Grazioso Salvare Dashboard Documentation

Jon Wickerd CS-340 6/22/24

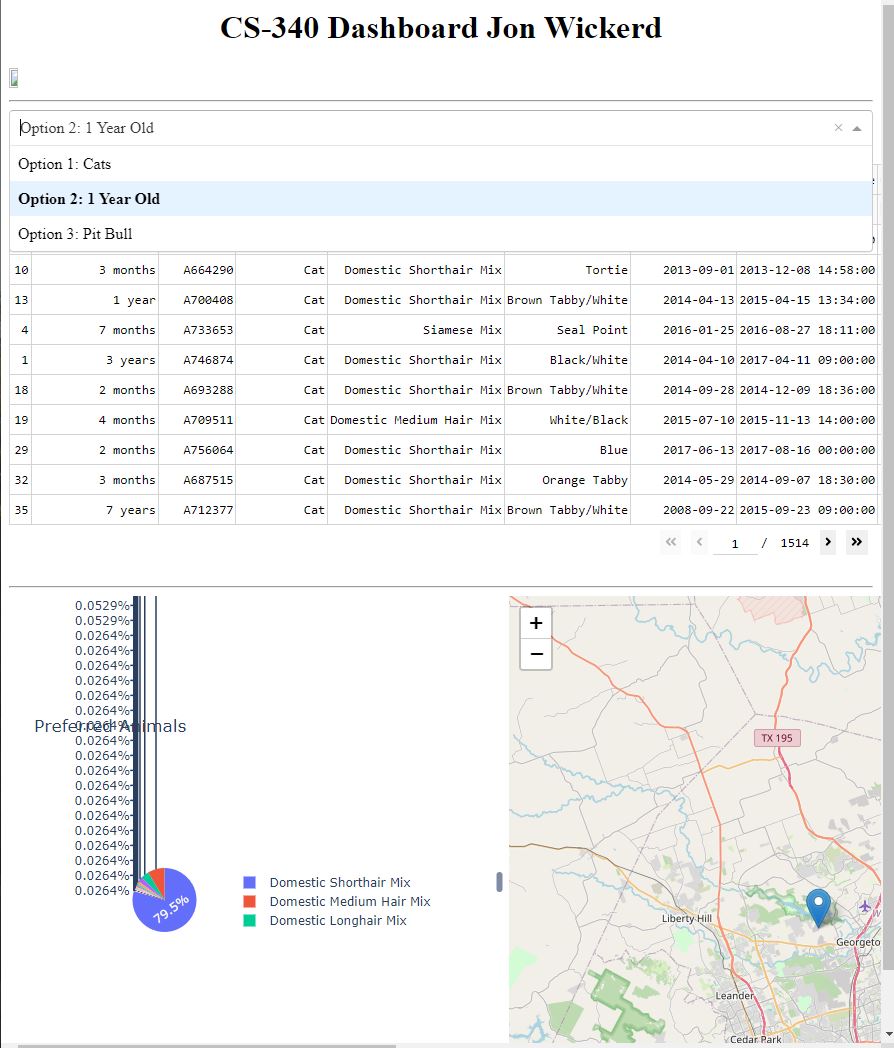
# Project Overview

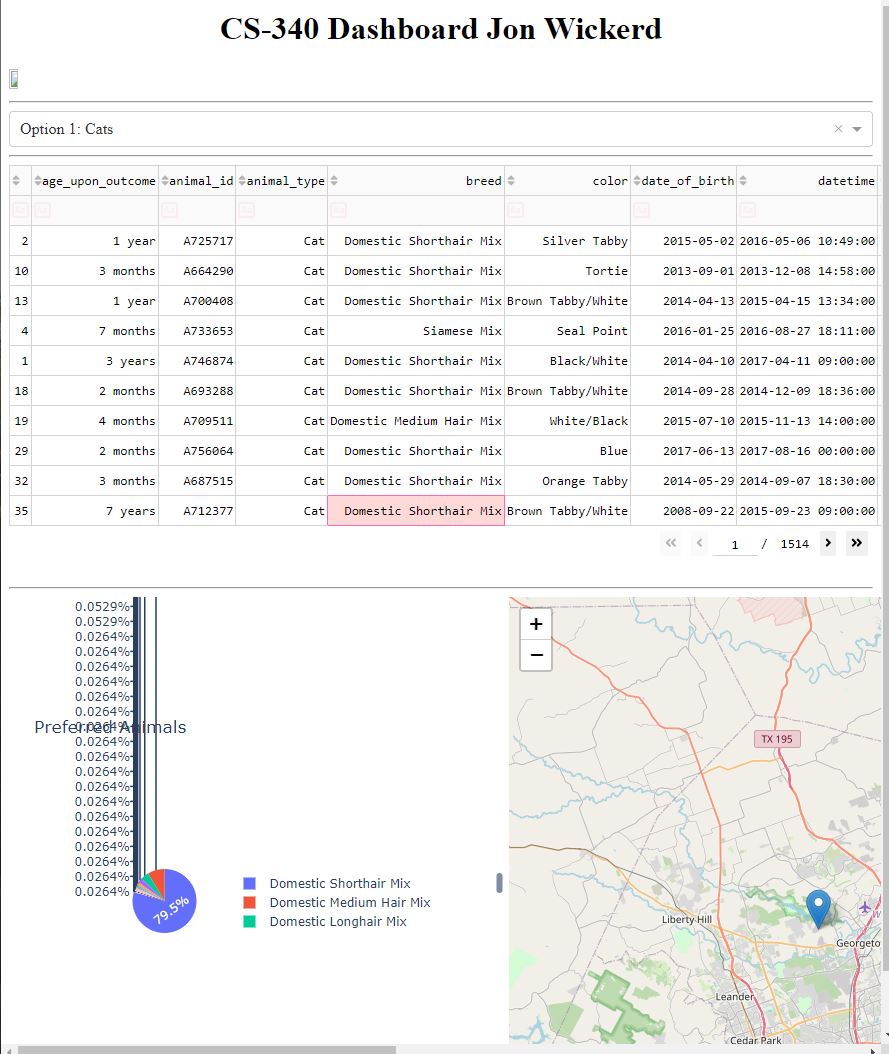
The Grazioso Salvare Dashboard project aims to create an interactive web application that allows users to explore and analyze data from the Austin Animal Center. This includes filtering data based on animal types, ages, and breeds, displaying data in a tabular format, visualizing preferred animal breeds through a pie chart, and showcasing animal locations on a map.

## Screenshots

Below are screenshots demonstrating the functionality of the dashboard after applying different filters:

Filter Applied: Option 2 - 1 Year Old:





Filter Applied: Option 1 - Cats:

## Tools Used

**Dash:** A Python framework for building analytical web applications. It was chosen for its ability to create interactive dashboards with complex visualizations and dynamic user interactions.

**Plotly Express:** Used for creating interactive charts and graphs, including the pie chart displaying preferred animal breeds.

**Dash Leaflet:** Integrated for displaying geographic data with interactive maps.

**Pandas:** Used for data manipulation and analysis, particularly for handling data retrieved from MongoDB.

**MongoDB:** Used as the database to store and retrieve data from the Austin Animal Center. MongoDB was chosen for its flexibility with unstructured data (JSON format), ease of integration with Python via pymongo, and scalability for potential future data growth.

## MongoDB Integration

MongoDB was selected as the model component due to several key capabilities:

**Schema Flexibility:** MongoDB's document-based data model allows for storing data in JSON-like documents, which is well-suited for storing diverse types of animal records from the Austin Animal Center.

**Scalability:** MongoDB's horizontal scaling capabilities ensure that the database can handle large volumes of data efficiently as the application scales.

**Integration with Python:** MongoDB integrates seamlessly with Python using pymongo, providing robust querying and data manipulation capabilities directly within Python scripts.

## Dash Framework

Dash was chosen as the framework for the web application due to its:

**Pythonic Nature:** Allows developers to leverage their Python skills for both frontend (UI) and backend (callbacks) development.

**Component-Based Structure:** Facilitates modular development and reusability of components like dropdowns, tables, graphs, and maps.

**Interactive Capabilities:** Provides built-in support for interactivity through callbacks, enabling real-time updates based on user input without page reloads.

## Project Completion Steps

**Data Retrieval:** Data was retrieved from MongoDB using a Python CRUD module (AnimalShelter) and loaded into a Pandas DataFrame.

**Dashboard Layout:** Developed the dashboard layout using Dash HTML components for structuring, dropdowns, tables, graphs, and maps.

**Callbacks:** Implemented callbacks using Dash's @app.callback decorator to update components dynamically based on user interactions (e.g., dropdown selection, table row selection).

**Visualization:** Created interactive visualizations such as the pie chart for preferred animal breeds and geographic locations on the map using Plotly Express and Dash Leaflet.

## Challenges Faced

**Data Filtering Logic:** Ensuring correct data filtering logic based on dropdown selections required careful consideration of DataFrame operations and MongoDB query structure.

**Visualization Scaling:** Managing the scaling of visualizations like the pie chart to avoid disproportionate representation of small data subsets.

# Conclusion

The Grazioso Salvare Dashboard provides a user-friendly interface for exploring and analyzing Austin Animal Center data interactively. Leveraging MongoDB for data storage and retrieval, and Dash for web application development, ensures robust performance and scalability for future enhancements.