Iowa State University

Electrical and Computer Engineering

Team: sdmay21-31

Advisor: Dr. Goce Trajcevski

Thomas Beckler, Bradley Gannon, Benjamin Huinker, Gabriel Huinker, Koushhik Kumar, Cristina Marquez, Jacob Spooner

Proximity vs. Diversity in Heterogeneous Datasets

Introduction

Problem

Researchers have difficulty clustering and visualizing a large and multivariate (heterogenous) datasets, combining:

- Weighted impacts on proximity
- Weighted impacts on diversity

They need a tool that efficiently identifies patterns in clusters in the data.

Solution

- Create a web application
- Use customizable algorithms to relativise the data
- Allow uploading of various datasets
- Analyze datasets with given proximity and diversity parameters.

Design Requirements

Functional Requirements

- A website to interact with stored data
- Secure and Efficient data storage
- Users determine which attributes require proximity/diversity
- Users can change the weight of attributes
- Allow users to query heterogeneous datasets
- Allow users to upload and manage datasets

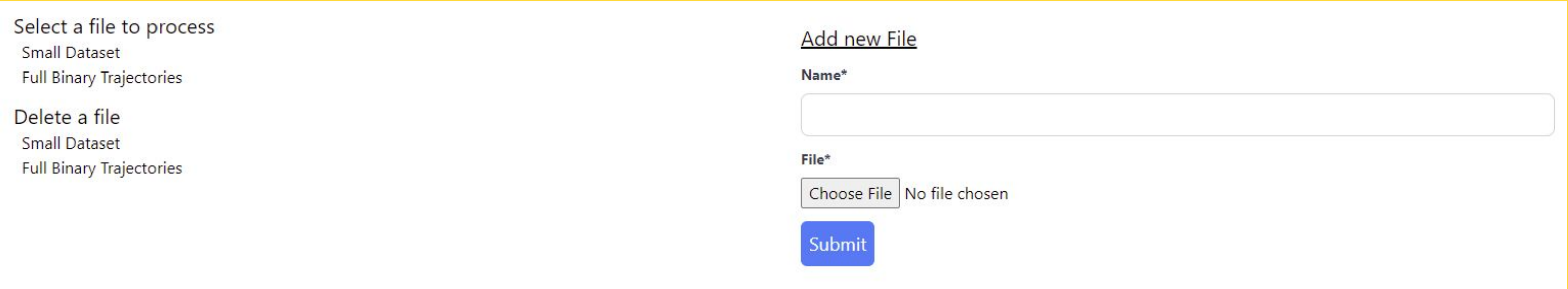
Non-Functional Requirements

- Easy to navigate
- Simple and intuitive design
- Fast render and process times
- Public access

Operating Environment

- Web Application

Website - Adding a Dataset



Select a file to process
Small Dataset
Full Binary Trajectories

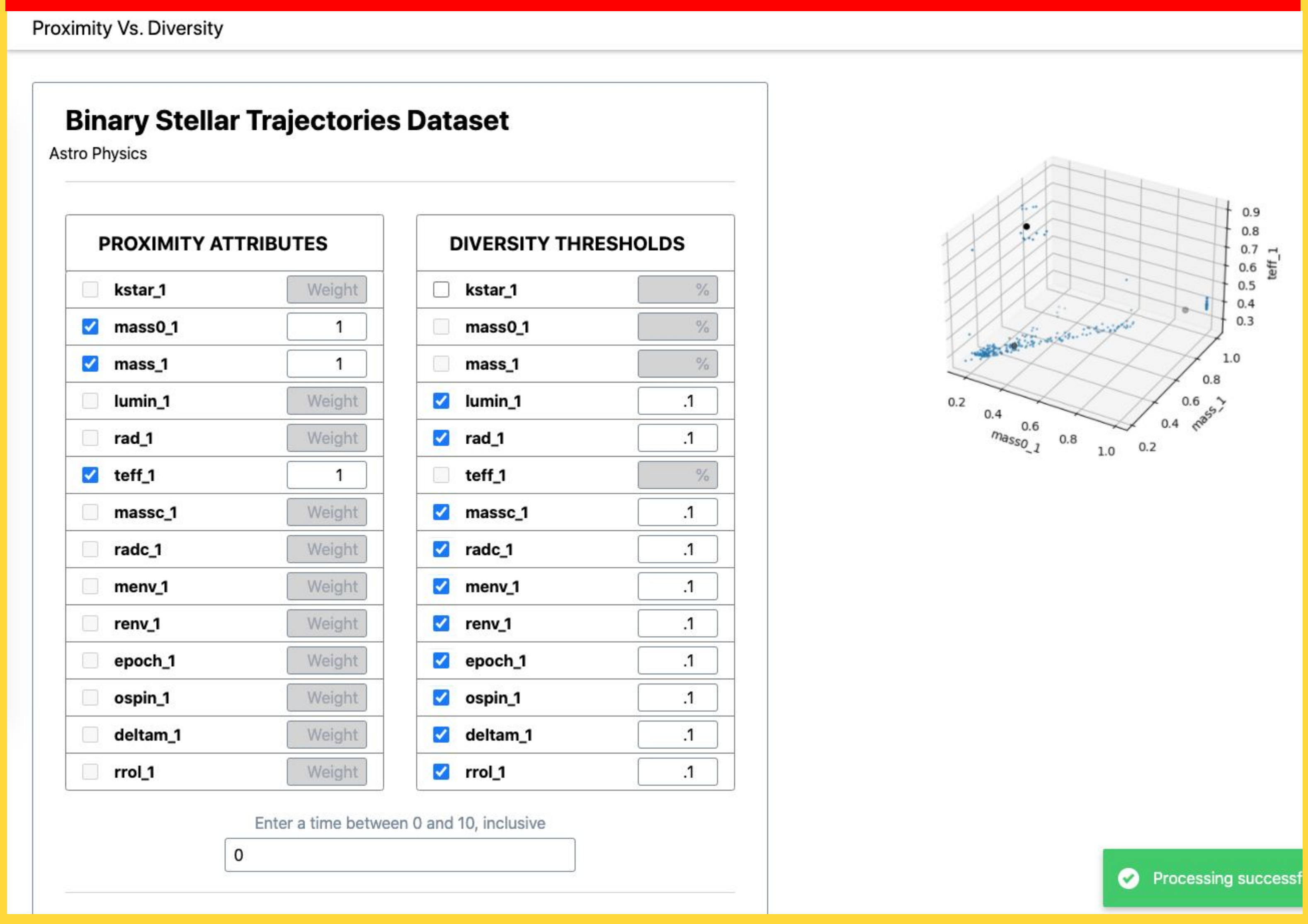
Delete a file
Small Dataset
Full Binary Trajectories

Add new File

Name*

File*

Website - Processing Page



Proximity Vs. Diversity

Binary Stellar Trajectories Dataset

Astro Physics

	PROXIMITY ATTRIBUTES	
□	kstar_1	Weight
☑	mass0_1	1
☑	mass_1	1
□	lumin_1	Weight
□	rad_1	Weight
☑	teff_1	1
□	massc_1	Weight
□	radc_1	Weight
□	menv_1	Weight
□	renv_1	Weight
□	epoch_1	Weight
□	ospin_1	Weight
□	deltam_1	Weight
□	rrol_1	Weight

	DIVERSITY THRESHOLDS	
□	kstar_1	%
□	mass0_1	%
□	mass_1	%
☑	lumin_1	.1
☑	rad_1	.1
□	teff_1	%
☑	massc_1	.1
☑	radc_1	.1
☑	menv_1	.1
☑	renv_1	.1
☑	epoch_1	.1
☑	ospin_1	.1
☑	deltam_1	.1
☑	rrol_1	.1

Enter a time between 0 and 10, inclusive

0

Testing

Testing Environment

- **Local Development**

Django Test Cases and Manual Testing was used to verify the functionality of the following areas.

- **Algorithms**
- **Queries**
- **Business Logic**
- **Frontend Integration and Functionality**

Intended Users and Uses

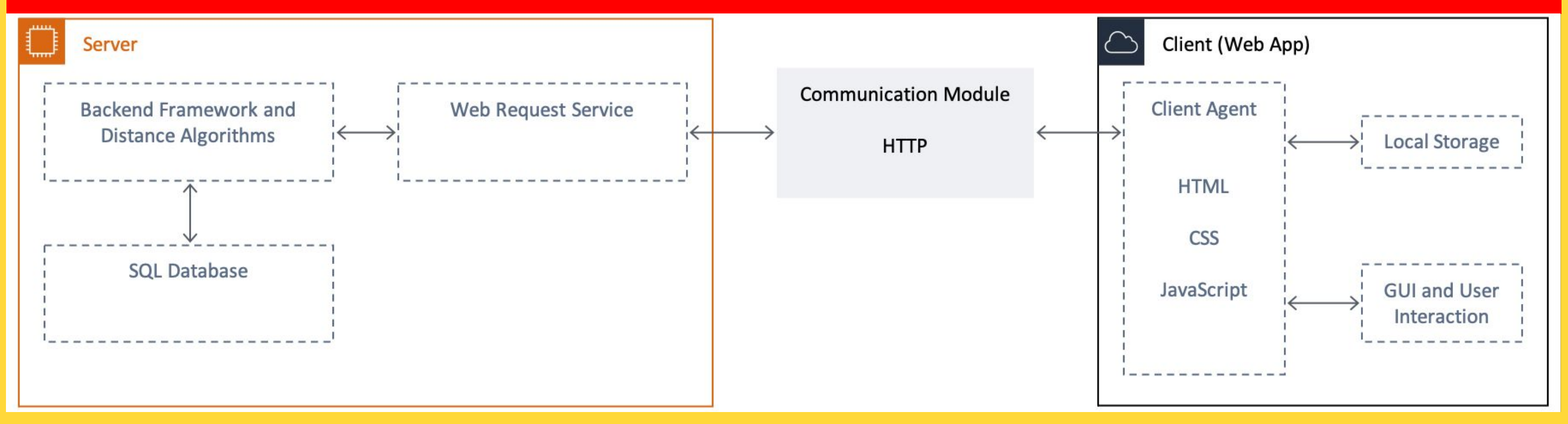
Intended Users

- Astrophysicists
- General public

Intended Use Cases

- Clustering Data
- Visualize nodes & clustering patterns

Block Diagram



Technologies

Client

- HTML / CSS / JS
- Tailwind CSS

Server

- Django
- Gunicorn
- nginx
- Memcached
- RabbitMQ

Database

- PostgreSQL

Algorithm

- NumPy
- Pandas
- Pyclustering
- Matplotlib

Primary Frameworks

Primary Frameworks and Technologies:

- Django

  Backend web framework written in python

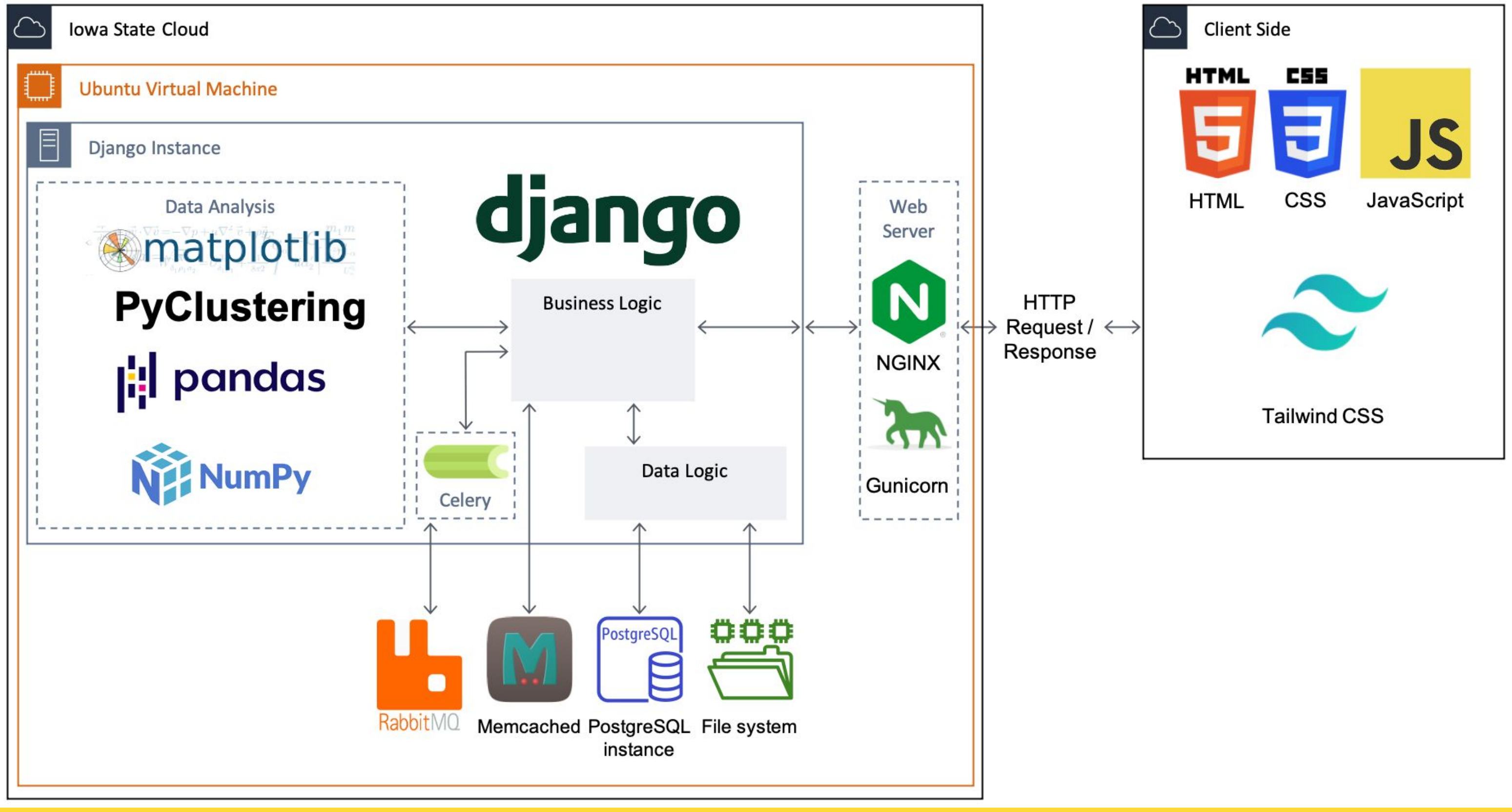
- Postgres

  Database storage allowing JsonField

- Pyclustering

  Python clustering package with customizable distance metrics

Design Architecture



Engineering Standards and Design Practices

IEEE 802.3-2018

- Web application uses wireless/ethernet network

P2989

- Login authentication required to access database

IEEE 23026-2015

- Maintenance required for the website

Project Resources

Single Core Ubuntu 20.04 50GB VM hosted on Iowa State Servers