

## CS 340 7-2 Project Two README

### About the Project/Project Title:

#### Grazioso Salvare Rescue Dashboard:

Is a Jupyter Dash dashboard that connects to the Austin Animal Center (AAC) MongoDB dataset via our reusable CRUD module (aac\_crud.py). It helps Grazioso Salvare quickly identify matching animals across different categories including breed, sex, and age profiles for the three required rescue types (Water, Mountain/Wilderness, Disaster). The UI utilized interactive radio filters, a dynamic DataTable, and a breed-distribution pie chart (with my additional “Other” aggregation to maximize visual appeal), as well as a geolocation map with selectable markers and popups.

#### Motivation:

The previous modules have all led up to this final project. With the CRUD module and earlier implementations, we gained the ability to manage and identify data within our AAC system, but a clear visual representation was still missing.

Our latest development addresses that gap by providing a fully visual interface. This enables both technical users and non-technical staff to easily view and interpret the data. Additionally, they can run simple queries to expand the dataset, and any new results will automatically be reflected in the dashboard.

Essentially, the dashboard serves as an effective tool for presenting data in a clear, visual format.

#### Getting Started:

To run the project locally or in Apporto we must do the following:

Verify that your setup has already integrated our aac\_crud.py module, this information can be found in our previous project readme.

When it’s completed and you verify all that information you may continue.

#### Place the required files in one folder:

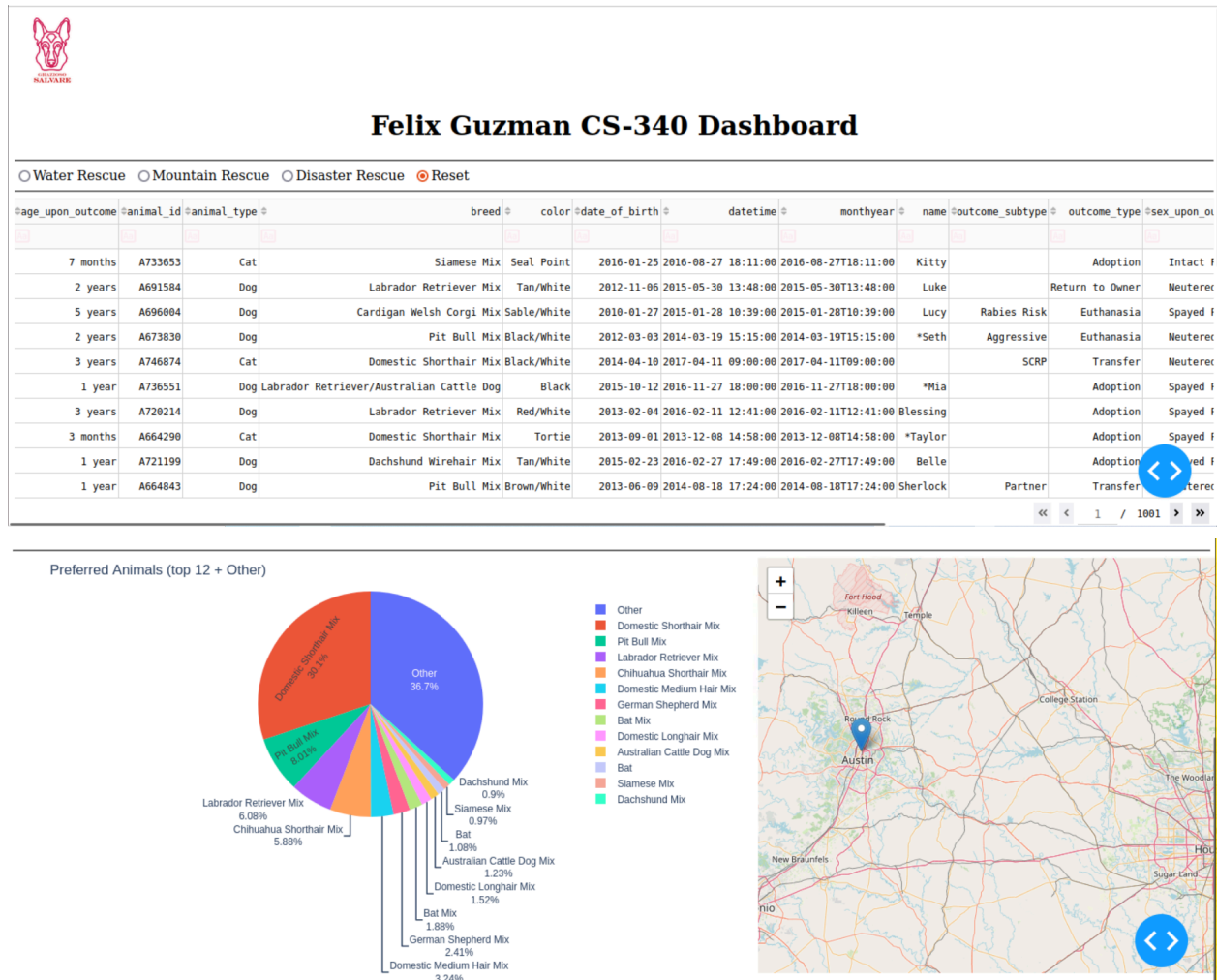
Ideally you would do this inside of a Jupyter notebook which can be opened by typing “jupyter notebook” in the terminal.

**Required Files:** ProjectTwoDashboard.ipynb, aac\_crud.py, and Grazioso Salvare Logo.png.

## Once that's done:

Open ProjectTwoDashboard.ipynb in Jupyter/Apporto and run it. The notebook hard-codes aacuser credentials for the classroom environment so make sure to change connection parameters in aac\_crud.py if you're using a different MongoDB instance.

If ran correctly you will receive a link (Ip Address) which you can click and will lead you to this:



## Installation:

The tools needed to properly utilize this module include the following:

**Python 3+:** To install Python you can travel to their official website and download it directly based on the operating system you have.

<https://www.python.org/downloads/>

Note: This template has been adapted from the following sample templates: [Make a README](#), [Best README Template](#), and [A Beginners Guide to Writing a Kickass README](#).

Once Python is installed you must go into the terminal and use pip to install any necessary dependencies which include the following:

**Shell Commands (Installation): PyMongo:** pip install pymongo | **Jupyter Notebook:** pip install jupyter

### **Additional Packages:**

If you followed the aac\_crud.py instructions you should already have Python3, PyMongo, and Jupyter installed, so here is what you will need for this new project.

**Dash-Leaflet | Plotly | Pandas | Matplotlib**

### **Installation Steps:**

**Open the terminal and type:** pip install dash-leaflet plotly pandas matplotlib

### **Explanation:**

As the requirement states we must use Python, and beyond that it's a portable language that's perfect for the task. Given our setup, Jupyter Notebook is the most convenient and adaptable environment for development and testing. PyMongo is the officially supported MongoDB driver for Python and provides direct support for authentication and CRUD operations, so installing it simplifies our Python and MongoDB interactions.

### **Updated Explanation:**

The old explanation still applies but for updated dependencies we need them all because we import them in our code to create our visual dashboard. Dash-leaflet, plotly, pandas, and matplotlib are all utilized to create it.

### **Usage:**

I will cover the usage case of an employee or anyone wanting to utilize our visualizer and how.

Assuming they have already opened the dashboard they have the following options:

**Select a rescue type:** Water Rescue, Mountain Rescue, Disaster Rescue, or Reset (default).

### **Technical Explanation of our Dashboard:**

Essentially, our dashboard callback uses aac\_crud to query MongoDB for matching records and returns them to the DataTable giving us the information we gathered.

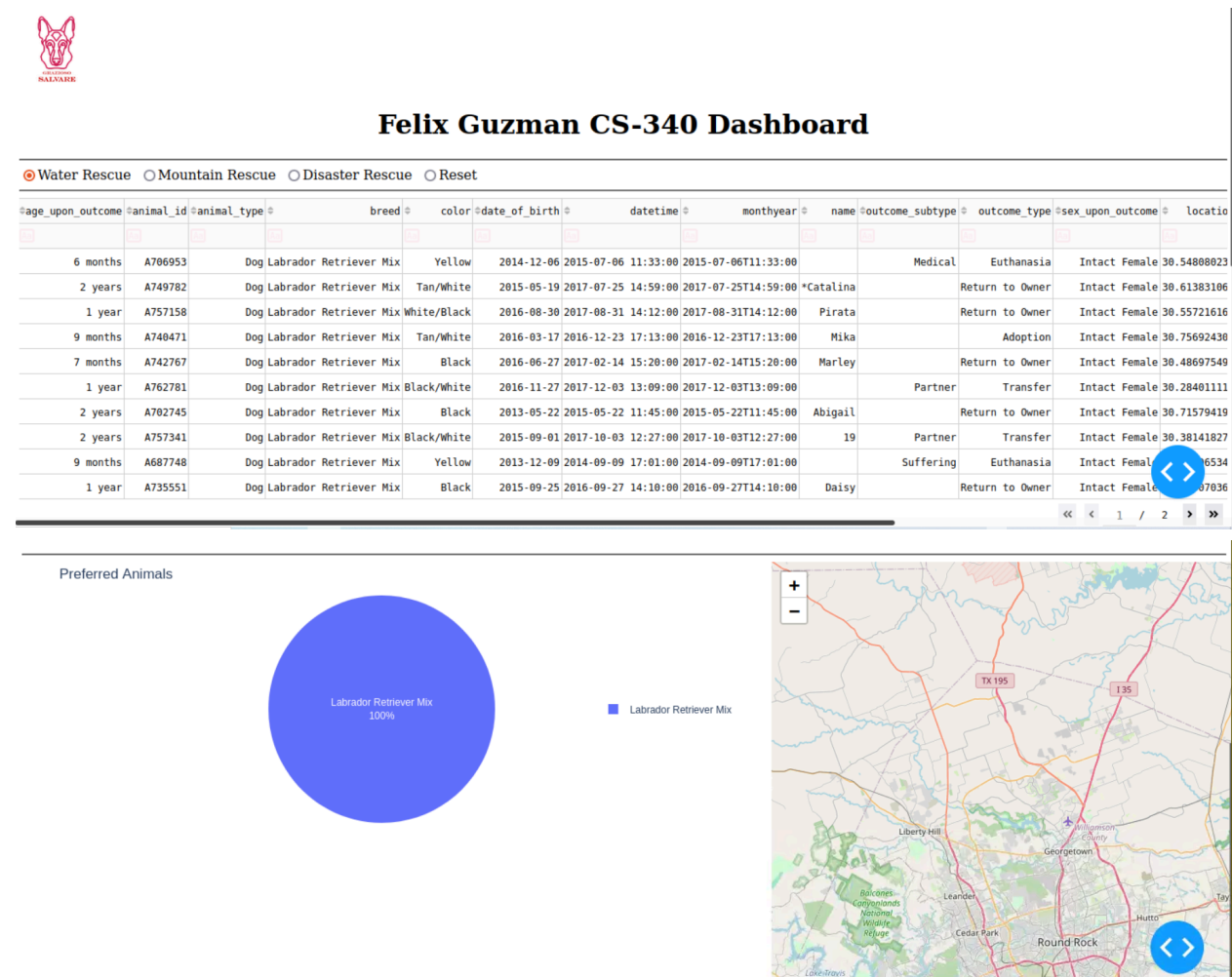
Note: This template has been adapted from the following sample templates: [Make a README](#), [Best README Template](#), and [A Beginners Guide to Writing a Kickass README](#).

Our info is distributed in pie charts which rebuild and aggregate our data into several slices for each breed. Due to visuals I decided to create a different group called the "Other" group which gathers the unique breeds into one category.

The map uses location\_lat and location\_long from the selected rows and selecting a row highlights it and opens a popup with the animal's name and breed.

### Water Rescue behavior:

As requested in the specifications we adhere to filtering to the established parameters. This includes rescue types as water, the selected preferred breeds, preferred sex, and training age in weeks between 26 and 156 via our own variables, as well as previously established variables/parameters gathered from our dataset. (You can see this below)



Note: This template has been adapted from the following sample templates: [Make a README](#), [Best README Template](#), and [A Beginners Guide to Writing a Kickass README](#).

## Mountain Rescue behavior:

As requested in the specifications we adhere to filtering to the established parameters. This includes rescue types as mountain, the selected preferred breeds, preferred sex, and training age in weeks between 26 and 156 via our own variables, as well as previously established variables/parameters gathered from our dataset. (You can see this below)

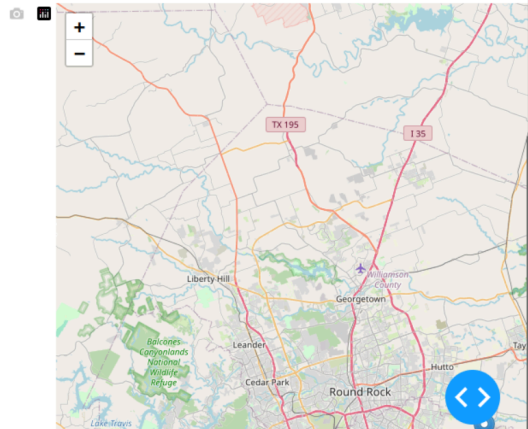
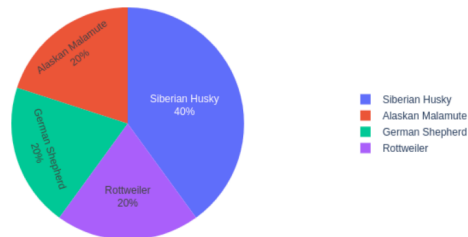


## Felix Guzman CS-340 Dashboard

☐ Water Rescue ☒ Mountain Rescue ☐ Disaster Rescue ☐ Reset

age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat
2 years	A708726	Dog	Alaskan Malamute	Sable/White	2013-07-30	2015-08-02 17:24:00	2015-08-02T17:24:00	Papa		Return to Owner	Intact Male	30.4309339291938
6 months	A765461	Dog	German Shepherd	Sable	2017-07-20	2018-01-22 11:54:00	2018-01-22T11:54:00	Sargent		Return to Owner	Intact Male	30.40668985085
2 years	A728165	Dog	Rottweiler	Black	2015-05-31	2017-09-23 11:23:00	2017-09-23T11:23:00	Zeke		Return to Owner	Intact Male	30.466577208743
2 years	A721834	Dog	Siberian Husky	Brown/White	2014-03-05	2016-03-23 16:23:00	2016-03-23T16:23:00		Suffering	Euthanasia	Intact Male	30.5680998448899
2 years	A704101	Dog	Siberian Husky	Black/White	2013-06-01	2015-06-02 16:41:00	2015-06-02T16:41:00	Lobo		Return to Owner	Intact Male	30.4263764229275

Preferred Animals



## Disaster Rescue behavior:

As requested in the specifications we adhere to filtering to the established parameters. This includes rescue types as disaster, the selected preferred breeds, preferred sex, and training age in weeks between 20 and 300 via our own variables, as well as previously established variables/parameters gathered from our dataset. (You can see this below)

Note: This template has been adapted from the following sample templates: [Make a README](#), [Best README Template](#), and [A Beginners Guide to Writing a Kickass README](#).

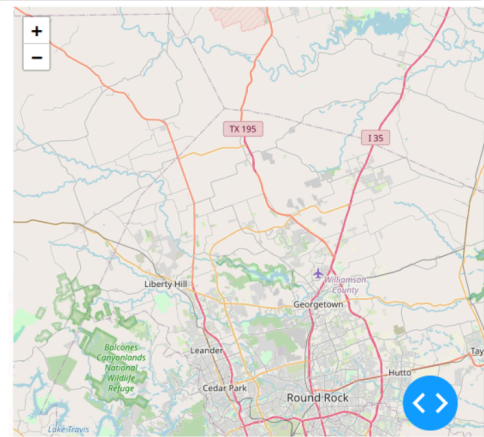
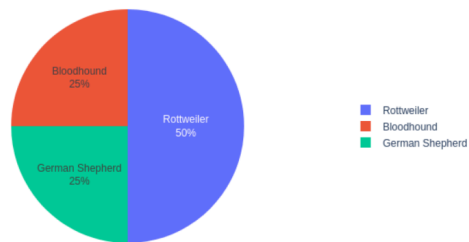


## Felix Guzman CS-340 Dashboard

☐ Water Rescue ☐ Mountain Rescue ☒ Disaster Rescue ☐ Reset

age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat
4 years	A712291	Dog	Bloodhound	Red	2011-09-20	2015-09-22 15:43:00	2015-09-22T15:43:00	Boomer		Return to Owner	Intact Male	30.2709983761287 -97
6 months	A765461	Dog	German Shepherd	Sable	2017-07-20	2018-01-22 11:54:00	2018-01-22T11:54:00	Sargent		Return to Owner	Intact Male	30.40668985085 -9
4 years	A694614	Dog	Rottweiler	Black/Brown	2011-01-01	2015-01-01 14:25:00	2015-01-01T14:25:00	Striker		Return to Owner	Intact Male	30.329873203611 -97
2 years	A728165	Dog	Rottweiler	Black	2015-05-31	2017-09-23 11:23:00	2017-09-23T11:23:00	Zeke		Return to Owner	Intact Male	30.466577208743 -97

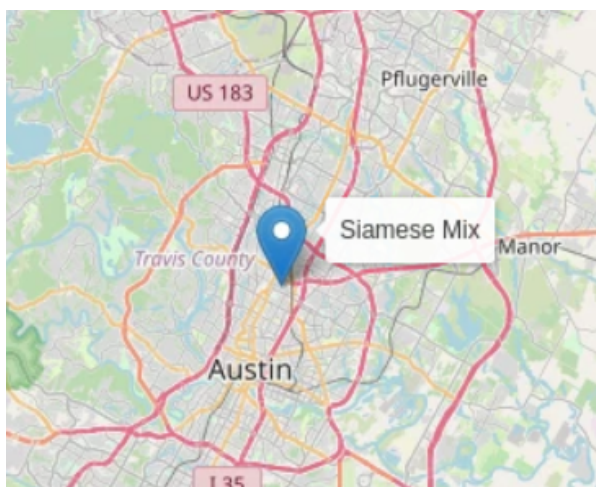
Preferred Animals



### Map interaction:

Additionally, selecting a row highlights it and shows its marker popup on the map with name and breed as it is shown below:

### Map Closeup:



Note: This template has been adapted from the following sample templates: [Make a README](#), [Best README Template](#), and [A Beginners Guide to Writing a Kickass README](#).

## Code Example

Concise, key snippets that show how the dashboard queries and prepares the pie chart.

Filter query used by callbacks

## Tests:

Manual tests to verify CRUD + dashboard:

The easiest way to verify is to use the following command to get a list of expected documents in the database then you just verify by visually checking that our graphs match the data: **db.read({})**

## Our database read command results:

```
In [2]: db.read({})

Out[2]: [{ '_id': ObjectId('6886edb0092dfd39b95b3001'),
  'rec_num': 4,
  'age_upon_outcome': '7 months',
  'animal_id': 'A733653',
  'animal_type': 'Cat',
  'breed': 'Siamese Mix',
  'color': 'Seal Point',
  'date_of_birth': '2016-01-25',
  'datetime': '2016-08-27 18:11:00',
  'monthyear': '2016-08-27T18:11:00',
  'name': 'Kitty',
  'outcome_subtype': '',
  'outcome_type': 'Adoption',
  'sex_upon_outcome': 'Intact Female',
  'location_lat': 30.3188063374257,
  'location_long': -97.7240376703891,
  'age_upon_outcome_in_weeks': 30.8225198412698},
  { '_id': ObjectId('6886edb0092dfd39b95b3002'),
  'rec_num': 5,
```

You could also go to our old README document to examine the tests and queries we utilized, you could use similar queries to modify our database and see the changes visually.

## Screenshots:

I supplied most of the necessary screenshots above **and made any necessary updates**.

## Roadmap/Features (Optional):

With this README we have completed our final project for this course. I'm assuming if there was a need for it some additional features would include an easy CSV exporter that would allow a real company to manage the visualization data much easier. Other than that the possibilities are endless.

## Contact:

Your name: **Felix Guzman**

Note: This template has been adapted from the following sample templates: [Make a README](#), [Best README Template](#), and [A Beginners Guide to Writing a Kickass README](#).