_length" : "1 year",
        "verification_status" : "Not Verified",
        "purpose" : "debt_consolidation",
        "annual_inc" : 20000,
        "home_ownership" : "RENT",
        "fico_range_low" : 650,
        "total_acc" : 10,
        "revol_util" : 50,
        "loan_amnt" : 6000
    },
    "value" : {
        "bad" : 3,
        "total" : 27,
        "ratio" : 0.111111111111111
    }
}

/* 2 */
{
    "_id" : {
        "emp_length" : "1 year",
        "verification_status" : "Not Verified",
        "purpose" : "debt_consolidation",
        "annual_inc" : 20000,
        "home_ownership" : "RENT",
        "fico_range_low" : 650,
        "total_acc" : 10,
        "revol_util" : 60,
        "loan_amnt" : 6000
    },
    "value" : {
        "bad" : 7,
        "total" : 23,
        "ratio" : 0.304347826086957
    }
}

/* 3 */
{
    "_id" : {
        "emp_length" : "1 year",
        "verification_status" : "Not Verified",
        "purpose" : "debt_consolidation",
        "annual_inc" : 20000,
        "home_ownership" : "RENT",
        "fico_range_low" : 650,
        "total_acc" : 10,
        "revol_util" : 70,
        "loan_amnt" : 6000
    },
    "value" : {
        "bad" : 4,
        "total" : 18,
        "ratio" : 0.222222222222222
    }
}
```

Later, I can use this table to estimate risk for a given loan. If I'm happy with the ratio, I buy a note/stake in the loan. If you're confused, this is the right side of the block diagram.

Step 3. Scrape secondary market loans

The secondary market allows investors to buy or sell notes that have been issued and that might already have a history of of payments.

(1) There are typically (I've never seen less) hundreds of thousands of notes available on the secondary market
(2) FolioFN Provides a way to filter all these notes
(3) You can then download .csv file containing only filtered notes or all notes on the secondary market (400k in the screenshot)



403,064 of 492,788 notes

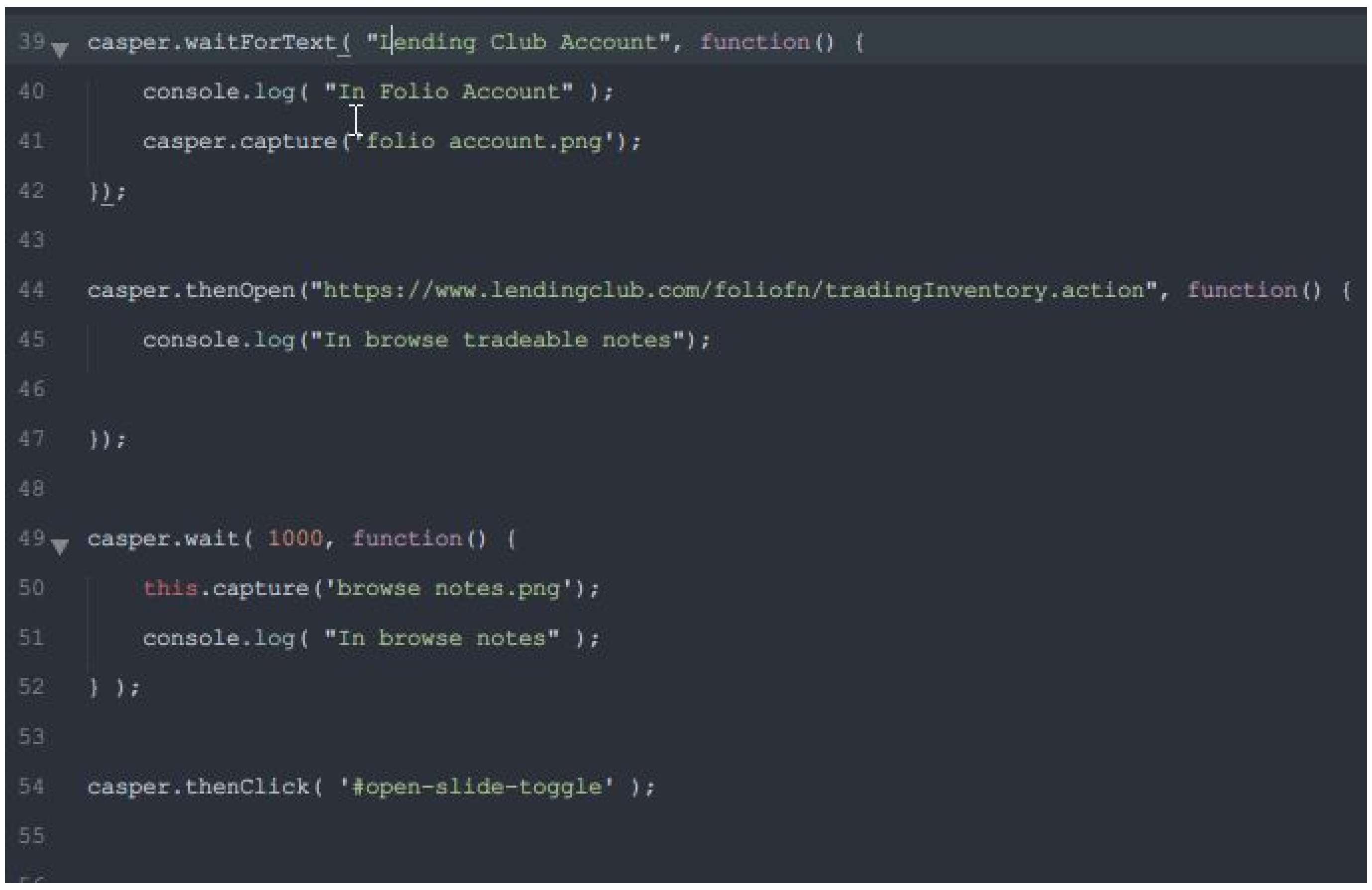
Interest Rate: From 4.0% To 29.0%
Remaining Payments: From: 1 To: 60
Asking Price: From 0.00 To Any
Loan Term: [✓ 36 months] [✓ 60 months]
Recent Credit Score: 600 or Less – 850; ≤600 - 850
Yield to Maturity: From Any % To Any %
Outstanding Principal: From 0.00 To Any
Credit Score Change: [✓ ↗] [✓ ↘] [✓ →]
Markup / Discount: 100% Discount – 70+% Markup; (100%) - 15%
Loan Status: [✓ Issued] [✓ Current] In Grace, Late 16-30, Late 31-120, [✓ Never Late]
Original Note Amount: From 25 To Any
Exclude Loans: Exclude Loans I have already invested in

						CENT EDIT ORE	DAYS SINCE PAYMENT	REMAINING PAYMENTS	
						5-759	1	1	$
						0-754	1	3	$
						5-749	22	5	$
						0-684	14	5	$
						0-624	9	1	$
						5-559	15	1	$
						5-769	8	5	$
						5-629	15	5	$
						5-749	24	5	$
						5-699	15	5	$
						5-569	18	5	$
						0-664	18	5	$
						0-684	17	5	$
					↘	0-694	23	5	$
7618226	$25	10.99%	36	Current	↘	660-664	4	5	$

It's easy to go straight to downloading the full inventory. However, the full inventory doesn't have a way to filter out loans that I already have a stake in. For this reason, I used the "Exclude Loans I have already invested in" filter and only ever tried processing a set of filtered loans. At the time I came up with this, the Foliofn API didn't exist.

To start, I would go to this page each day, apply my filters, and then download the .csv. After .maybe 2 days I was sick of it. I did my Googling, and found PhantomJS (and CasperJS)

CasperJS lets me pretend to be a user clicking things in a browser. I used it to log into Lending Club, go enter the filters I normally do by hand, and download the .CSV. It was pretty easy to set up and even easier to start using.



```
casper.waitForText( "lending Club Account", function() {
    console.log( "In Folio Account" );
    casper.capture('folio account.png');
});

casper.thenOpen("https://www.lendingclub.com/foliofn/tradingInventory.action", function() {
    console.log("In browse tradeable notes");

});

casper.wait( 1000, function() {
    this.capture('browse notes.png');
    console.log( "In browse notes" );
} );

casper.thenClick( '#open-slide-toggle' );

```

Step 4. Automate things!

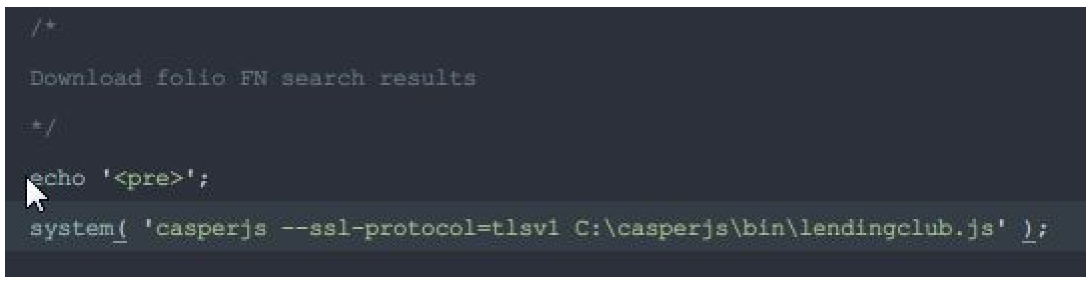
Loan buying backend

Now that I've got a list of notes that are currently for sale, a database collection of existing loans, and a database collection of predicted risk, I think it's time to put them all together!

Puts stuff together

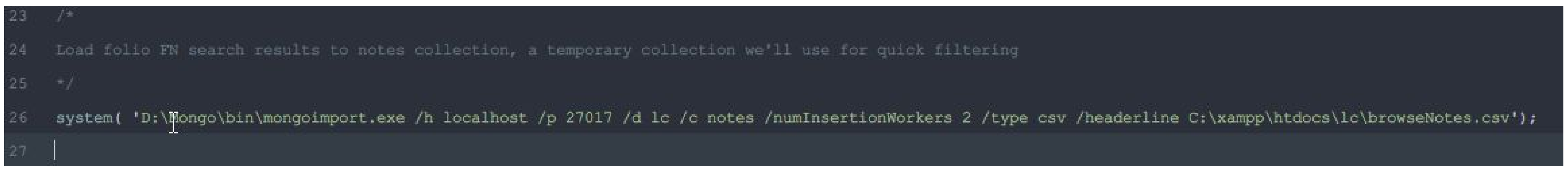
Here's a breakdown of what my backend does:

First, it runs the CasperJS script to download the notes for sale for me, saving me another step.



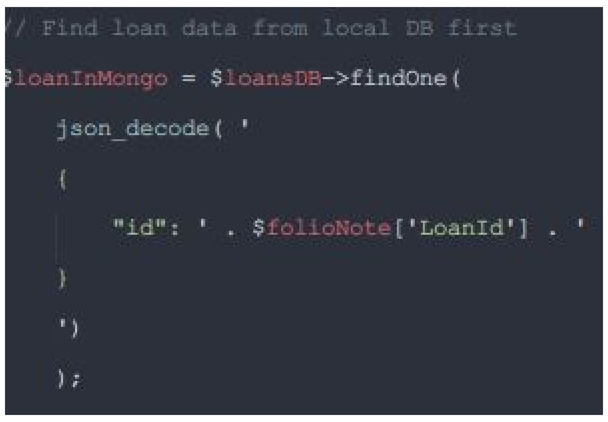
```
/*
Download folio FN search results
*/
echo '<pre>';
system( 'casperjs --ssl-protocol=tlsv1 C:\casperjs\bin\lendingclub.js' );
```

Then, I load all those notes into a temporary Mongo collection called *notes*. This step was really not necessary. I could've just parsed the csv lines in PHP but I wanted to do it differently.



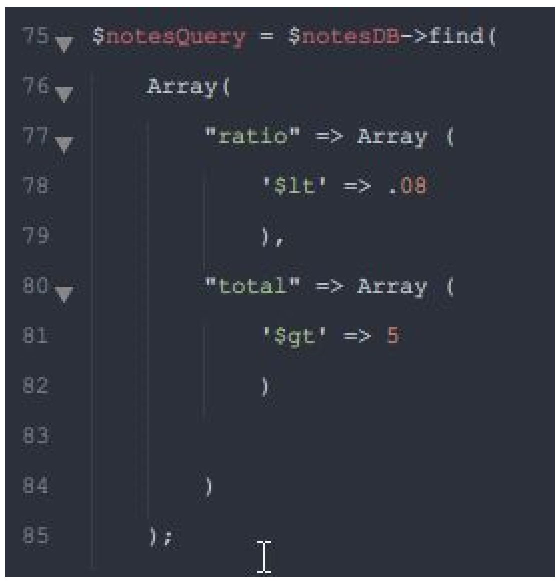
```
/*
Load folio FN search results to notes collection, a temporary collection we'll use for quick filtering
*/
system( 'D:\Mongo\bin\mongoimport.exe /h localhost /p 27017 /d lc /c notes /numInsertionWorkers 2 /type csv /headerline C:\xampp\htdocs\lc\browseNotes.csv');
```

Next I go through each of these notes for sale, and look up the original loan data if it exists (by using the LoadID)



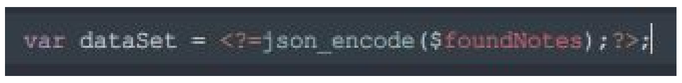
```
// Find loan data from local DB first
$loanInMongo = $loansDB->findOne(
    json_decode( '
    {
        "id": ' . $folioNote['LoanId'] . '
    }
    ')
    );
```

If I get a match, I'll run the loan data against my predicted risk collection and add the risk to that note data. Then, I can go back and query all the notes that have some estimated risk. In this case, I want to find notes that are "safer" than 8% bad. I also don't want to judge based on a dataset of 1-2 notes (say a loan with borrower's annual income=21050, purpose=vacation, … some very specific traits)



```
$notesQuery = $notesDB->find(
    Array(
        "ratio" => Array (
            '$lt' => .08
            ),
        "total" => Array (
            '$gt' => 5
            )

        )
    );
```

Finally, I pass a group of notes that I might want to buy to the frontend



```
var dataSet = <?=json_encode($foundNotes);?>;
```

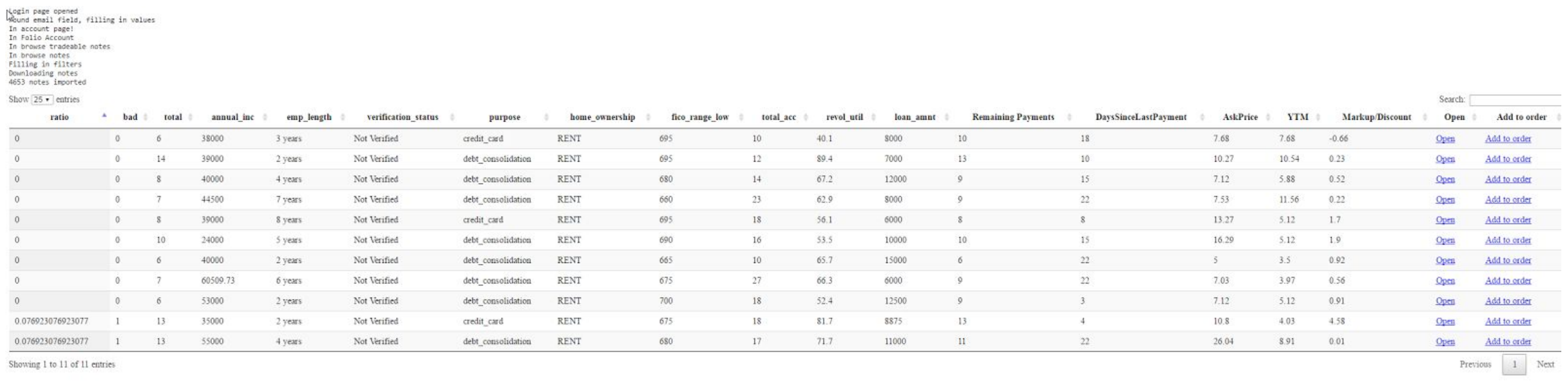
Loan buying front-end

So now that I've got all of the secondary markets for sale and I'm able to cross reference those to the loan data from before, I need to pick and choose which ones I'm actually going to buy. At the time there was no API to buy/sell notes on the secondary market so I had to go through potential notes one-by-one and buy them. I made it as easy as I could (down to 1-2 clicks). It could have been automated further.

I used DataTables in the frontend. I had used it for work and really liked it so I ended up re-using it here. Being able to sort by columns without a lot of configuration was a nice feature. I could've done this in a Python GUI but web is easiest to implement for me.

The backend (PHP) did a whole lot of automation for me. It

- cleared out the temporary notes database
- called the CasperJS crawler
- loaded the resulting CSV of notes available on the secondary market into MongoDB
- Checked each note by looking up its loan data from the Loans collection and using the person_analysis collection
- Filtered whatever was a match by ratio (see the left 3 columns)
- Gave me links to Open (see more details about the note) or Add to order (add to FolioFn cart)



```
Login page opened
Found email field, filling in values
In account page!
In Folio Account
In browse tradeable notes
In browse notes
Filling in filters
Downloading notes
4653 notes imported
```

Show 25 entries

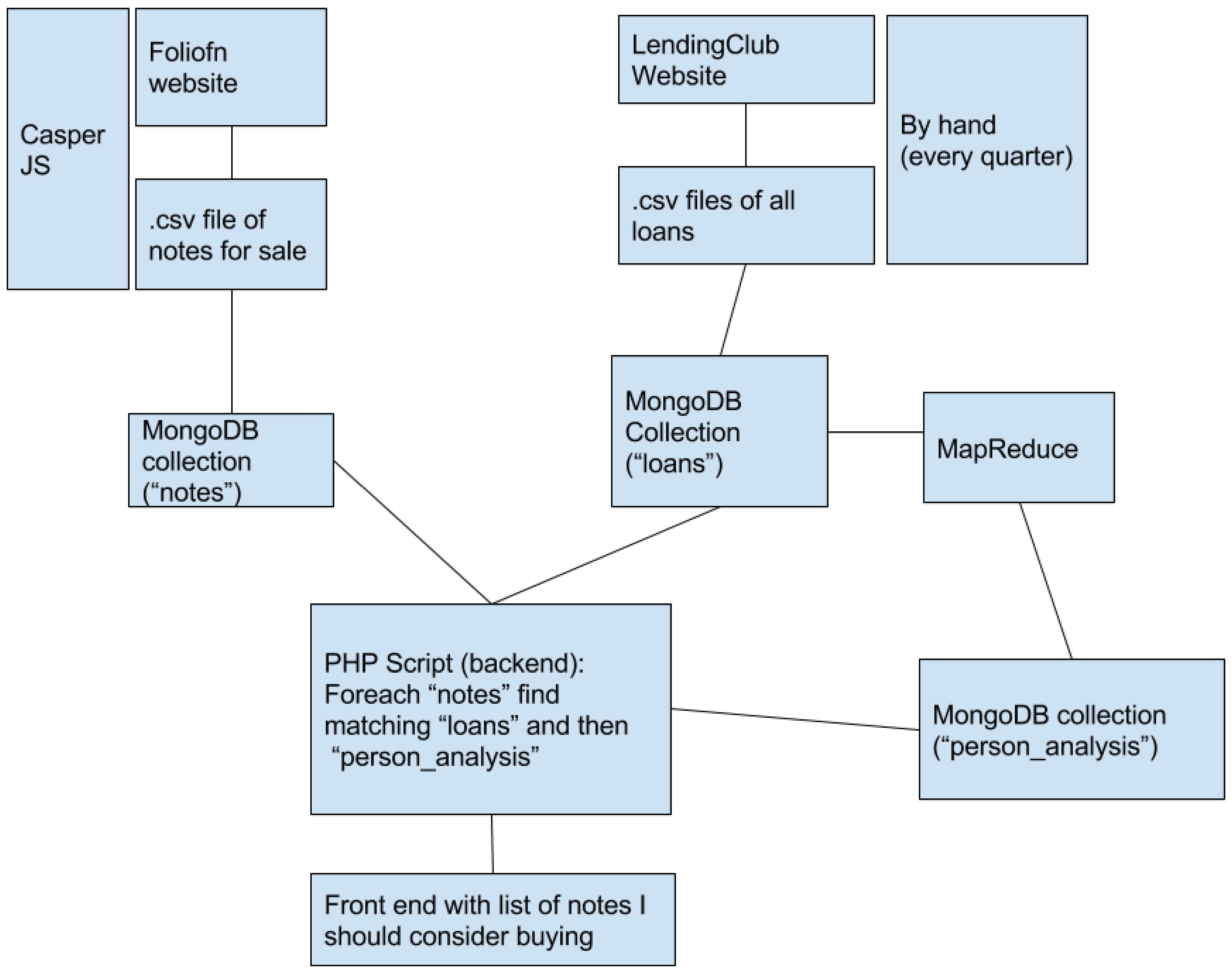
Search:

ratio	bad	total	annual_inc	emp_length	verification_status	purpose	home_ownership	fico_range_low	total_acc	revol_util	loan_amnt	Remaining Payments	DaysSinceLastPayment	AskPrice	YTM	Markup/Discount	Open	Add to order
0	0	6	38000	3 years	Not Verified	credit_card	RENT	695	10	40.1	8000	10	18	7.68	7.68	-0.66	Open	Add to order
0	0	14	39000	2 years	Not Verified	debt_consolidation	RENT	695	12	89.4	7000	13	10	10.27	10.54	0.23	Open	Add to order
0	0	8	40000	4 years	Not Verified	debt_consolidation	RENT	680	14	67.2	12000	9	15	7.12	5.88	0.52	Open	Add to order
0	0	7	44500	7 years	Not Verified	debt_consolidation	RENT	660	23	62.9	8000	9	22	7.53	11.56	0.22	Open	Add to order
0	0	8	39000	8 years	Not Verified	credit_card	RENT	695	18	56.1	6000	8	8	13.27	5.12	1.7	Open	Add to order
0	0	10	24000	5 years	Not Verified	debt_consolidation	RENT	690	16	53.5	10000	10	15	16.29	5.12	1.9	Open	Add to order
0	0	6	40000	2 years	Not Verified	debt_consolidation	RENT	665	10	65.7	15000	6	22	5	3.5	0.92	Open	Add to order
0	0	7	60509.73	6 years	Not Verified	debt_consolidation	RENT	675	27	66.3	6000	9	22	7.03	3.97	0.56	Open	Add to order
0	0	6	53000	2 years	Not Verified	debt_consolidation	RENT	700	18	52.4	12500	9	3	7.12	5.12	0.91	Open	Add to order
0.076923076923077	1	13	35000	2 years	Not Verified	credit_card	RENT	675	18	81.7	8875	13	4	10.8	4.03	4.58	Open	Add to order
0.076923076923077	1	13	55000	4 years	Not Verified	debt_consolidation	RENT	680	17	71.7	11000	11	22	26.04	8.91	0.01	Open	Add to order

Showing 1 to 11 of 11 entries

Previous 1 Next

Block Diagram


Casper JS
Foliofn website
.csv file of notes for sale
LendingClub Website
.csv files of all loans
By hand (every quarter)
MongoDB collection ("notes")
MongoDB Collection ("loans")
MapReduce
PHP Script (backend): Foreach "notes" find matching "loans" and then "person_analysis"
MongoDB collection ("person_analysis")
Front end with list of notes I should consider buying

Along came Pandas

The Mongo+PHP+CasperJS system was great for a while. But then I found Pandas, a nice library for Python that made data munging and analysis super easy. I won't write about all of it here, but it's on GitHub (https://github.com/dharik/Lending-Club). In fact, all of the above is there now too.

Lessons Learned

Use the best tool for the job.

In the first step (Get the dataset), I said that the loan data was a good fit for a relational database and that I instead used MongoDB just for the sake of learning. This was like getting a new drill and trying to use it as a hammer. I'm glad I got to use MongoDB, other options would have been better. In fact, PostgreSQL only weeks after I started with MongoDB added support for JSON data types.

Now, I could have used MongoDB in a way that it's better suited, such as putting each note as a subdocument for the loan it belongs to. But I stopped due to time and LendingClub having some big problems.

Use version control.

While I did my "analysis," all along the way I was modifying my files and not saving copies of them anywhere. There was some good stuff I found that I ended up erasing just because I wanted to try something else. It would've helped to go back to a previous state.

What next

It was a fun project. I doubt I'll continue working on it because LendingClub isn't doing so hot anymore. Some sketchy activities and also their website+API going offline frequently so I'm not even able to run my scripts.

Computer vision looks fun, challenging, and practical. I'd love to try out some GPU processing.