The Doghouse Project

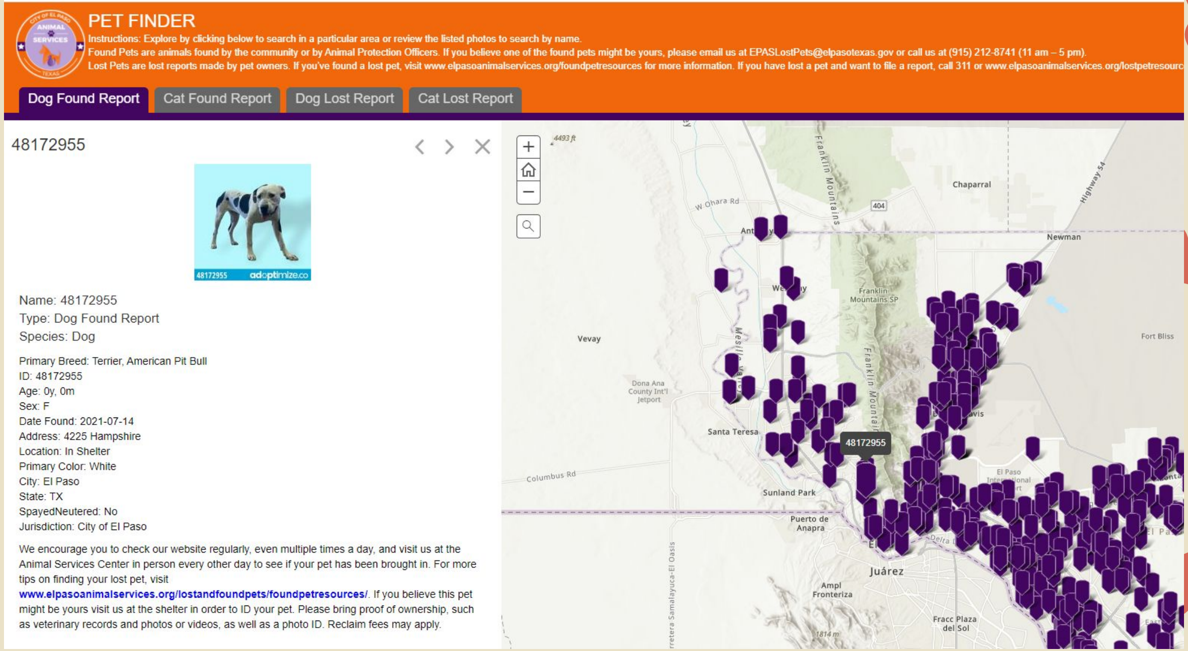
Leaflet Interactive Map

Who's in the Doghouse:
Alex, Amy, Beth, Jeanna, and Jordan

Our Inspiration…Jordan's Doggos

Additional Inspiration

- We were inspired by El Paso, Texas city's page that has an interactive Lost Pet Finder map
- The map includes markers that show where pets were found
- The pets are identified using unique IDs
- The markers include pop-ups with information for each pet including: picture, type, breed, color, etc



PET FINDER

Instructions: Explore by clicking below to search in a particular area or review the listed photos to search by name.

Found Pets are animals found by the community or by Animal Protection Officers. If you believe one of the found pets might be yours, please email us at EPASLostPets@elpasotexas.gov or call us at (915) 212-8741 (11 am – 5 pm).

Lost Pets are lost reports made by pet owners. If you've found a lost pet, visit www.elpasoanimalservices.org/foundpetresources for more information. If you have lost a pet and want to file a report, call 311 or www.elpasoanimalservices.org/lostpetresourc

Name: 48172955
Type: Dog Found Report
Species: Dog

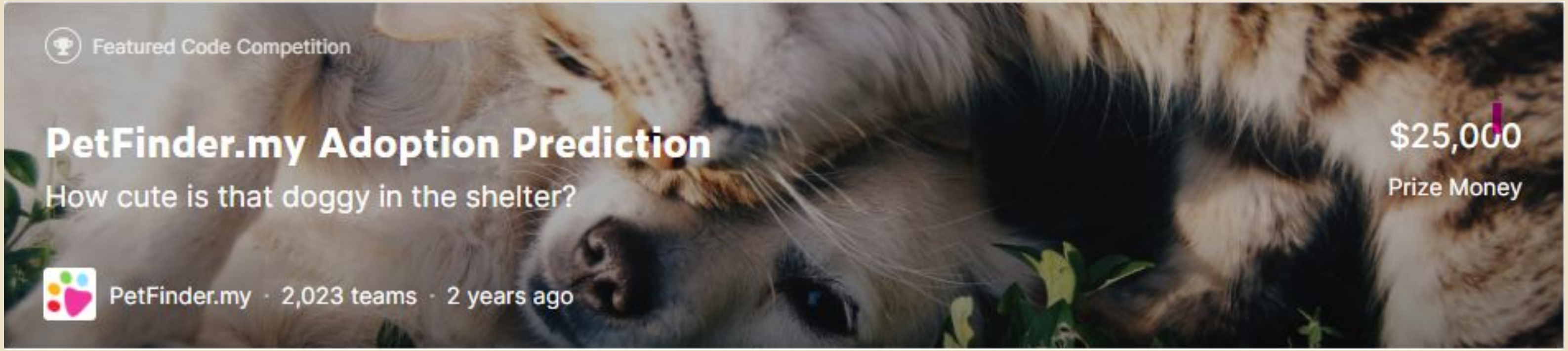
Primary Breed: Terrier, American Pit Bull
ID: 48172955
Age: 0y, 0m
Sex: F
Date Found: 2021-07-14
Address: 4225 Hampshire
Location: In Shelter
Primary Color: White
City: El Paso
State: TX
SpayedNeutered: No
Jurisdiction: City of El Paso

We encourage you to check our website regularly, even multiple times a day, and visit us at the Animal Services Center in person every other day to see if your pet has been brought in. For more tips on finding your lost pet, visit www.elpasoanimalservices.org/lostandfoundpets/foundpetresources/. If you believe this pet might be yours visit us at the shelter in order to ID your pet. Please bring proof of ownership, such as veterinary records and photos or videos, as well as a photo ID. Reclaim fees may apply.

Source link

$$$ Final Inspiration $$$

- When searching Kaggle for datasets we found a competition where a group used Petfinder API to collect pet adoption data.
- This group is in the running to win $25k!!!



Featured Code Competition

PetFinder.my Adoption Prediction

How cute is that doggy in the shelter?

PetFinder.my · 2,023 teams · 2 years ago

$25,000

Prize Money

Source

Project Process:

I

Data Sources

II

ETL

III

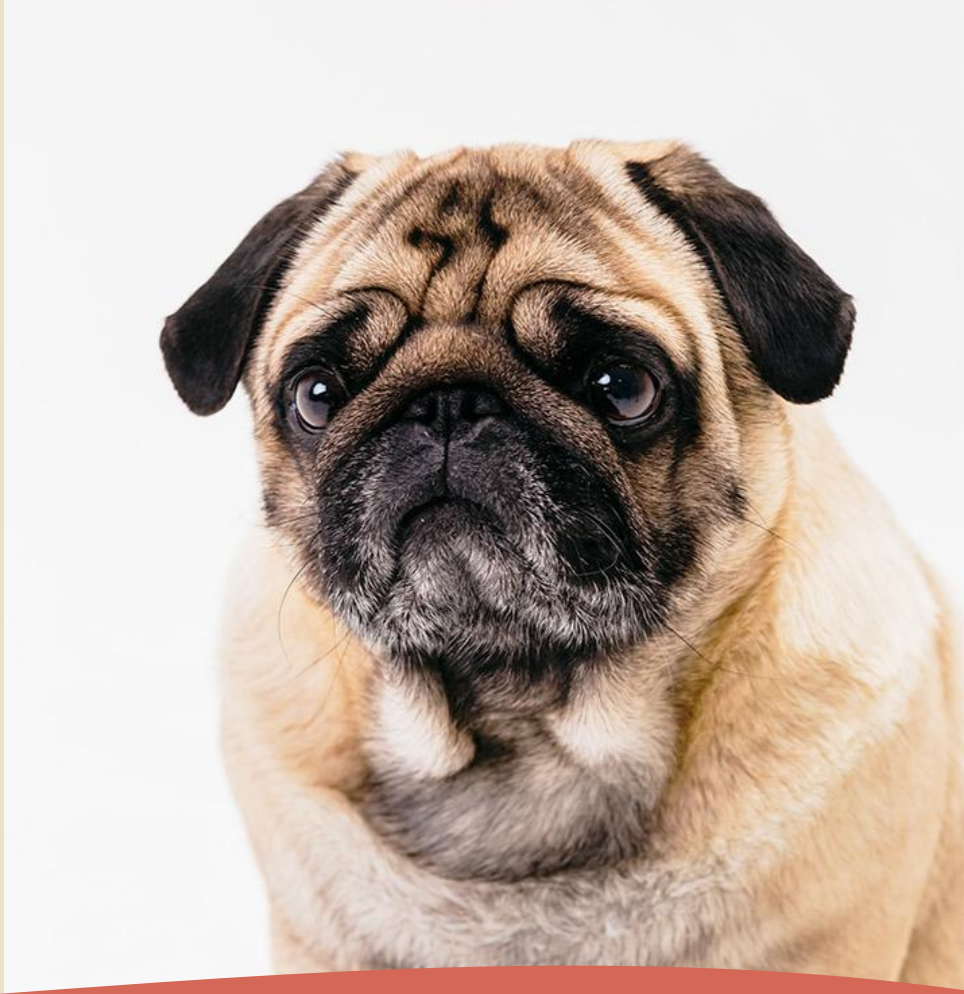
JS and Leaflet

IIII

Final Display

Data Sources

- Used CSV for US Capital Lat Longs from a GitHub repo
- Petfinder API
  - Used to pull data for available dogs available for adoption
  - Utilizes both a standard API Key and a secret key
- Petpy
  - Python library used to interact with the Petfinder API Key and secret key



ETL

Extract

Performed API calls and loaded CSVs

Transform

Cleaned data, dropped columns, replaced null values, created new columns, updated headers, reformatting for Geocoding

Load

Used PostgreSQL to load and store the data

Exercise Time

Amy moved and Alex got a new (free) bike

Puppy Time

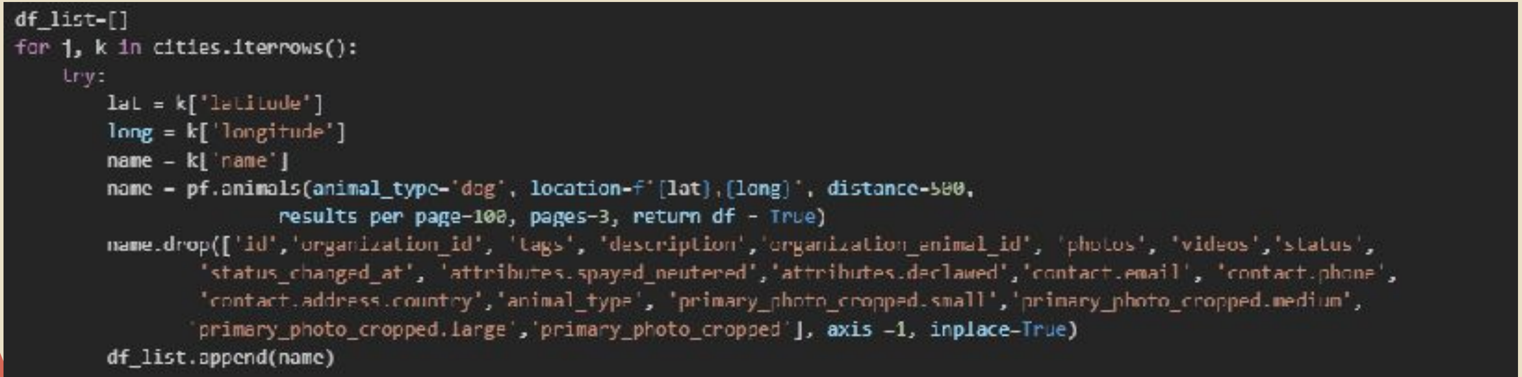
Jordan played with the pups

Play Time

Beth went to Hilton Head Island and Jeanna went camping

Extract and Transform

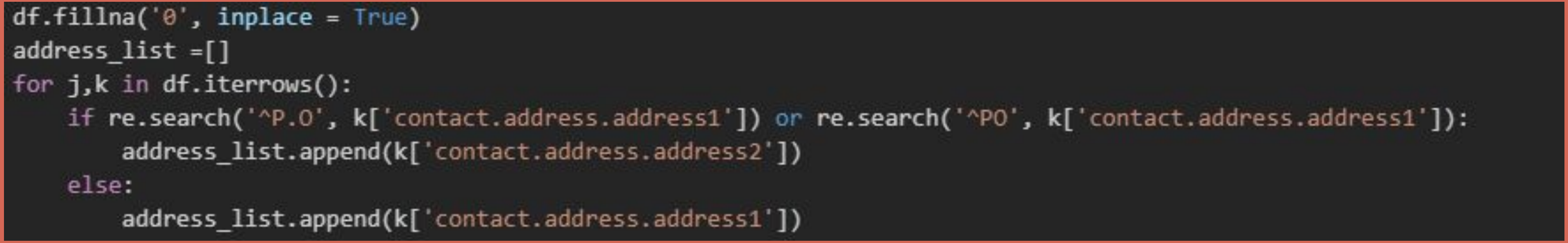
- Performed 50 Petfinder API calls and returned 50 separate dataframes using Latitude and Longitude to search City names
- Dropped unnecessary columns
- Stored dataframes in lists using city name for each dataframe
- Concatenated lists of dataframes to form one giant dataframe with ~14k rows of data



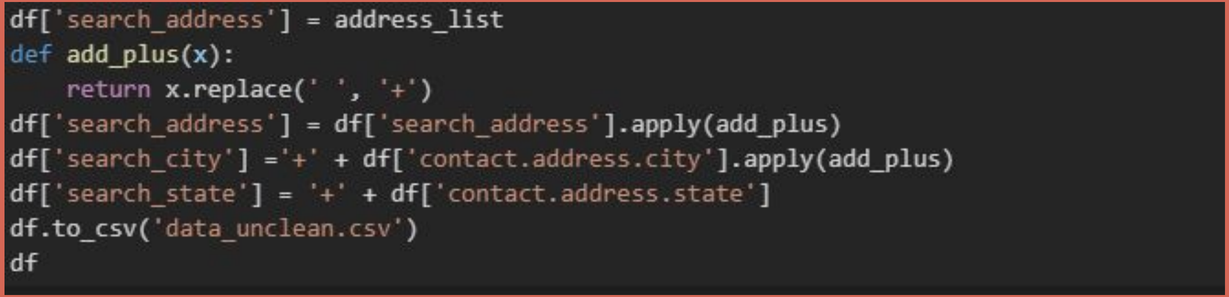
```
df_list=[]
for j, k in cities.iterrows():
    try:
        lat = k['latitude']
        long = k['longitude']
        name = k['name']
        name = pf.animals(animal_type='dog', location=f'{lat},{long}', distance=500,
                          results_per_page=100, pages=3, return_df = True)
        name.drop(['id','organization_id', 'tags', 'description','organization_animal_id', 'photos', 'videos','status',
                   'status_changed_at', 'attributes.spayed_neutered','attributes.declawed','contact.email', 'contact.phone',
                   'contact.address.country','animal_type', 'primary_photo_cropped.small','primary_photo_cropped.medium',
                  'primary_photo_cropped.large','primary_photo_cropped'], axis =1, inplace=True)
        df_list.append(name)
```

Extract and Transform

- Looped through/created addresses and replaced P.O. Boxes with “address2”
- Looped through new addresses, converted addresses to the appropriate Google Geocode API format

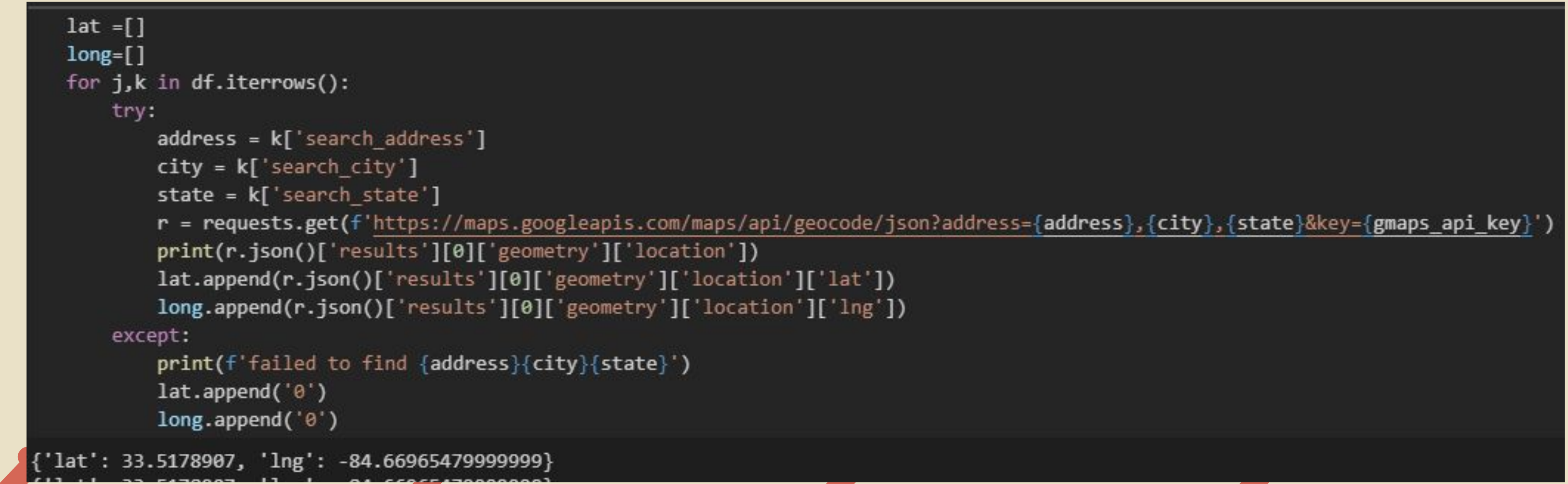


```python
df.fillna('0', inplace = True)
address_list =[]
for j,k in df.iterrows():
    if re.search('^P.O', k['contact.address.address1']) or re.search('^PO', k['contact.address.address1']):
        address_list.append(k['contact.address.address2'])
    else:
        address_list.append(k['contact.address.address1'])
```



```python
df['search_address'] = address_list
def add_plus(x):
    return x.replace(' ', '+')
df['search_address'] = df['search_address'].apply(add_plus)
df['search_city'] ='+' + df['contact.address.city'].apply(add_plus)
df['search_state'] = '+' + df['contact.address.state']
df.to_csv('data_unclean.csv')
df
```

Extract and Transform... again



```
lat =[]
long=[]
for j,k in df.iterrows():
    try:
        address = k['search_address']
        city = k['search_city']
        state = k['search_state']
        r = requests.get(f'https://maps.googleapis.com/maps/api/geocode/json?address={address},{city},{state}&key={gmaps_api_key}')
        print(r.json()['results'][0]['geometry']['location'])
        lat.append(r.json()['results'][0]['geometry']['location']['lat'])
        long.append(r.json()['results'][0]['geometry']['location']['lng'])
    except:
        print(f'failed to find {address}{city}{state}')
        lat.append('0')
        long.append('0')
```

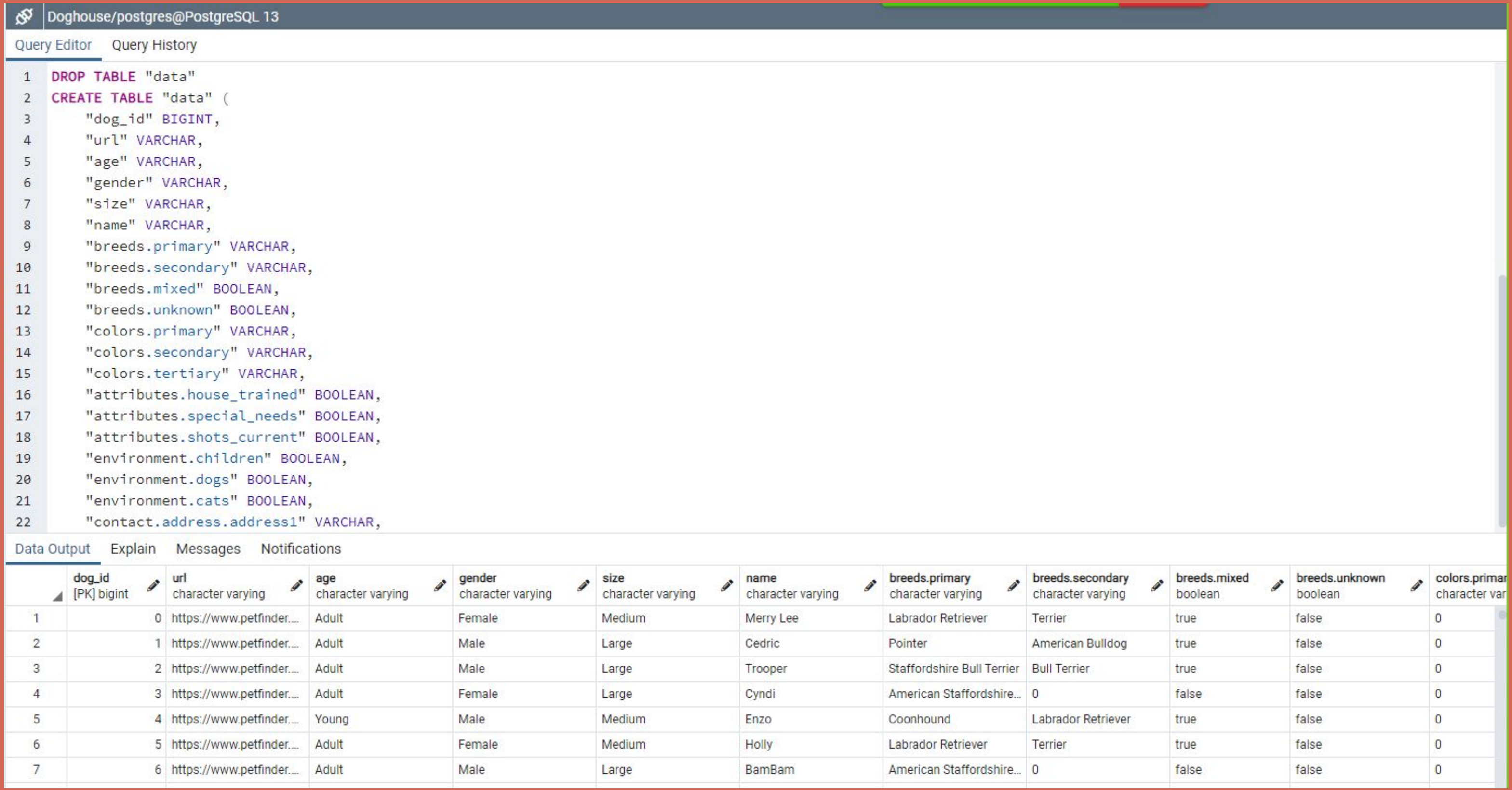
```
{'lat': 33.5178907, 'lng': -84.66965479999999}
```

ETL: LOAD

- Read CSV using Pandas
- Dropped unnecessary columns… again!
- Created database and table schema in PostgreSQL
- Created connection in JupyterNotebook using Pandas to load data to DB

Primarily used the DB to warehouse the data.

ETL: LOAD continued…



```
DROP TABLE "data"
CREATE TABLE "data" (
    "dog_id" BIGINT,
    "url" VARCHAR,
    "age" VARCHAR,
    "gender" VARCHAR,
    "size" VARCHAR,
    "name" VARCHAR,
    "breeds.primary" VARCHAR,
    "breeds.secondary" VARCHAR,
    "breeds.mixed" BOOLEAN,
    "breeds.unknown" BOOLEAN,
    "colors.primary" VARCHAR,
    "colors.secondary" VARCHAR,
    "colors.tertiary" VARCHAR,
    "attributes.house_trained" BOOLEAN,
    "attributes.special_needs" BOOLEAN,
    "attributes.shots_current" BOOLEAN,
    "environment.children" BOOLEAN,
    "environment.dogs" BOOLEAN,
    "environment.cats" BOOLEAN,
    "contact.address.address1" VARCHAR,
```

	dog_id [PK] bigint	url character varying	age character varying	gender character varying	size character varying	name character varying	breeds.primary character varying	breeds.secondary character varying	breeds.mixed boolean	breeds.unknown boolean	colors.primar character var
1	0	https://www.petfinder....	Adult	Female	Medium	Merry Lee	Labrador Retriever	Terrier	true	false	0
2	1	https://www.petfinder....	Adult	Male	Large	Cedric	Pointer	American Bulldog	true	false	0
3	2	https://www.petfinder....	Adult	Male	Large	Trooper	Staffordshire Bull Terrier	Bull Terrier	true	false	0
4	3	https://www.petfinder....	Adult	Female	Large	Cyndi	American Staffordshire...	0	false	false	0
5	4	https://www.petfinder....	Young	Male	Medium	Enzo	Coonhound	Labrador Retriever	true	false	0
6	5	https://www.petfinder....	Adult	Female	Medium	Holly	Labrador Retriever	Terrier	true	false	0
7	6	https://www.petfinder....	Adult	Male	Large	BamBam	American Staffordshire...	0	false	false	0

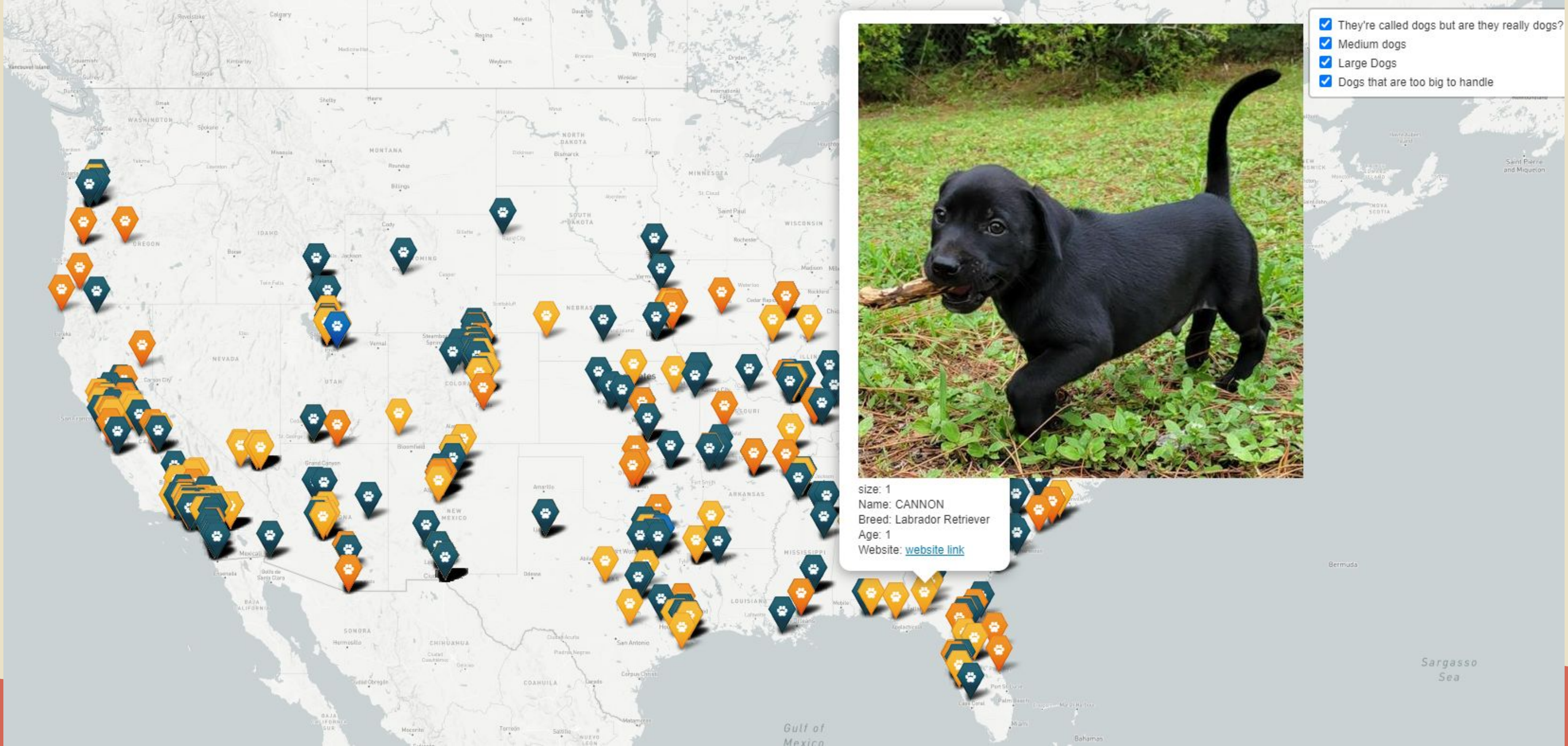
III

JavaScript and Leaflet

JavaScript and Leaflet

- Created a base map using mapbox
- Initialized the layers for dog sizes
- Used leaflet control function to display the clickable layers
- Created specialized icons using the ExtraMarker library
- Stored the clean Petfinder data in a Javascript file that we called in the index file
- Looped through the data and created markers
- Assigned markers to the layer groups
- Assigned the layer groups to the map
- Finally we searched our interactive map to find our new, adorable BFF!

Final Display



☑ They're called dogs but are they really dogs?
☑ Medium dogs
☑ Large Dogs
☑ Dogs that are too big to handle

size: 1
Name: CANNON
Breed: Labrador Retriever
Age: 1
Website: website link

Pros and Cons

Pros

- Visually appealing
- Consumer driven final output
- Petpy library saved the day
- Comfortable using python for the bulk of the ETL
- Customizing the map markers

Cons

- API call specifications were challenging
- Data lacked lat longs before Geocoding
- Extensive data cleaning
- Dataset was too big to run on our computers and could only use a test set
- Had to conform cleaned data CSV to JSON format for Leaflet

Conclusion

The Doghouse Project can help people find their 4-legged bestfriends!

Thank you.

Questions?

Resources

Data

- Petfinder API Dogs for Adoption
  - https://www.petfinder.com/developers/
- US Capitals Lat Long Data
  - https://github.com/jasperdebie/VisInfo/blob/master/us-state-capitals.csv

Google Slide Template

- https://slidesgo.com/theme/indoor-games-for-dogs#search-dog&position-0&results-5
  - Stock images and slide formatting