**README: Austin Animal Center Outcomes Dashboard**

**Project Description**

This project aims to create an interactive dashboard for the Austin Animal Center Outcomes dataset. The dashboard displays data in the form of an interactive table and two charts - a geolocation chart and a chart of our choice. The dashboard allows users to filter data based on rescue type and preferred dog breeds using radio items or dropdowns. The filtered data is displayed in real-time in the table and the charts, allowing the user to explore and gain insights into the data.

Graphical user interface, application, Word

Description automatically generated

**Tools Used**

The following tools were used to create the dashboard:

* **Python 3.8**
* **Jupyter Notebook**
* **Dash framework**
* **MongoDB**

The choice of tools was based on their suitability for the project. Python was chosen due to its ease of use and extensive libraries for data processing and visualization. Jupyter Notebook was used to develop and test the code in an interactive environment.

Dash was chosen as the framework to build the web application. It provides a simple and efficient way to create reactive, web-based applications. The framework is based on Flask, Plotly.js, and React.js, which are popular technologies in web development.

MongoDB was chosen as the model component for the project due to its flexibility, scalability, and ease of integration with Python. MongoDB is a NoSQL document-oriented database that stores data in a JSON-like format. This makes it easy to store and manipulate data in Python using its driver, PyMongo.

**Steps to Reproduce**

To reproduce the dashboard, follow these steps:

* Clone the GitHub repository using the following command:

**git clone** [**https://github.com/username/repo.git**](https://github.com/username/repo.git)

* Install the required libraries by running the following command:

**pip install -r requirements.txt**

* Start Jupyter Notebook by running the following command:

**jupyter notebook**

* Open the ProjectTwoDashboard.ipynb file and run the code.
* Access the dashboard by navigating to http://localhost:8050/ in a web browser.
* After running the code, you should get these features:

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application

Description automatically generated

All widgets that are interactive and available to use

**Resources**

The following resources and software applications were used in the development of the dashboard:

* Python 3.8: https://www.python.org/
* Jupyter Notebook: https://jupyter.org/
* Dash: https://dash.plotly.com/
* MongoDB: https://www.mongodb.com/
* PyMongo: https://pypi.org/project/pymongo/
* Plotly.js: https://plotly.com/javascript/
* React.js: https://reactjs.org/
* Bootstrap: https://getbootstrap.com/

**Challenges**

One of the main challenges encountered during the project was working with MongoDB. As a document-oriented database, it required a different approach to working with data compared to traditional relational databases. However, this was overcome by familiarizing ourselves with MongoDB's query language and PyMongo's API.

Another challenge was developing the interactive widgets and ensuring that the dashboard was responsive to user inputs. This was addressed by carefully designing the dashboard layout and using the Dash framework's built-in callbacks to update the charts and table in real-time.

Finally, deploying the dashboard to a web server required additional configuration and security considerations. However, this was addressed by following the Dash documentation and best practices for web deployment.