CODEX core use cases

This document contains the “key path” scenarios. The intent is to describe the primary user-flow through the application so that completeness can be tested before launch.

Second priorities: expand utility to larger data sets and make more robust. Critical features for key scenario/use-case

Explain-this UI

- Input: 2+ Features and 1 Selection
- Output: interactive decision tree, accuracy of predictions based on the Feature(s) used
- Visualize *why* the Selection is unique
- Explain *how* the answer was derived
- Save a Model that can be used to categorize another data set

Session/state-saving

- Save state and let me continue where I left off (state in URL? Save data on server? Cookies + server? Accounts?)

Heat maps

- Make heatmaps an overlay/mode of scatterplots (could initially at least highlight the Heatmaps where the Scatterplot selections are, so you can compare Scatterplots to Heatmaps)
- Fix Heatmap helper text to show full coordinates as well as the count/density in that heat-bin
- Make Heatmaps more like Density plot with more granular size of the heat-bins

Chart management and Interactions

- Make it easy to copy/paste the settings used for Clustering so that they can be used when in code later (this is not Code-exporter)
- the clusters produced by a clustering algorithm need to be grouped and have some indication of how they were created (e.g. settings for clustering). Need to separate each “set” of clusters in the UI so you can tell which sets logically go together (e.g. you shouldn’t mix a cluster from algo1 with a cluster from algo2)
- select-color on charts (red) is similar to some of the cluster chart colors (e.g. purply-red) so it’s hard to tell what’s selected across charts when the clustering color is active
- update hide/show on chart vs. selected (as input to algorithms) for Features and Selections
- Indicate which graphs and algorithms are possible when a Feature set is selected
- Box-zoom / Chart interaction improvements

MVP High-level features

1. Pose questions of the graphs so that they can answer things for you

- heat vs. scatter -> histogram

1. “Explain this”
- “Applied classification”
- Take a selection and :
- make decision-tree classifier on this selection vs. everything else
- visualize why it’s different or unique
- “which of the 1000 values matter?”
- LUKAS HAS AN EXAMPLE HE CODED
- also explain *how* the answer was derived

1.1. Filter out garbage
- what’s **good/bad**
- can’t do *anything* without filtering your data set first
- relationships between Features to make sure that it makes sense
- min/max/etc. for a Feature
- derived Features
- mask things out that are garbage
- anything empty/null etc.
- normalization
- manual filtration
- interactive sortable heatmap of values

1.5. Directed classification/regression
- don’t optimize hyper-parameters for me
- but GUIDE me

1.5. Train a model to predict X (Classification/Regression)
What did I learn? How well did it work? Cross-validate. Generalized test error…
even on the same features you can use the Classification as a way to LEARN ABOUT THE DATA

1.5. Reliability:
- Error handling so it’s not busted
- Saving sessions
- Selecting Features & Selections and interacting with algorithms and charts

2. “**More like this**”

2. Dimensionality reduction:
- “not minimal”
- high user need because most people won’t do it
- back-end needs this in order to do clustering, but the user doesn’t need it critically right now

3. Real-time algorithms that show each round so you can decide when to stop

CODEX “seed utility” (2) key path scenario

(1) The user goes to codex.jpl.nasa.gov and sees the CODEX user-interface. User Imports a file and decides to load the Chemcam demo data to explore— (2) during the load there’s a loading indicator where the Feature list will go (or where the file will be listed, if Feature list is moved).

(3) User plots four Features on Scatterplots (Ti vs. Al, Fe, Mg) to find outliers. (4) Each plot can sweep between Scatterplot mode and Heatmap (Contour?) mode to get a better sense of whether clusters of dots that look like outliers are actually dense.

(5) User will explore a few outlier data points to compare across the various Scatterplots by circling them and seeing where the groups lie across other graphs.

(6) User box-zooms into the region of interest and pans around to make sure that their selection is as intended.

(7) User flips one of the Scatterplots into Density Plot (Heat Map?) mode to make sure that the outlier is really an outlier group and not a dense cluster.

(8) User saves one of the outlier groups as a Selection called “what is this”.

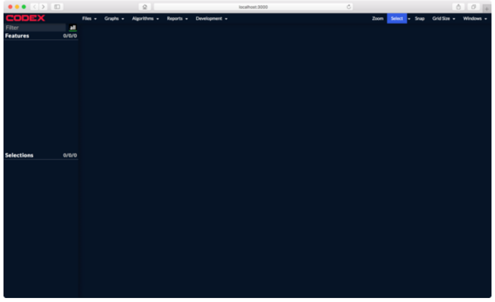
(9) User chooses the Explain This workflow and is prompted to include some Features and Selection. User selects the four Features and “What is this” Selection. [We also need a label…how is that defined?]

(10) The UI shows that the Explain This algorithm is running; When it’s done, the results show up.

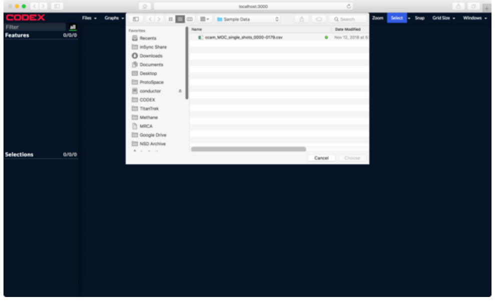
(11) User can quickly see how “What is this” Selection is distinct from the rest of the points in the file. This includes a decision tree, an accuracy rating (vs. depth of analysis), and a summary of how much each Feature contributed to the distinctiveness of the Selection. User can also quickly see how changing the tree depth would impact the decision tree.

Key path scenario

1. Open the URL



2. Open ChemCam example data and view loading indicator



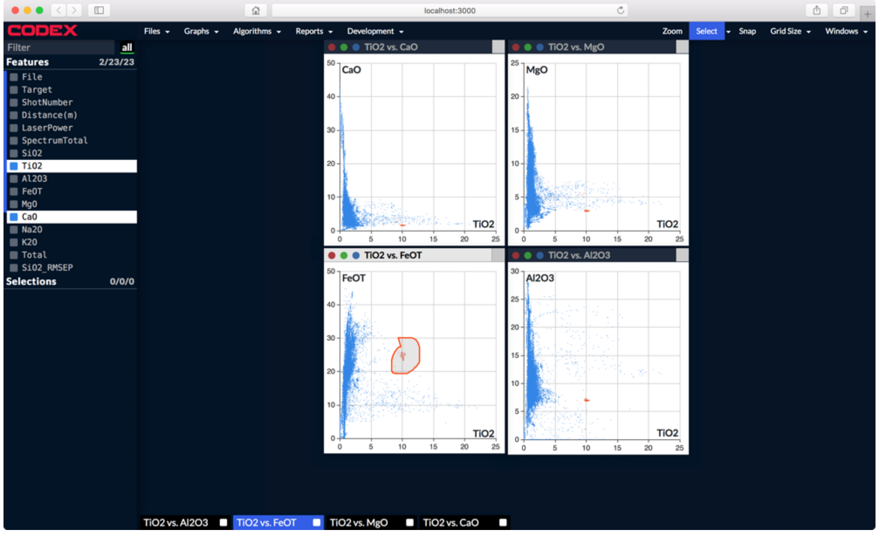
2b. Features are loaded into left hand panel

3. Make some scatter plots and heatmaps
V1: Feature -> Chart (can quickly make multiple scatterplots like this)
V2: New chart -> Add Features (might need a Correlogram to do n x m scatterplots)

4. Sweet between Scatterplot and Heatmap modes

5. Brushing to compare points across plots.

6. Zoom in on a chart (box-zoom and pan)

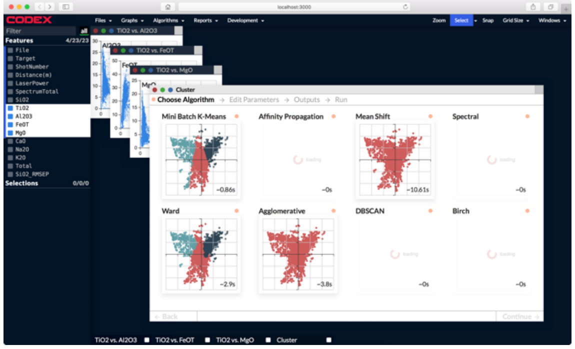


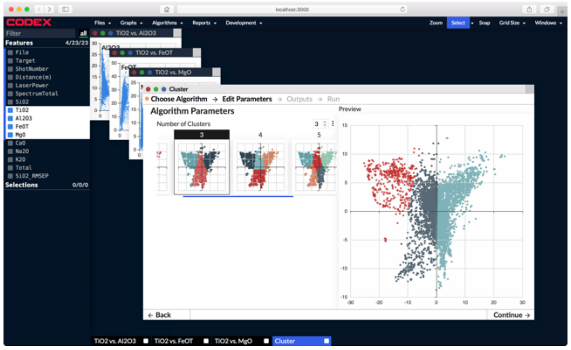
7. Flip one of the scatterplots into a heat map/density map

8. Save the current Selection called “what is this”

9. Select those same four Features and one Selection and “Explain This”, plus the “labels” Category Label.

13. choose parameters

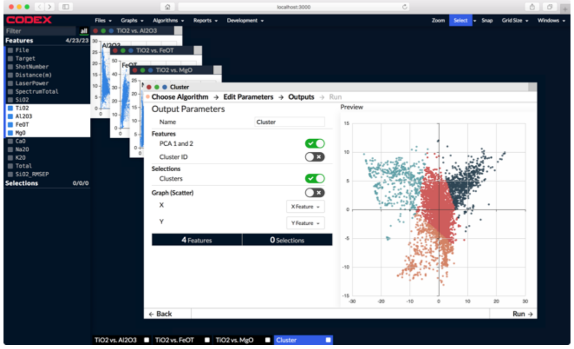




Algorithm Parameters

Number of Clusters 3

14. start clustering and close the app (Save the user’s place) (Do we need to also allow intentional named-session saves on top of an auto-save of the current state?)



Output Parameters

Name Cluster

Features

PCA 1 and 2 ☑

Cluster ID ☒

Selections

Clusters ☑

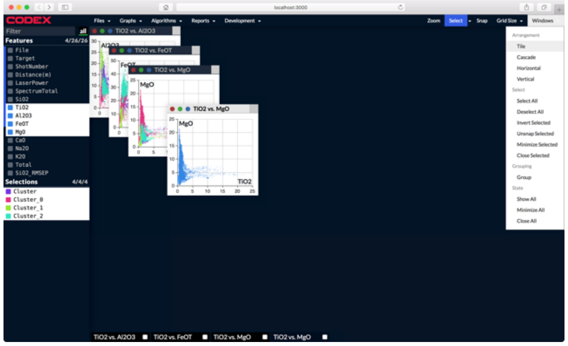
Graph (Scatter) ☒

X X Feature

Y Y Feature

15. User returns to completed clustering run

16. Color original scatter plots by the new clusters. Tile windows.



17. User checks the settings used to generate the clusters by clicking widget icon in the title of the Cluster-group header in the Selection panel.

Next the user wants to compare SL classification to the Clustering.
18. Supervised learning (SL): classification. User selects Features and Selections/rows. Options of various SL algorithms with previews (like Clustering, but no visualization). Once processed, an output is a Model (need to figure out what stats/summary visualization to give here).