Data Products

Flask, Jinja, Bootstrap & Bokeh

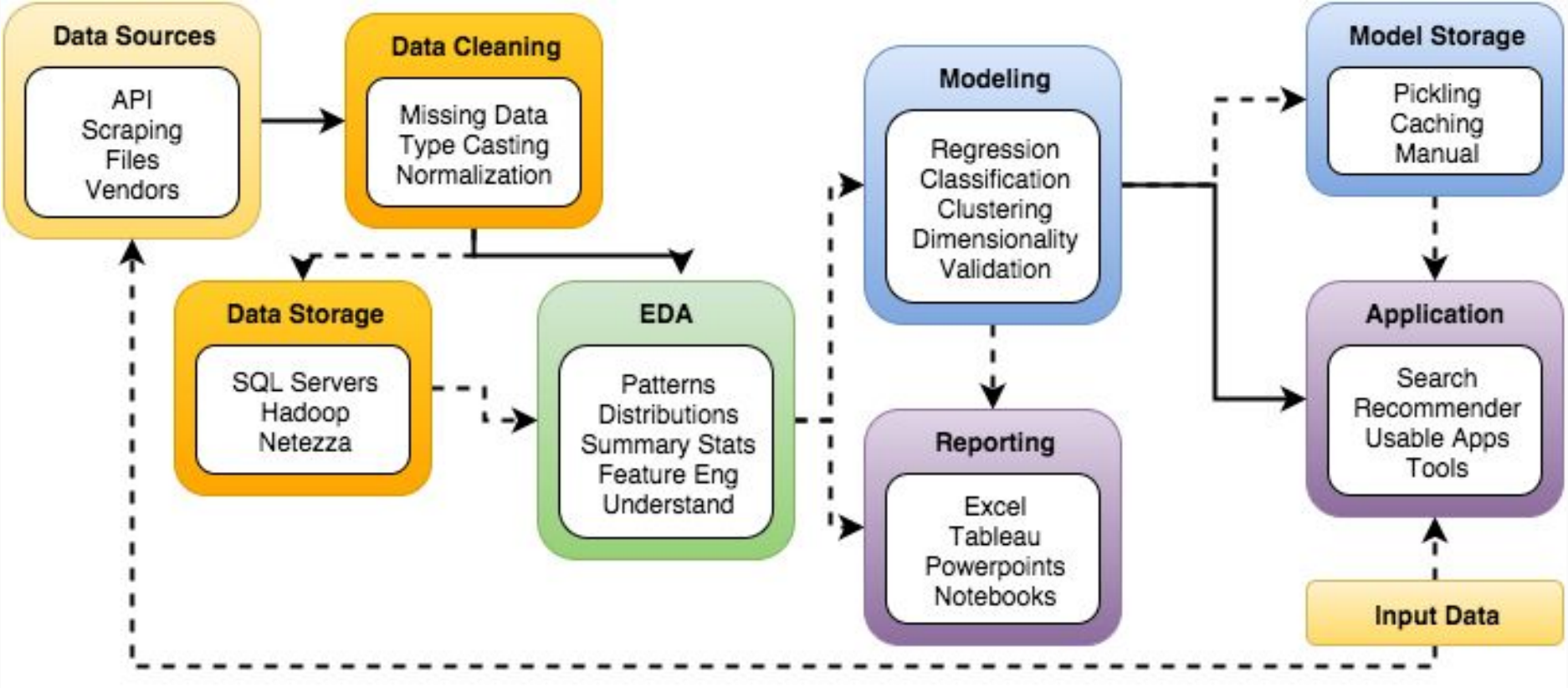
galvanize

Objectives:

- Describe example data product workflows
- Implement simple webpages using HTML and Flask
- Describe the HTTP methods GET and POST & list the differences
- Build a cross-platform, modern website using the Bootstrap framework
- Embed plots in your website using the bokeh package

galvanize

What do you deliver as a data scientist?



Example capstones

galvanize

Furniture seeker (furniture recommender): Lili Yao

Recthetrail.com (trail recommender): Jade Tabony

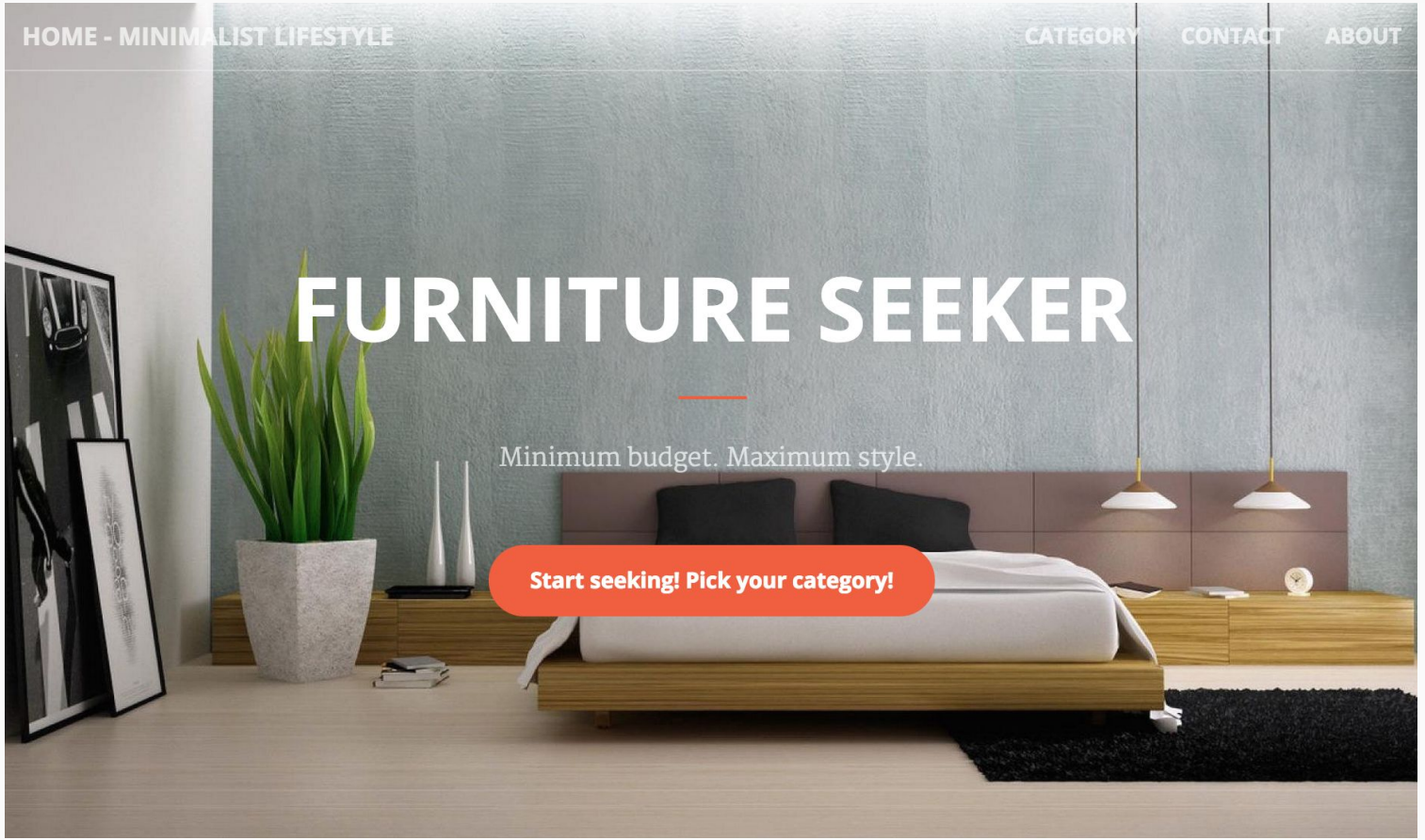
Artorithmia.com (art recommender): Aaron Lichtner

Signedge.tech (street sign classifier): Brian McAdams

Rapid-rescue.com (medical emergency predictor): Nick Buker

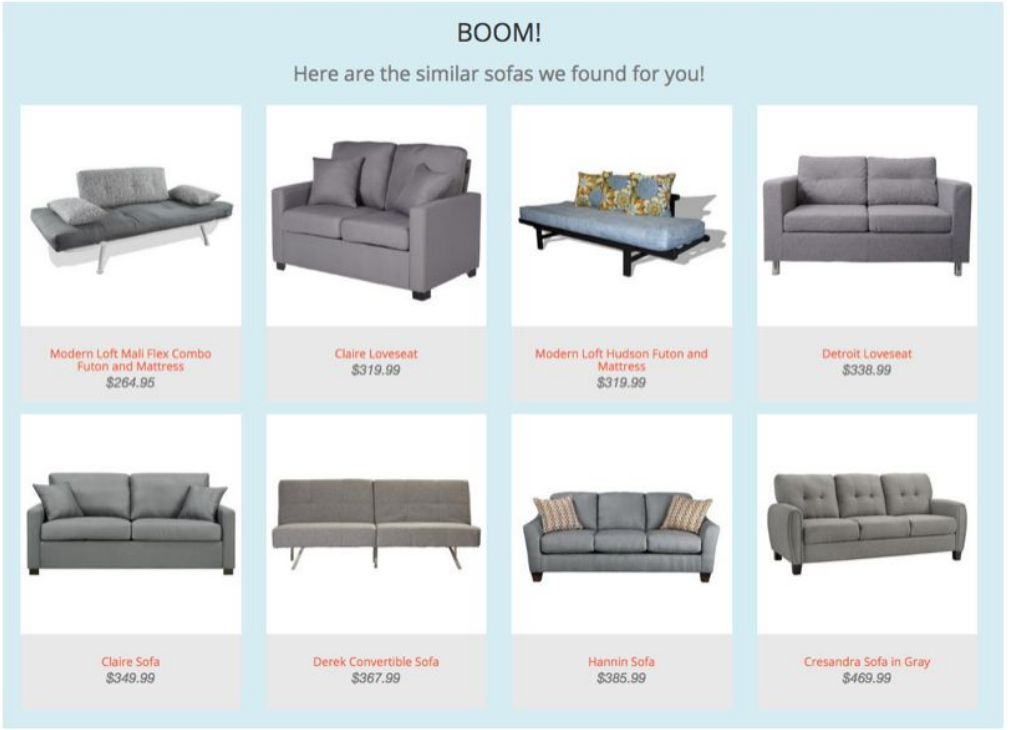
Capstone example (Lily Yao)

galvanize



Capstone example (Lily Yao)

galvanize



galvanize

Applications run in a web browser

Frameworks include:

- Django (yay)
- Ruby on Rails (boo :-))
- Flask (Microframework)
- ...

HTTP protocol review

The Hypertext Transfer Protocol (HTTP) enables communications between clients and servers.

The two most common HTTP methods are:

- GET: Requests data from a server (Default method in http & flask)
- POST: Submits data to server

Other HTTP Request Methods:

- PUT - Updates data on server
- DELETE - Deletes data on server

Important differences: see table at this w3 link

HTML (Hyper Text Markup Language)

- Based on markup tags
- Each tag describes different document entity

CSS (Cascading Style Sheets)

- Describes how HTML is displayed on screen
- Assigns style properties to (sections of) your site
- Can control the layout of multiple web pages all at once

Your main reference today is W3 Schools. HTML, CSS, JavaScript, BootStrap, ...

HTML example

galvanize

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Page Title</title>
    </head>
    <body>
    <!-- page content -->
        <h1>My Page</h1>
        <p>
            All the things I want to say.
        </p>
        <p style="color: purple; text-align: right;">
            My right-aligned purple text.
        </p>
    </body>
</html>
```

Inline CSS in “style” attribute will overwrite class and element styles from CSS files

- Flask is a microframework
- Aims to keep the core simple
- Flask won't make many decisions for you
- Decisions that it does make are easy to change
- Tries not to implement components that are available through other libraries
- Uses template language Jinja2 (similar to the Django template engine but provides Python-like expressions)

Install:

```
pip install flask

pip install Jinja2
```

Simple Flask app

galvanize

```
1  from flask import Flask
2  app = Flask(__name__)
3
4  # home page
5  @app.route('/')
6  def index():
7      return '''
8          <!DOCTYPE html>
9          <html>
10             <head>
11                 <meta charset="utf-8">
12                 <title>Page Title</title>
13             </head>
14           <body>
15             <!-- page content -->
16             <h1>My Page</h1>
17             <p>
18                 All the things I want to say.
19             </p>
20           </body>
21         </html>
22         '''
23
24 if __name__ == '__main__':
25     app.run(host='0.0.0.0', port=8080, debug=True)
26
```

At the IP “0.0.0.0” on port 8080 (port 80 is default for HTTP, port 8080 is common alternative for testing purposes)

At the root of the website (/) it will serve the index page defined by the html displayed in the index() function

Open in browser 'http://localhost:8080/' or 'http://0.0.0.0:8080/'

galvanize

Routing is binding URLs to Python functions.

URL can contain variables (they are passed to bound function).

```
@app.route('/user/<username>')
def show_user_profile(username):
    # show the user profile for that user
    return 'User %s' % username

@app.route('/post/<int:post_id>')
def show_post(post_id):
    # show the post with the given id, the id is an integer
    return 'Post %d' % post_id
```

URLs can then be generated for each function with url_for(fct_name, **kwargs)

```
26
27  url_for('index')
28  url_for('login')
29  url_for('login', next='/')
30  url_for('profile', username='John Doe')
```

By default, a route only answers to GET requests, but you can add the 'methods' argument to the route() call.

```
@app.route('/path', methods=['GET', 'POST'])
```

Templates

galvanize

Templates allow us to dynamically generate HTML files from text files (instead of dealing with python strings)

```python
from flask import Flask, render_template
from random import random
app = Flask(__name__)


@app.route('/')
def index():
    n = 100
    x = range(n)
    y = [random() for i in x]
    return render_template('table.html', data=zip(x, y))


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080, debug=True)

```

Templates: Dynamic HTML generation

galvanize

```
<table border="1">
  <thead>
    <th>x</th>
    <th>y</th>
  </thead>
  <tbody>
    {% for x, y in data %}  <!--start for loop over variable data-->
      <tr>                  <!-- on each row -->
        <td>{{ x }}</td>    <!-- write variable x -->
        <td>{{ y }}</td>    <!-- write variable y -->
    </tr>                   <!-- end the row -->
    {% endfor %}            <!-- end for loop -->
</tbody>
</table>
```

Template rendering with variables

galvanize

Method `flask.render_template(template_name_or_list, context)` accepts context – the variables that should be available in the template.

- render_template('table.html', data=zip(x, y))
- render_template('hello.html', name=name)

From inside templates you can access request and session objects

- request.form['username']
- request.args.get('key', '') #like dict.get()
- request.cookies.get('username')
- session['username']

Bootstrap is a popular front-end web framework combining HTML, CSS, & JavaScript.

- Easy way to develop modern web pages
- Cross-platform, including mobile (responsive)
- Downloadable templates available at startbootstrap.com (and others)
- High quality results
- Free & open source

Twitter Bootstrap: Getting started

galvanize

Start Bootstrap is resource with free Bootstrap themes and templates.

- Download a theme from startbootstrap.com & unzip
- You can start with bare template
- Match the file structure to Flask:
  - Move the js, css, and fonts to 'static' folder
  - Move .html files to 'templates' folder
- Create flask application file .py
- Edit content in .html template files
- Run application
- Use the same .html template for all pages
- Don't forget to add routes and links to connect all new pages

Bokeh is a python library to create interactive plots.

- Display your data in a more pleasing way than a static image
- Update charts easily
- Users can interact with your charts
- Install using `conda install bokeh` with all the dependencies that Bokeh needs
- If you have installed all dependencies you can use `pip install bokeh`. (It does not install the examples)

Bokeh

galvanize

To use Bokeh in our website we need to use the respective css and js libraries:

```
<link rel="stylesheet" href="http://cdn.pydata.org /bokeh/release/bokeh-0.11.1.min.css" type="text/css" />
<script type="text/javascript" src="http://cdn.pydata.org /bokeh/release/bokeh-0.11.1.min.js"></script>
```

In your Python app:

```
from bokeh.plotting import figure
p = figure(**kwargs)
```

Bokeh produces embeddable JavaScript that will render plot:

```
from bokeh.embed import components
script, div = components(plot)
return render_template('dashboard.html', script=script, div=div)
```

Add plot to template: {{ script | safe }} {{ div | safe }} (This means “don’t escape”)

Resources: Bokeh Quickstart, Bokeh Embedding

Interactive plotting libraries

galvanize

Bokeh

D3.js

Morris.js

Plot.ly

And more

Don't steal content:

- Plenty of free-to-use images are available.
  - Google search options: filter images by usage rights
  - Flickr: license options in search
  - https://unsplash.com/ and similar services
- Give your sources credit!

Hosting

galvanize

Free options include:

- PythonAnywhere
- Heroku

Paid options include:

- AWS (free with credits)
- DigitalOcean

Django vs Flask vs Pyramid

galvanize

Discussion on different Python web frameworks

https://www.airpair.com/python/posts/django-flask-pyramid