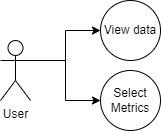
Use Case #1: The user can view the clustered repository data with each cluster color coded.

Use Case #2: The user can select the two metrics to display the clustered data on.



| Component | Priority | Requirement Name | Requirement Description |
| --- | --- | --- | --- |
| API | High | Handle GET Requests | The API should be able to handle GET requests which will return all post-clustering data |
| API | High | Handle POST Requests | The API should be able to handle POST requests which will return post-clustering data on the two input metrics |
| API | Med | Handle Invalid URLs | The API should return code 404 if a non-existent subdomain access attempt is made |
| API | Low | Load Balancing (Scalability) | The API should have some form of load balancing in case many API calls are made at the same time |
| Model | High | Cluster Data | The model should be able to cluster the repositories based on the numeric metrics. |
| Model | low | Calculate Feature Importance | The model should calculate feature importance and return a list of top features to be used in the front end as suggestion tool. |
| Front End | High | Display Data | The Front End should display the clustered data the machine learning algorithm returns on a graph |
| Front End | High | Send/Retrieve Data | The Front End should send two metrics to the API with a POST request and recieve the data returned by the API |
| Front End | Med | Change Metrics | The Front End should allow the user to change the metrics that the clustered data is shown on |
| Front End | Med | Show Repo Name on Hover | The Front End should show a popup displaying the repository's name when the user hovers over it on the graph |