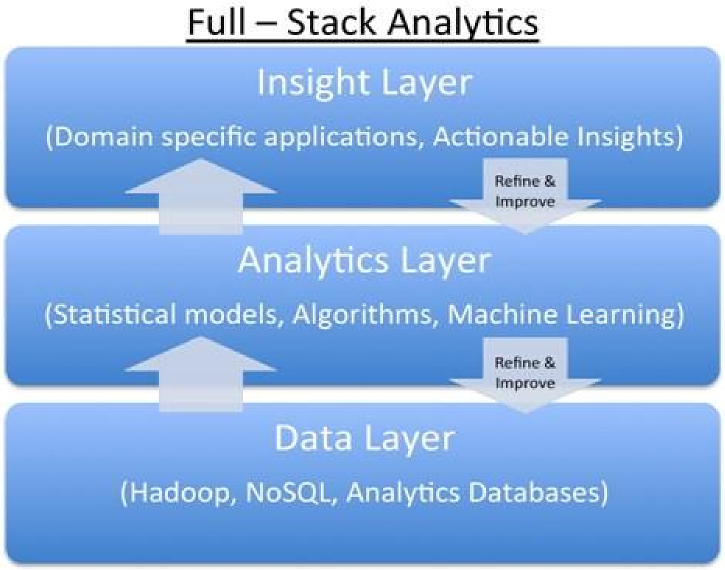
Deploying and Evaluating Data Products

https://db.tt/JIYOoiPu

Josh Levy, PhD

About Vast

Data and analytics for considered purchases (vehicles, homes, ...)



White Label Marketplaces, Market Reports, Sales Apps

Pricing, Supply, Demand, Recommendations, Behavior

Inventory (rapid churn), Consumer Behavior

Graphic: Chip Hazard (Flybridge Capital Partners)
http://www.kdnuggets.com/2014/05/stacking-deck-next-wave-opportunity-big-data.html

Maturity levels for turning "models" into "products"

1. Can we deploy one model into a production environment?
2. Given two models that perform similar functions can we evaluate which is better?
3. Can we operationalize model training?

Goals for Mature Data Products

New version of a model automatically

- trained
- deployed
- evaluated

Traffic automatically routed to top performer

Outline for this talk

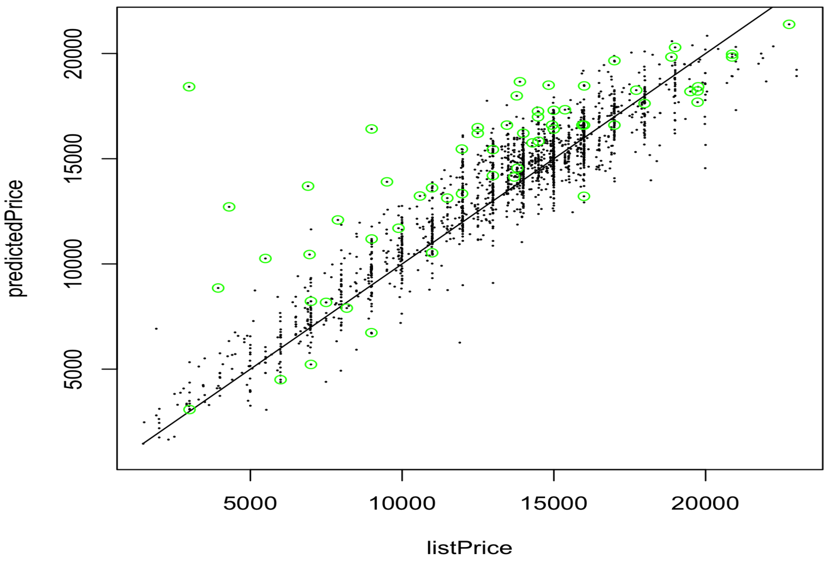
1. Can we deploy one model into a production environment?
2. Given two models that perform similar functions can we evaluate which is better?

Deploying models is hard

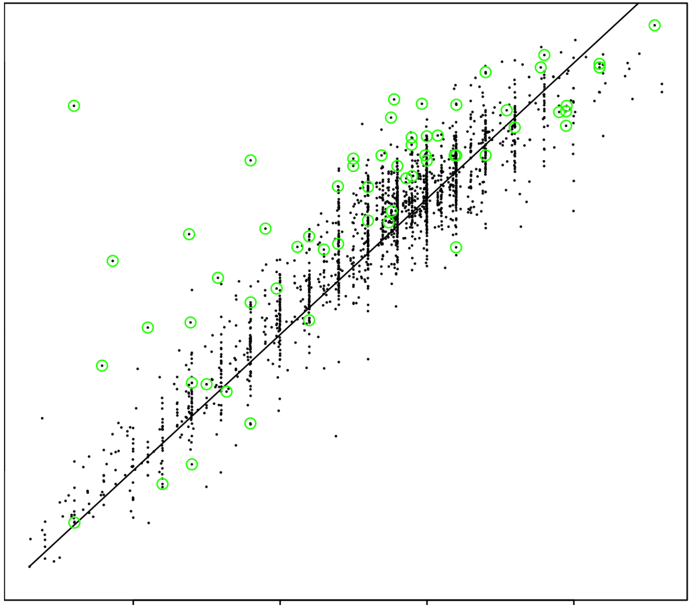
Conway's Law helps to explain why

> *Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations*

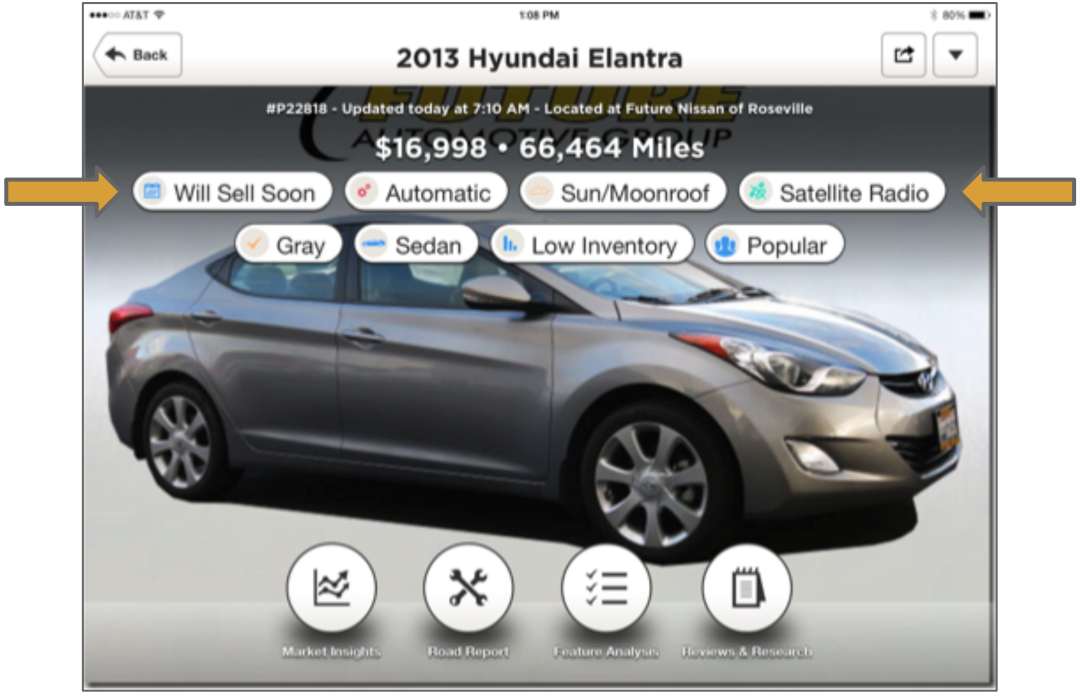
Vast has put together a Data Science team that thinks about training and validating **models**, running **experiments**



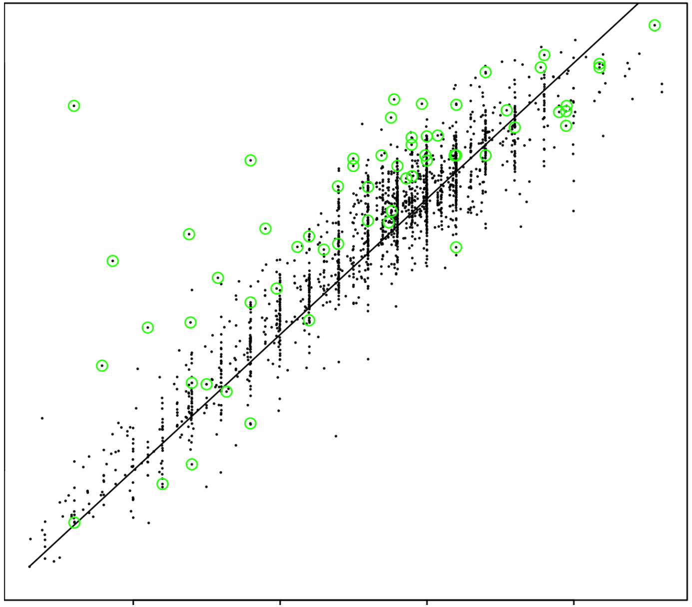
Models don't exist in isolation.



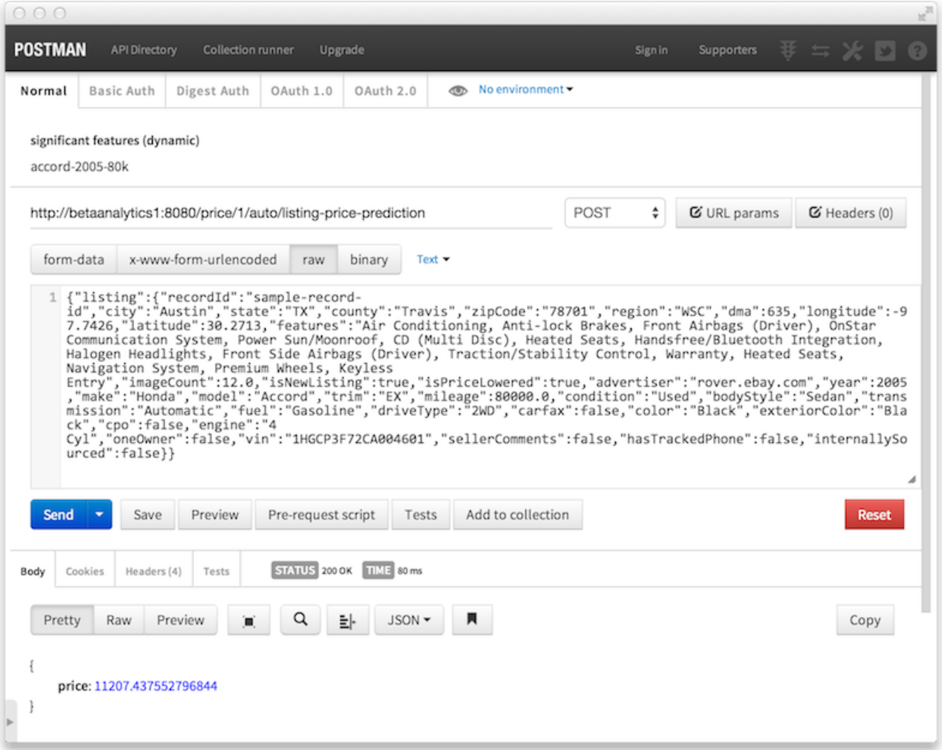
Add value when exposed in a product.



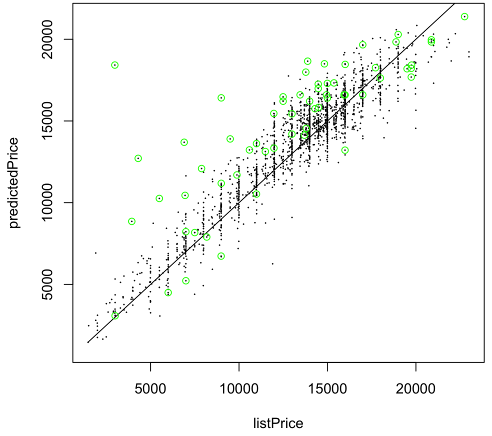
Models don't exist in isolation.

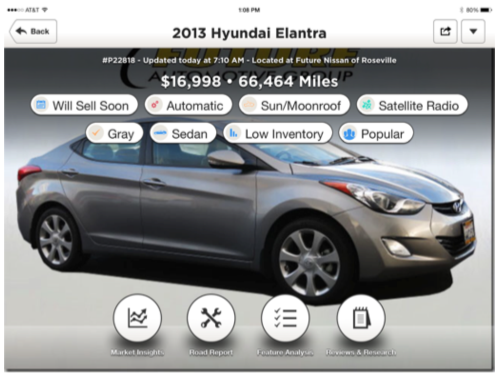


Add value when exposed as a product.



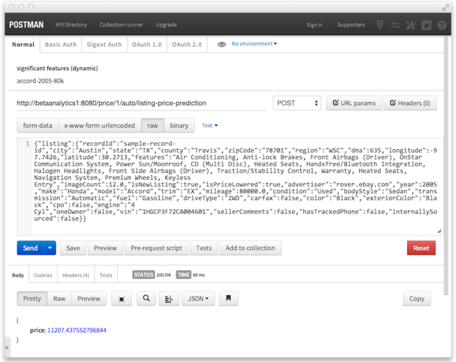
Communication barriers between these teams





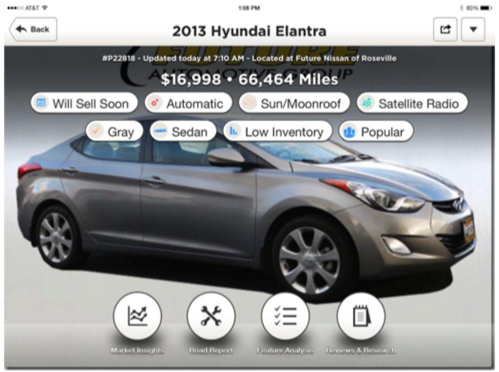
???

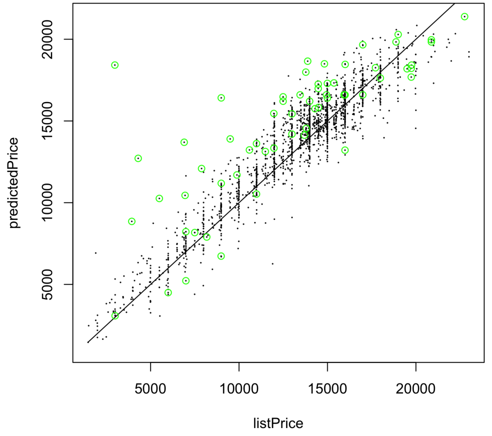
???



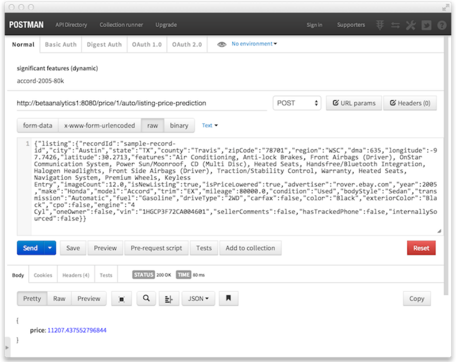
Simplified Problem

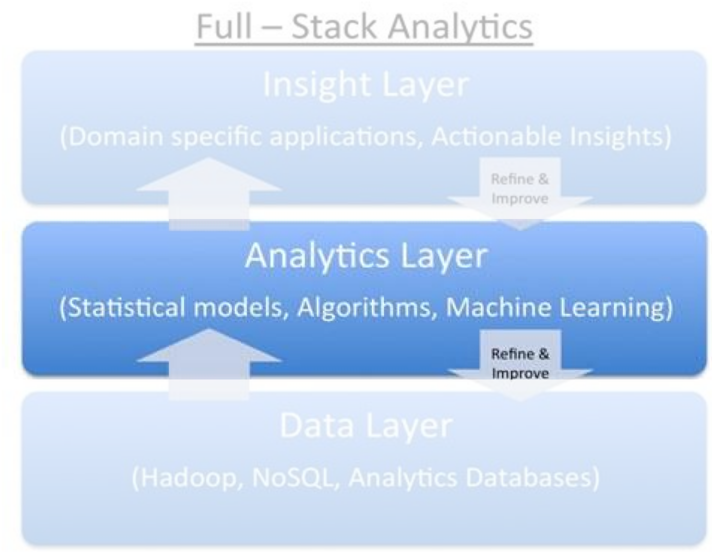
Expose models as services internal apps and external customers can access.



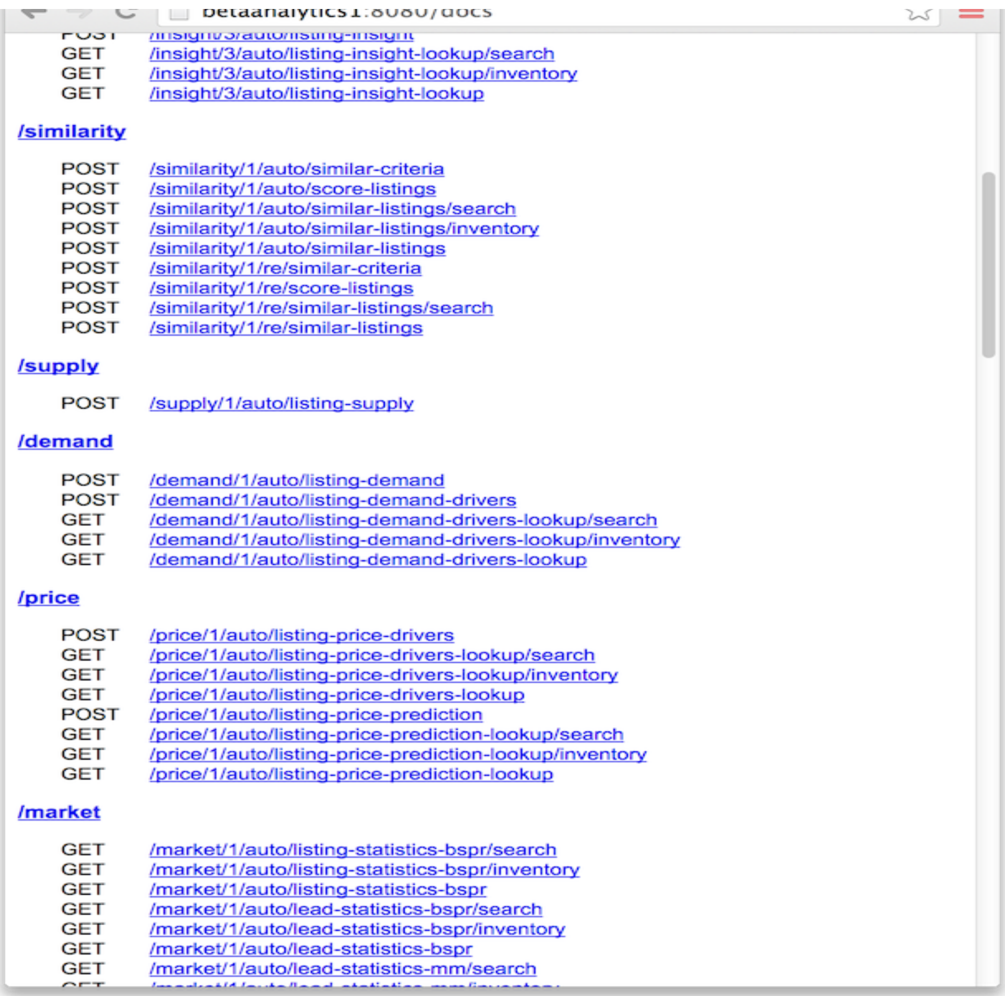


???





Data Science and Engineering collaborate to build Analytics Layer



betaanalytics1:8080/docs/price/1/auto/listing-price-drivers

POST /price/1/auto/listing-price-drivers

Retrieve price drivers for the auto listing in the POST body.

Query Parameters

modelName
  (optional) the name of the analytical model to be used [default]
includeModelContext
  (optional) whether or not to include model context in the response [true, false]
callerId
  (optional) a user-friendly identifier for the originating application
partnerId
  (optional) a user-friendly identifier for the originating partner

Samples

Request JSON
Response JSON

analytics-service-server-5.8.0-86652


```
{
- listing: {
    recordId: "sample-record-id",
    city: "Austin",
    state: "TX",
    county: "Travis",
    zipCode: "78701",
    region: "WSC",
    dma: 635,
    longitude: -97.7426,
    latitude: 30.2713,
    features: "Air Conditioning, Anti-lock Brakes, Front Airbags (Driver), OnStar Communication System, Power Sun/Moonroof, CD (Multi Disc),
    Heated Seats, Handsfree/Bluetooth Integration, Halogen Headlights, Front Side Airbags (Driver), Traction/Stability Control, Warranty,
    Heated Seats, Navigation System, Premium Wheels, Keyless Entry",
    imageCount: 12,
    isNewListing: true,
    isPriceLowered: true,
    advertiser: "rover.ebay.com",
    price: 20000,
    year: 2010,
    make: "Honda",
    model: "Accord",
    trim: "EX",
    mileage: 80000,
    condition: "Used",
    bodyStyle: "Sedan",
    transmission: "Automatic",
    fuel: "Gasoline",
    driveType: "2WD",
    carfax: false,
    color: "Black",
    exteriorColor: "Black",
    cpo: false,
    engine: "4 Cyl",
    oneOwner: false,
```

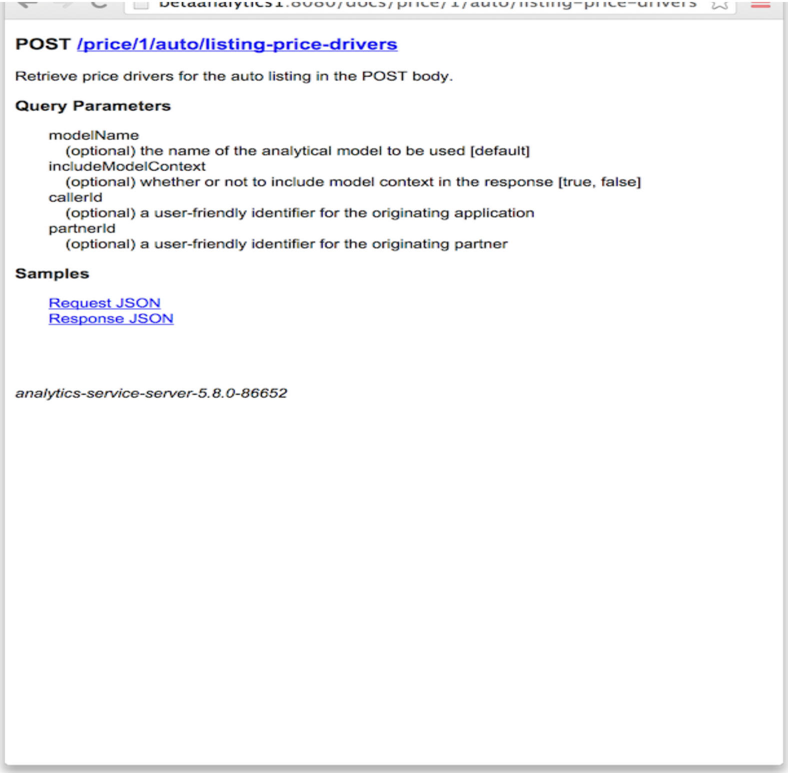


```
[
- {
    featureName: "Wheel Type",
    featureValue: "Premium",
    baselineValue: "Alloy",
    featurePrice: 671
  },
- {
    featureName: "Navigation System",
    featurePrice: 258.4500477481595
  },
- {
    featureName: "Exterior Color",
    featureValue: "Black",
    featurePrice: 0
  }
]
```

Exposing Models as Services

Communications challenges between scientists and engineers

- Human language & concepts
- Technology platforms



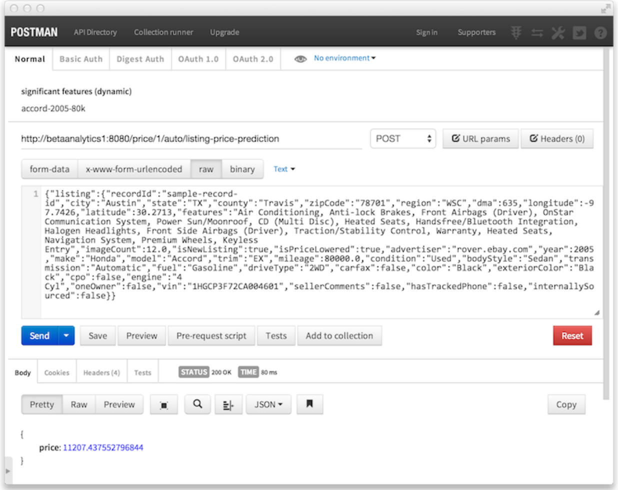
Communication between Humans

Looking for common ground between

Data Scientists thinking about Experiments, Training and Validation

Engineers thinking about Scalability, Deployment, Reliability, and Monitoring

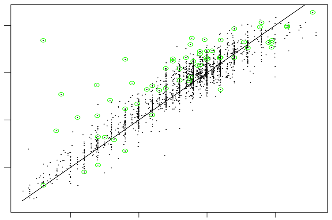
Technology Platforms



Back-end engineering

- Typically JVM (Scala)
- Apps and other platforms can call JSON over HTTP services

Technology Platforms



IP[y]: IPython Interactive Computing

Data Science: "Use the most comfortable tool for the job."

- Typically Python (sklearn) or R for models trained on inventory (millions of rows)
- Hadoop (scalding or streaming) when working with user behavior, systems exhaust

Most comfortable tool for the job

DS team started CS PhD-heavy, DIY mindset

Growth from MS in Business Analytics program

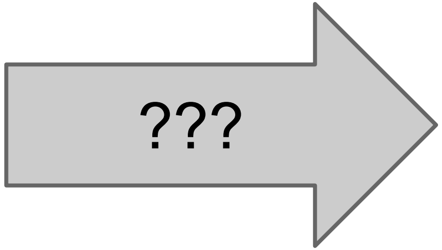
- Strong stats background, productive in R
- Different hiring standards than back end engineering

Expectations for MS in Business Analytics

We're learning to push "Data Janitor" work onto engineering team.

Assuming MSBAs take clean data in, and produce models.

Those models need to become products, but MSBAs don't need to write production scala.



Answers from Vast in chronological order

1. Rewrite scoring function for JVM
2. Export as PMML
3. Expose as JSON over HTTP or WebSockets from Python or R

1. **Rewriting Scoring Function**

Pro

- DevOps is a freebie
- Additional implementations of an interface can be cheap

Con

- First version of a model is expensive
  - Back to Conway's law - difficult communication
- Worry about transcription errors

1. Rewriting Scoring Function

Best Practice

Scientist writes code to generate some model representation.

Engineer writes code to read model and score live data; Exposes that code as service.

2. **Export as PMML**

Pro

- Cheaper, easier than rewrite
  - Scientist generates PMML
  - Engineer plugs in off the shelf runner
- DevOps still free

Con

- Limited to doing things that can be expressed in PMML
- Jumped through hoops for feature transformations

PMML Example: Auto Pricing Model

Training: split according to domain knowledge

- (make, model)
- (age, location)
- (completeness of data)

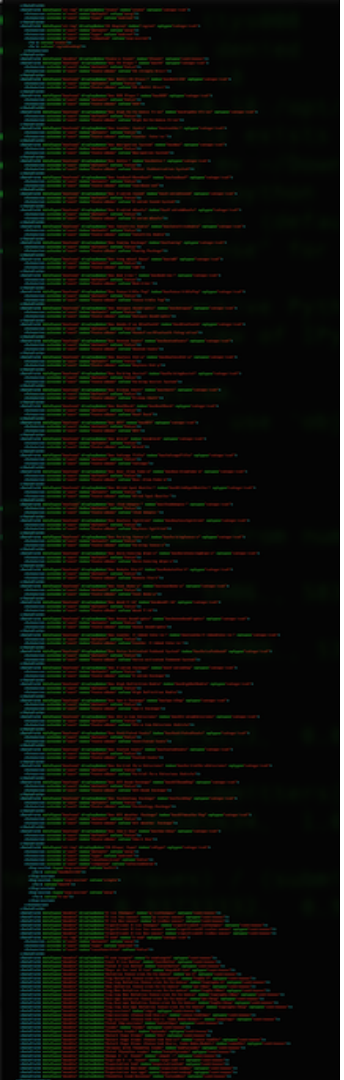
LASSO on each group

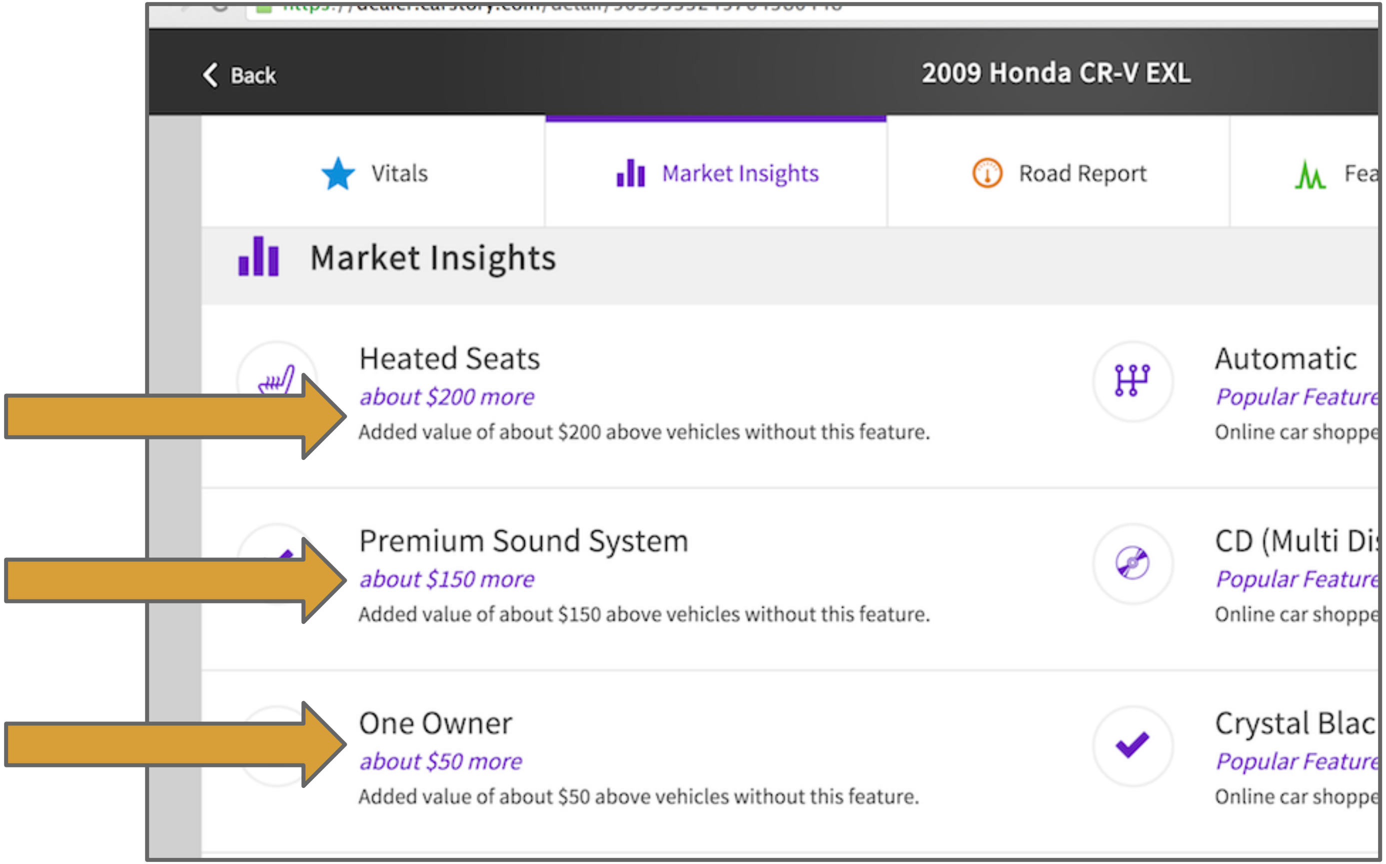
- Predicts total price
- Decomposes price by features

Auto Pricing Model

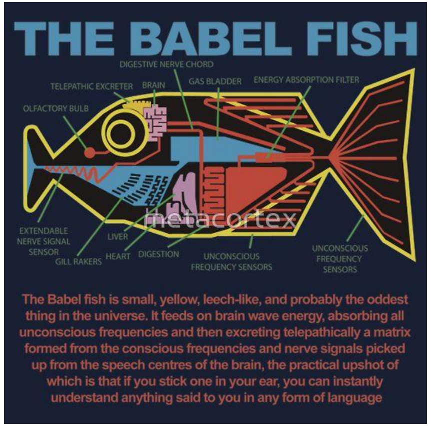
One PMML Decision Tree per make, model

- Internal nodes from manual splits
- ~500 leaf nodes. Each is a regression model.




Back
2009 Honda CR-V EXL
Vitals
Market Insights
Road Report
Market Insights
Heated Seats
about $200 more
Added value of about $200 above vehicles without this feature.
Premium Sound System
about $150 more
Added value of about $150 above vehicles without this feature.
One Owner
about $50 more
Added value of about $50 above vehicles without this feature.
Automatic
CD (Multi Di
Crystal Blac

3. **Expose as JSON over HTTP**



Pro

- Everyone speaks in their native tongue
- More natural data xform than PMML
- automate this: yhat, Wolfram, Azure

Con

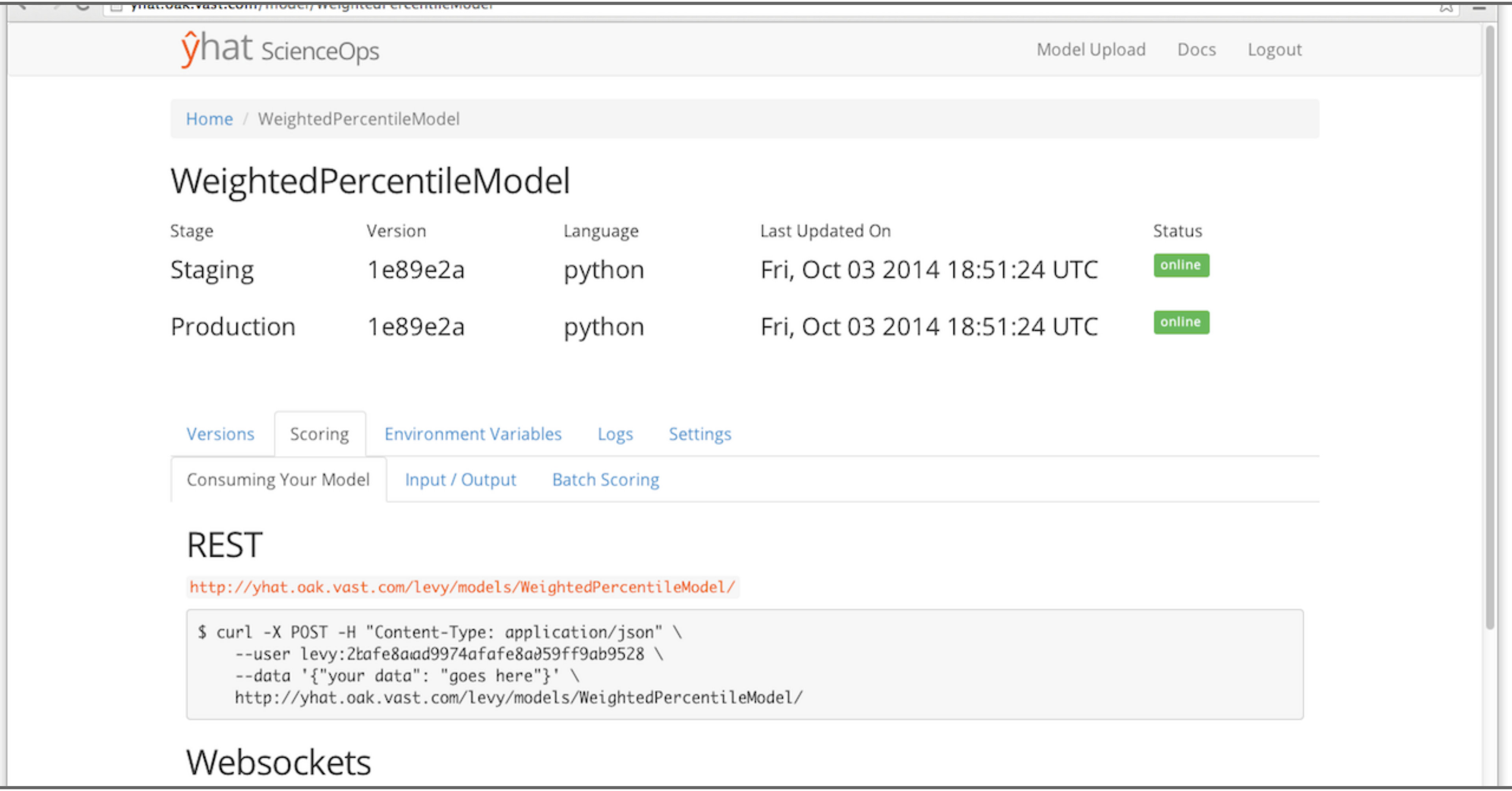
- Dev/Ops needs to harden something new

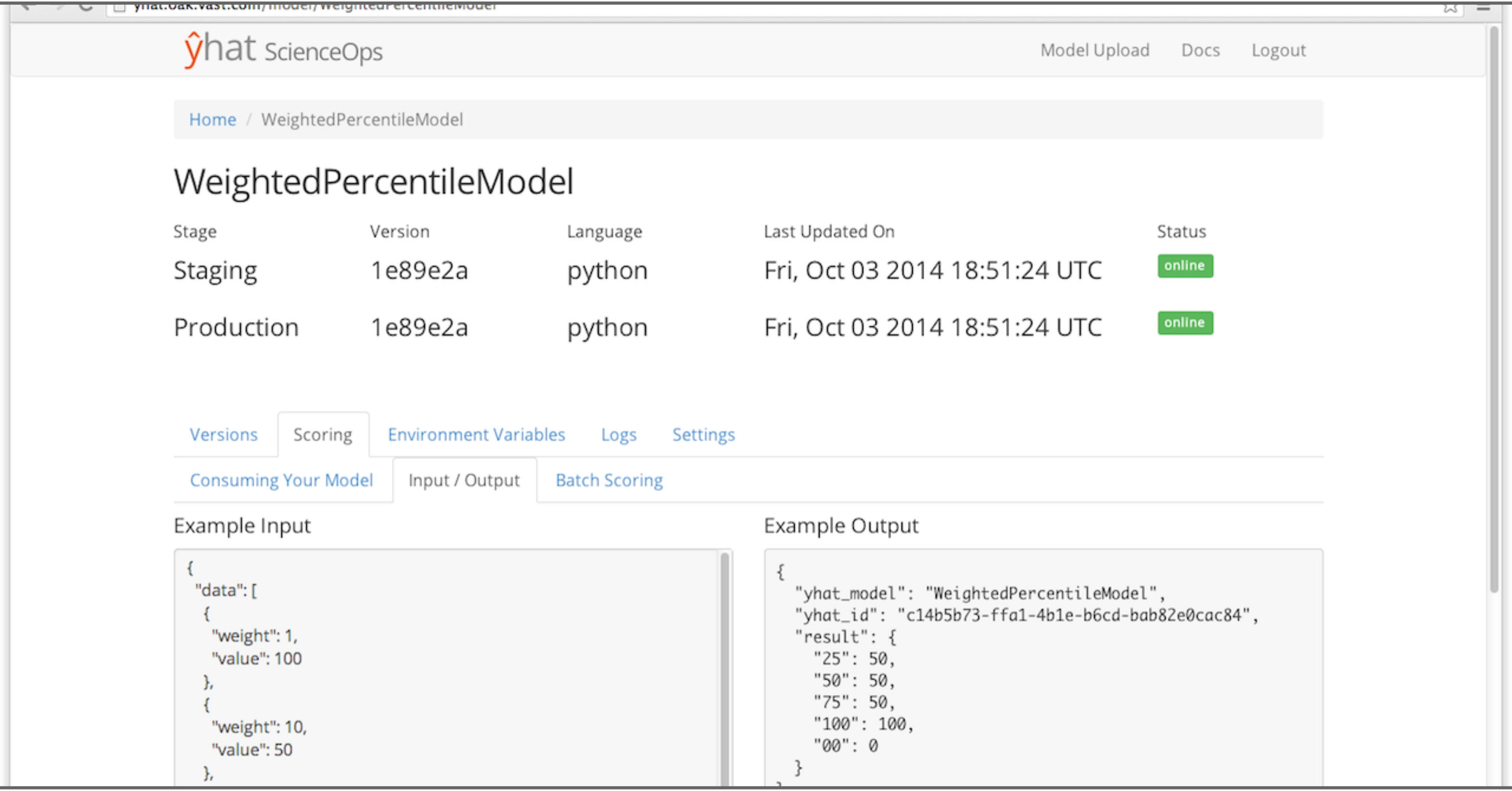
yhat_models — vim — 94×42

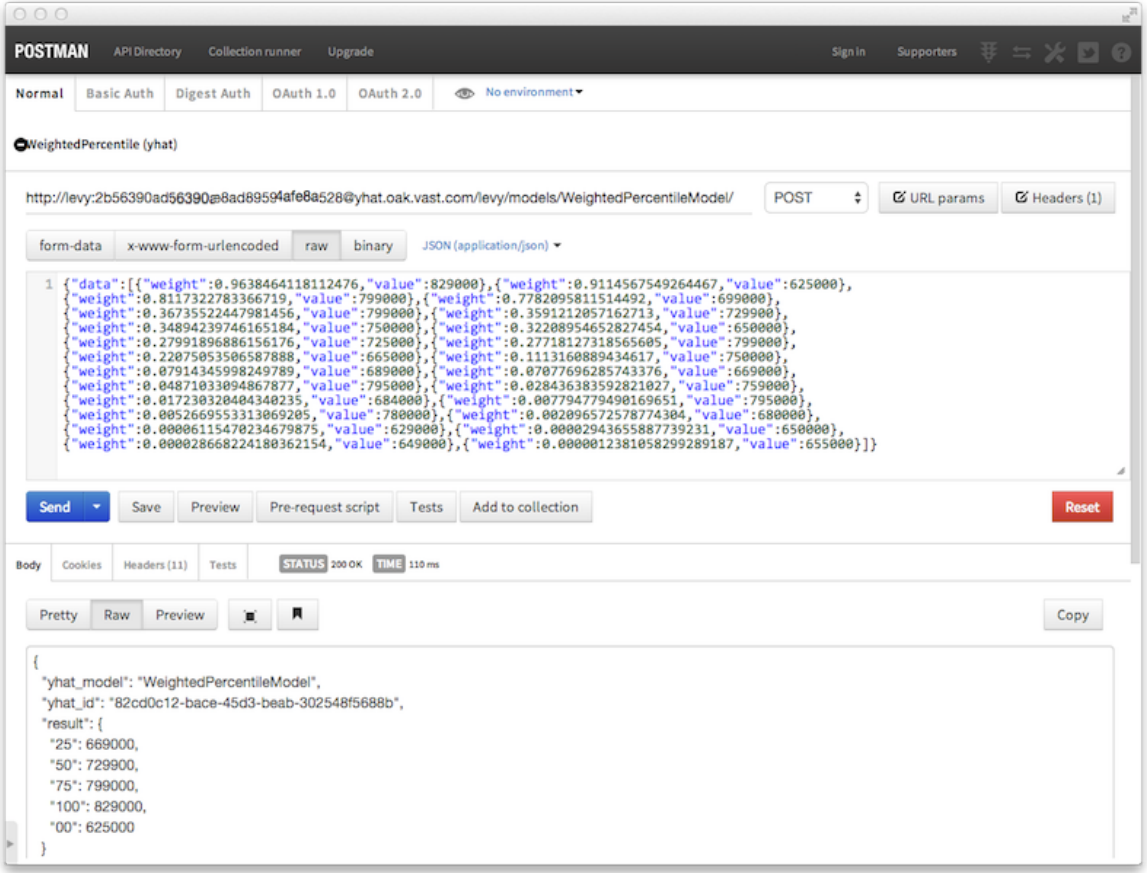
```python
import json
from numpy import array
from operator import itemgetter
from yhat import Yhat, YhatModel , preprocess

class WeightedPercentileModel(YhatModel):


    @preprocess(in_type=dict, out_type=dict)
    def execute(self, data):
        d = sorted(data['data'], key=itemgetter('value'))
        values = array(map(itemgetter('value'), d))
        weights = array(map(itemgetter('weight'), d), dtype=float)
        weights /= weights.sum()
        cs = weights.cumsum()
        results = {'00': values[0], '100': values[-1]}

        for w0, v0, w1, v1 in zip(cs, values, cs[1:], values[1:]):
            print "%0.6f\t%0.6f\t%0.6f\t%0.6f" % (w0, v0, w1, v1)
            if ('75' not in results):
                if (w0 < 0.75) and (w1 >= 0.75):
                    results['75'] = v1
                elif (w0 >= 0.75):
                    results['75'] = v0
                if ('50' not in results):
                    if (w0 < 0.50) and (w1 >= 0.50):
                        results['50'] = v1
                    elif (w0 >= 0.50):
                        results['50'] = v0
                    if ('25' not in results):
                        if (w0 < 0.25) and (w1 >= 0.25):
                            results['25'] = v1
                        elif (w0 >= 0.25):
                            results['25'] = v0
            else:
                break
        return results

if __name__ == "__main__":
    yh = Yhat(USERNAME, API_KEY, "http://yhat.oak.vast.com")
    result = yh.deploy("WeightedPercentileModel", WeightedPercentileModel, globals(), True)
:
```

4 lines of boilerplate to wrap python code in YhatModel

2 more lines to deploy model

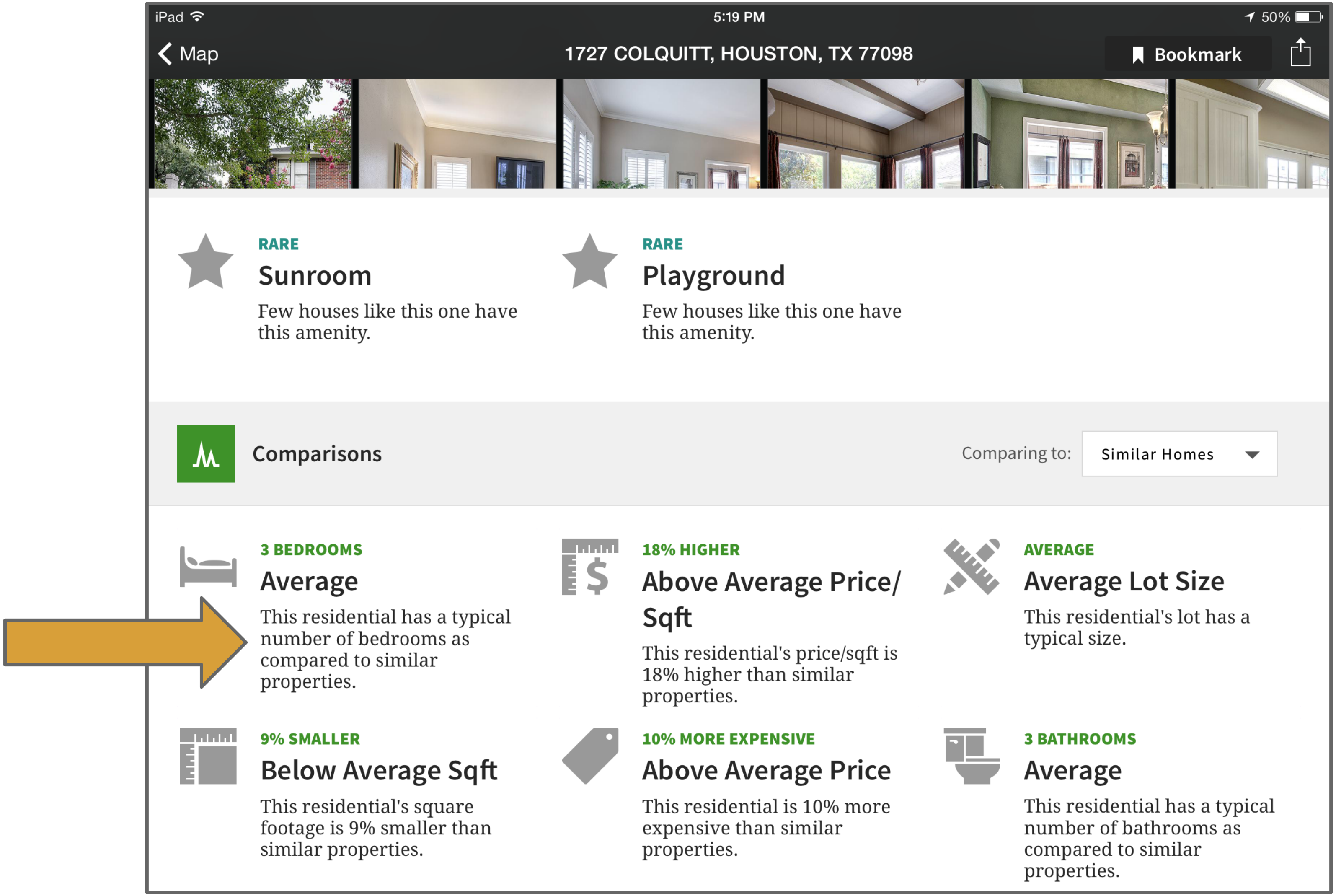

yhat ScienceOps
Model Upload
Docs
Logout
Home / WeightedPercentileModel
WeightedPercentileModel
Stage
Version
Language
Last Updated On
Status
Staging
1e89e2a
python
Fri, Oct 03 2014 18:51:24 UTC
online
Production
1e89e2a
python
Fri, Oct 03 2014 18:51:24 UTC
online
Versions
Scoring
Environment Variables
Logs
Settings
Consuming Your Model
Input / Output
Batch Scoring
REST
http://yhat.oak.vast.com/levy/models/WeightedPercentileModel/
$ curl -X POST -H "Content-Type: application/json" \
--user levy:2bafe8aad9974afafe8a959ff9ab9528 \
--data '{"your data": "goes here"}' \
http://yhat.oak.vast.com/levy/models/WeightedPercentileModel/
Websockets


ŷhat ScienceOps
Model Upload
Docs
Logout
Home / WeightedPercentileModel
WeightedPercentileModel
Stage
Version
Language
Last Updated On
Status
Staging
1e89e2a
python
Fri, Oct 03 2014 18:51:24 UTC
online
Production
1e89e2a
python
Fri, Oct 03 2014 18:51:24 UTC
online
Versions
Scoring
Environment Variables
Logs
Settings
Consuming Your Model
Input / Output
Batch Scoring
Example Input
{
"data": [
{
"weight": 1,
"value": 100
},
{
"weight": 10,
"value": 50
},
Example Output
{
"yhat_model": "WeightedPercentileModel",
"yhat_id": "c14b5b73-ffa1-4b1e-b6cd-bab82e0cac84",
"result": {
"25": 50,
"50": 50,
"75": 50,
"100": 100,
"00": 0
}



Deployed model exists as a service, could be integrated into apps

Our engineers wrap it in another service layer so they can control authentication, logging, ...


iPad
5:19 PM
50%
Map
1727 COLQUITT, HOUSTON, TX 77098
Bookmark
RARE
Sunroom
Few houses like this one have this amenity.
RARE
Playground
Few houses like this one have this amenity.
Comparisons
Comparing to:
Similar Homes
3 BEDROOMS
Average
This residential has a typical number of bedrooms as compared to similar properties.
18% HIGHER
Above Average Price/Sqft
This residential's price/sqft is 18% higher than similar properties.
AVERAGE
Average Lot Size
This residential's lot has a typical size.
9% SMALLER
Below Average Sqft
This residential's square footage is 9% smaller than similar properties.
10% MORE EXPENSIVE
Above Average Price
This residential is 10% more expensive than similar properties.
3 BATHROOMS
Average
This residential has a typical number of bathrooms as compared to similar properties.

Deployment Recommendations

New Projects: yhat from the start

Existing Projects: Tempting to continue using PMML or Rewrite

- Want to deploy from python / R as soon as you want to do something new
- But then someone has to support both

Better to deploy directly from training code

Deployment Recommendations

Always name your models. Allow multiple models that perform the same function to coexist.

- Sometimes want different customers on different instances of the model
  - trained on public data only
  - trained on public data + proprietary from customer X
- Allows competition

Outline for this talk

1. Can we deploy one model into a production environment?
2. Given two models that perform similar functions can we evaluate which is better?

Three types of evaluation

Depending on the nature of the model use

1. Direct evaluation against ground truth
2. Indirect evaluation against business metrics
3. Human judgment

Evaluating price predictions against ground truth

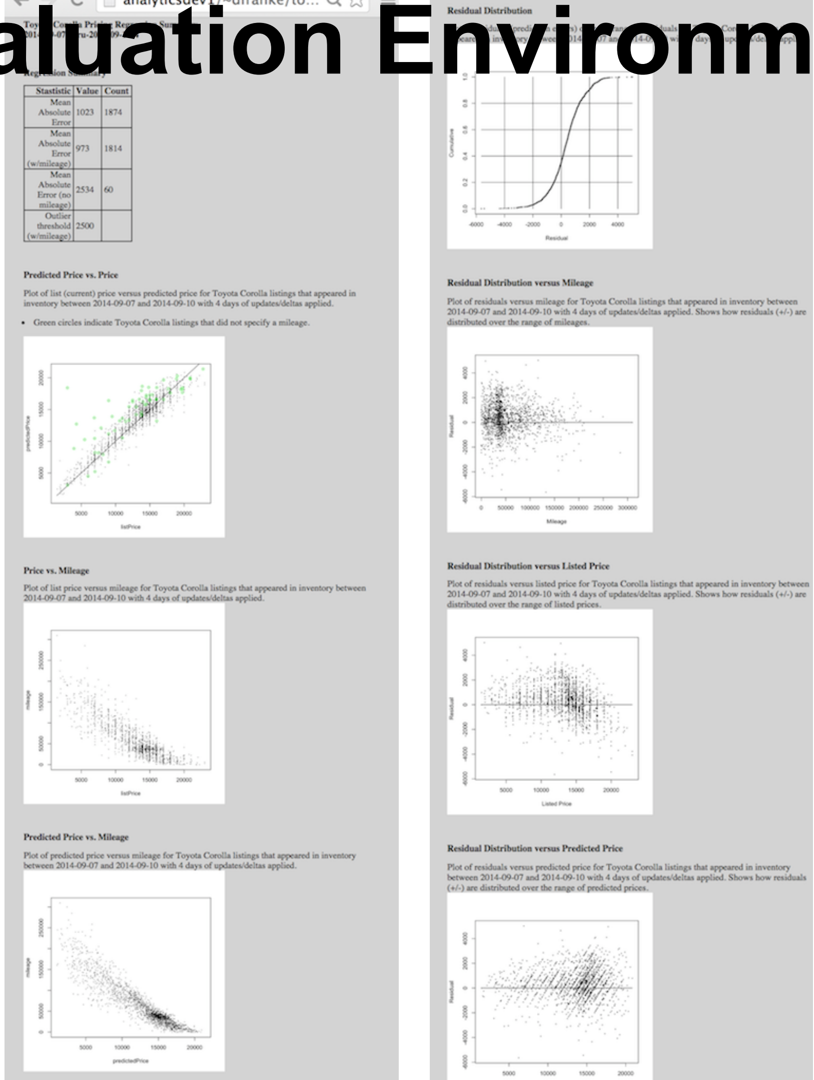
Vehicle listings come off the feed with a price.

Given a choice between two models, choose the one that minimizes mean absolute error.

Direct Evaluation Environment

Summary of global measures

Drill-down graphs



Residual graphs

- CDF
- Drill-down

Outliers (w/mileage)

Date/VIN	Prediction Error
2014-09-08-4-1NXBR32E93Z103371	-5639
2014-09-07-4-2T1BU4EE9BC551738	-5438
2014-09-08-4-2T1BR32E84C260490	5021
2014-09-07-4-5YFBU4EE0DP119963	4958
2014-09-07-4-2T1BU4EE5AC354421	4644
2014-09-09-4-2T1BU4EE0CC822669	-4343
2014-09-09-4-5YFBU4EE0DP090142	-4224
2014-09-09-4-5YFBU4EE6DP157066	4111
2014-09-10-4-5YFBURHE3EP033914	4095
2014-09-09-4-2T1BURHE0EC114305	-4070
2014-09-09-4-5YFBU4EE5DP159455	4034
2014-09-07-4-1NXBR32E88Z021820	-3899
2014-09-10-4-2T1BR32E66C627776	3866
2014-09-09-4-2T1BURHE2EC133602	-3772
2014-09-10-4-5YFBURHE6EP010613	-3689
2014-09-10-4-JTDBR32E060065495	3589
2014-09-09-4-5YFBURHE4EP016152	-3506
2014-09-08-4-2T1BU4EE7BC642622	3449
2014-09-07-4-5YFBU4EE4DP121621	3430
2014-09-09-4-1NXBU4EEXAZ199444	3360
2014-09-09-4-5YFBU4EE7DP207893	3330
2014-09-09-4-5YFBURHE0EP056678	-3324
2014-09-09-4-5YFBURHE9EP013005	-3324
2014-09-08-4-2T1BU4EE8BC741286	3281
2014-09-07-4-5YFBU4EE9DP157546	3226
2014-09-07-4-5YFBU4EE3DP156506	3221
2014-09-10-4-2T1BURHE7EC208102	3211
2014-09-07-4-2T1BU4EE4BC604300	-3162
2014-09-08-4-2T1BURHE3EC150103	3126

Links to upstream data for outliers

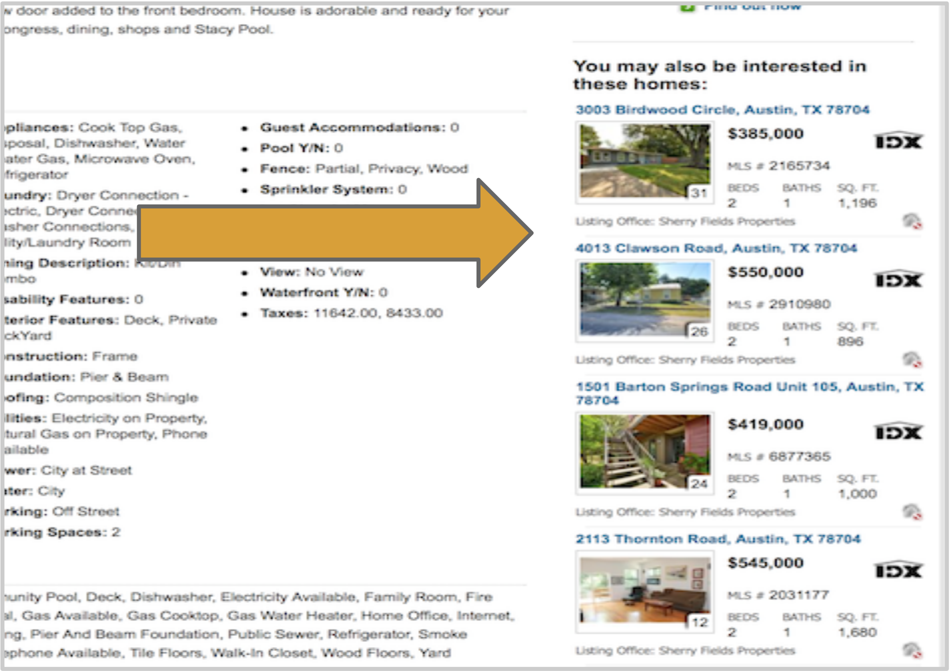
Allows manual investigation

- Is the model to blame for the outlier?
- Is the data garbage?

Indirect Evaluation

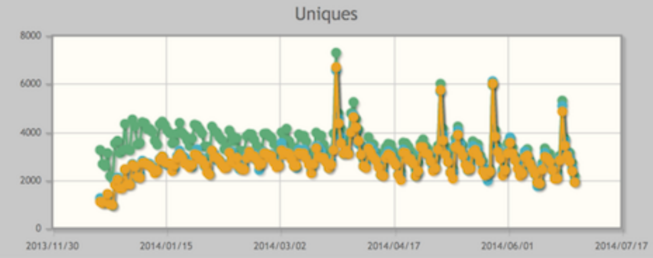
Recommendations on details page

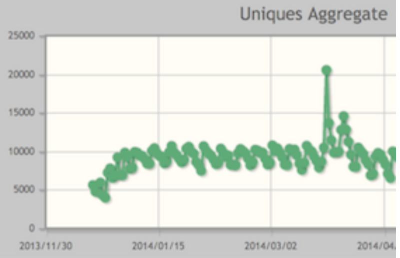
- Can't measure relevance or quality
- Conversion rate is important to business



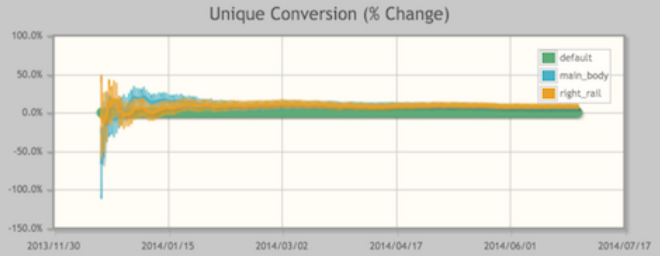
Test Summary	
Test:	recom_homes
Publisher:	
Start Date:	2013-12-20
Total Uniques:	1746503
% Users:	0%

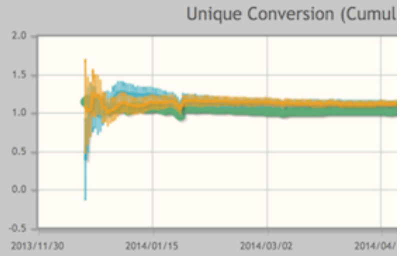
Uniques		
Variant	% Alloc	Uniques
default	0%	664093
main_body	0%	547809
right_rail	0%	534601



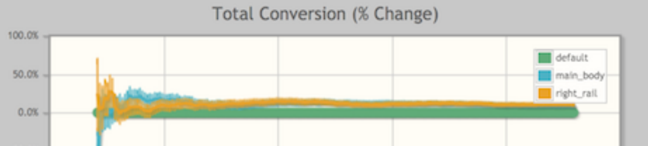


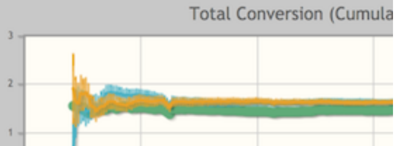
Unique Conversion			
Variant	Value	%-Change	P-Value
default	1.01%	-	-
main_body	1.08%	6.69%	.0001
right_rail	1.09%	7.99%	.0000





Total Conversion			
Variant	Value	%-Change	P-Value
default	1.44%	-	-

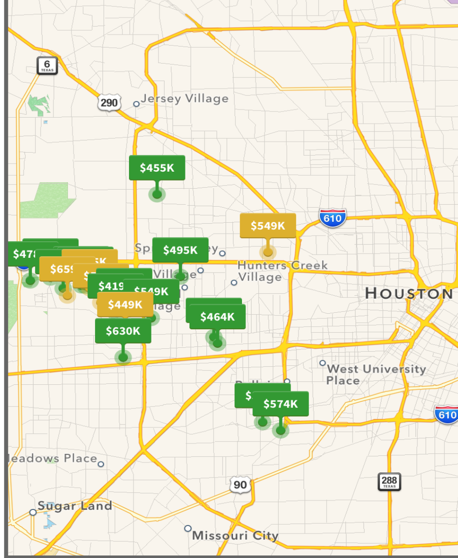


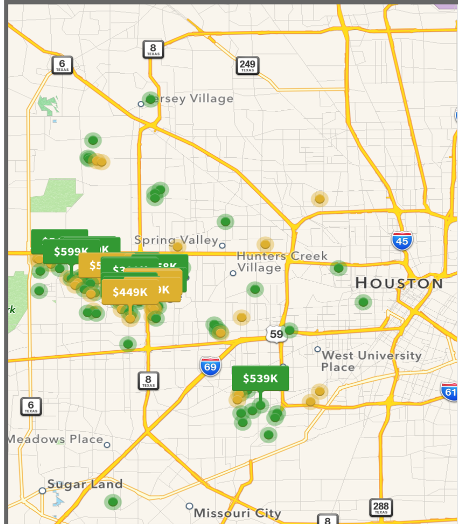


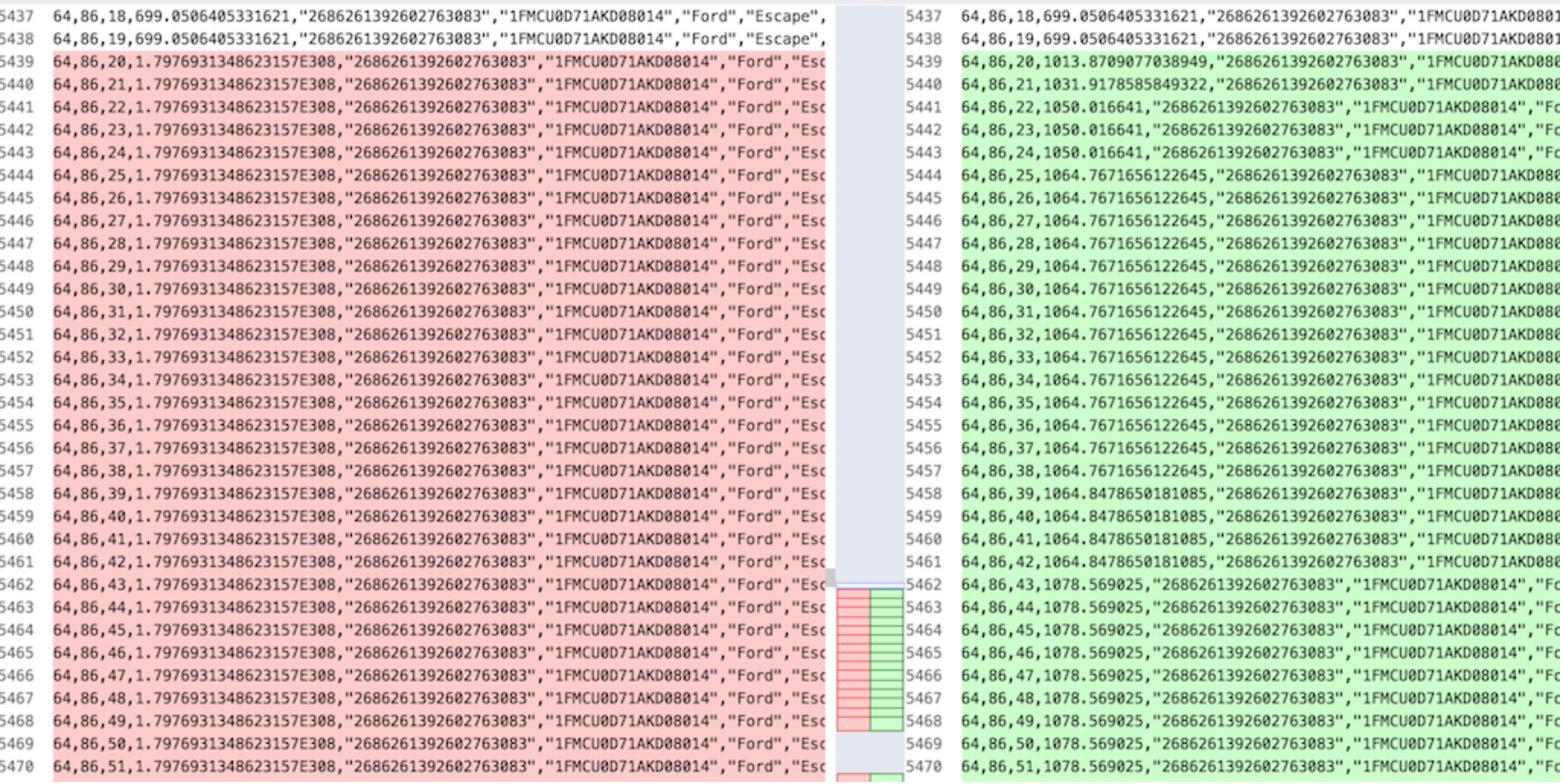
Human Judgment

Sort in early versions of HomeStory

- Can't measure "relevance"
- Cold start: Not enough traffic to optimize conversion rate yet
- Best we can do is "not embarrassing"
- Be good enough to attract traffic, set up future experiments







We use tools for bulk side-by-side comparisons.

Evaluation Recommendations

Do as much as you can

- Direct evaluation against truth
- Indirect evaluation against business metrics
- Human judgment

Goals

- Data driven decisions
- As much automation as possible

Plugs

Back end engineering team is hiring

Looking for Strong Junior / Senior Java/Scala person

Resumes to Olivier: omodica@vast.com