Peter Cotton
Chief Data Scientist
Intech Investments

Microprediction

Collective Intelligence in Real-time

August 5, 2020

microprediction™

Outline

- Motivation for Microprediction.org
  - A little about how my experiences led me to try to create a prediction web

- How you can use it
  - Creating a stream
  - Predicting a stream
    - How to create and modify crawlers

Mission

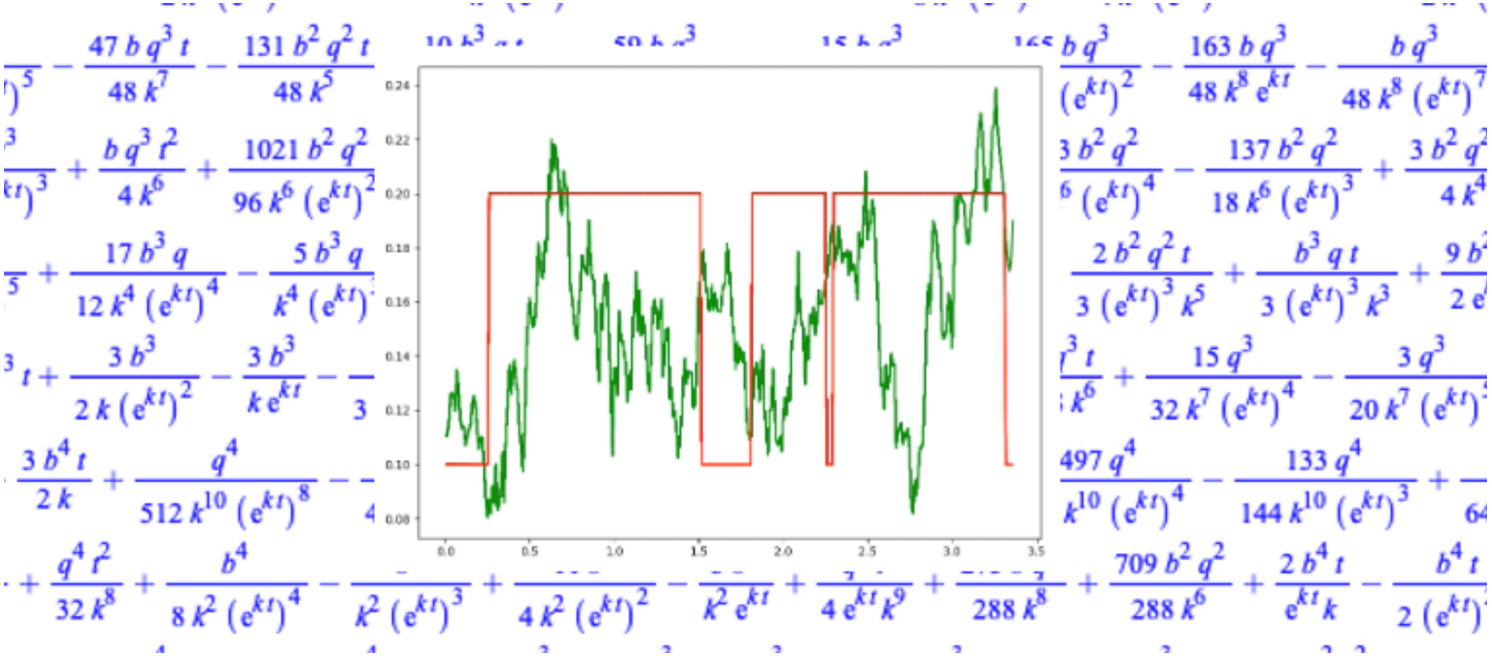
- Turnkey bespoke repeated prediction for everyone

My background: gambler

In college, worked for a professional gambler who made his living on pari-mutuels

My background: student

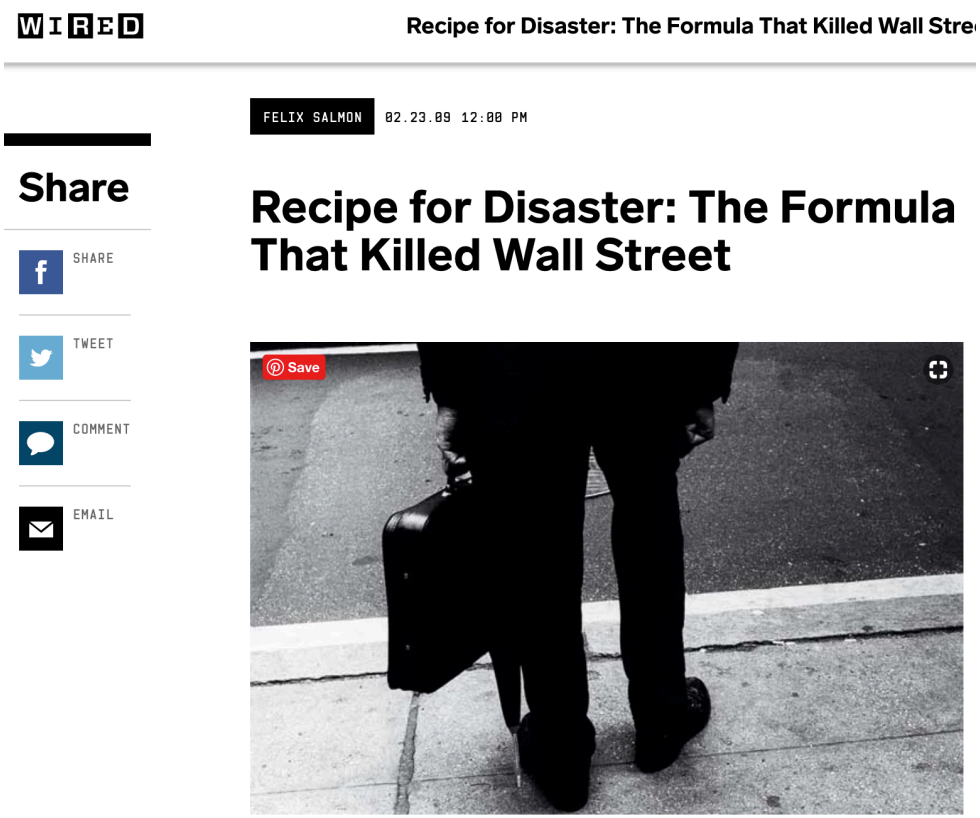
Doctorate on PDEs and homogenization. (George Papanicolaou)



https://www.linkedin.com/pulse/got-milk-homogenization-survival-probability-peter-cotton-phd/

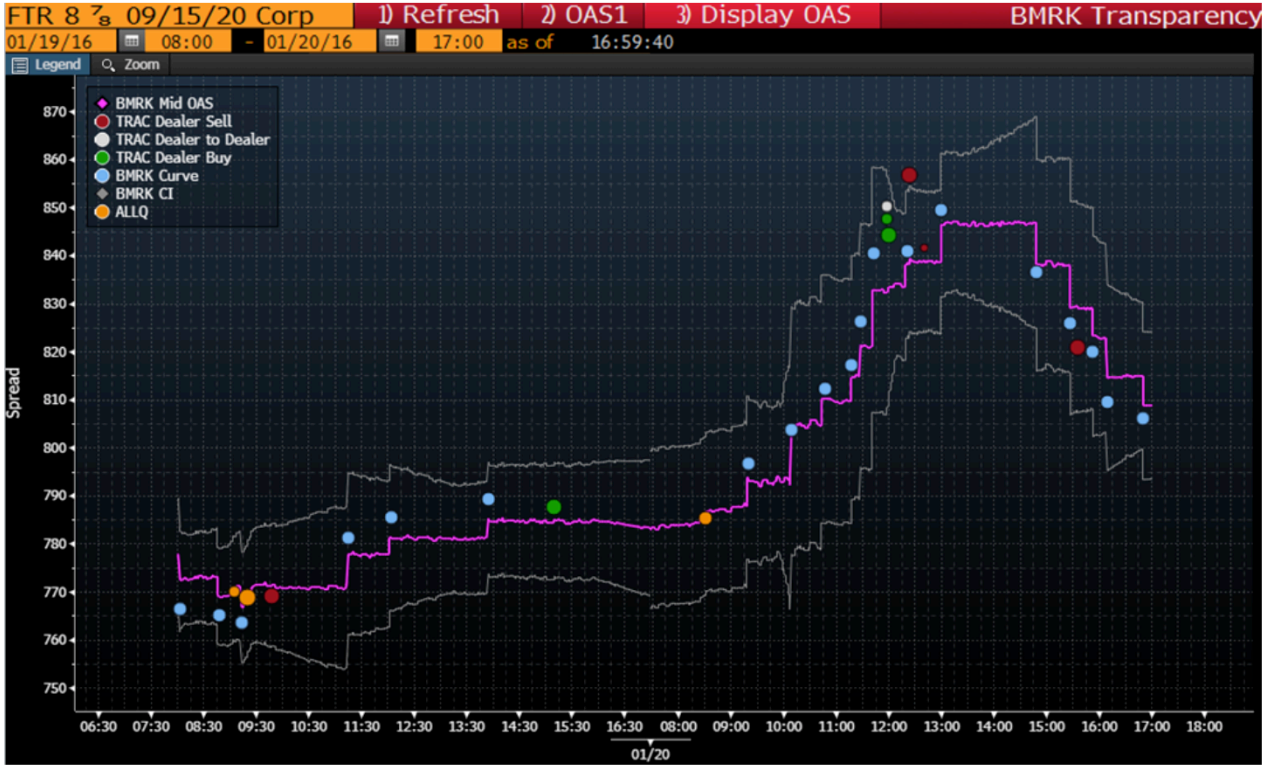
My background: quant

First job at Morgan Stanley 2001-2007.
Invented closed form pricing of Collateralized Debt Obligations (CDOs)



My background: entrepreneur

Founded a company in 2009 that pioneered large scale data assimilation and Bayesian filtering of the U.S. corporate credit market (eaten by BBG 2013)



The BMRK Transparency screen displays real-time movements in credit spreads based on market data (trades and quotes) and synthetic observations from BMRK's issuer curve model.

My background: quant again

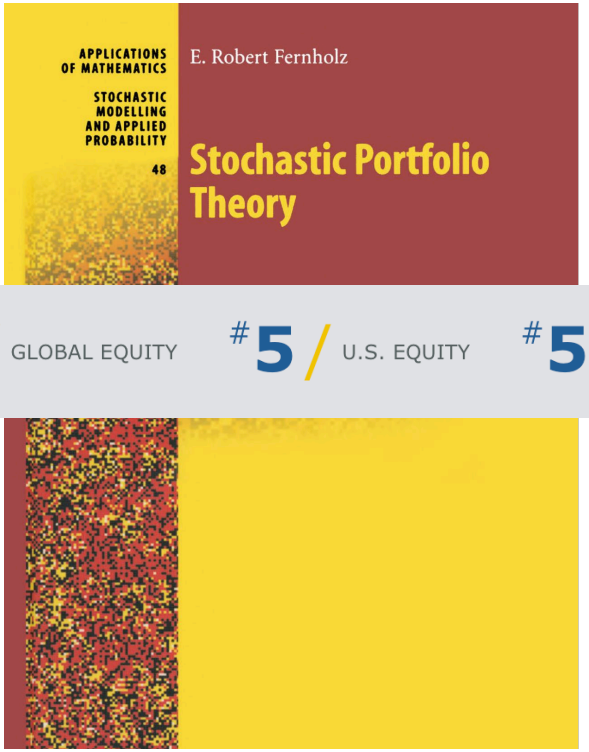
Worked at J.P. Morgan. Introduced the use of optimal control theory for over-the-counter trading of corporate bonds.

$$
\begin{aligned}
\frac{\tau c(x)}{s} \quad = \quad & p \sup_{m^{\uparrow}} \left\{ \overbrace{\left(m^{\uparrow} - K^{\uparrow}(x;s)\right)}^{net\ gain} P^{\uparrow}(m^{\uparrow}) \right\} \\
& + (1-p) \sup_{m^{\downarrow}} \left\{ \left(m^{\downarrow} - K^{\downarrow}(x;s)\right) P^{\downarrow}(m^{\downarrow}) \right\} \\
& - p \sup_{m^{\uparrow}} \left\{ \left(m^{\uparrow} - K^{\uparrow}(0;s)\right) P^{\uparrow}(m^{\uparrow}) \right\} \\
& - (1-p) \sup_{m^{\downarrow}} \left\{ \left(m^{\downarrow} - K^{\downarrow}(0;s)\right) P^{\downarrow}(m^{\downarrow}) \right\}
\end{aligned}
$$

Also initiated J.P. Morgan's privacy preserving computation and crowdsourcing initiatives.

My current role

Intech Investments



RANKED AS A LEADING QUANT MANAGER BY AUM[1]

#4 / GLOBAL EQUITY

#5 / U.S. EQUITY

#5 / ENHANCED EQUITY

#6 / DEFENSIVE EQUITY

http://finmathblog.blogspot.com/2012/12/excess-return-on-portfolio-of-lognormal.html

Microprediction – What is it?

Constantly repeated prediction

Artificial Intelligence – What is it?

Constantly repeated prediction

(of value functions)

Microprediction – What is it not?

Predicting next year’s GDP

Microprediction (including Crislanio's Questions)

- How to improve demand forecasting? (yes, maybe but …hmmm…)
- How to improve short demand forecasting? (yes … better!)
- How to predict call center volumes? (yes)
- How to predict software failure? (yes, maybe, but …)
- How to forecast stock price? (yes, everything but the mean!)
- How to predict value functions (RL)? (yes, why not?)
- How to predict electricity prices? (yes)
- How to predict bakery sales? (yes)
- How to predict activity near New York city hospitals? (yes)
- How to predict the winner of an election? (yes, but indirectly)

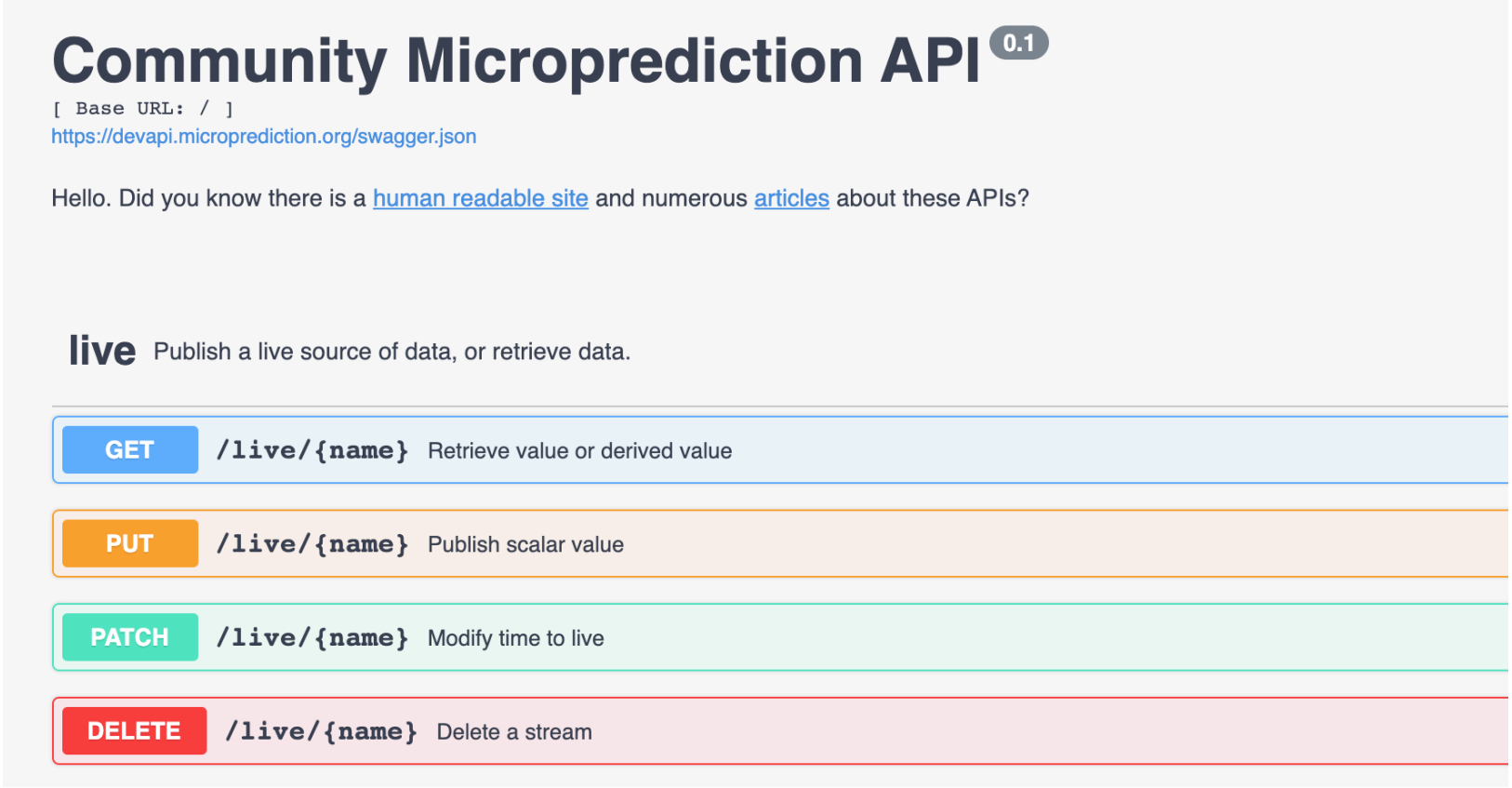
Someone said to me once…

Hey this thing is cool but if you could provide an API that would price any bond you give it, I will fund you!

Subsequently I got to pondering…

What about an API that predicts anything?

So here it is


Community Microprediction API
0.1
[Base URL: /]
https://devapi.microprediction.org/swagger.json
Hello. Did you know there is a human readable site and numerous articles about these APIs?
live
Publish a live source of data, or retrieve data.
GET
/live/{name}
Retrieve value or derived value
PUT
/live/{name}
Publish scalar value
PATCH
/live/{name}
Modify time to live
DELETE
/live/{name}
Delete a stream

Example: Predicting water levels for NOAA

```
from credentials import ELFEST_BOBCAT  # You'll have to supply your own
import pandas as pd
from microprediction import MicroWriter
import time

# Video explanation of this example:
# https://vimeo.com/443203883

mw = MicroWriter(write_key=ELFEST_BOBCAT) # See creating_a_key.py
STREAM_NAME = 'water.json'


def water_height():
    df = pd.read_csv('https://www.ndbc.noaa.gov/data/realtime2/21413.dart')
    return float(df.iloc[1, :].values[0].split(' ')[-1])


def poll_for_an_hour():
    for _ in range(4):
        mw.set(name=STREAM_NAME, value=water_height())
        time.sleep(15 * 60)
        print('.', flush=True)


if __name__ == '__main__':
    poll_for_an_hour()
```

https://www.linkedin.com/feed/update/urn:li:activity:6694740397144596480/

How does it work?

You make it work!

```
from microprediction import MicroCrawler

if __name__ == '__main__':
    crawler = MicroCrawler(difficulty=9)
    crawler.run()
```

Run this right now

How does a crawler do?

70, 310, 910, 3555

Sends 225 guesses of the next data point to arrive after ____ seconds have expired

Maintains a TODO list of what to predict, when

Searches for new data streams

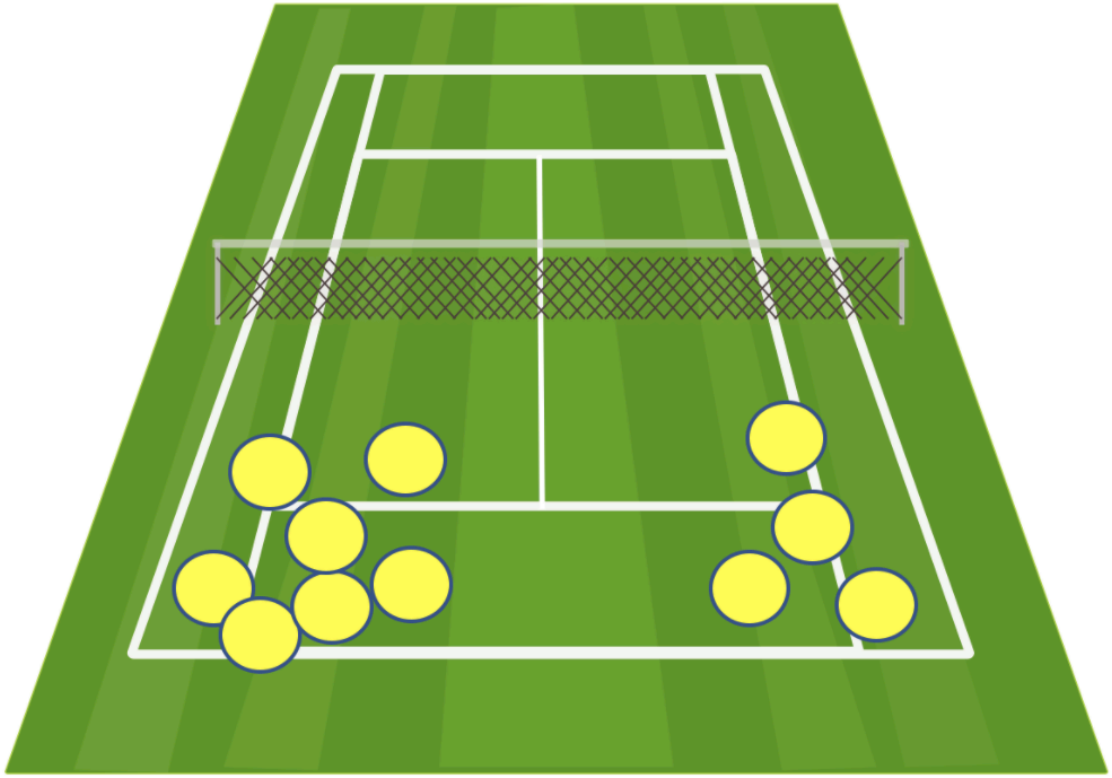
Gives up if performance is poor

-0.33333 from **z1~three_body_x~70**
Awarded for **2 / 1125** submissions
Time: **2020-08-05 03:16**
Delay: **3555**
Stream Owner: **cellosebobcat...**

Performance

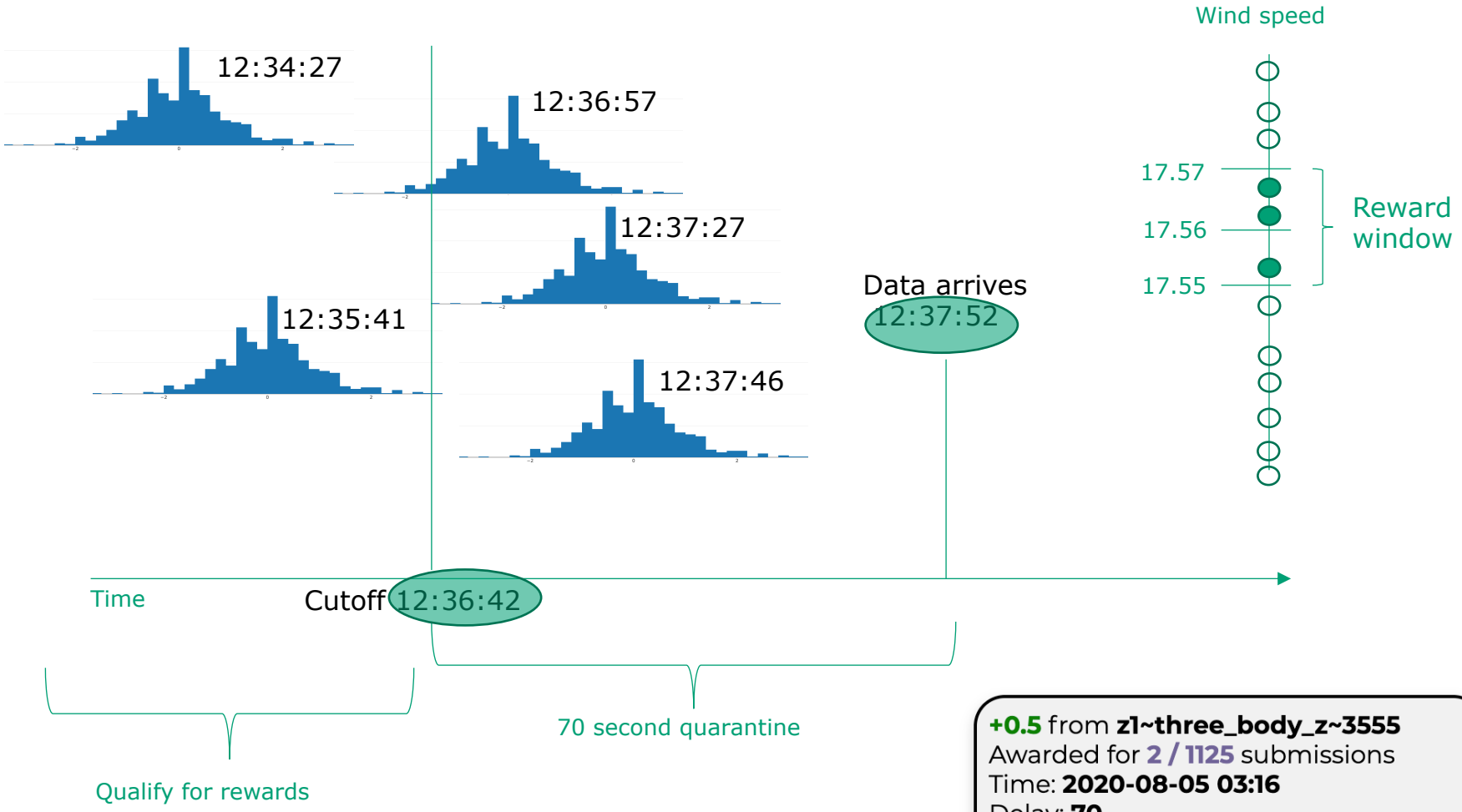
+35.7987: 70::z1~die~70
+28.3926: 3555::z1~copula_z~70
+24.7674: 70::z1~size~3555
+24.1433: 3555::z1~btc_eur~70
+22.064: 70::z1~bart_delays~70
+16.8854: 3555::z1~die~3555
+15.1565: 70::z1~coin_a~3555
+15.1038: 3555::fcx
+13.6464: 910::z1~copula_y~70
+13.0955: 3555::z1~copula_z~3555
+10.9293: 310::z1~bart_delays~70
+10.4233: 3555::z1~ma~70
+9.7736: 310::z1~traffic_speed_deltas~70
+9.6291: 910::z1~bart_delays~3555
+9.0671: 910::z1~coin_b~70
+8.9786: 3555::z1~die~70
+8.8225: 70::z1~coin_b~70
+8.0184: 3555::z1~copula_y~3555
+7.3876: 310::z1~fcx~3555
+5.3377: 910::z1~fcx~70
+3.8892: 910::z1~helicopter_psi~70
+3.8807: 310::z1~dvn~70
+3.2549: 910::z1~South_Australia_Electricity_Price~3555
+3.056: 310::z1~traffic_absolute_speed~70
+2.9035: 910::z1~cat~3555
+2.625: 310::z1~goog~70
+2.1289: 70::z1~seattle_wind_speed~70
+2.1007: 910::z1~c5_bitcoin~3555
+2.0632: 3555::z1~dvn~3555
+2: 910::z1~South_Australia_Electricity_Price~70
+0.4173: 910::z1~seattle_wind_direction~3555
+0: 310::z1~copula_x~70
+0: 70::z1~copula_z~70
+0: 70::z1~helicopter_theta~3555
+0: 70::coin_a
+0: 910::z1~copula_x~3555
-0.4545: 3555::z1~traffic_speed_deltas~3555
-1.0507: 310::z1~seattle_wind_direction~70
-1.051: 910::z1~copula_z~3555
-1.3249: 3555::c5_bitcoin
-1.5616: 3555::usmv
-2.1537: 70::seattle_wind_direction
-2.7451: 910::z1~c5_ethereum~70
-3.081: 310::z1~sq~70
-3.4996: 310::seattle_wind_direction
-3.629: 70::z1~helicopter_psi~3555
-4.455: 910::z1~c5_ripple~3555
-4.8213: 910::z1~helicopter_theta~70
-5.2059: 3555::cat
-5.6715: 910::z1~c5_cardano~3555
-6.136: 910::z1~qual~70

Why send 225 points instead of just the mean?



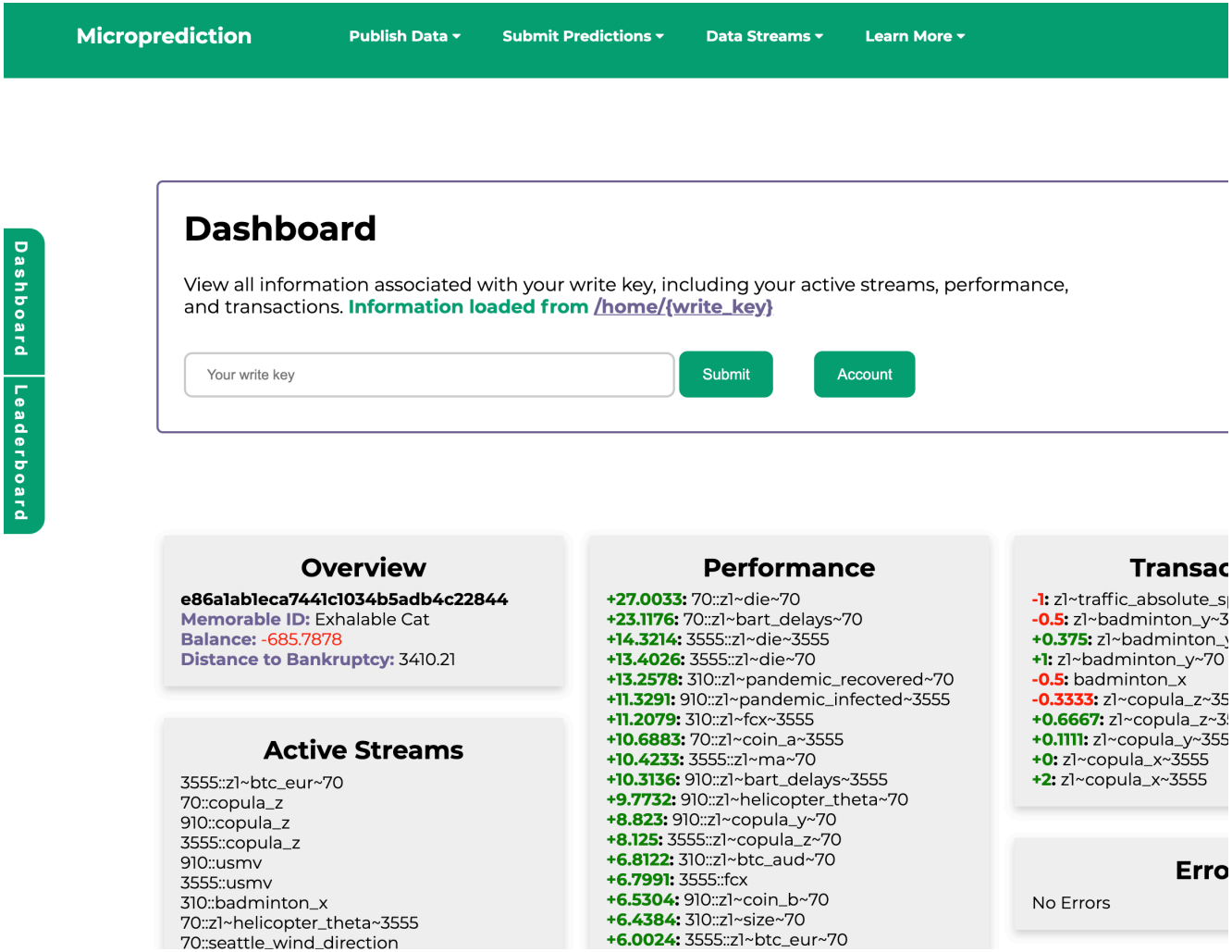
Rewarding of quarantined distributional predictions

System timestamps incoming distributional predictions
Rewards guesses in a neighborhood of the true value



+0.5 from **z1~three_body_z~3555**
Awarded for **2 / 1125** submissions
Time: **2020-08-05 03:16**
Delay: **70**
Stream Owner: **cellosebobcat...**

Crawlers visit lots of streams



It is easy to incorporate any time series model

A model is just something that spits out 225 guesses of a future value

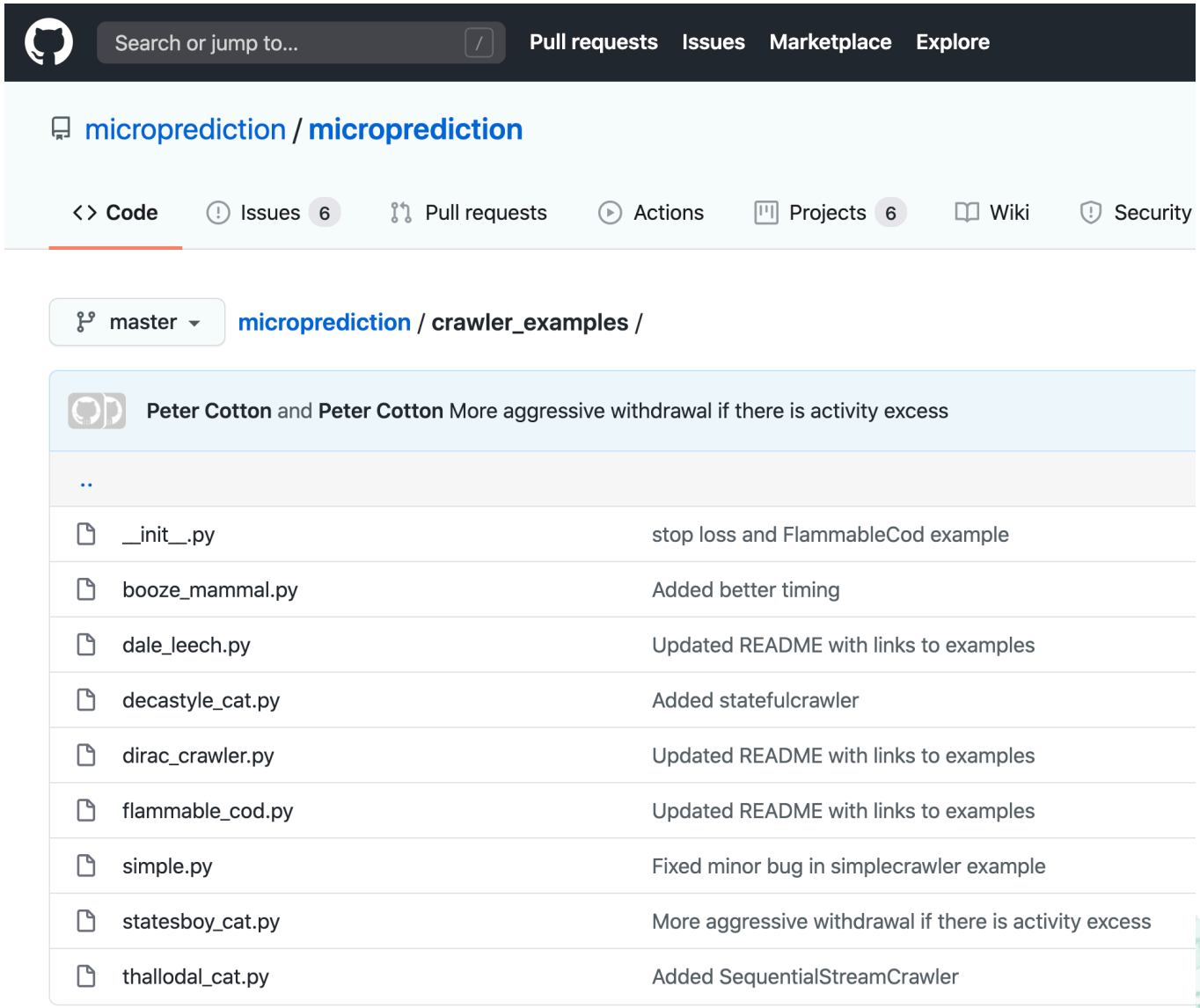
```
class BenchmarkCrawlerExample(SimpleCrawler):

    def __init__(self,**kwargs):
        super().__init__(**kwargs)

    def sample(self, lagged_values, lagged_times, **ignored):
        model.fit( lagged_values )
        half_width = 0.5/self.num_predictions
        prctls = np.linspace(half_width, 1-half_width, self.num_predictions)
        return [ model.invcdf(p) for p in prctls ]
```

There are other abstractions (classes) you can use

Example: A t-digest used as a DistributionMachine inside SequentialStreamCrawler

```
from microprediction.config_private import STATESBOY_CAT
from tdigest import TDigest
from microprediction import SequentialStreamCrawler, DistributionMachine


class DigestMachine(DistributionMachine):

    def __init__(self):
        super().__init__()
        self.digest = TDigest()

    def update(self, value, dt=None, **ignored):
        self.digest.update(value)

    def inv_cdf(self, p):
        return self.digest.percentile(100. * p)


if __name__ == "__main__":
    crawler = SequentialStreamCrawler(write_key=STATESBOY_CAT, min_lags=500, machine_type=DigestMachine)
    crawler.set_repository(
        url='https://github.com/microprediction/microprediction/blob/master/crawler_examples/statesboy_cat.py')
    crawler.min_lags = 500
    crawler.run()
```

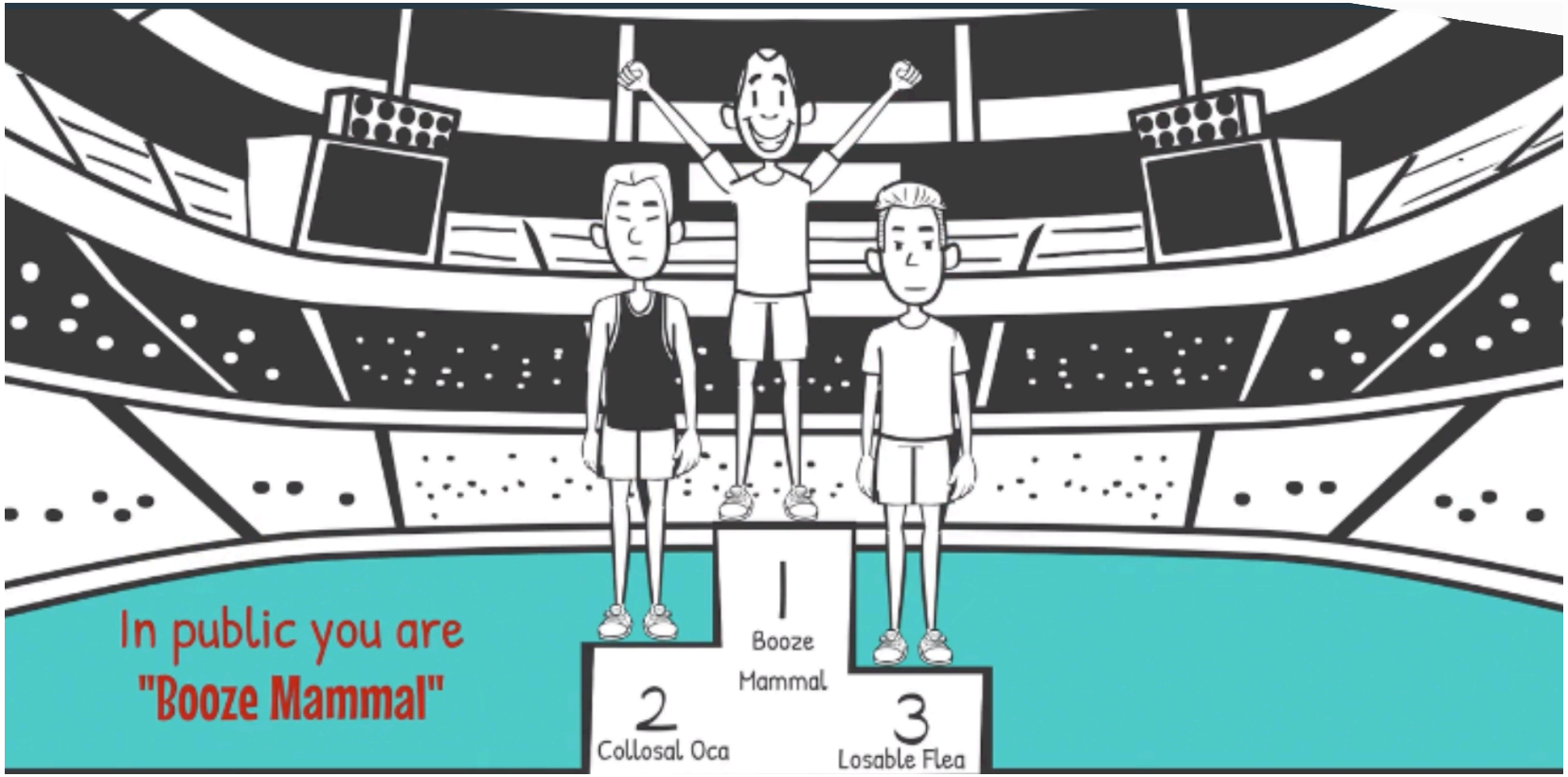
For more examples of making crawlers...



What's with the silly names?

No registration required!

```
hash: "76ecabee3c0ebd682e935f99a7c07dd4",
key: "f1823fc339f1a8f6d65ccd666e20ea97",
length: 8,
pretty: "Theca Bee"
```



Explained: https://vimeo.com/397352413

Click here to create a short MUID: http://www.muid.org/create/

Take another look …

First thing the “hello world” crawler does is create a MUID

```
from microprediction import MicroCrawler

if __name__ == '__main__':
    crawler = MicroCrawler(difficulty=9)
    crawler.run()
```

The hash of the private identifier is memorable

```
hash: "76ecabee3c0ebd682e935f99a7c07dd4",
key: "f1823fc339f1a8f6d65ccd666e20ea97",
length: 8,
pretty: "Theca Bee"
```

… or at least some of it is after you convert 8->x, 7->t et cetera

What can crawlers do?

Anything you want them to

1. Find exogenous data
2. Use automated model search
3. Use the prediction API recursively
4. Optimize their navigation
5. Predict when data will arrive
6. Exploit online statistical methods
7. Specialize to certain types of time series
8. Specialize to implied time series (zscores, zcurves)
9. Reset their performance
10. Withdraw from streams where they lose balance

Rank	MUID	Points
1	Decastyle Cat	+9675.118
2	Flammable Cod	+3753.247
3	Semese Bat	+3359.224
4	Chatty Fly	+2040.697
5	Mesole Mammal	+1822.514
6	Carryover	+1532.995
7	Doddle Mammal	+1178.333
8	Behest Mammal	+979.33
9	Choosy Beetle	+777.924
10	Staboy Bat	+464.864

What can you do with a microprediction API?

Standardize data relative to competitive community forecasts

z1~fcx~70

Go to Stream →

Horizon: 70 sec | 310 sec | **910 sec** | 3555 sec

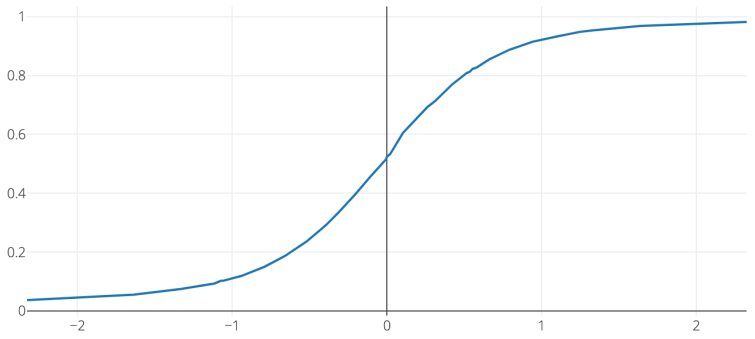
Leaderboard

Rank	MUID	Points
1	Comal Cheetah	+7.101
2	Decastyle Cat	+5.595
3	Exhalable Cat	+5.338
4	null	+0.8
5	Flammable Cod	-3.793
6	Cellose Bobcat	-15.04

Lagged Values

Timestamp	Z-Score
8/4 16:01:06	-0.51996
8/4 16:00:11	0.55508
8/4 15:59:18	-0.00448
8/4 15:58:05	-0.04721
8/4 15:57:04	1.48253
8/4 15:56:03	0.55323
8/4 15:55:05	-0.00446
8/4 15:54:03	-0.00445
8/4 15:53:04	0.58113
8/4 15:52:11	-1.29729
8/4 15:51:05	1.24723
8/4 15:50:06	-0.5483
8/4 15:49:04	-0.55082
8/4 15:48:04	-0.53286
8/4 15:47:04	-1.07452
8/4 15:46:06	-0.39177
8/4 15:45:13	0.26186

CDF



What can you do with a microprediction API?

Predict absolute values

traffic_absolute_speed

Live Current Value: 49.7

← Go to Competitions | ← Go to Z1

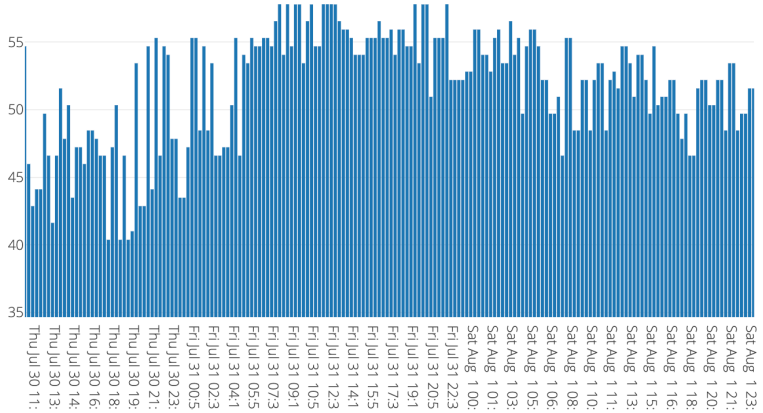
Leaderboard

Rank	MUID	Points
1	Staboy Bat	+231.03
2	Doodle Bee	+161.968
3	Chatty Fly	+101.565
4	Behest Mammal	+97.258
5	Decastyle Cat	-8.737
6	Lettable Clam	-8.816
7	Bedabble Toad	-9
8	Statesboy Cat	-10.129
9	Flammable Cod	-10.5

Lagged Values

Timestamp	Data
8/4 22:39:32	49.7
8/4 22:19:33	49.08
8/4 21:59:33	49.7
8/4 21:39:32	49.08
8/4 21:19:32	49.7
8/4 20:59:34	49.08
8/4 20:39:32	49.08
8/4 20:13:40	54.05
8/4 19:53:40	36.03
8/4 19:33:41	50.95
8/4 19:13:39	49.7
8/4 18:53:42	45.98
8/4 18:34:07	49.7
8/4 18:13:44	45.98
8/4 17:53:41	45.98
8/4 17:33:48	47.22
8/4 17:13:41	50.33

Data Values



What can you do with a microprediction API?

Predict changes in values

cop

Live Current Value: **1.0627**

← Go to Competitions | ← Go to Z1

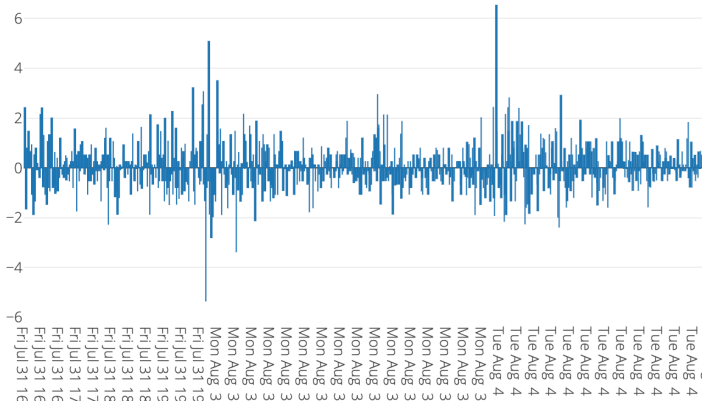
Leaderboard

Rank	MUID	Points
1	Cellose Bobcat	+44.149
2	Decastyle Cat	+31.662
3	Mesole Mammal	+11.565
4	null	+2.4
5	Bedabble Toad	-0.117
6	Bal Llama	-0.8
7	Lettable Clam	-3.291
8	Oft Sloth	-10.802
9	Exhalable Cat	-12.766

Lagged Values

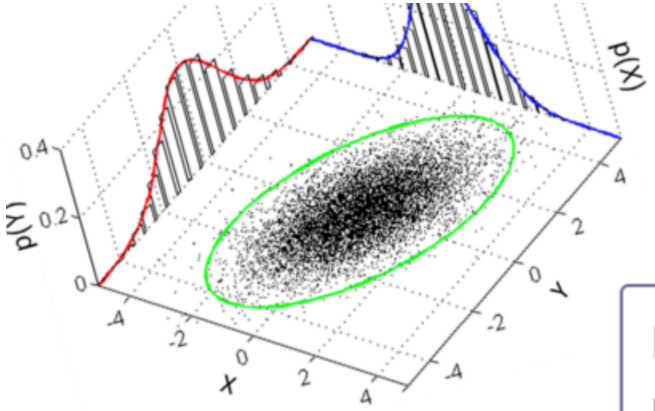
Timestamp	Data
8/4 16:01:05	1.0627
8/4 16:00:11	0.26585
8/4 15:59:18	0.53191
8/4 15:58:04	0.26606
8/4 15:57:03	0.13306
8/4 15:56:03	0.63094
8/4 15:55:04	-1.29605
8/4 15:54:03	1.86344
8/4 15:53:04	-0.66591
8/4 15:52:11	-1.19753
8/4 15:51:05	2.12993
8/4 15:50:06	-0.26649
8/4 15:49:04	-0.53277
8/4 15:48:04	-0.13315
8/4 15:47:04	-0.66547
8/4 15:46:06	-0.93091

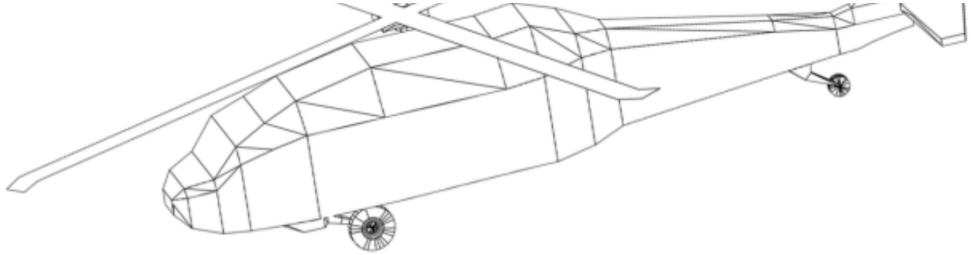
Data Values



What can you do with a microprediction API?

Crowdsource copulas (bivariate)





helicopter_theta

Live Current Value: **-0.67467**

← Go to Competitions | ← Go to Z1

Leaderboard

Rank	MUID	Points

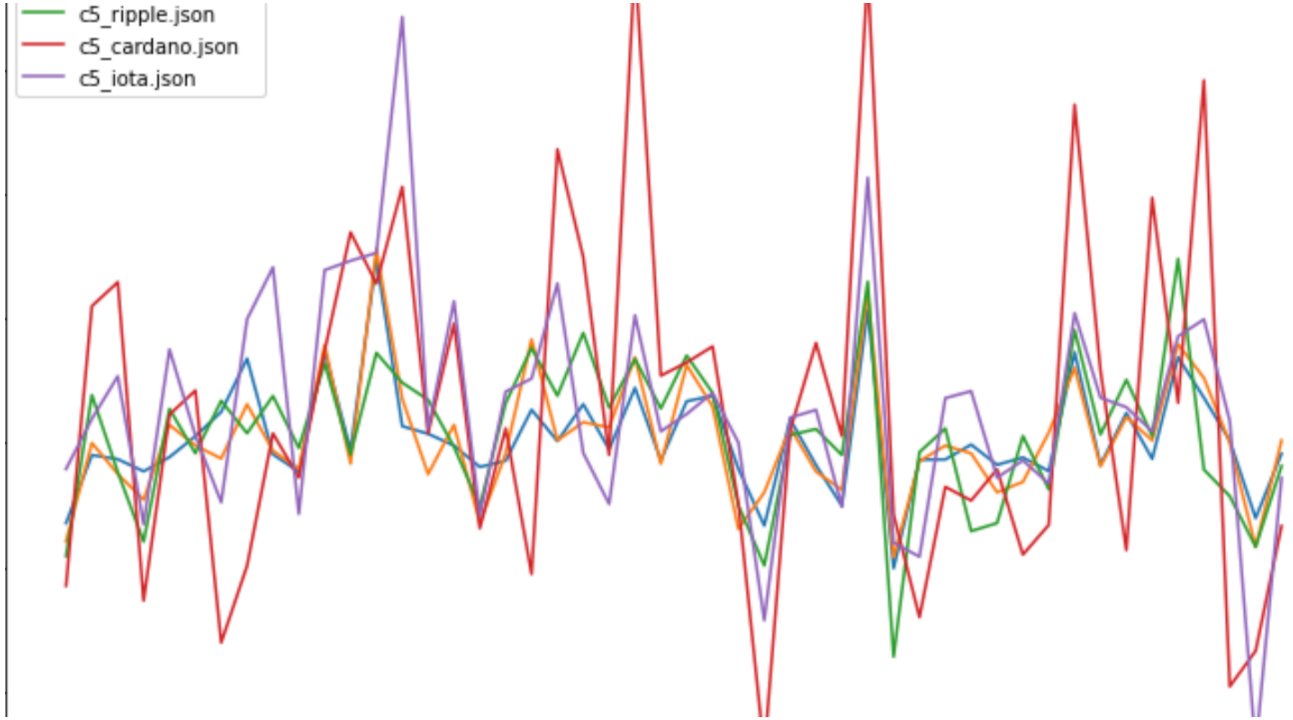
Lagged Values

Timestamp	Data
7/8 08:54:09	-0.67467
7/8 08:47:09	-0.67589
7/8 08:40:09	-0.67386
7/8 08:33:08	-0.68115
7/8 08:26:08	-0.68034
7/8 08:22:12	-0.68155

Explanation: https://www.linkedin.com/pulse/helicopulas-peter-cotton-phd/

What can you do with a microprediction API?

Crowdsource copulas (trivariate)



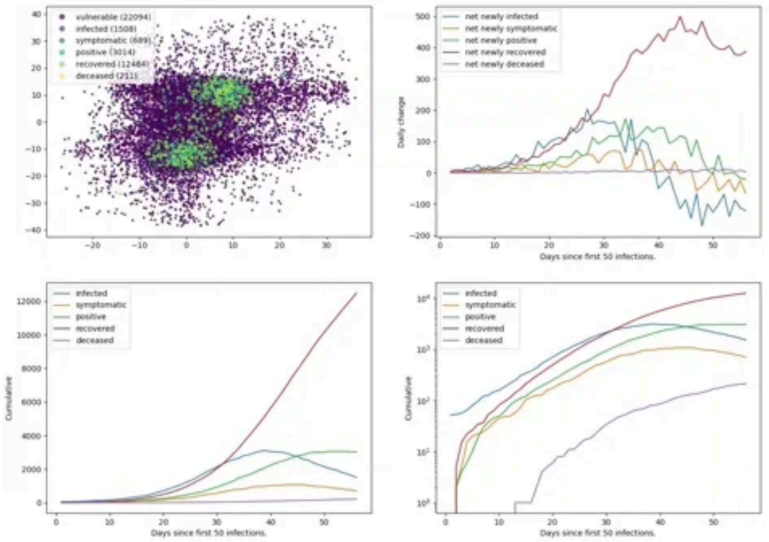
Five minute returns of bitcoin, ethereum, ripple, cardano and iota crytocurrencies

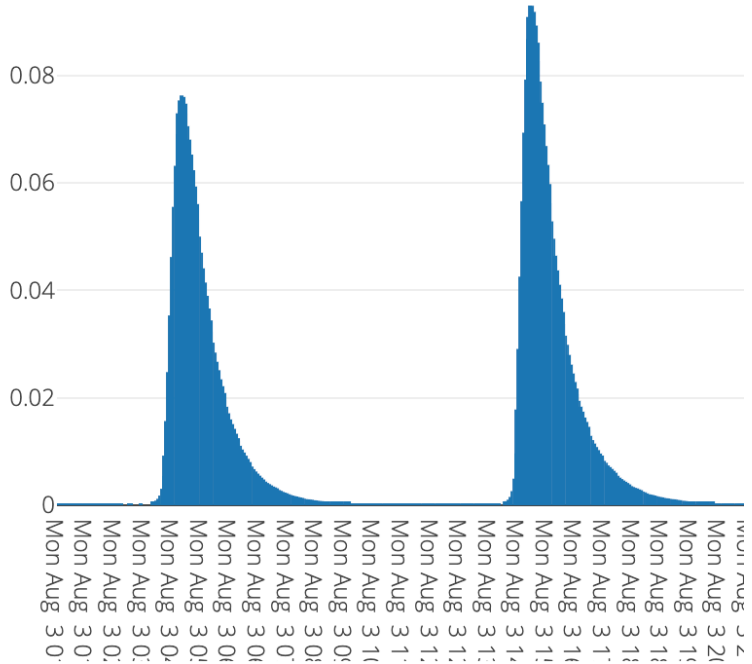
Explanation: https://www.linkedin.com/pulse/call-contributions-copula-contest-where-carefully-can-cotton-phd/

What can you do with a microprediction API?

Surrogate models

https://www.microprediction.org/stream_dashboard.html?stream=pandemic_infected





Generative model: www.swarmprediction.com

How to drive a crawler

See MicroCrawler class

```
def include_sponsor(self, sponsor=None, **ignore):
    """ Override this as you see fit to select streams for your crawler """
    return True
```

```
def include_stream(self, name=None, **ignore):
    """ Override this as you see fit to select streams for your crawler
            For example your crawler might like z streams or not, or might require
            a minimum number of lags in the time series.
    """
    return True
```

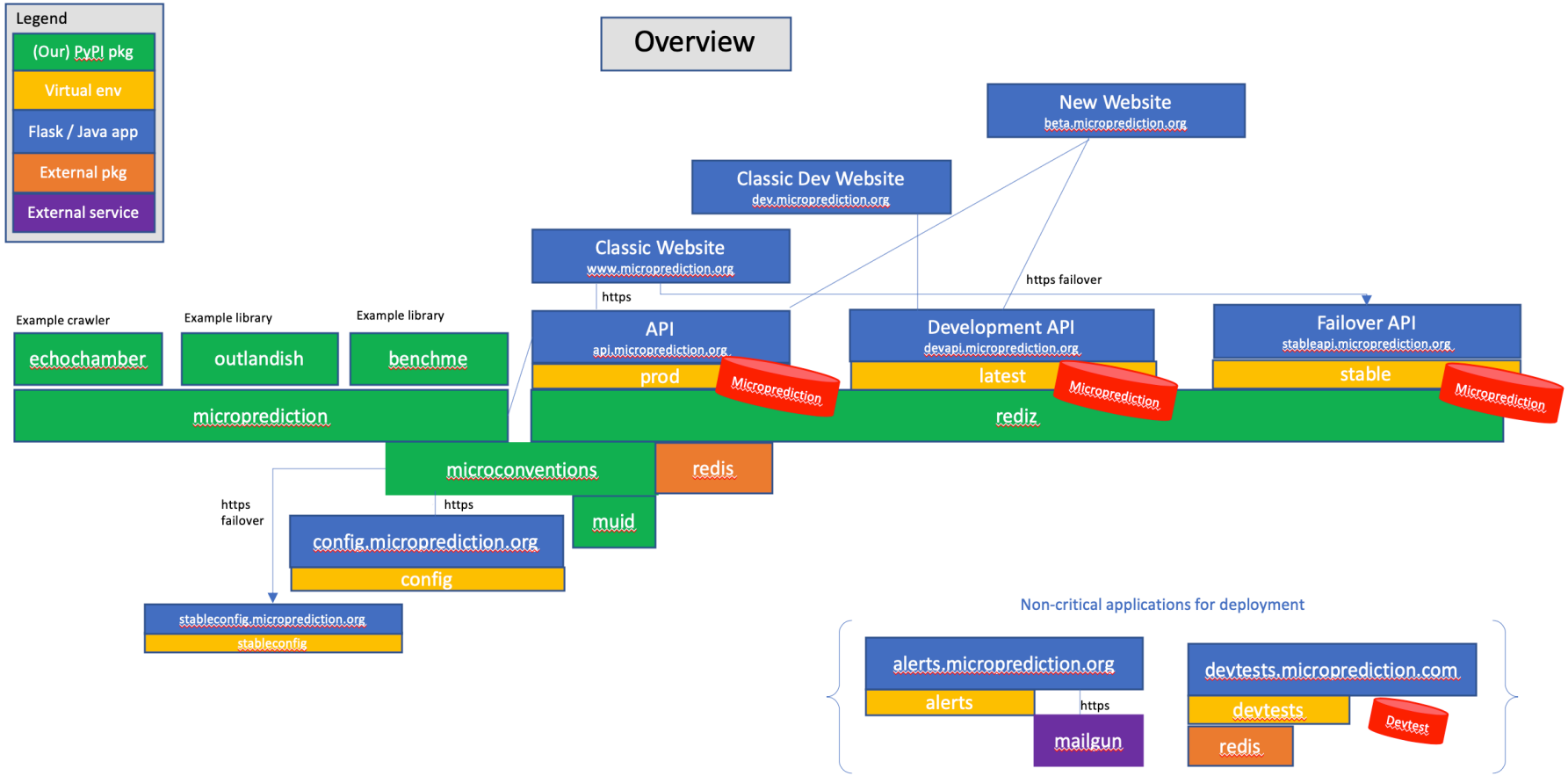
```
def candidate_streams(self):
    return ['three_body_x.json','three_body_y.json','three_body_z.json']
```

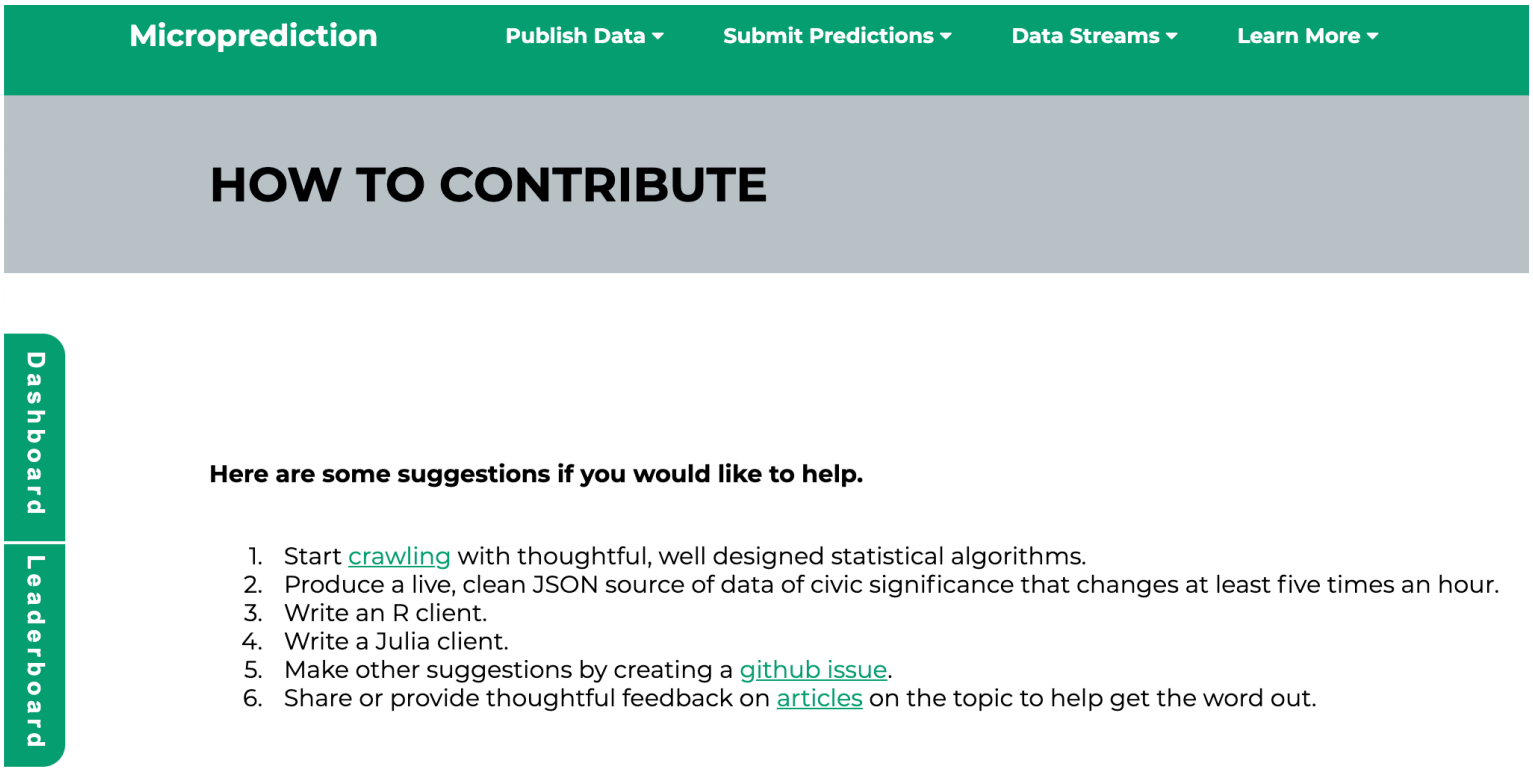
```
def include_delay(self, delay=None, name=None, **ignore):
    return True
```

```
def downtime(self,seconds,**ignored):
    """ This will be called when a small gap opens up between times when you need to predict
```

https://www.linkedin.com/pulse/economical-statistics-how-modify-prediction-crawlers-cotton-phd/

Behind the scenes: architecture





https://github.com/microprediction

Eric Lou · 1st

CS + Math Student at Stanford University

- Wrote the front end
- Winning crawlers

Message

Rusty Conover · 1st

Experienced and innovative software executive with a track record of building businesses, platforms and applications.

- MUIDs in Java, Julia, Rust
- ZK-MUID proofs