



# Proximity vs. Diversity

Team: SDMay21-31

Advisor/Client: Goce Trajcevski

Gabe Huinker, Jacob Spooner,  
Benjamin Huinker, Cris Marquez, Brad Gannon,  
Koushik Kumar, Thomas Beckler

# Project Vision

- Problem
  - Large Multivariate/Heterogeneous Datasets
  - Clustering and Visualizing
  - Weighted Proximity and Diversity
- Solution
  - Web Application
  - Customizable algorithms
  - Customizable distance metrics
- End Users
  - Researchers
  - Public



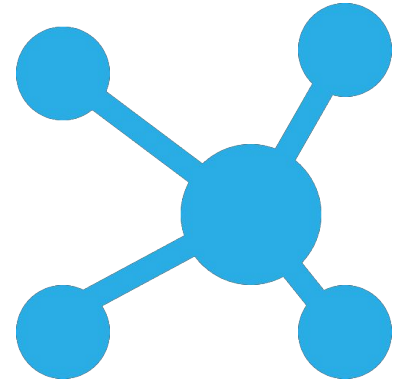
# Technical Challenges



Large datasets on  
small machine



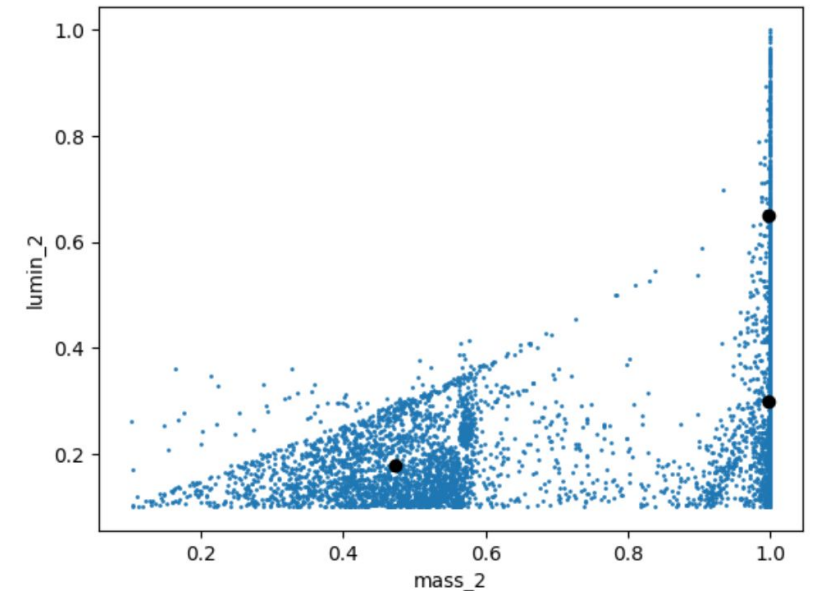
Dynamic data and  
simulations



Idea of Proximity and  
Diversity

# Proximity vs. Diversity

- Time instances from a time series
  - Time interpolation
- Heterogeneous data and Relativization
  - Min and Max across their attributes in a simulation
- Proximity - Weighted Euclidean Distance
  - $D(br,bb) = \sqrt{W1 * (M1b/M2b - M1r/M2r)^2 + W2 * (L1b/L2b - L1r/L2r)^2}$
- Diversity - Threshold base



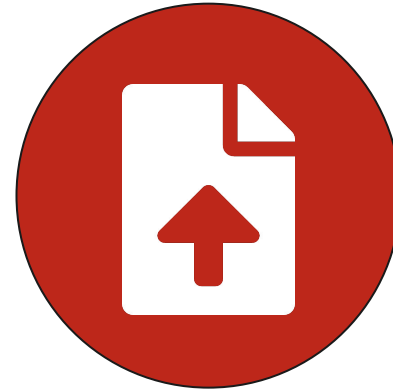
# Functional Requirements



Web Application



Secure data storage



Dynamic Datasets

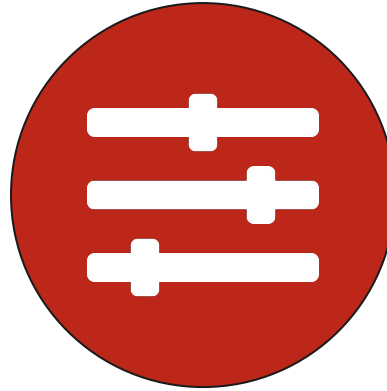


Weighted Proximity  
and Diversity

# Non-Functional Requirements



Publicly accessible



Intuitive Interface:  
Controls & Visuals



Fast Render Times

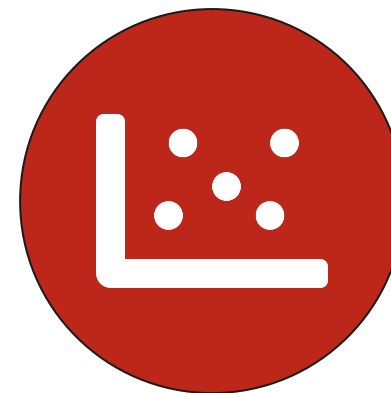
# Constraints



No Funding:  
ISU-Provided VM



Large Datasets



2D and 3D graphs  
only

# Standards

- IEEE 802.11-2020
- 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





# Risks and Risk Management

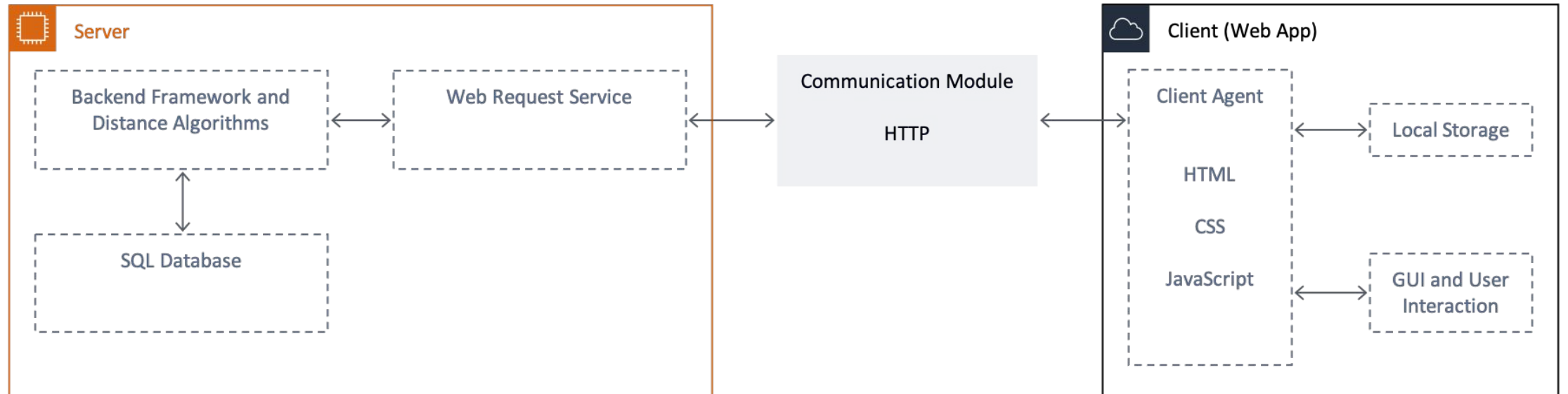
- Security Risks
  - Production environment breaches
  - Queries and algorithms
- Performance Risks
  - Queries and algorithms
- Risk Mitigation Strategies
  - Use features of Django
  - Create tests for runtimes

# Task Decomposition

- Database Design
- Local and Production environment setup
- UI Implementation
- UI Functionality
- Backend Endpoints
- Algorithms
- Algorithm Query
- Data visualization
- Help Documentation

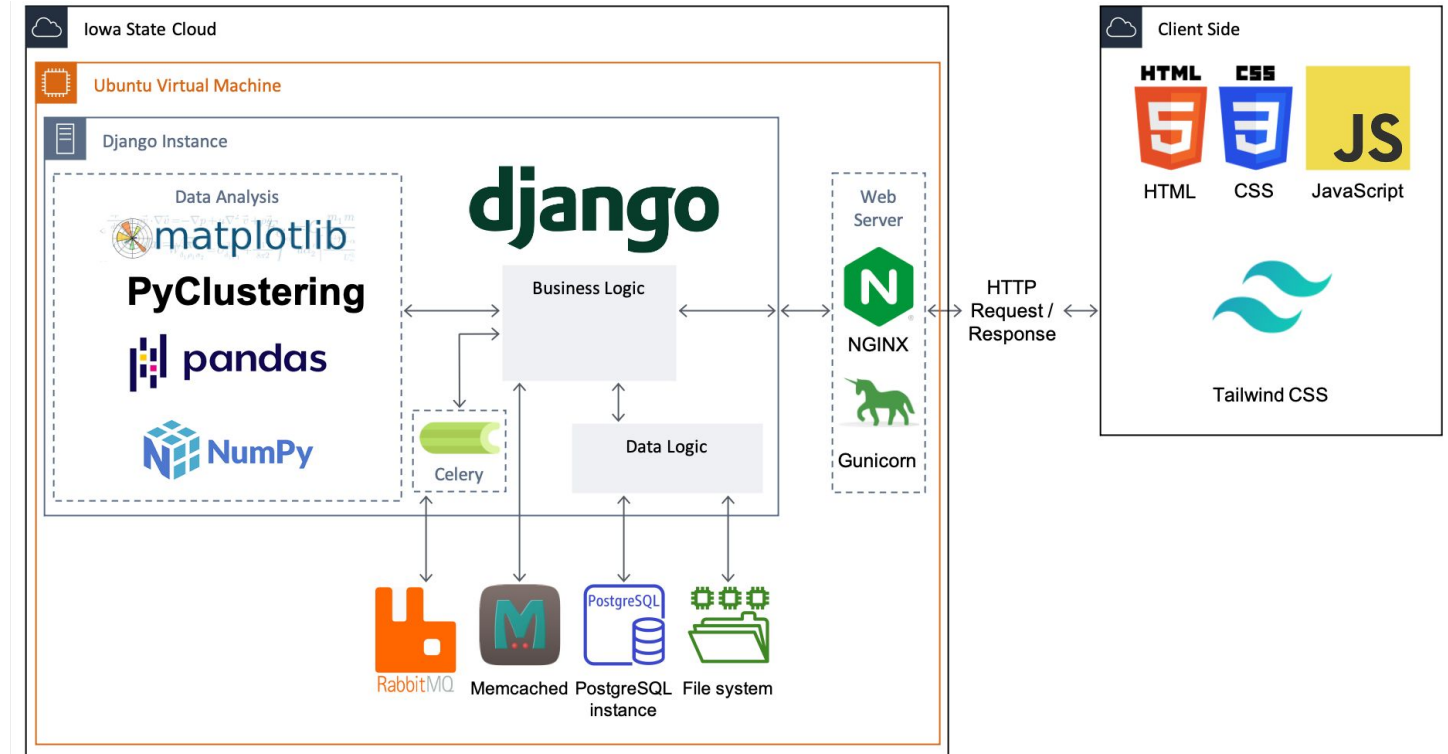
# High-level System Design

- Web Application
- Client-Server Model
- HTTP Communication



# Primary Frameworks

- Django (Backend)
- PostgreSQL (Database)
- nginx + gunicorn (Web server)
- Memcached (Caching)
- RabbitMQ + Celery (Async Work)
- Matplotlib (Data Visualization)
- Tailwind CSS (CSS Framework)
- Pyclustering, NumPy, Pandas (Data analysis)



# UI

- Proximity Attribute Selection
- Diversity Threshold Selection
- Algorithm Selection
- Data visualization
- Data table
- Admin interface

**Binary Trajectories 2 Dataset**  
Astrophysics

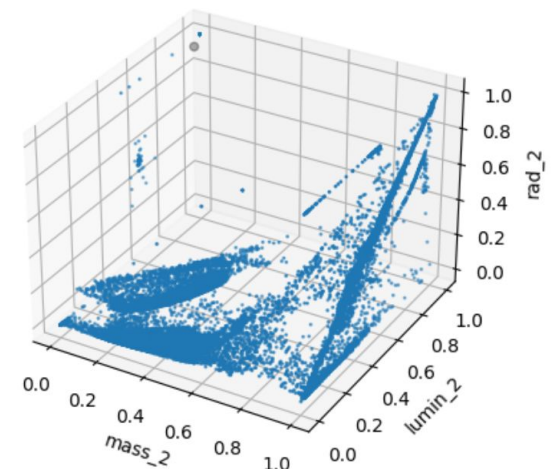
PROXIMITY ATTRIBUTES		DIVERSITY THRESHOLDS	
<input type="checkbox"/> kstar_2	4	<input type="checkbox"/> kstar_2	%
<input type="checkbox"/> mass0_2	2	<input type="checkbox"/> mass0_2	%
<input checked="" type="checkbox"/> mass_2	50	<input type="checkbox"/> mass_2	0.3
<input checked="" type="checkbox"/> lumin_2	33	<input type="checkbox"/> lumin_2	%
<input type="checkbox"/> rad_2	Weigh	<input type="checkbox"/> rad_2	%
<input type="checkbox"/> teff_2	Weigh	<input type="checkbox"/> teff_2	%
<input type="checkbox"/> massc_2	Weigh	<input type="checkbox"/> massc_2	%
<input type="checkbox"/> radc_2	Weigh	<input checked="" type="checkbox"/> radc_2	0.1
<input type="checkbox"/> menv_2	Weigh	<input type="checkbox"/> menv_2	%
<input type="checkbox"/> renv_2	Weigh	<input type="checkbox"/> renv_2	%
<input type="checkbox"/> epoch_2	Weigh	<input type="checkbox"/> epoch_2	%
<input type="checkbox"/> ospin_2	Weigh	<input type="checkbox"/> ospin_2	%
<input type="checkbox"/> deltam_2	Weigh	<input type="checkbox"/> deltam_2	%
<input type="checkbox"/> rrol_2	Weigh	<input type="checkbox"/> rrol_2	%

Enter a time between 0 and 1001, inclusive  
300

KMeans  
Enter number of clusters between 1 and 20, inclusive  
3

Select up to 3 attributes

**Process**



**Django administration**

Home · Authentication and Authorization · Users

APP  
Datasets [+ Add](#)

AUTHENTICATION AND AUTHORIZATION  
Groups [+ Add](#)  
Users [+ Add](#)

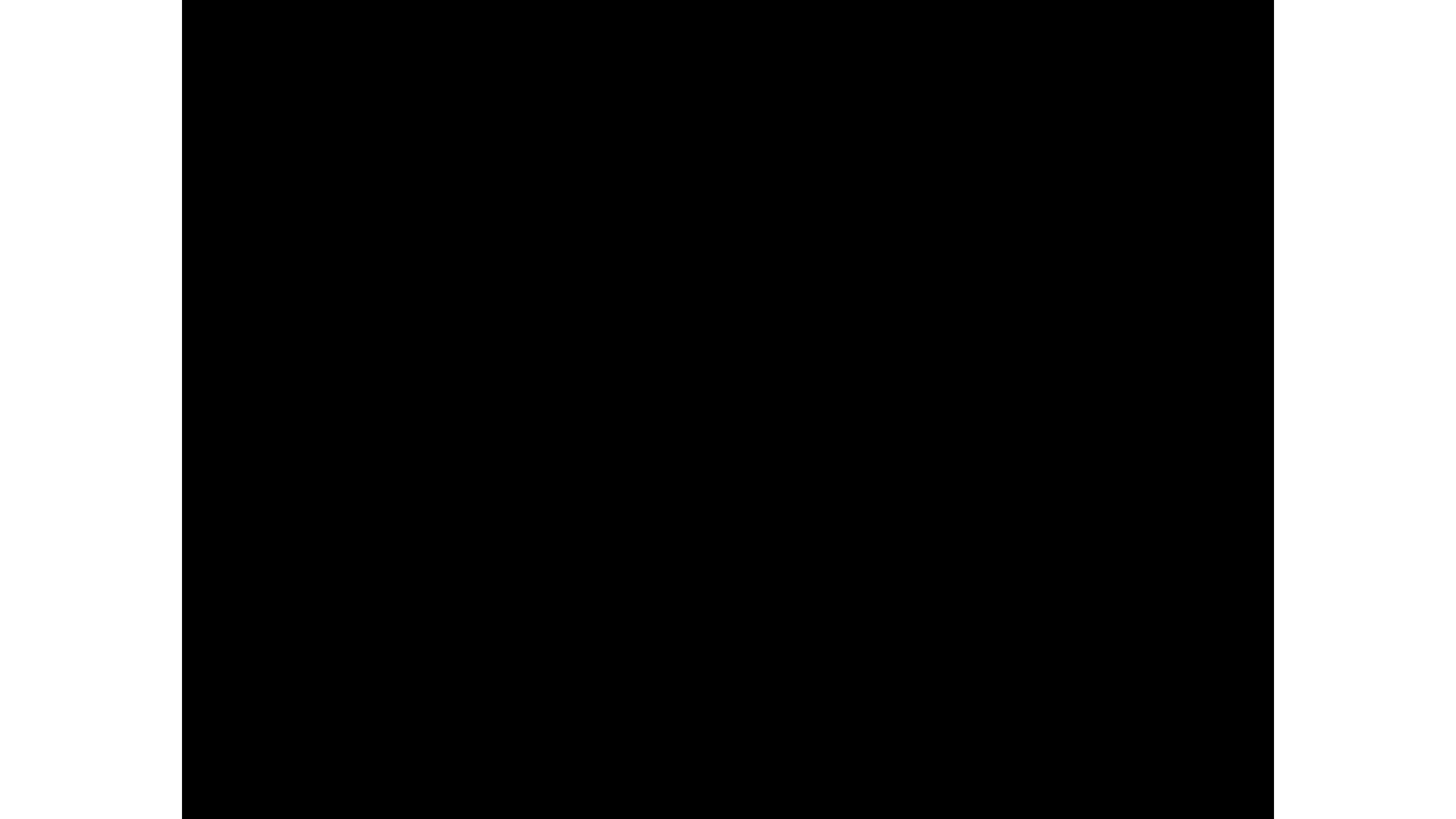
Select user to change

Q  Search

Action:  Go 0 of 1 selected

<input type="checkbox"/> USERNAME	EMAIL ADDRESS
<input type="checkbox"/> astro	astro@email.com

1 user



# Project Plan - Progress Metrics/Tracking

- Progress is largely monitored through task completion
- Track progress through Github, Slack, and Google Drive
  - Codebase: Github
  - Tasks: Kanban - Github Project
  - Communication: Slack
  - Documentation: Google Drive



# Project Plan - Milestones

1. Architecture Design (Oct. 25)
2. Finalize Design Document V1 (Nov. 15)
3. Implement UI (Feb. 23)
4. Finalize Database Design (Mar. 25)
5. Finalize Algorithm Solutions (Apr. 4)
6. Complete Testing (Apr. 20)
7. Final Software Version (Apr. 24)
8. Finalize Design Document (Apr. 25)



# Unit Testing

- Algorithms - Create Clustering Data
  - (Weighted) Euclidean algorithm
  - Relativised node attributes
- Business Logic - Validate and Sanitize
  - Validate and ensure correct request
- Database Queries - TimeFrame Query
  - One node per simulation
  - Correct index corresponding to time\_percentage

django

# Interface and Acceptance Testing

- 9 Pages
- Distance Functions
- User Interface
- Visualizations
- Assigning Weights and Thresholds
- Fast Response Times

django

# Conclusion

- **What we Accomplished:**
  - Proximity vs. Diversity with K-means
  - Dynamic Datasets
  - Secure data storage and fast response times
- **Where to go from here:**
  - Higher dimensional graphs
  - Multiple clustering algorithms
  - Accommodate larger datasets and user loads



# Conclusion - Team Member Contributions

- **Gabe Huinker:**

- Server and System Architecture
- Algorithm Design and Implementation
- Database Design and Implementation
- Dataset Management

- **Jacob Spooner:**

- Backend Setup and Maintenance
- Communication with Front End
- Data Validation and Processing
- Operation Manual

- **Brad Gannon:**

- Algorithm Design and Implementation
- Algorithm Research
- Data Visualization Experimentation

- **Koushhik Kumar:**

- Frontend Design and Implementation
- Designed Improved (Current) UI
- Coded & Styled Index & Datasets pages

- **Thomas Beckler:**

- Designed Initial Index Page
- Frontend Developer
- User Manual
- Basic Poster Outline

- **Cristina Marquez:**

- Initial Index Page Design and Implementation
- Frontend Developer
- Styled User Manual

- **Ben Huinker:**

- Modified seed code to seed other databases
- Tested seeding and frontend
- Created test databases for project