

# Django/Python/Machine Learning Web Application (Presentation #2)

Chris Winsor

5/27/2020

Prepared for Metrowest Boston Developers Machine Learning Group

Available from [https://github.com/cwinsor/django\\_103\\_plasticc\\_and\\_ux](https://github.com/cwinsor/django_103_plasticc_and_ux)

# Where we are...

## Last Time:

- Django Basic Web Application (Tic-Tac-Toe)

[https://github.com/cwinsor/django\\_102\\_pluralsight/blob/master/django\\_web\\_app\\_framework\\_intro.pdf](https://github.com/cwinsor/django_102_pluralsight/blob/master/django_web_app_framework_intro.pdf)

<https://app.pluralsight.com/library/courses/django-fundamentals-update/table-of-contents>

## This time:

- Django Web Application with Machine Learning model, Kaggle dataset, Postgres, Google Charts

[https://github.com/cwinsor/django\\_103\\_plasticc\\_and\\_ux/blob/master/presentation.pptx](https://github.com/cwinsor/django_103_plasticc_and_ux/blob/master/presentation.pptx)

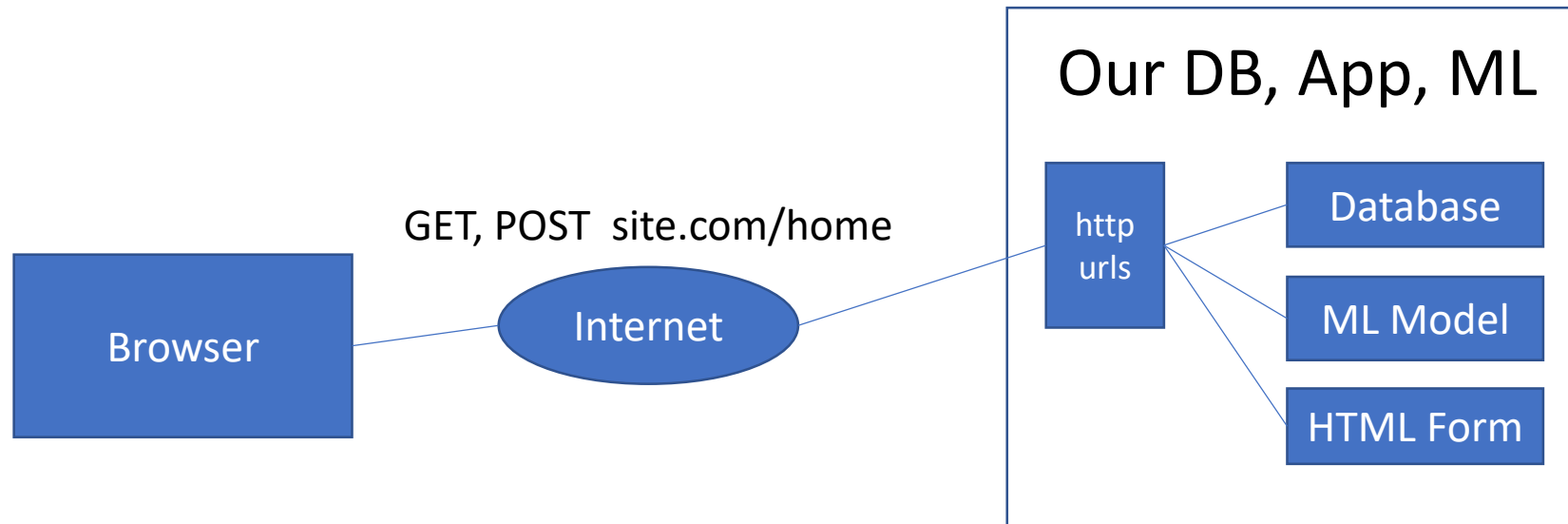
# Goals:

- Basic Web App Design with Server-side Python (Django)
- Integrating ML into Web Application
- Deploy to cloud
- A bit of client-side (Google Charts, Bootstrap)
- Why:
  - A means to express your work (anyone with a browser) vs  
Jupyter Notebook, .ppt/.doc, Vimeo, Kaggle
  - Practice, Practice, Practice

# Django Framework

## Game Application

## Server-side Python

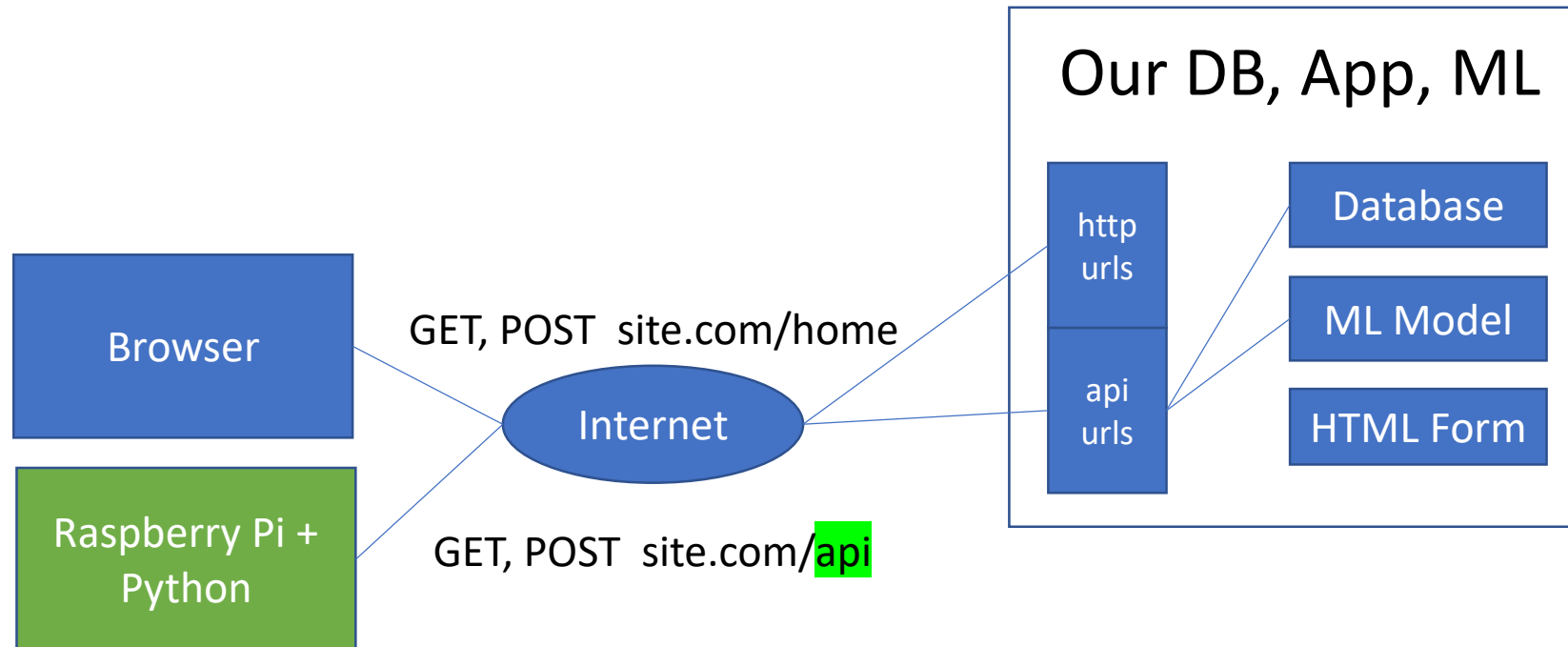


# Where we can take this...

- Today
  - ML as part of web application
- Tomorrow: Internet-of-Things
  - Front-end is Raspberry Pi running Python
  - POST data to back-end
  - GET classifications/recommendations from backend database, ML model
- Day 3: Access OTHER web APIs
  - Google Maps
  - Geolocation
  - NASA
  - Chuck Norris Jokes
  - Speech-to-Text
- Day 4: Offer our OWN api to subscribers
  - see Gene's presentation <LINK?> <https://github.com/MetrowestBostonDevelopersMLGroup/MeetingPresentations>
  - and article <https://a16z.com/2020/02/16/the-new-business-of-ai-and-how-its-different-from-traditional-software/>

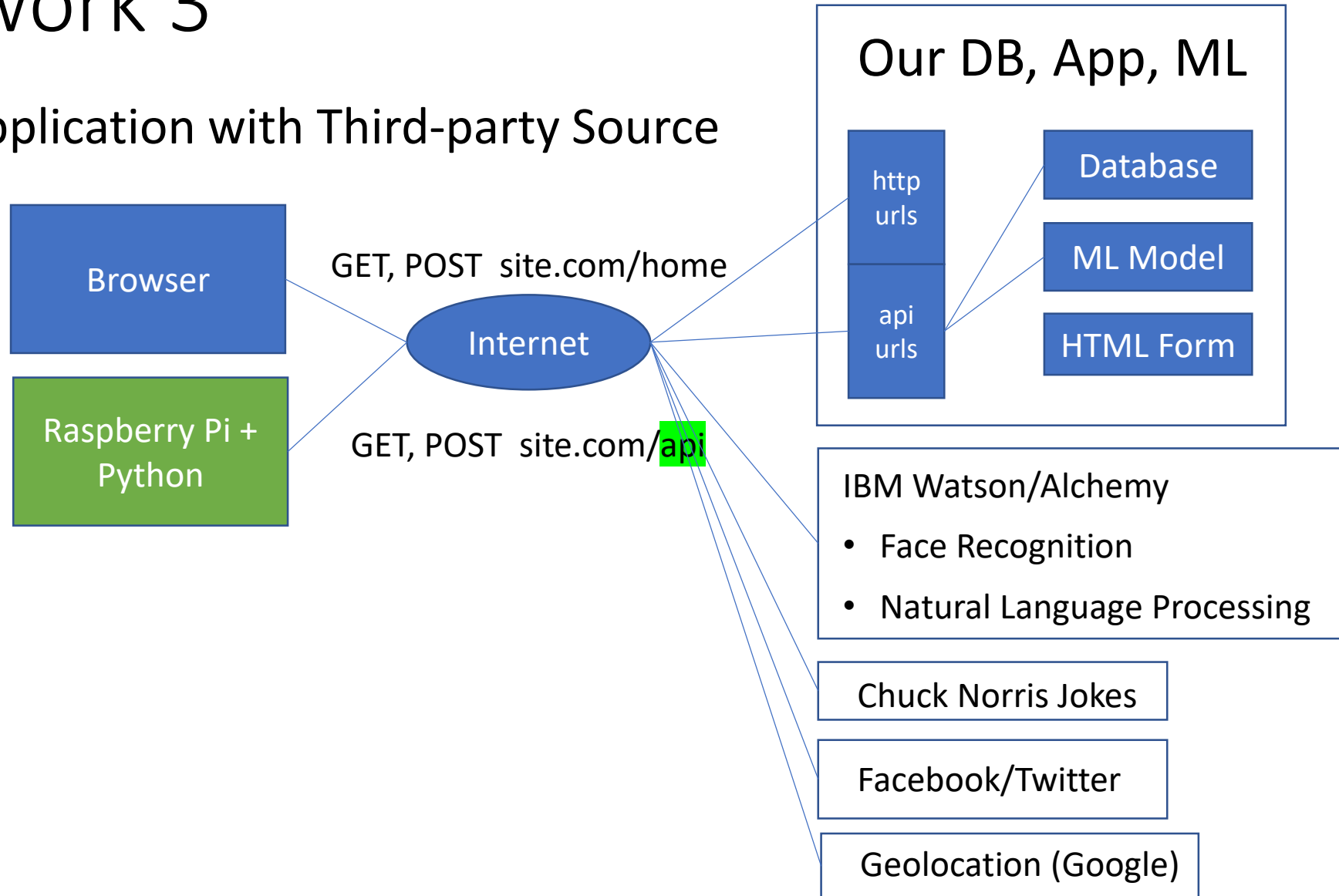
# Framework 2

## Internet of Things



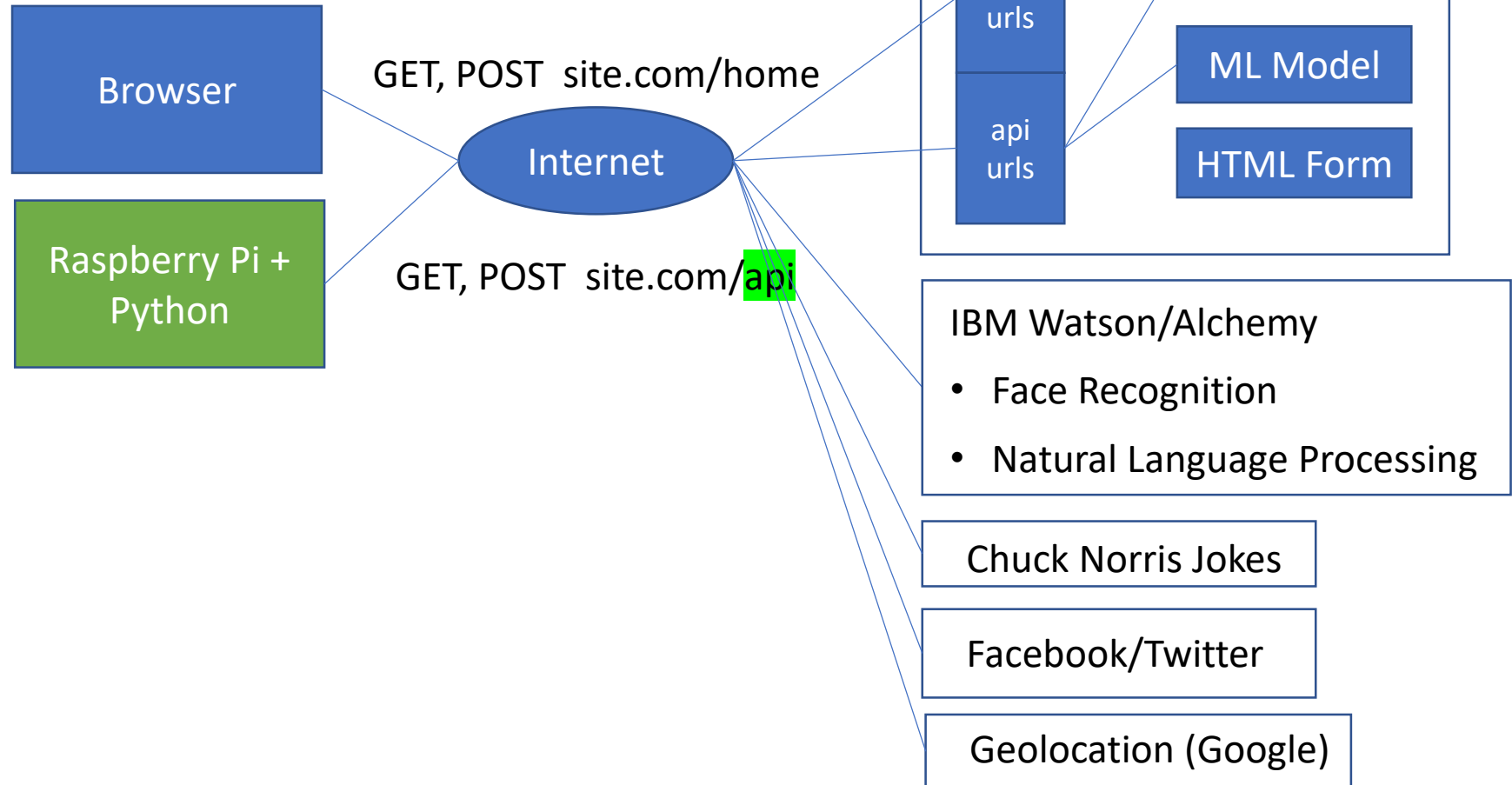
# Framework 3

## Application with Third-party Source



# Framework 4

## Application with Third-party Source





# There's so much out there...

- 18 Fun APIs For Your Next Project <https://medium.com/@vicbergquist/18-fun-apis-for-your-next-project-8008841c7be9>
- 15 APIs developers need to know <https://www.creativebloq.com/web-design/apis-developers-need-know-121518469>
- 9 free/cool web APIs to use in your next project <https://rapidapi.com/collection/cool-apis>
- ...the list goes on

“If you need some intelligence in your app, you'd be silly to build the NLP and other technology on your own. Instead, focus on what your app will do with that intelligence.” <https://www.creativebloq.com/web-design/apis-developers-need-know-121518469>

But we digress

One step at a time...

# “StarChaser”



- Game where you compete to classify stars (Supernova, Pulsar, etc)
- Leaderboard with head-to-head competition
- Python/Django backend with:
  - Keras/Tensorflow Machine Learning
  - Kaggle dataset, Postgres
- References and Links to explain what's going on

# Best Practices in Web App Design

- Define the Problem
- Identify Needs of User, Organization
- Do Some Research
- Define Your Solution
- Test Your Design

## To Include:

- Identify Audience
- Key Points
- Wireframe and Review

## References:

Getting Started in UX Design by Kurt Krumme <https://app.pluralsight.com/library/courses/getting-started-ux-design/table-of-contents>

UX Design Creating Wireframes by Susan Simkins <https://app.pluralsight.com/library/courses/ux-design-creating-wireframes/table-of-contents>

# Target Audience:

- Engineering Manager (Hiring Manager)
- Machine Learning / Vision Architect
- General Passers-by (recruiters, friends, networking contacts)

# Personas



## Data / ML Architect

- Expanding team to develop and deploy new algorithms
- Needs to preserve current tools and process
- Needs to define metrics/success
- Needs infrastructure/tool roadmap
- Needs trained/skilled hands



## Director of Engineering

- Wants to add Machine Learning to Web App
- Needs to deliver on schedule
- Needs to deliver quality
- Predictability is paramount



## Passers-by

Recruiters:

- “Can I place him?”
- “What does he do?”

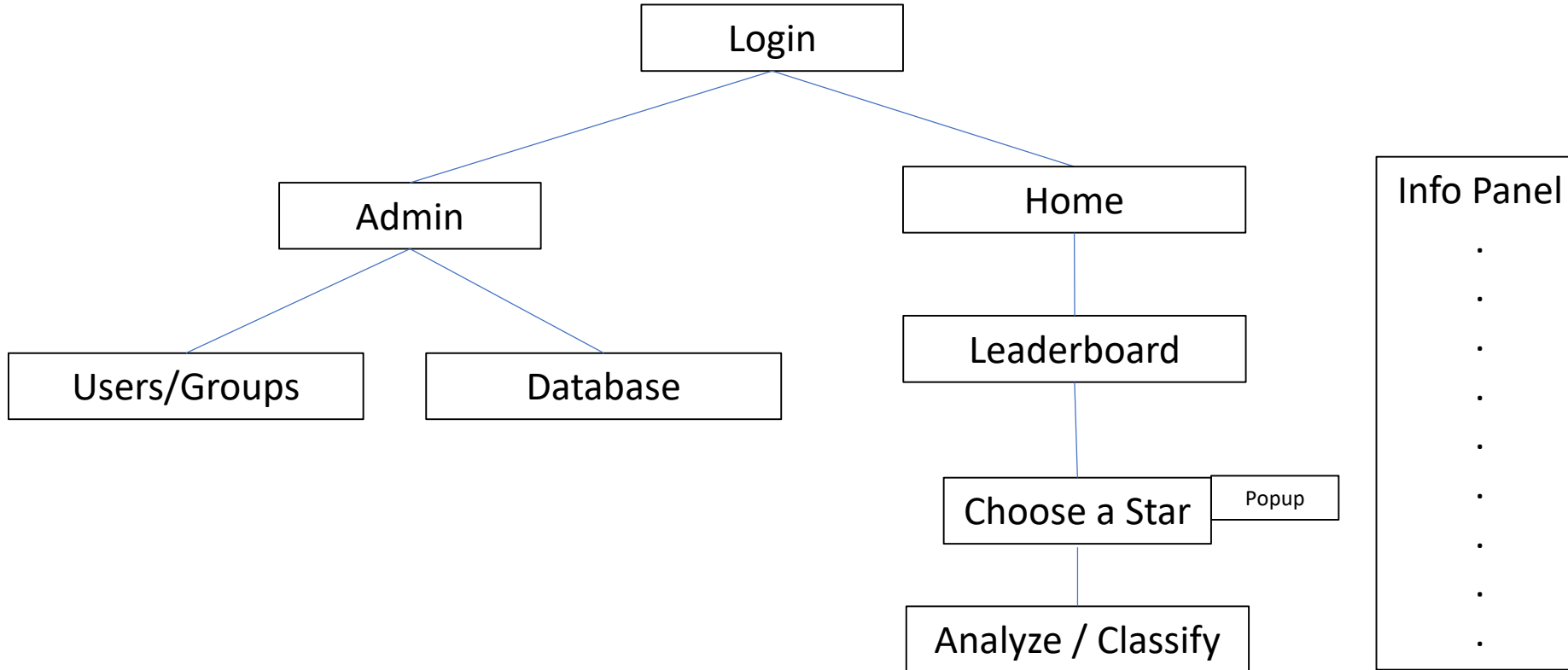
Professional Contacts / Network

- Elevator pitch

# Priority Content

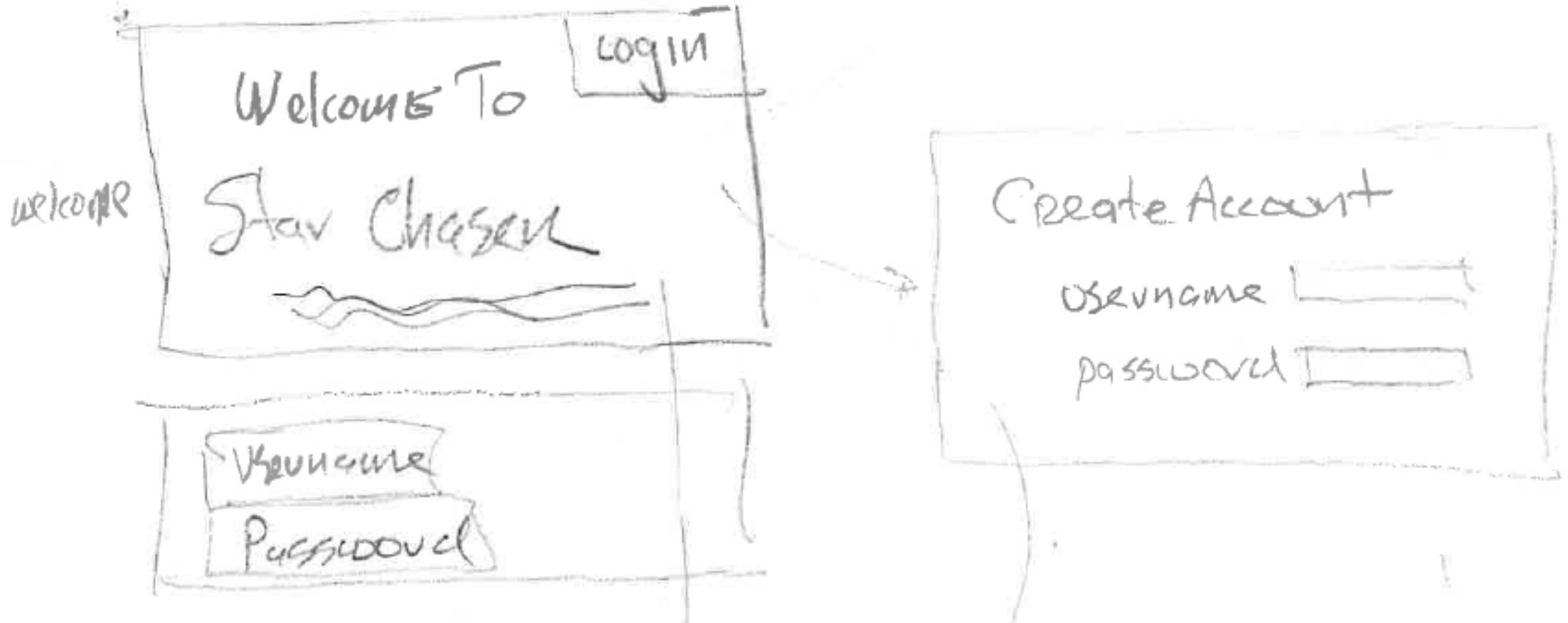
- “Info” panels explaining what/why
  - Context (why) of the application (first page)
  - Background on PLAsTiCC
  - Metrics – why important and reference
  - Data Processing – constructing the model (slide)
  - Dataset – timeseries and size/scale
  - Tensorflow Model – B.Trotta and K.Boone approach summary and other approaches used. Reference/link to my other slidesets.
  - Tensorflow background
  - Structure of website (slides from this presentation)
- “Popup” on make\_bid page:
  - Highlight merge of TensorFlow ML results with local database

# Proposed Site Map





# Wireframes



player  
(home)

Welcome Alice

Stars You Chased

name	class	Winnings
star 2	12	91200
star 3	13	4300

LEADER BOARD

Rank	Name	\$
1	ALICE	13,500
2	CHRIS	2100
3	Rob	300

CHASE A STAR

Select A Star from the  
list or get a **New List**

① 210734  
302124  
312150  
2111  
⑩ 2111

PICK  
STAR

Choose Wisely!

B. Trotter  
P(class)

1 . . . . 13

K. Boone  
P(class)

1 . . . . 13

Name your Star

"star1"

Help

Info

CHASE IT

Help

Info

Context

Help(game) → Pop-up

Info (technology behind the scenes) → Pop-up

Info (astronomy) ——— Link/Ref

Info (Kaggle PLAS-T, CC) → Link/Ref.

Info (B. Trotta) — Ref.

Info (K. Boone) — Ref.

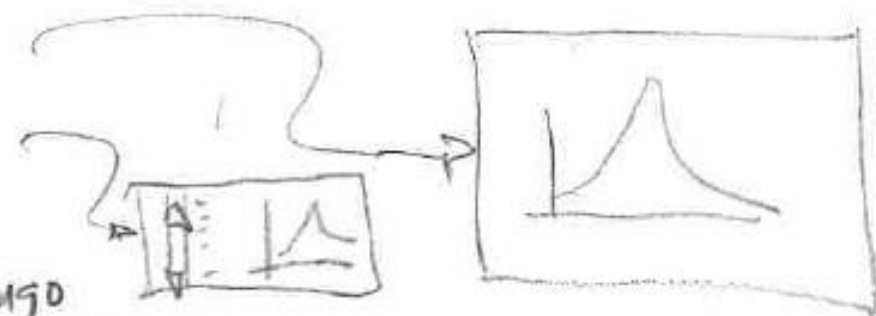
Info (this star)

Info (example stars)

Info METRICS

Info WEB APP, Django

Info TIMESERIES data



# PLACE Your Bets!

[Help](#)
[Info](#)
[Logout](#)

You have \$100 and bet what class of star you think it is.

Star

STAR CLASS

place  
bits

PAYOUT

1	2	3	4	5	6	7	8	9	10	11	12	13
1M	100k	50k	20k	10k	5k	1k	500	200	100	50	20	10

ADVICE:

B. Trotta

--	--	--	--	--	--	--	--	--	--	--	--	--

K. Boone

--	--	--	--	--	--	--	--	--	--	--	--	--

YOUR BET:

--	--	--	--	--	--	--	--	--	--	--	--	--

SUBMIT

TOTAL

--

WELL DONE!

You have won \$ ~ on your \$100 bet

Class	1	2	3	4	5	6	7	8	9	10	11	12	13
Payoff	\$1M	100k	50k	20k	.	.	.	100	.	.	.	.	\$20

ADVICE  
B. Trotter  
K. Boone


Your Bet

							20						
--	--	--	--	--	--	--	----	--	--	--	--	--	--

Results

							WINNER						
--	--	--	--	--	--	--	--------	--	--	--	--	--	--

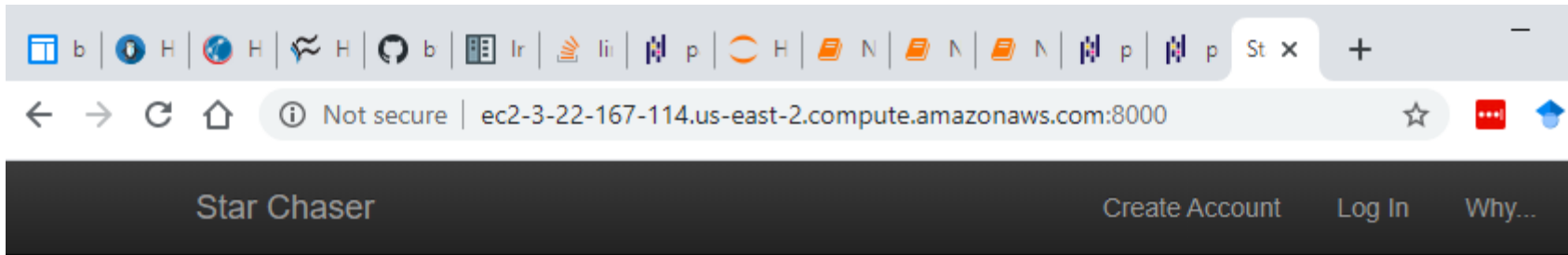
\$\$\$

							\$2000						
--	--	--	--	--	--	--	--------	--	--	--	--	--	--

TOTAL Winnings: \$2000

Play Again

Home



## Let's play Starchaser!

In this game you are an astronomer tasked with identifying different types of stars.  
Is it a supernova? A pulsar? Or maybe there was a lensing event.  
Top researchers will help you decode the astronomical observations.

Good luck on your new role, astronomer!

b

H

H

H

b

lr

li

p

H

N

N

N

p

p

St

x

+

←

→

↻

🏠

ⓘ

Not secure

|

ec2-3-22-167-114.us-east-2.compute.amazonaws.com:8000/player/home/

|

☆

⋮

🔒

Star Chaser

Logout

Why...

# Welcome, cwinzor

Stars you chased:

Your total score is 0

Leaderboard:

- Player 2: bob 0
- Player 3: susan 0
- Player 4: kyle 0
- Player 5: dpwinsor 0
- Player 1: cwinzor 0

Select a star



## Choose a star:

Available:

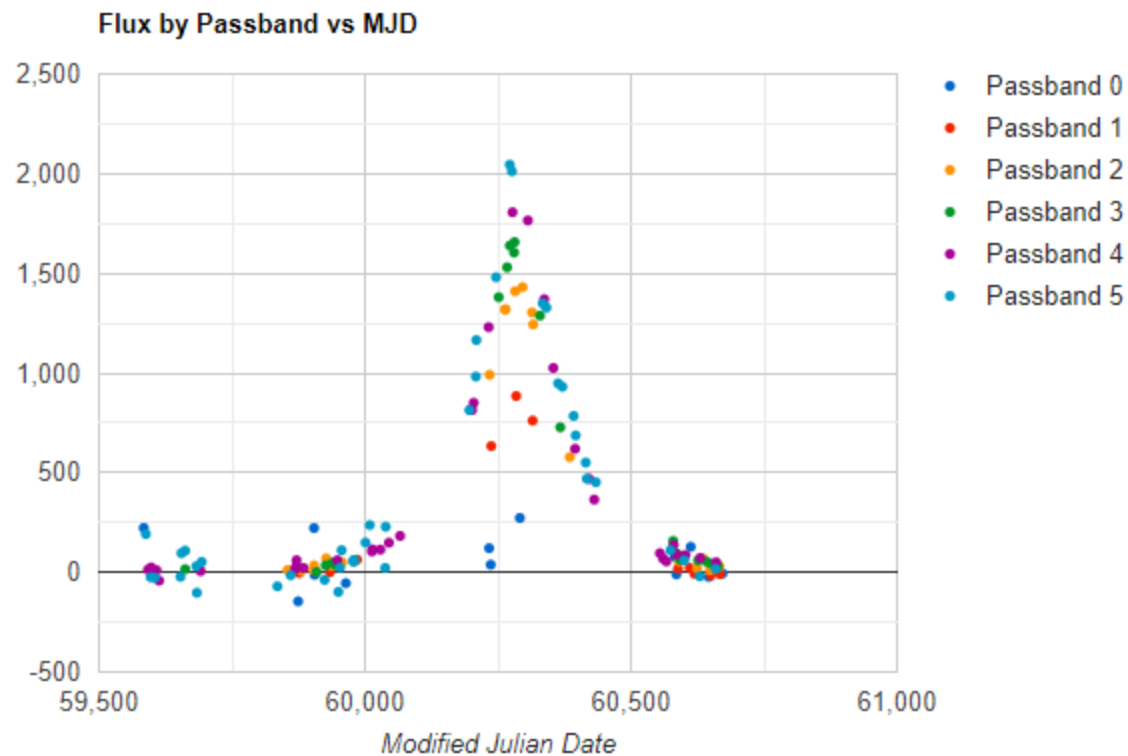
Star ID	Show Chart	New Deal	Place Bet
• 21906139	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 102667737	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 223791	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 216970	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 104209892	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 47726502	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 6660231	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 52175350	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 72735	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 124679	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 118117242	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 123488700	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 93394704	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 104765443	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>

## Choose a star:

Available:

123488700

Star ID	Show Chart	New Deal	Place Bet
• 21906139	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 102667737	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 223791	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 216970	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 104209892	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 47726502	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 6660231	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 52175350	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 72735	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 124679	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 118117242	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 123488700	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 93394704	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 104765443	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>

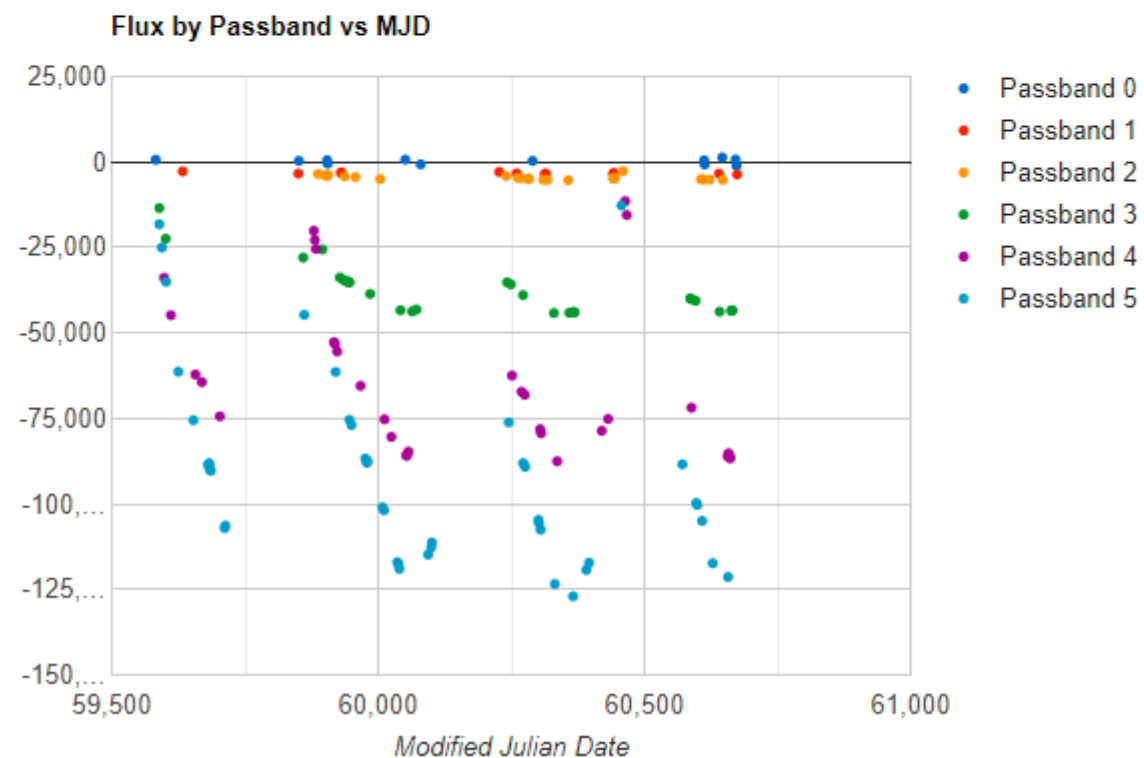


## Choose a star:

Available:

104765443

Star ID	Show Chart	New Deal	Place Bet
• 21906139	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 102667737	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 223791	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 216970	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 104209892	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 47726502	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 6660231	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 52175350	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 72735	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 124679	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 118117242	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 123488700	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 93394704	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 104765443	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>



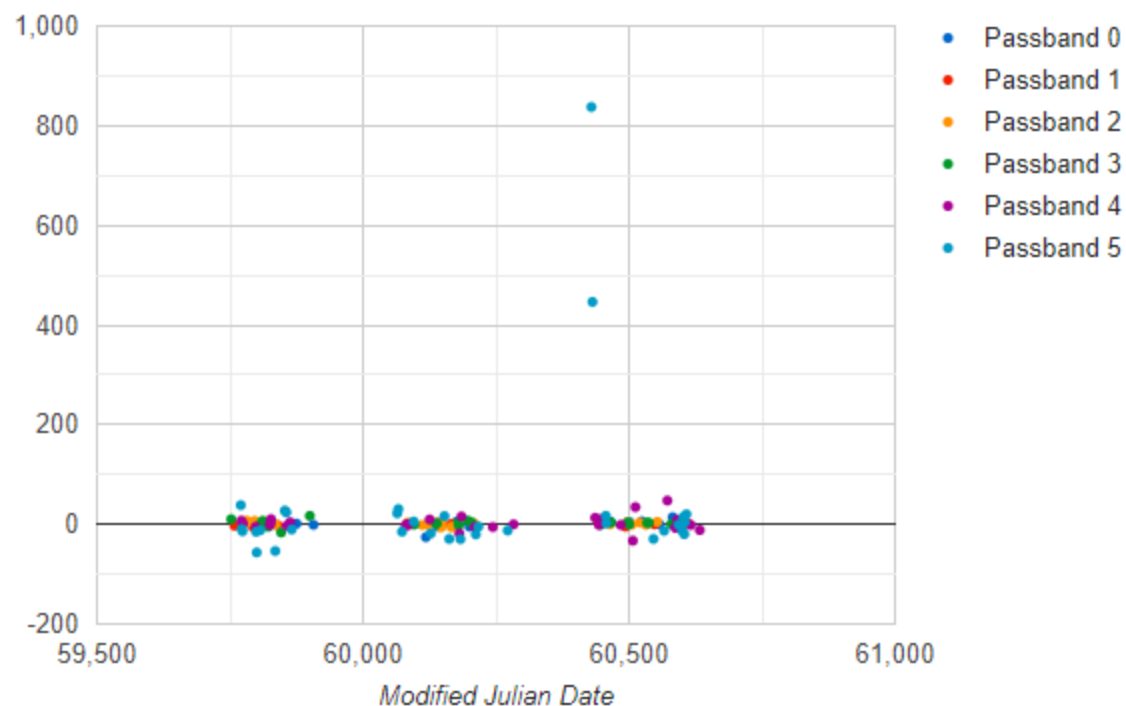
## Choose a star:

Available:

93394704

Star ID	Show Chart	New Deal	Place Bet
• 21906139	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 102667737	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 223791	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 216970	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 104209892	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 47726502	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 6660231	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 52175350	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 72735	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 124679	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 118117242	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 123488700	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 93394704	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 104765443	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>

Flux by Passband vs MJD

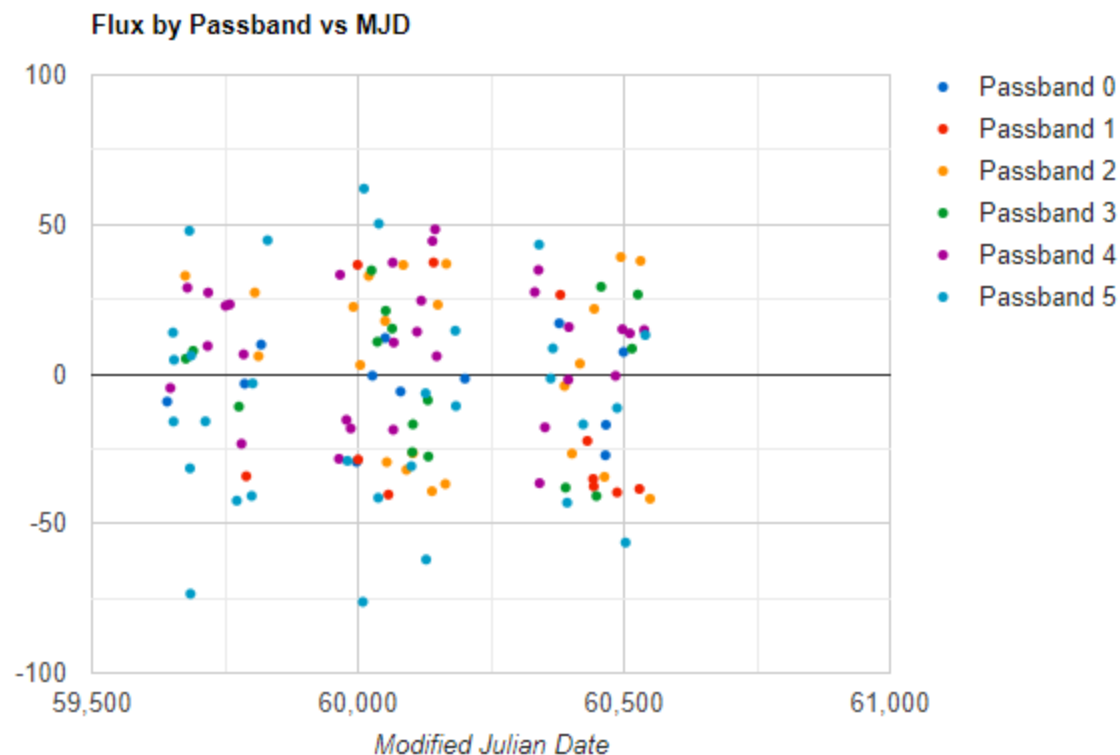


## Choose a star:

Available:

52175350

Star ID	Show Chart	New Deal	Place Bet
• 21906139	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 102667737	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 223791	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 216970	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 104209892	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 47726502	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 6660231	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 52175350	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 72735	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 124679	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 118117242	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 123488700	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 93394704	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>
• 104765443	<a href="#">Chart</a>	<a href="#">Deal</a>	<a href="#">Bet</a>



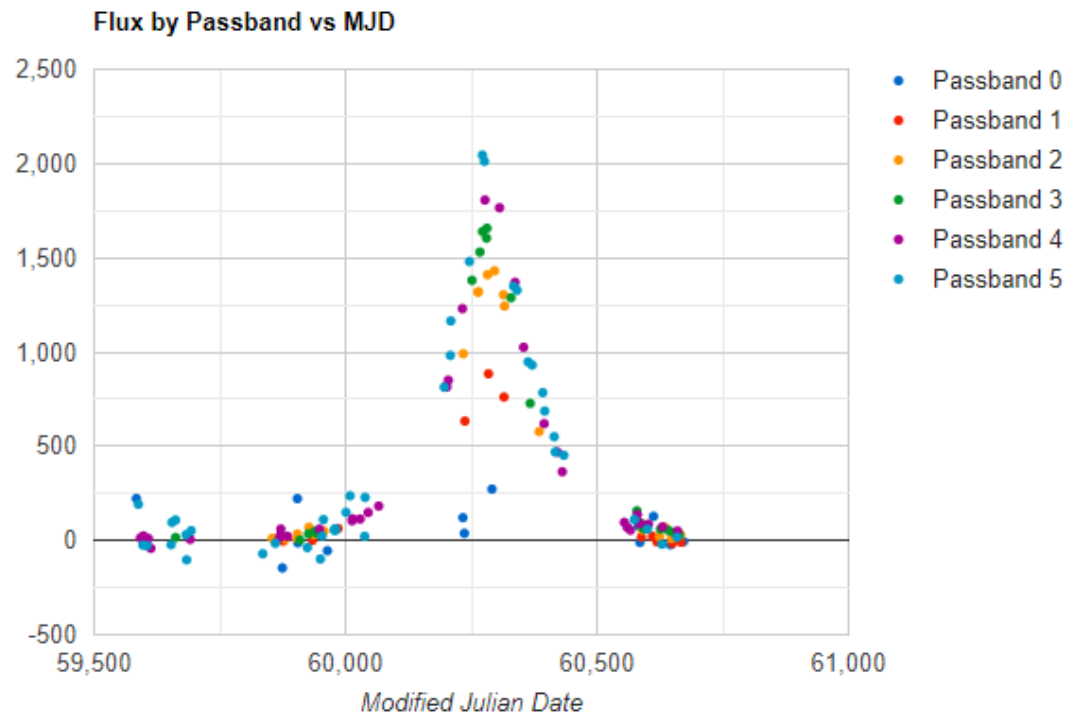
# Place your Bets!

Guidance is provided by astronomers B.Trotta and K.Boone using TensorFlow AI.

Class	B.Trotta's Bet	K.Boone's Bet	Your Bet (\$)
Supernova R	0	0	10
Supernova 1	0	0	30
Supernova 1b,c	3	0	35
Supernova 2	20	8	20
Binary (Spectro)	35	2	5
Eclipsing Binary	25	10	0
Astrometric Binary	10	40	0
Microlensing	7	35	0
Rotational Pulsar	0	5	0
X-Ray Pulsar	0	0	0
Type K	0	0	0
Type L	0	0	0
Type M	0	0	0
Total Bet			0

Submit

123488700



[blog\\_](#)[How](#)[How](#)[How](#)[btrot](#)[Instal](#)[linux](#)[pand](#)[Hom](#)[NB6c](#)[NB6c](#)[NB7c](#)[pand](#)[pand](#)[S](#)

Not secure | ec2-3-22-167-114.us-east-2.compute.amazonaws.com:8000/player/well\_done/



Star Chaser

[Logout](#)

[Why...](#)

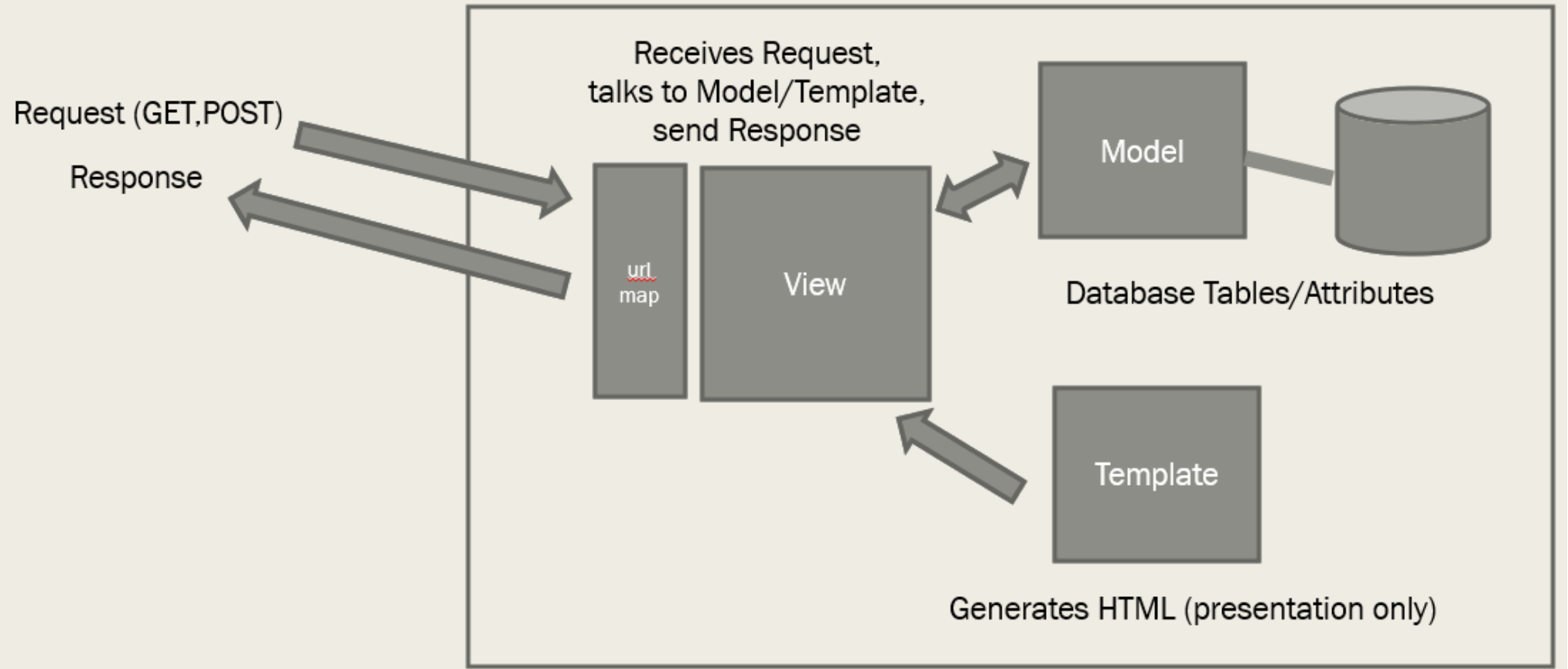
# Well done!

[Home](#)

# From last time

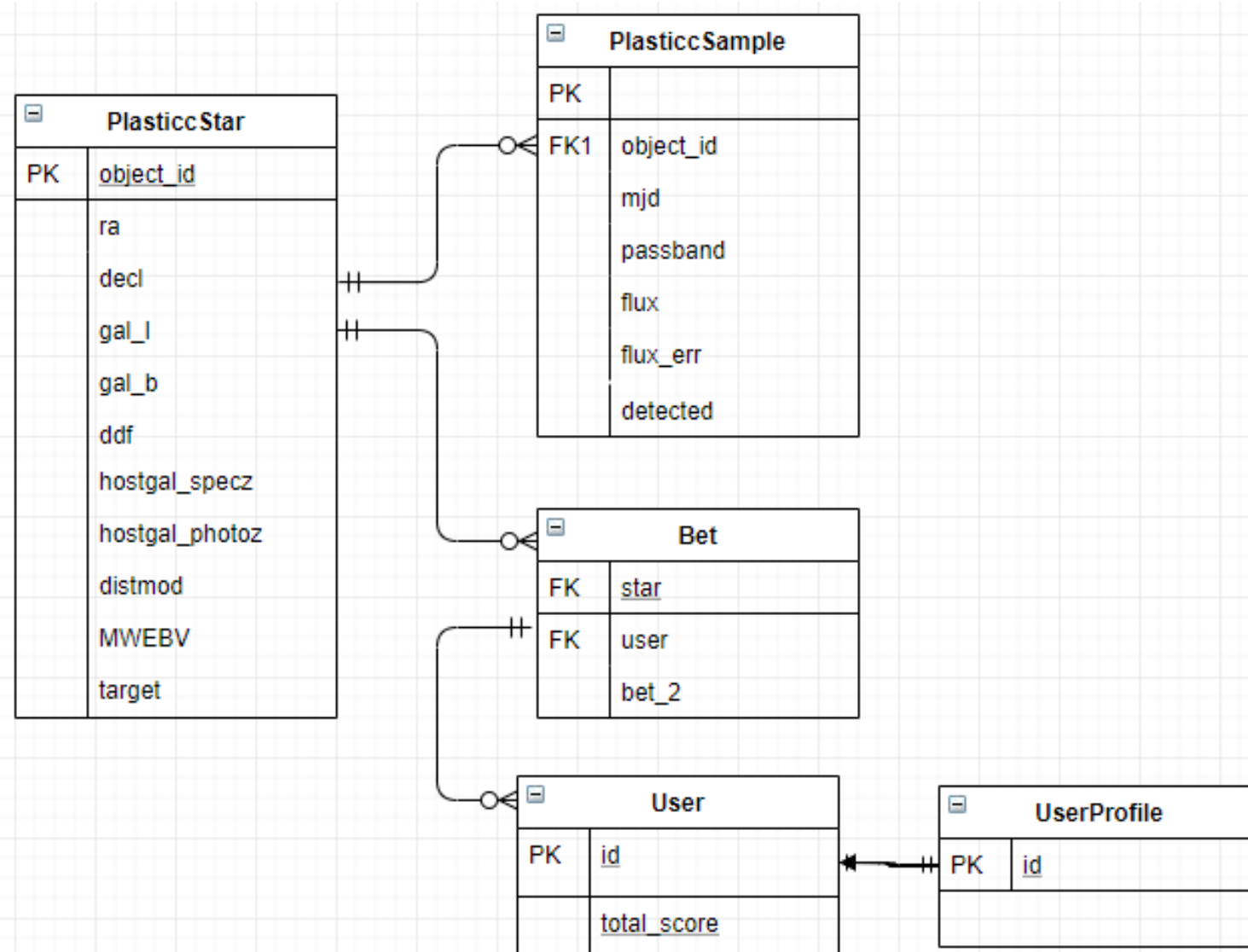
## Model Template View

Similar to MVC





# Schema



# pick\_star (view)

GET only (no template/POST)

Display a list of stars to choose from

Chart one star (chosen by user from list)

Two parameters on URL

- <http://...?p1=x&p2=y>

```
30 def pick_star(request):
31
32     # param 1 - list of star IDs comma delimited
33     if 'param1' in request.GET:
34         starlist_np = np.fromstring(param1, dtype=int, sep=',')
35     else:
36         starlist_np = PlasticcStar.objects.random_set()
37
38     # param2 = star to display chart
39     if 'param2' in request.GET:
40         star_to_display = request.GET['param2']
41     else:
42         star_to_display = ''
43
44     # build comma-delimited string
45     starlist_string = ''
46     for star_id in starlist_np:
47         starlist_string = '{}{}'.format(starlist_string, star_id)
48
49     context = dict()
50     context['starlist_string'] = starlist_string
51     context['star_to_display'] = star_to_display
52
53     # for charting - get data for chart
54     [star_obj, timeseries_data_str] = get_chart_data(star_to_display)
55     context['timeseries_data_str'] = timeseries_data_str
56
57     return render(
58         request=request,
59         template_name="app_player/pick_star.html",
60         context=context)
```

# pick\_star (html)

```
10 <h1>
11 |   Choose a star:
12 </h1>
13
14
15 <div class="col-sm-4">
16 |   <h3>Available:</h3>
17
18 |   {% for star in starlist_obj %}
19 |
20 |       <div class="col-sm-4"> {{ star.star_id }} </div>
21 |       <div class="col-sm-2"><a
22 |           |       href="{% url 'player_pick_star' %}?param1={{ starlist_string }}&param2={{ star.star_id }}">Chart</a>
23 |       </div>
24 |       <div class="col-sm-2"><a href="{% url 'player_pick_star' %}">Deal</a></div>
25 |       <div class="col-sm-2"><a href="{% url 'place_bet' id=star.star_id %}">Bet</a></div>
26 |
27 |   {% endfor %}
28
29 <!--Div that will hold the chart-->
30 <div class="col-sm-6">
31 |   <h3>{{ star_to_display }}</h3>
32 |   <div id="chart_div_2">
33 |
34 |       <!-- load my charting javascript -->
35 |       {% if timeseries_data_str %}
36 |       <script type="text/javascript">
37 |           (function () { window.temp5 = {{ timeseries_data_str }} ; }) ();
38 |       </script>
39 |       <script src="{% static 'js/chart_timeseries.js' %}"></script>
40 |       {% endif %}
41 |   </div>
42 </div>
```

# place\_bets (view)

Template/POST

Get:

Call ML modes for recommendations

Create form and populate

POST

Receive filled in form from request.POST

Validate

Save and redirect, or

Re-display w/ errors from "is\_valid()"

URL has form

- <http://xyz.com/starid>

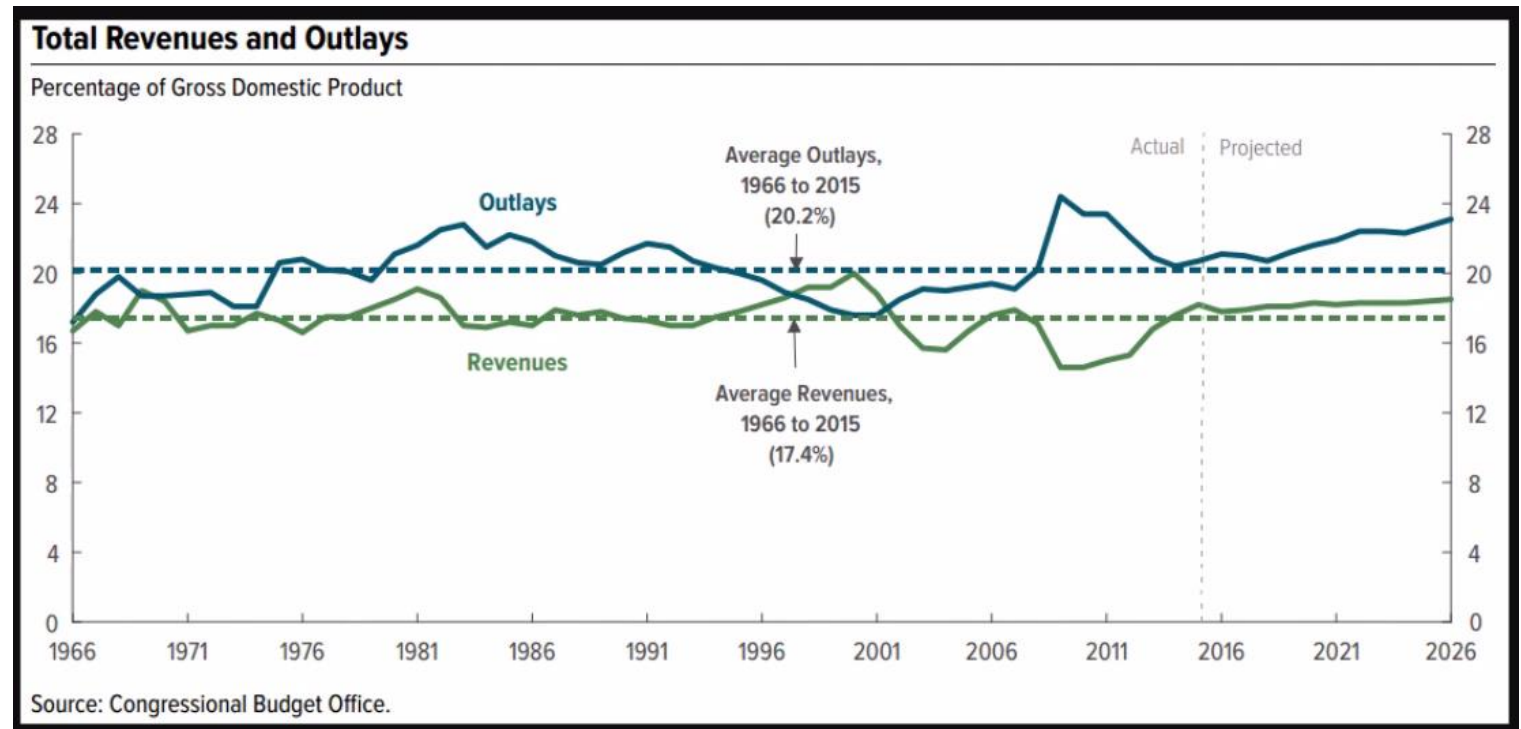
```
73 def place_bet(request, id):
74     star = get_object_or_404(PlasticcStar, pk=id)
75
76     if request.method == "POST":
77         form = BetForm(data=request.POST)
78
79         if form.is_valid():
80             form.save()
81             return redirect('player_well_done')
82     else:
83         df_btrotta = pd.DataFrame( )
84         df_kboone = pd.DataFrame( )
85
86         form = BetForm()
87         bet_form_set_disabled_fields(form, request, star)
88         bet_form_set_values_for_non_db_fields(form, request, star)
89         bet_form_set_reduction_fields(form, request)
90
91         # for charting - get chart data
92         [star_obj, timeseries_data_str] = get_chart_data(id)
93
94         context = {}
95         context['form'] = form
96         context['star_id'] = star.star_id
97         context['star_obj'] = star_obj
98         context['timeseries_data_str'] = timeseries_data_str
99
100     return render(
101         template_name="app_player/bet_form.html",
102         context=context)
```

## place\_bets (html)

```
10 <h1>Place your Bets!</h1>
15 <form method="post" action="{% url 'place_bet' id=star_id %}">
23     <div class="row">
24         <div class="col-sm-3"> <b>Class</b> </div>
25         <div class="col-sm-2"> <b>B.Trotta's Bet</b> </div>
26         <div class="col-sm-2"> <b>K.Boone's Bet</b> </div>
27         <div class="col-sm-2"> <b>Your Bet ($)</b></div>
28         <div class="col-sm-1"> <b></b> </div>
31     <div class="row">
32         <div class="col-sm-3"> Supernova R</div>
33         <div class="col-sm-2"> {{ form.bid_a2 }} </div>
34         <div class="col-sm-2"> {{ form.bid_a3 }} </div>
35         <div class="col-sm-2"> {{ form.bid_a }} </div>
36         <div class="col-sm-1"> {{ form.bid_a.errors }} </div>
37     </div>
38     (...)
145 <!-- the chart-->
147 <h3>{{ star_id }}</h3>
152 <!-- load charting javascript -->
153 {% if timeseries_data_str %}
154 <script type="text/javascript">
155     (function () { window.temp5 = {{ timeseries_data_str }} ; }) ();
156 </script>
157 <script src="{% static 'js/chart_timeseries.js' %}"></script>
158 {% endif %}
164 <font color="red"> <b> {{ form.non_field_errors }} </b> </font>
```

# “Data Visualization”

- Tons of libraries. High end (D3.js) to basic (Google Charts)
- Initial exploration is by Jupyter Notebook and/or Pandas/Matplotlib... see Titanic and PLAsTiCC #1
- For Web App I went with Google Charts

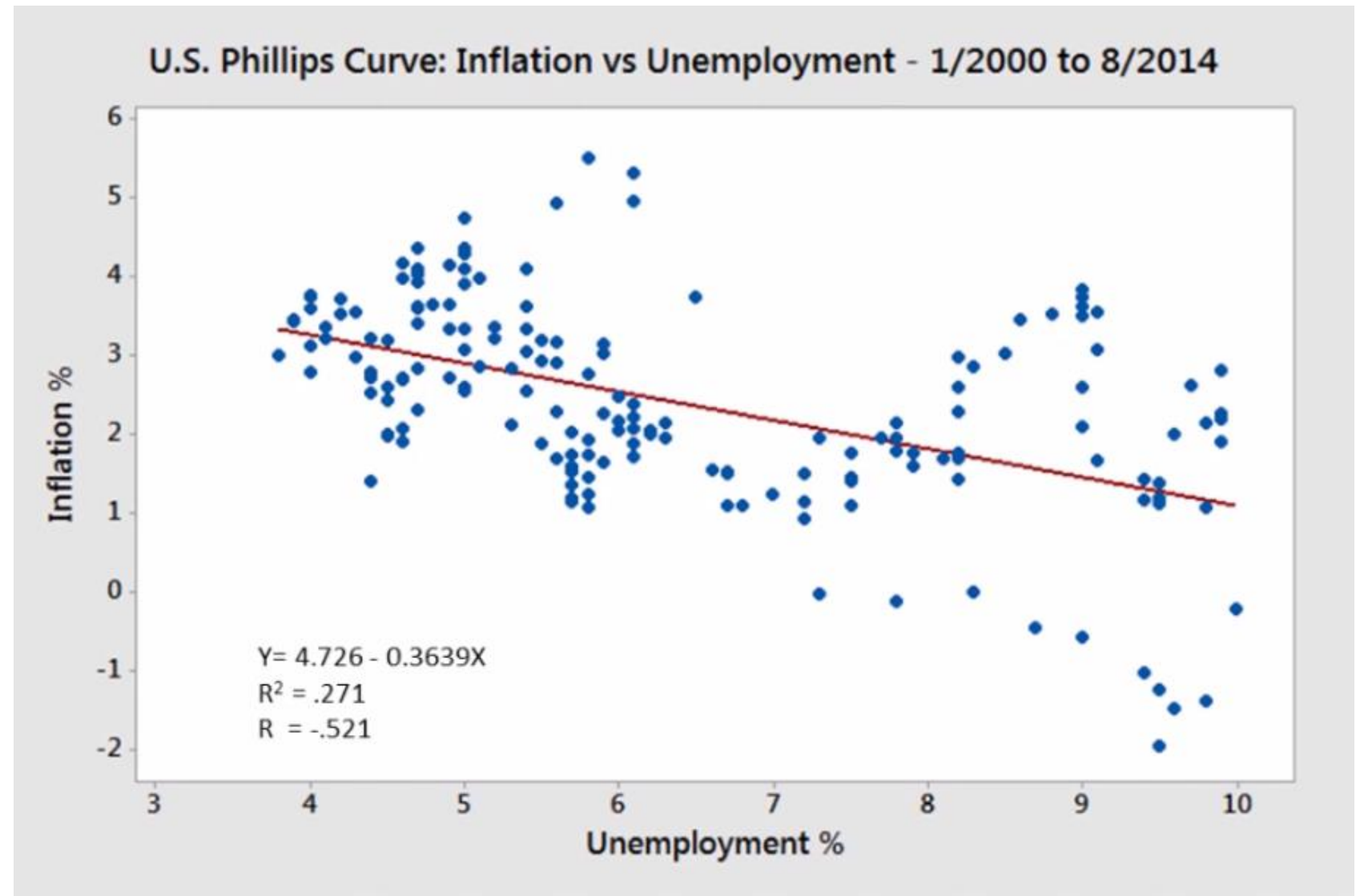


References:

<https://app.pluralsight.com/library/courses/build-first-data-visualization-google-charts>

<https://www.sitepoint.com/best-javascript-charting-libraries/>

# Why this is important...



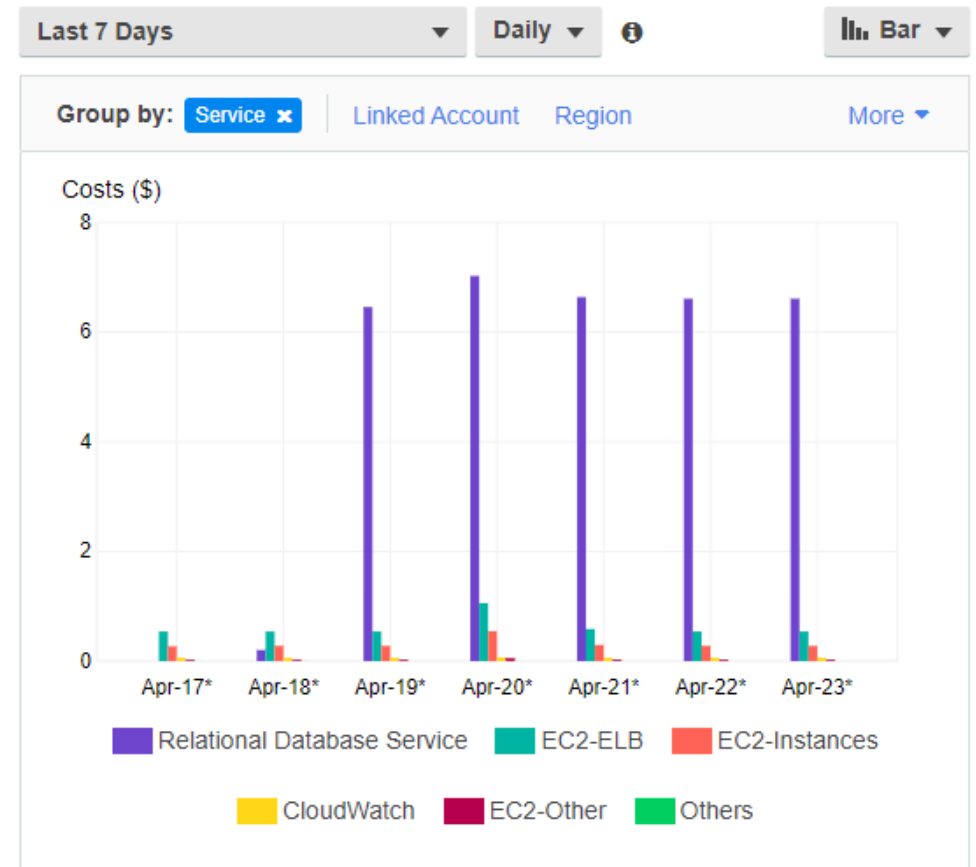
Source Data: FRED Database  
Inflation: CPI for All Urban Consumers

Reference:

<https://app.pluralsight.com/library/courses/build-first-data-visualization-google-charts>

# AWS EB (Elastic Beanstalk) vs ES2

- EB = EC2 + RDS + S3
- Scales with demand
- Follow instructions from <https://realpython.com/deploying-a-django-app-and-postgresql-to-aws-elastic-beanstalk/>
- Straightforward but not really...
- Baseline = \$8/day with no traffic and minimal data (ARRGH!)





# Second Try: Postgres directly on EC2

## 🔒 Daily costs

Apr 15, 2020 - May 21, 2020

Daily



Bar

Group by: None

Service

Linked Account

Region

Instance Type

Usage Type

Resource

Cost Category

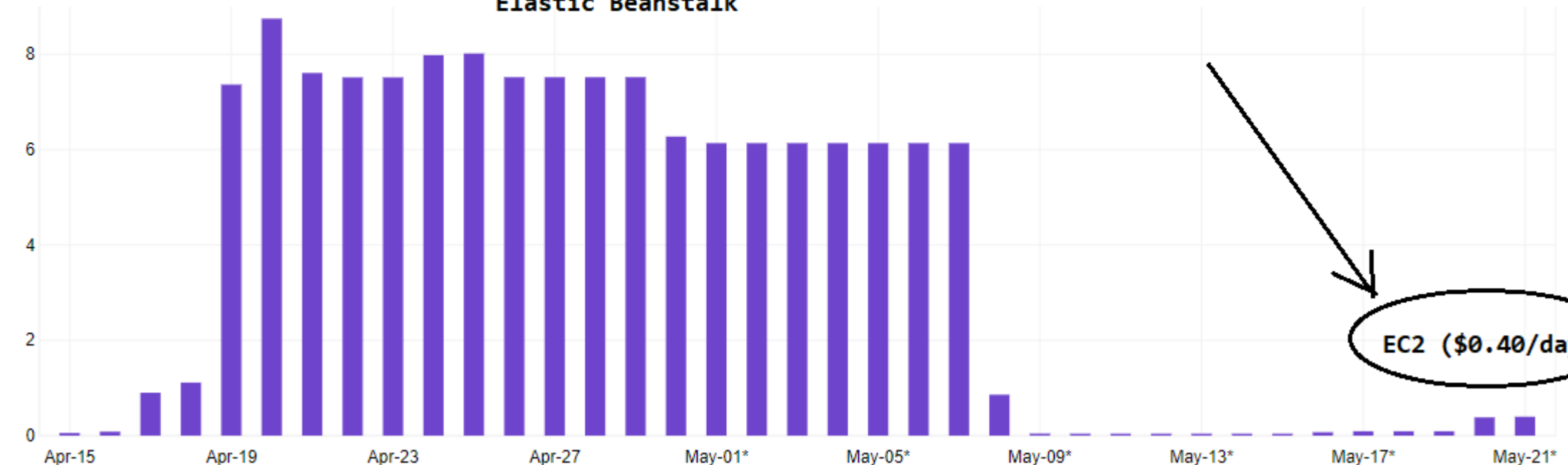
Tag

API Operation

More

Costs (\$)

Elastic Beanstalk



# Go Play !

- [www.cwinsor.us](http://www.cwinsor.us) (look for StarChaser)
- Or directly...
- <http://ec2-3-22-167-114.us-east-2.compute.amazonaws.com:8000/>

# Level of Effort and Takeaways

- A full featured Web App (admin, user accounts, Postgres, M.L., AWS)
- 2/23 to 5/22 (3 months). Lots of learning curve
- Started with Tic-tac-toe and:
  - Replaced out-of-box database with Postgres
  - Added external ML models, Kaggle dataset
  - Added Google Charts
  - Pushed to AWS
- Wireframe and personas super important for focus

# References

- Django:
  - Tic-Tac-Toe (Django Fundamentals by Reindert-Jan Ekker on Pluralsight) <https://app.pluralsight.com/library/courses/django-fundamentals-update/table-of-contents>
- Web UX/UI:
  - Getting Started in UX Design by Kurt Krumme <https://app.pluralsight.com/library/courses/getting-started-ux-design/table-of-contents>
  - UX Design Creating Wireframes by Susan Simkins <https://app.pluralsight.com/library/courses/ux-design-creating-wireframes/table-of-contents>
- Node, Postgres, Express (background only - could skim this)
  - “Build a CRUD single page application with Node, Express, Angular, Postgres” (Michael Herman)
  - <https://mherman.org/blog/postgresql-and-nodejs/> This is an example frontend/backend javascript web app with postgres db. It uses express web server/routing and (a little) angular on the front-end. You will use npm, express, node, browser trace/debug features. You will see javascript used on both client and server. This is very standard (server-side javascript) architecture.
- Front-end (suspect this will be handy in the future)
  - “Front-End Web Development Quick Start With HTML5, CSS, and JavaScript” (Shawn Wildermuth) <https://app.pluralsight.com/course-player?clipId=e5482b13-c204-4d52-89ec-94a1099592b0> Beginner HTML5, CSS, JavaScript – excellent

Thank You

# Best Pandas methods...

`.pivot()` given a nominal attribute, create an attribute for each value

`.merge()` merges two DataFrames  
(example...)

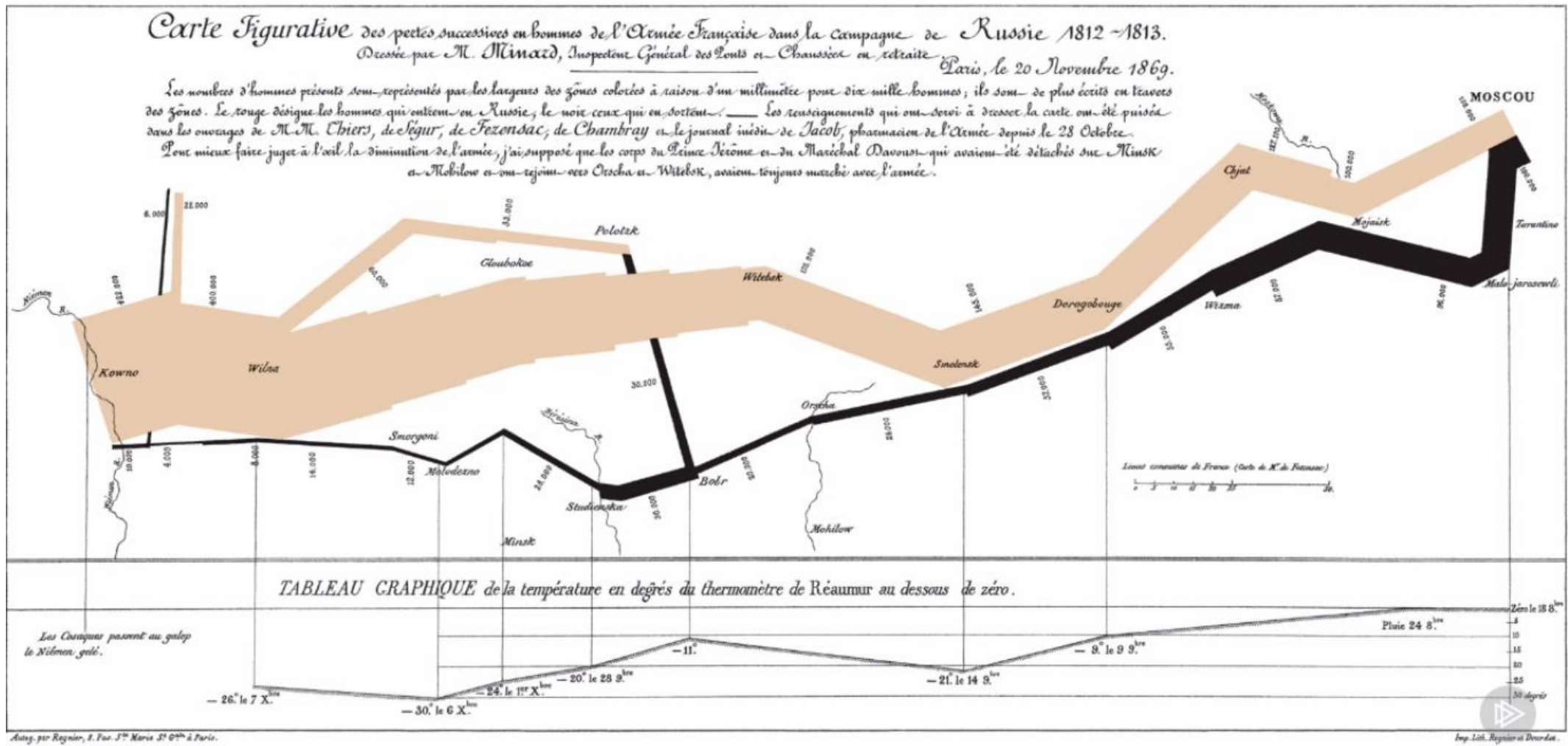
## Boolean indexing...

```
filter = reviews['score'] > 6.95  
reviews.loc[filter]
```

```
>>> df = pd.DataFrame({'foo': ['one', 'one', 'one', 'two', 'two',  
...                           'two'],  
...                   'bar': ['A', 'B', 'C', 'A', 'B', 'C'],  
...                   'baz': [1, 2, 3, 4, 5, 6],  
...                   'zoo': ['x', 'y', 'z', 'q', 'w', 't']})  
>>> df  
   foo  bar  baz  zoo  
0  one   A    1    x  
1  one   B    2    y  
2  one   C    3    z  
3  two   A    4    q  
4  two   B    5    w  
5  two   C    6    t
```

```
>>> df.pivot(index='foo', columns='bar', values='baz')  
bar  A  B  C  
foo  
one  1  2  3  
two  4  5  6
```

# Visualization



Reference:

<https://app.pluralsight.com/library/courses/build-first-data-visualization-google-charts>