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

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://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
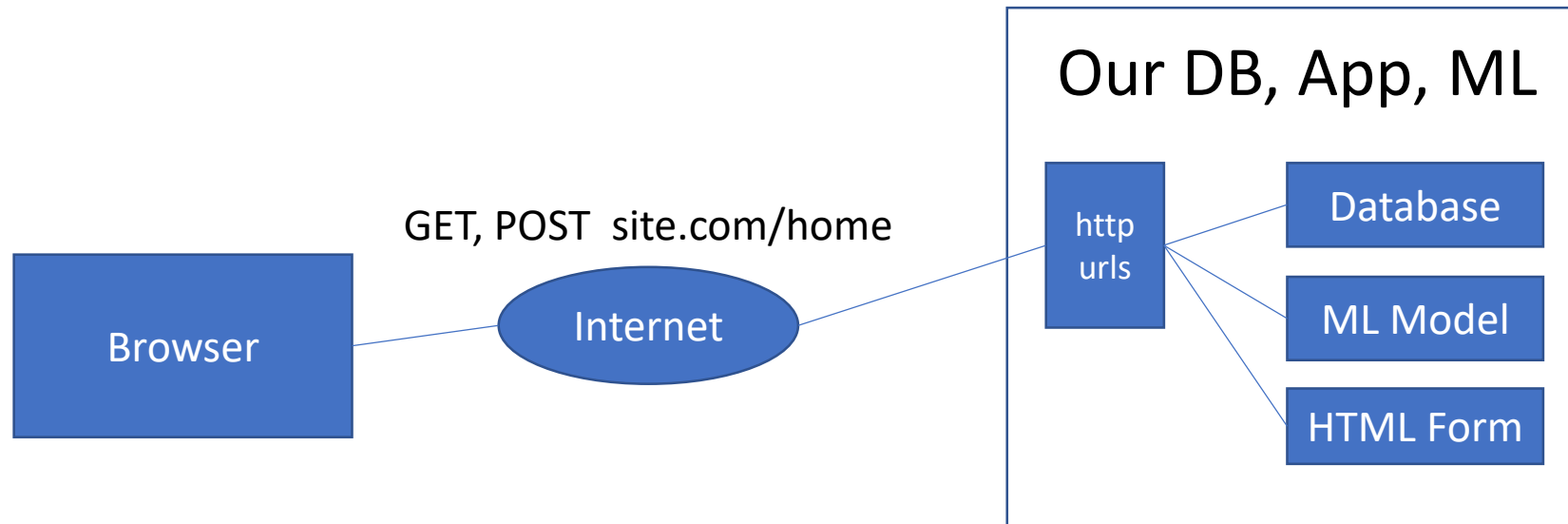
Goals:

- 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:
  - Make it available for Anyone (with access to a browser)
  - Practice, Practice, Practice

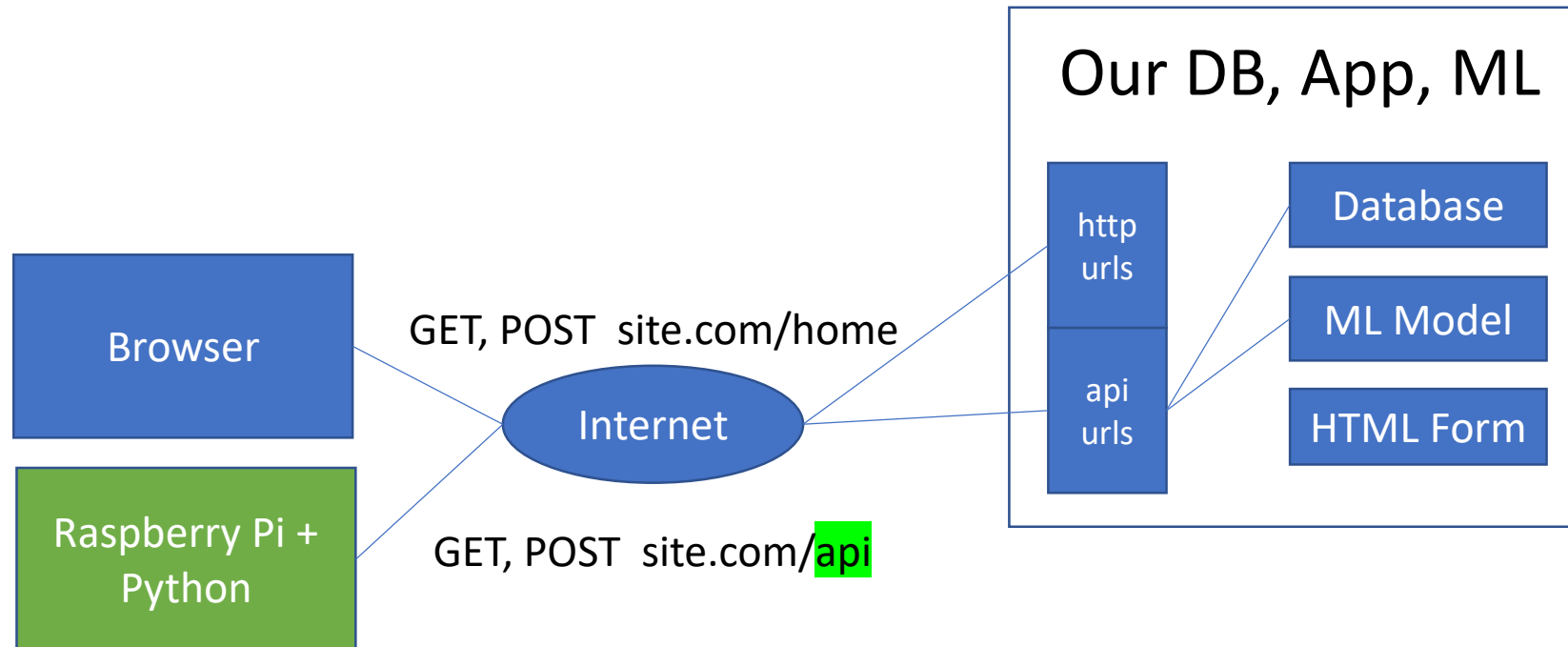# Django Framework

Game Application

Server-side Python

Our DB, App, ML

Browser

Internet

GET, POST site.com/home

http urls

Database

ML Model

HTML Form

Where we can take this…

- Today
  - ML as part of web application
- Tomorrow: Internet-of-Things
  - Front-end is Raspberry Pi running Python
  - Back-end "smarts" (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

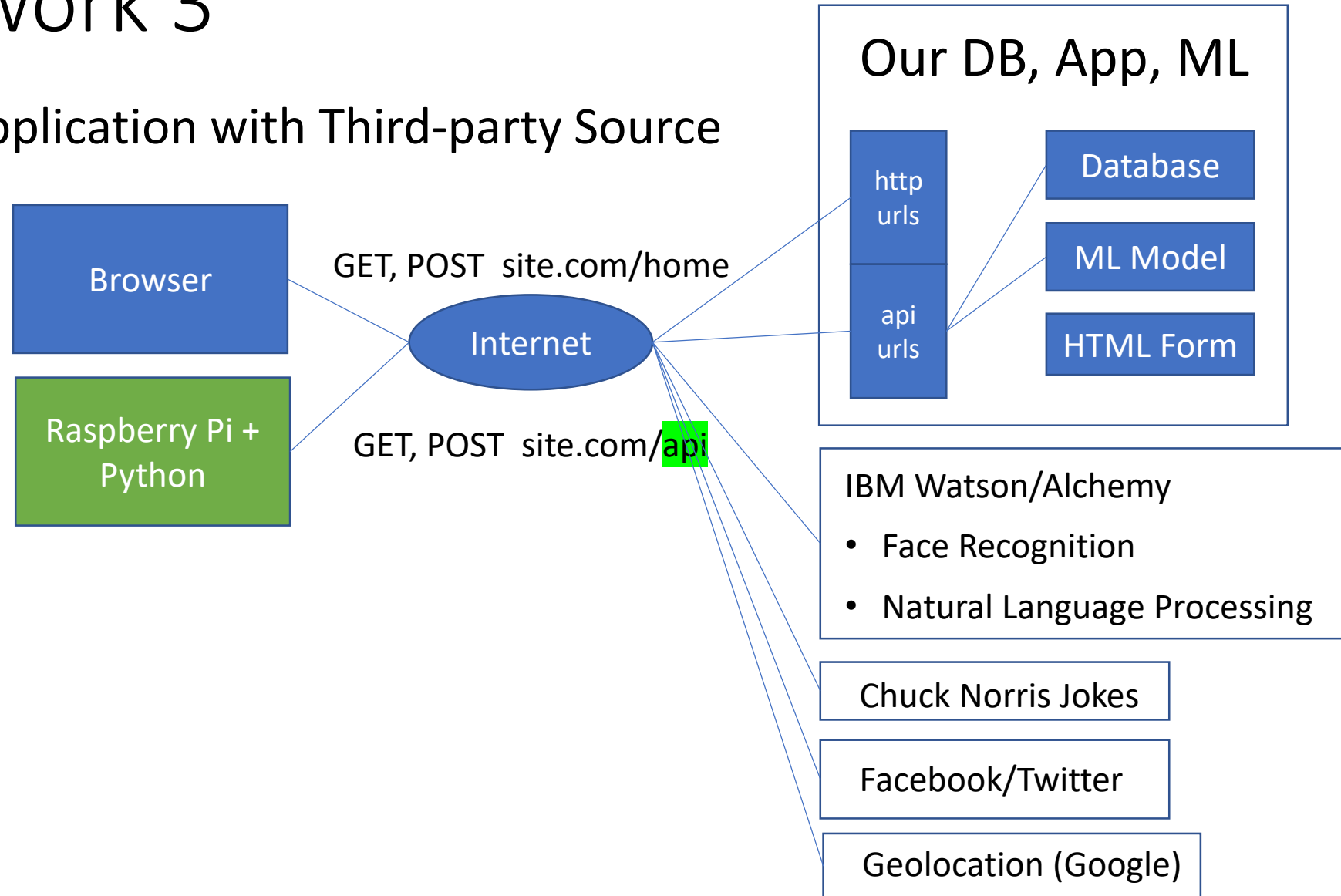## Internet of Things

Our DB, App, ML

Browser

Raspberry Pi + Python

Internet

GET, POST site.com/home

GET, POST site.com/api

http urls

api urls

Database

ML Model

HTML Form

Framework 3

Application with Third-party Source

Our DB, App, ML

Browser

Raspberry Pi + Python

GET, POST site.com/home

GET, POST site.com/api

Internet

http urls

api urls

Database

ML Model

HTML Form

IBM Watson/Alchemy
- Face Recognition
- Natural Language Processing

Chuck Norris Jokes

Facebook/Twitter

Geolocation (Google)

Framework 4

Our API to Subscribers

Application with Third-party Source

~~Our DB, App, ML~~

Browser

Raspberry Pi + Python

Internet

GET, POST site.com/home

GET, POST site.com/api

http urls

api urls

Database

ML Model

HTML Form

IBM Watson/Alchemy
- Face Recognition
- Natural Language Processing

Chuck Norris Jokes

Facebook/Twitter

Geolocation (Google)

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:
  - Kiras/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:
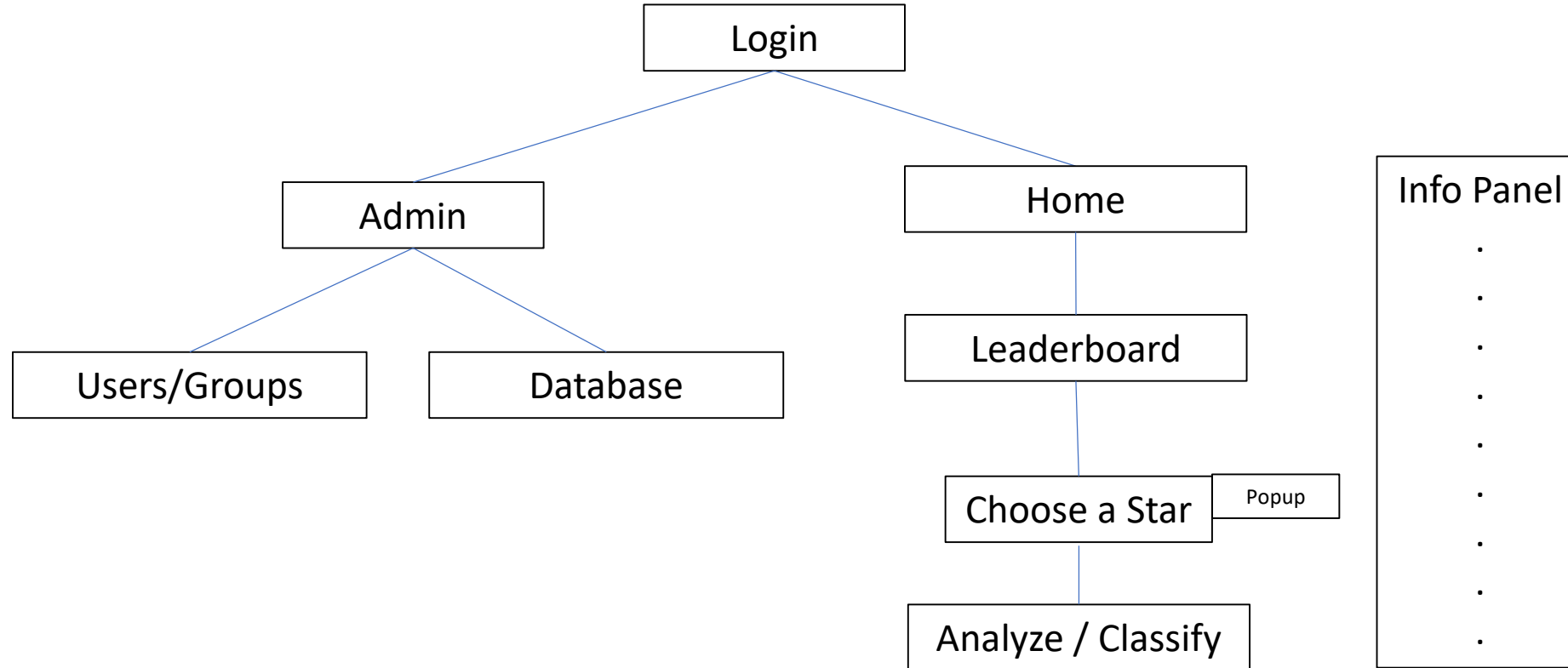
- "Can I place him?"

- "What does he do?"

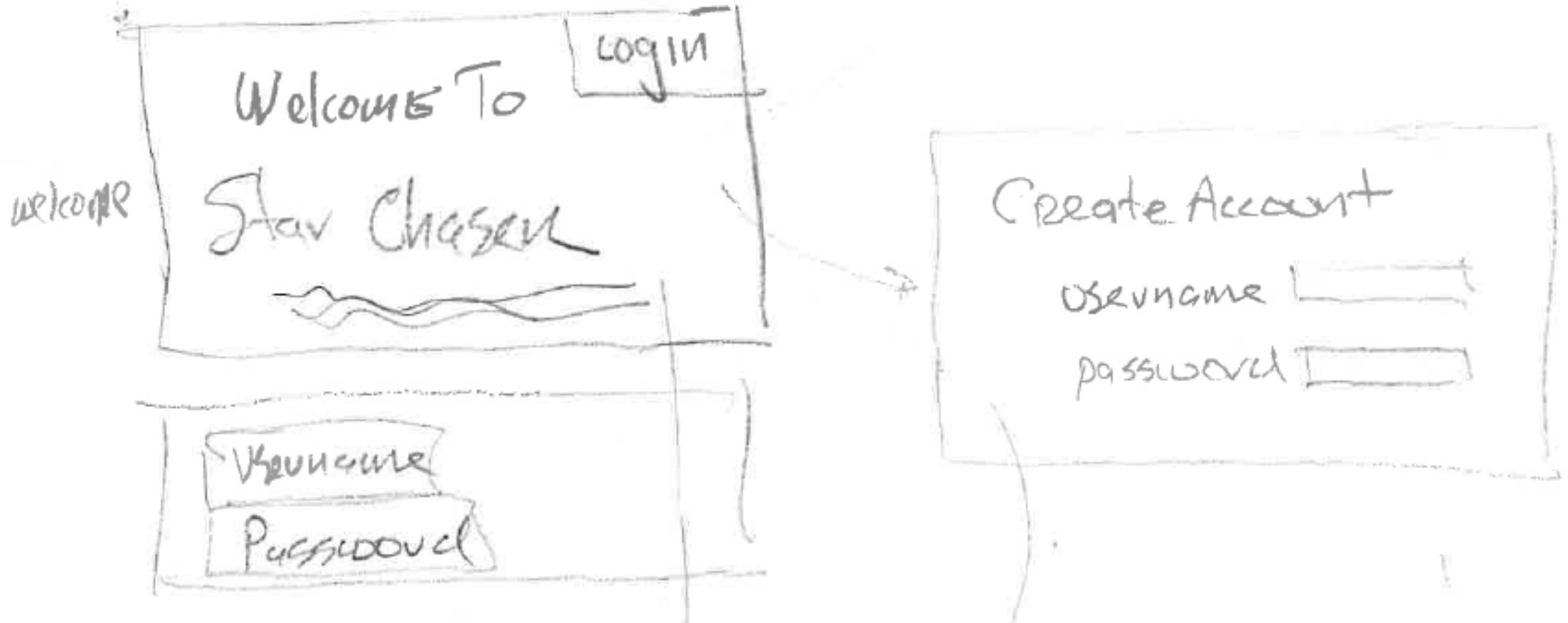Professional Contacts / Network

- 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

```
                              ┌──────────┐
                              │  Login   │
                              └──────────┘
                         ╱                    ╲
              ┌──────────┐                    ┌──────────┐
              │  Admin   │                    │   Home   │
              └──────────┘                    └──────────┘
             ╱            ╲                         │
  ┌──────────────┐  ┌──────────┐          ┌──────────────┐      ┌──────────────┐
  │ Users/Groups │  │ Database │          │ Leaderboard  │      │  Info Panel  │
  └──────────────┘  └──────────┘          └──────────────┘      │              │
                                                 │              │      .       │
                                          ┌──────────────┬─────┐│      .       │
                                          │ Choose a Star │Popup││      .       │
                                          └──────────────┴─────┘│      .       │
                                                 │              │      .       │
                                          ┌──────────────────┐  │      .       │
                                          │ Analyze / Classify│  │      .       │
                                          └──────────────────┘  │      .       │
                                                                 │      .       │
                                                                 └──────────────┘
```

Wireframes

player
(home)

Welcome Alice

Stars You Chased

name	Class	Winnings
star 2	12	$1200
star 3	13	$ 300

Chase A Star

LEADER BOARD

Rank	name	$
1	Alice	13,500
2	Chris	2100
3	Rob	300

Select A Star from the
list or get a [New List]

Choose Wisely!

B. Trotta   K. Boone
P(class)   P(class)

1 . - - - - 13   1 13

(PICK STAR)

1) 210734
302124
3 R1ju
un~
NO ~

Name your Star

["star 1"]

[Help] [Info]

[Chase It]

Help
Info

(Context)

K   D

Help (game) ———→ Pop. up
Info (technology behind the scenes) —→ Pop. up
Info (astronomy) ———— Link / Ref
Info (Kaggle PLAsTiCC) —→ 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
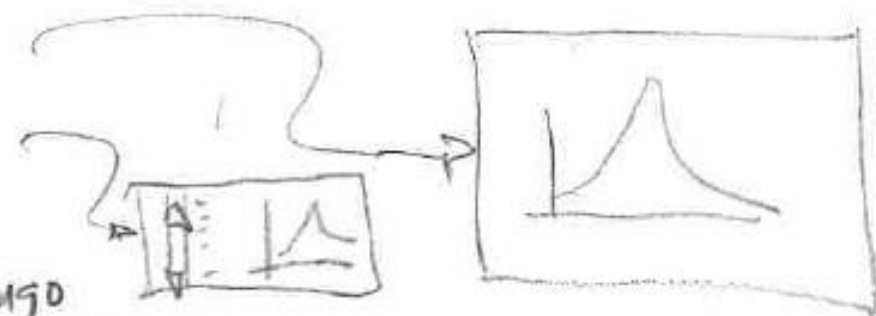
PLACE YOUR BETS!

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

place bets

STAR CLASS

1	2	3	4	5	6	7	8	9	10	11	12	13

Payout

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	20'	.	-	~	100	-	-	-	.	$20

Advice
B. Trotted
K. Boone

Your Bet

| | | | | | 20 | | | | | | |

Results

WINN
er

$$$

$2000

TOTAL WINNINGS : $2080

[Play Again] [Home]

Show the UI...

Implementation…

From last time

Model Template View
Similar to MVC

Request (GET,POST)
Response

Receives Request,
talks to Model/Template,
send Response

url map View

Model

Database Tables/Attributes

Template

Generates HTML (presentation only)
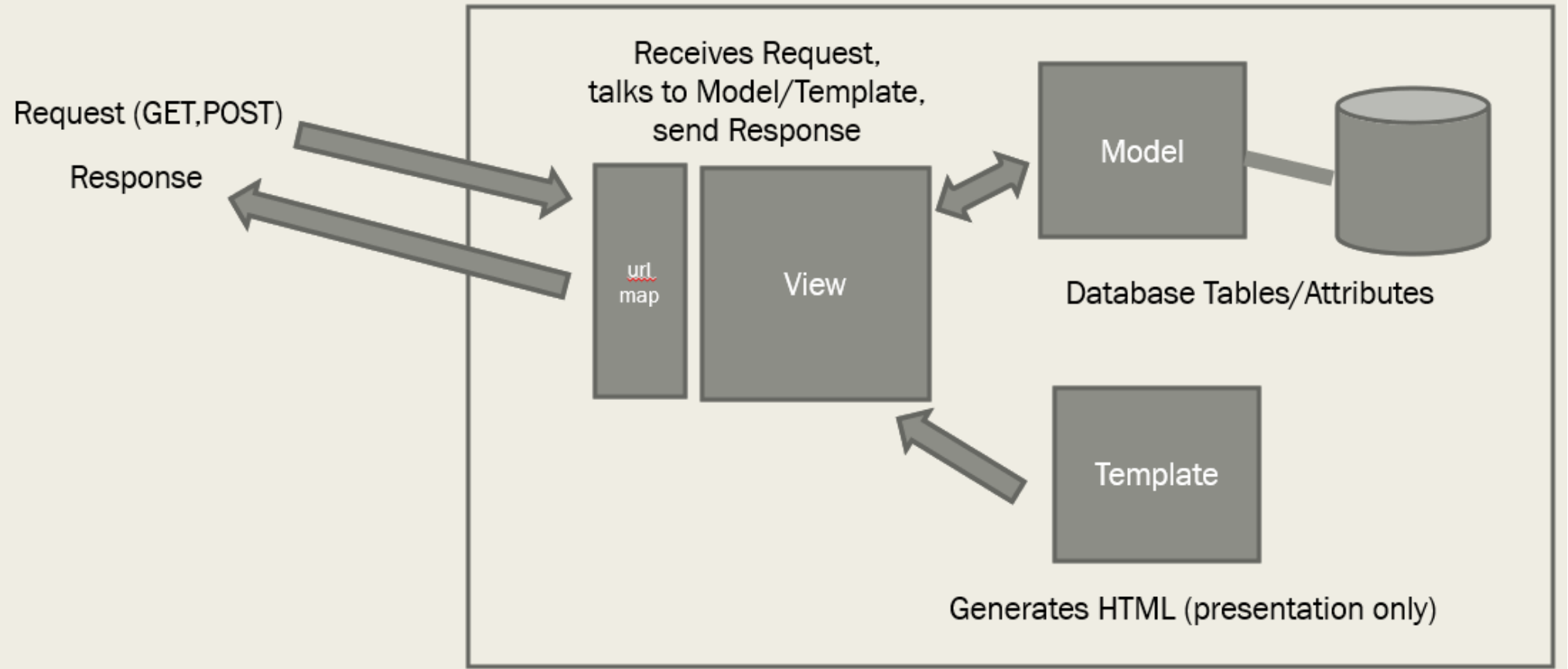
Schema
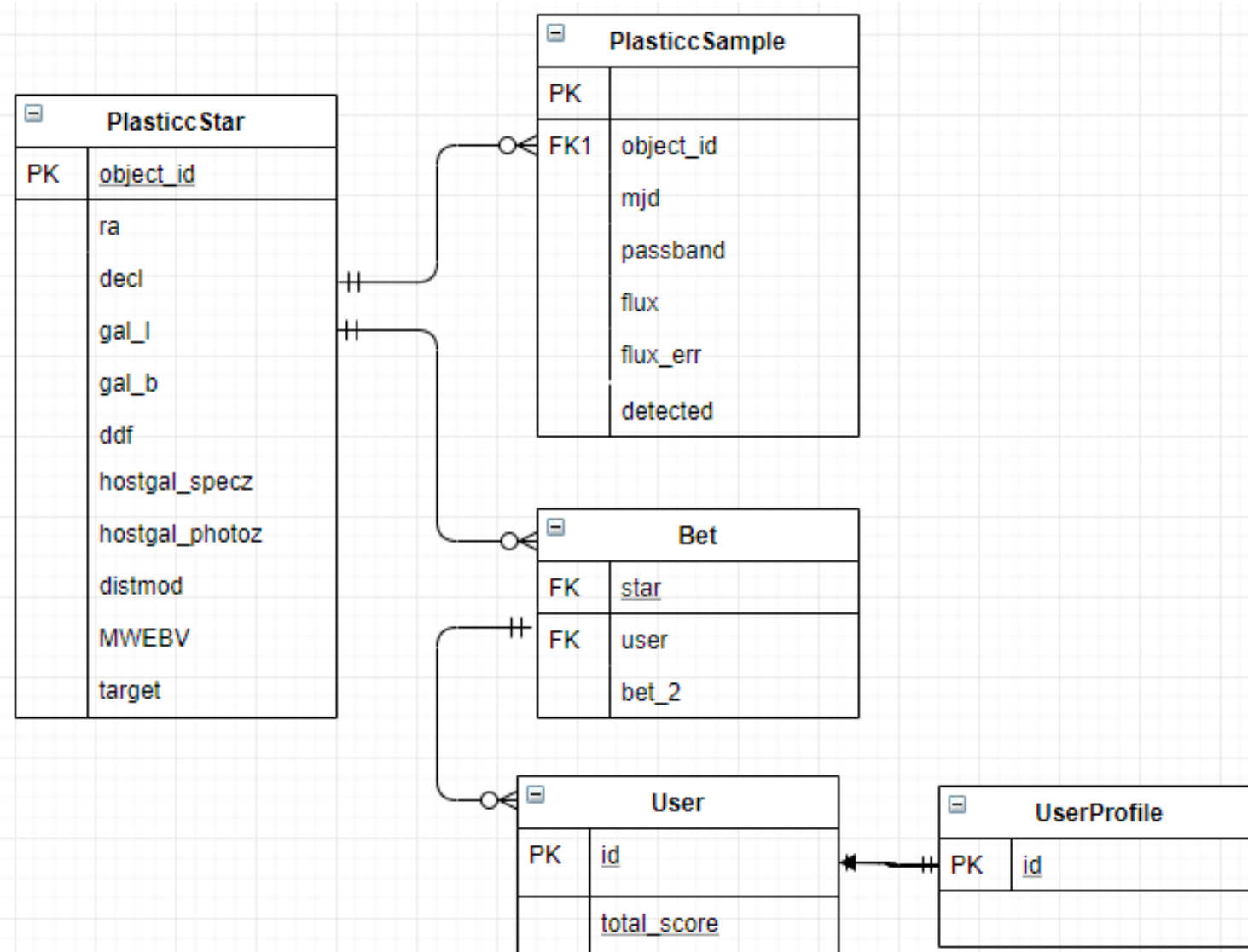
pick_star (view)

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

```python
def pick_star(request):

    # param 1 - list of star IDs comma delimited
    if 'param1' in request.GET:
        starlist_np = np.fromstring(param1, dtype=int, sep=',')
    else:
        starlist_np = PlasticcStar.objects.random_set()

    # param2 = star to display chart
    if 'param2' in request.GET:
        star_to_display = request.GET['param2']
    else:
        star_to_display = ''

    # build comma-delimited string
    starlist_string = ''
    for star_id in starlist_np:
        starlist_string = '{}{},'.format(starlist_string, star_id)

    context = dict()
    context['starlist_string'] = starlist_string
    context['star_to_display'] = star_to_display

    # for charting - get data for chart
    [star_obj, timeseries_data_str] = get_chart_data(star_to_display)
    context['timeseries_data_str'] = timeseries_data_str
    return render(
        request=request,
        template_name="app_player/pick_star.html",
        context=context)
```

pick_star (html)

```html
10    <h1>
11        Choose a star:
12    </h1>
15        <div class="col-sm-4">
16            <h3>Available:</h3>
27                {% for star in starlist_obj %}
30                    <div class="col-sm-4"> {{ star.star_id }} </div>
31                    <div class="col-sm-2"><a
32                        href="{% url 'player_pick_star' %}?param1={{ starlist_string }}&param2={{ star.star_id }}">Chart</a>
33                    </div>
34                    <div class="col-sm-2"><a href="{% url 'player_pick_star' %}">Deal</a></div>
35                    <div class="col-sm-2"><a href="{% url 'place_bet' id=star.star_id %}">Bet</a></div>
39                {% endfor %}
44        <!--Div that will hold the chart-->
45        <div class="col-sm-6">
46            <h3>{{ star_to_display }}</h3>
47            <div id="chart_div_2">
49                <!-- load my charting javascript -->
50                {% if timeseries_data_str %}
51                <script type="text/javascript">
52                    (function () { window.temp5 = {{ timeseries_data_str }} ; }) ();
53                </script>
54                <script src="{% static 'js/chart_timeseries.js' %}"></script>
55                {% endif %}
56            </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

```python
def place_bet(request, id):

    star = get object or 404(PlasticcStar, pk=id)

    if request.method == "POST":
        form = BetForm(data=request.POST)

        if form.is_valid():
            form.save()
            return redirect('player_well_done')
    else:
        df_btrotta = pd.DataFrame(        )
        df_kboone = pd.DataFrame(       )

        form = BetForm()
        bet_form_set_disabled_fields(form, request, star)
        bet_form_set_values_for_non_db_fields(form, request, star)
        bet_form_set_reduction_fields(form, request)


    # for charting - get chart data
    [star_obj, timeseries_data_str] = get_chart_data(id)

    context = {}
    context['form'] = form
    context['star_id'] = star.star_id
    context['star_obj'] = star_obj
    context['timeseries_data_str'] = timeseries_data_str
    return render(
        template_name="app_player/bet_form.html",
        context=context)
```

place_bets (html)

```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>

     (...)

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>
```
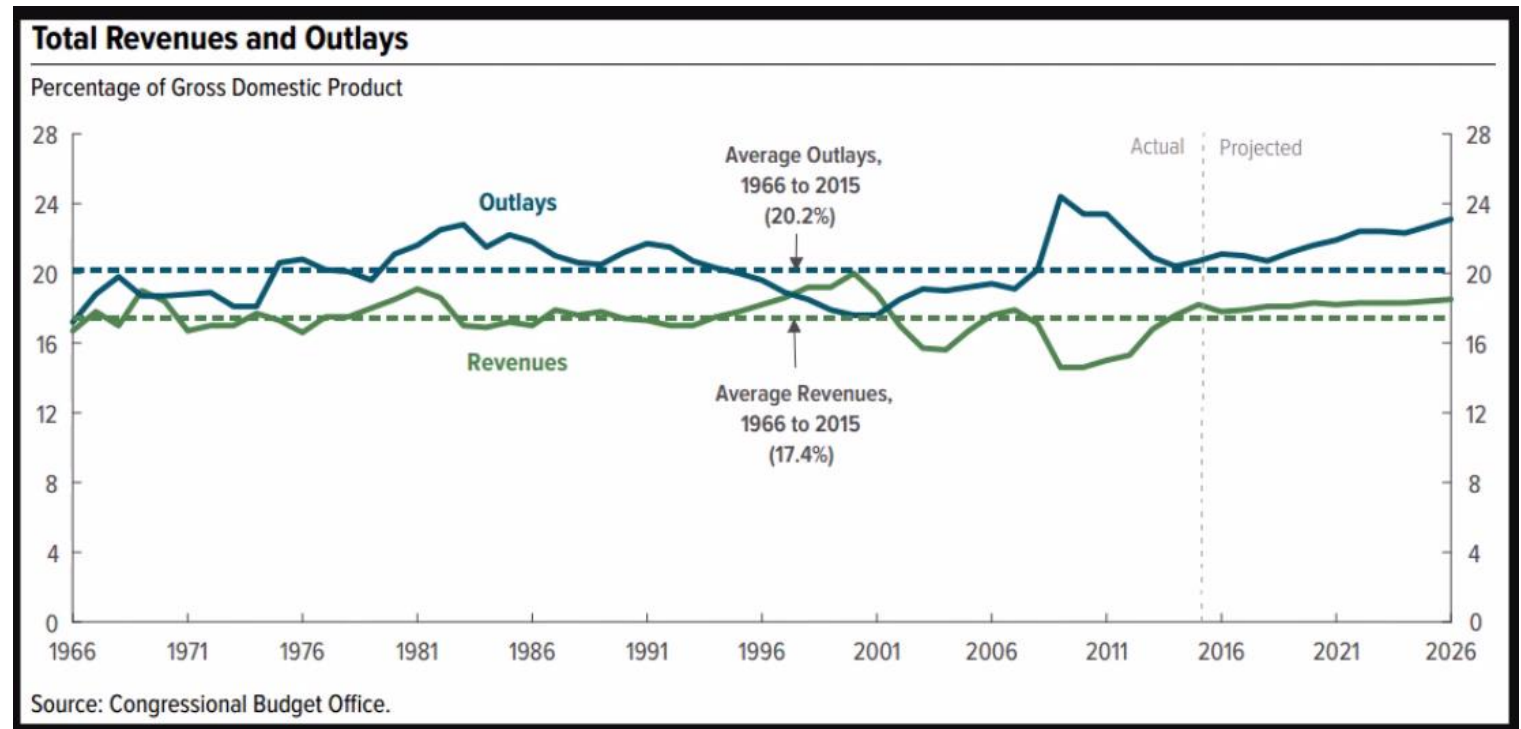
What I did not expect…

"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
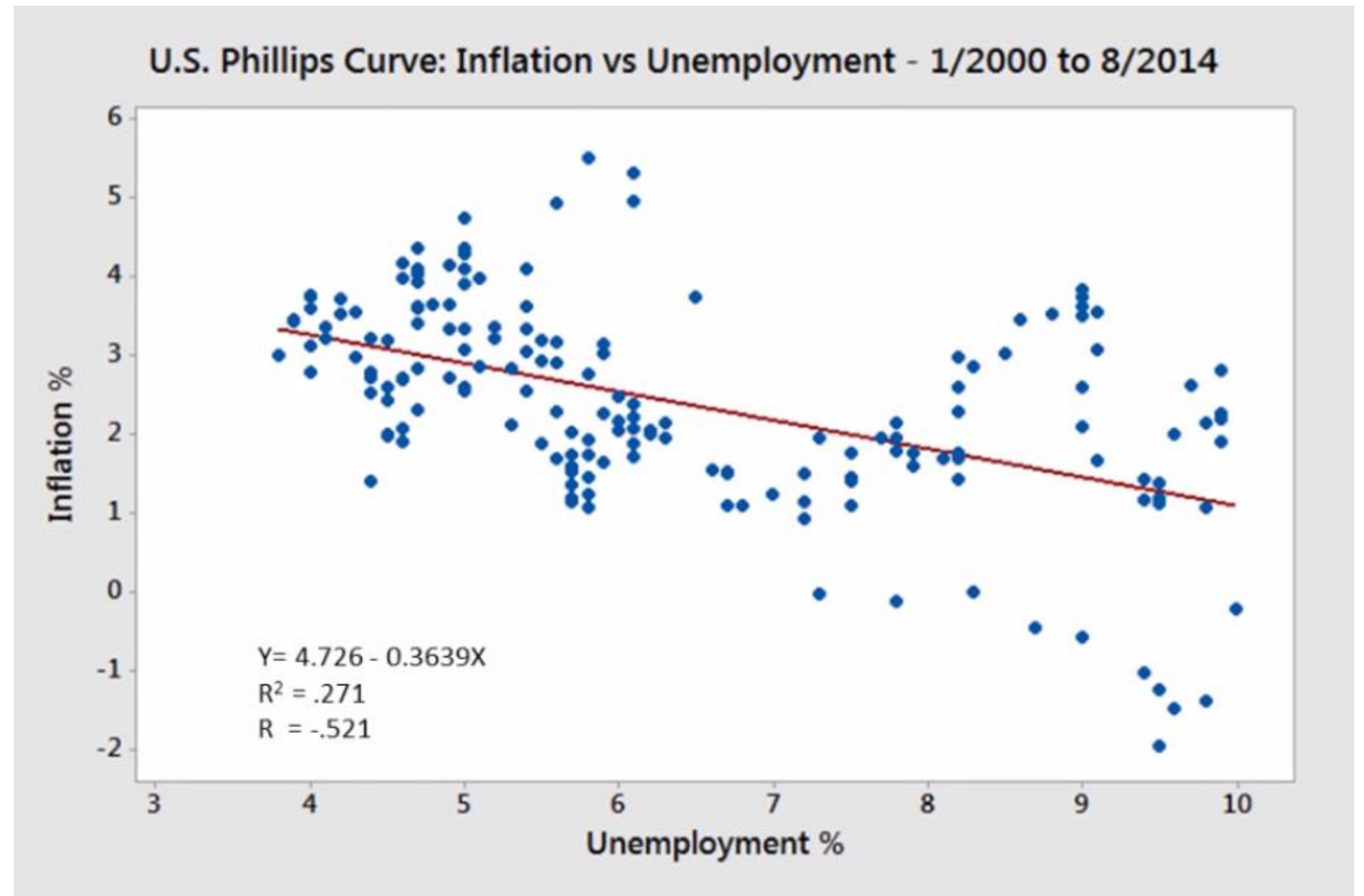- For Web App I went with Google Charts

Total Revenues and Outlays

Percentage of Gross Domestic Product

Source: Congressional Budget Office.

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…

U.S. Phillips Curve: Inflation vs Unemployment - 1/2000 to 8/2014

$Y = 4.726 - 0.3639X$
$R^2 = .271$
$R = -.521$

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!)
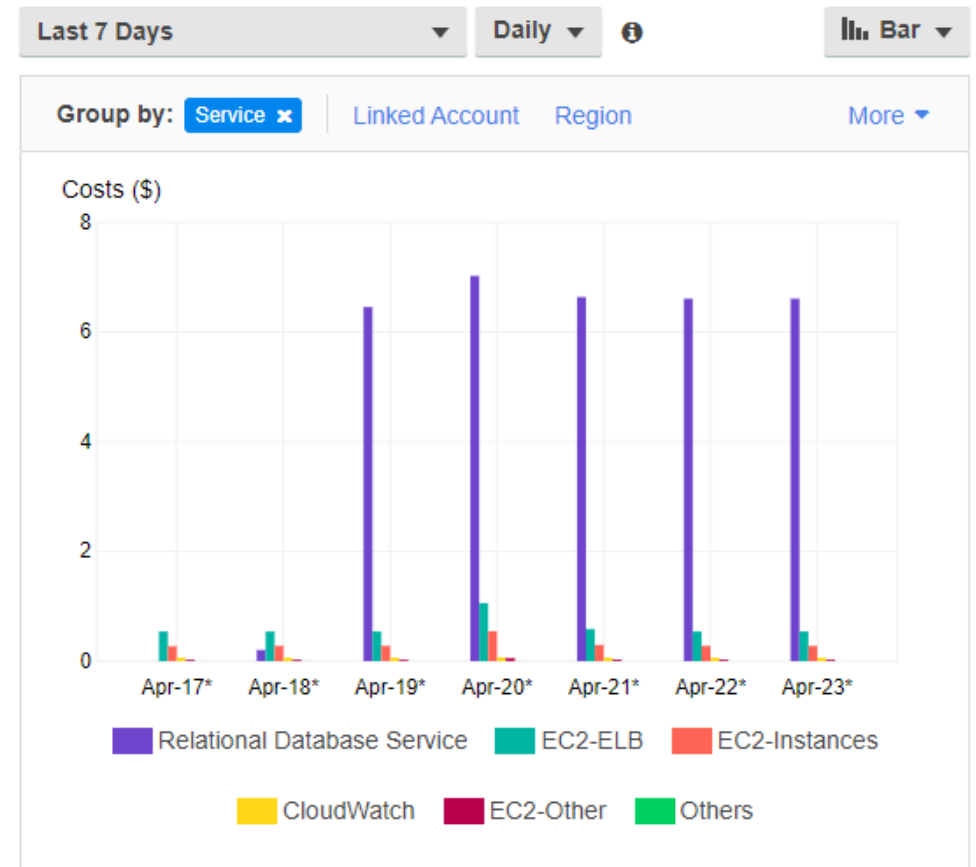
Second Try: Postgres directly on EC2
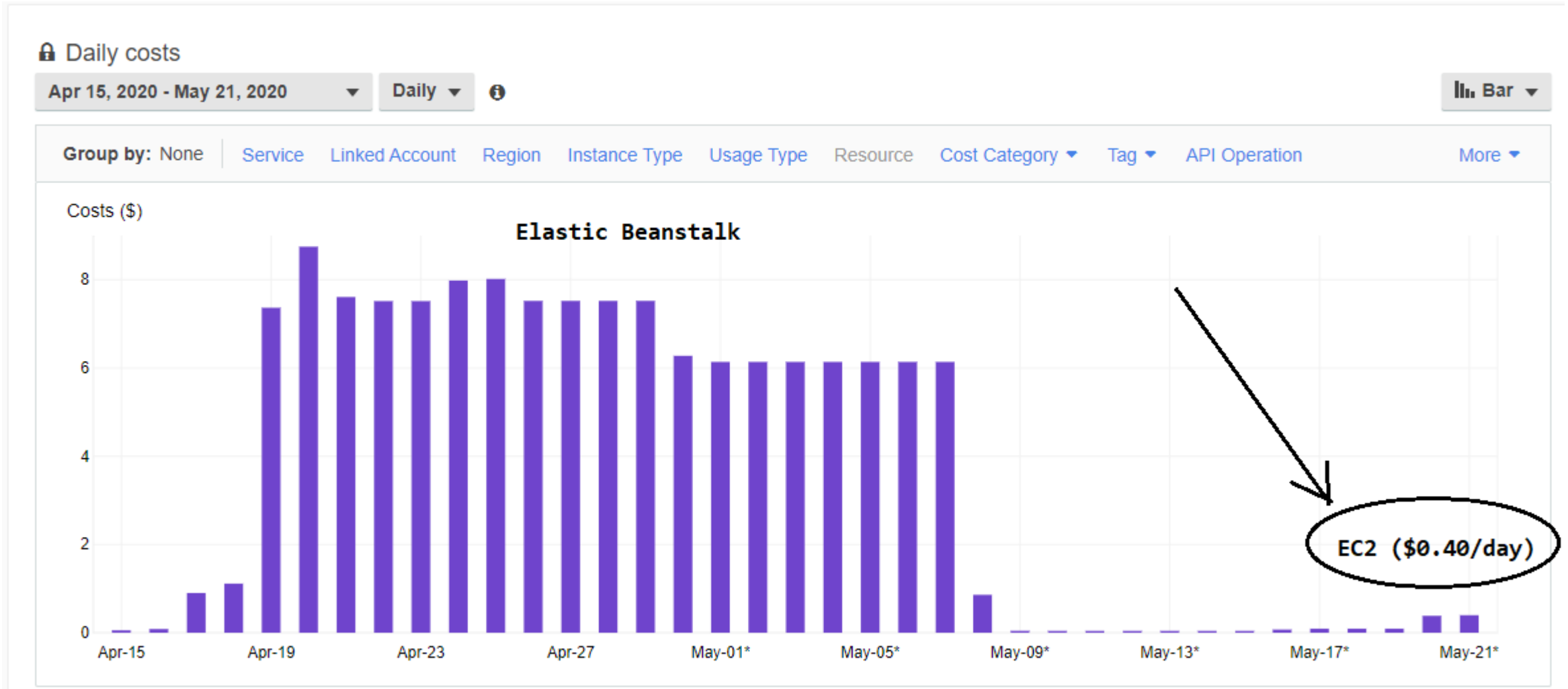
Go Play !

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

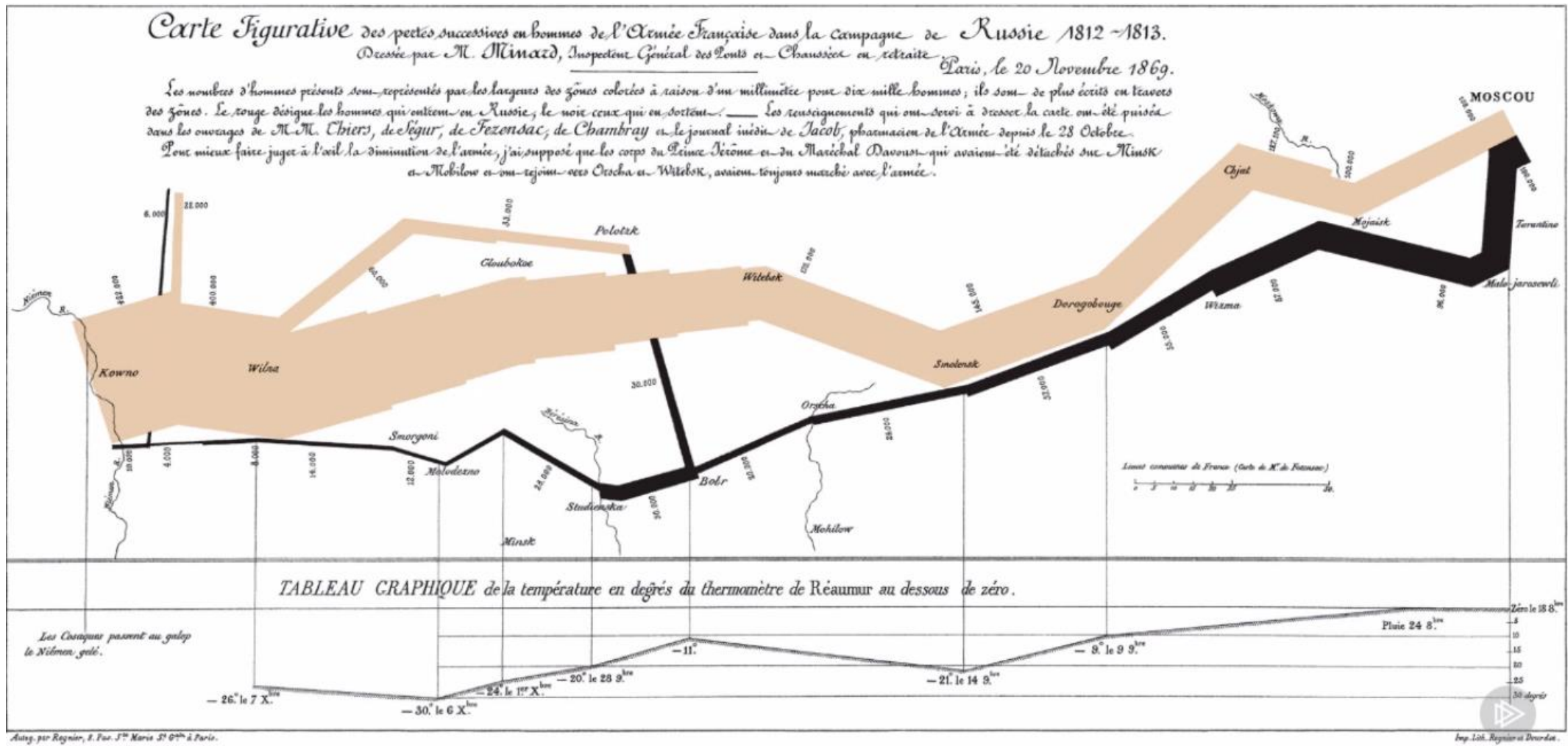# Level of Effort and Takeaways

- 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

- 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/
- Front-end (suspect this will be handy in the future)
  - "Front-End Web Development Quick Start With HTML5, CSS, and JavaScript" (Shawn Wildermuth)

Thank You

Visualization

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