SPC-DT Feature Wishlist

1. Minimize overlap of individual plots automatically (ideas)
2. Condense points that are near the center (i.e.,not edges) of rectangles. Coding
3. Resizable window (Coding)
4. Add frames between gray areas (Coding)
5. Confusion matrix pop out window for larger number of classes (Coding)
6. Show/output the worst-performing rectangle and cases that end up in it
   1. Auto-select decision area with the lowest purity (most misclassified) ideas not too small). (adjustable threshold for size)
7. Mark rectangles/cases as unclassifiable and/or need more data (new conceptually, refuse)
   1. Mark borderline cases that need more information (borderline)
8. User add a classification zone (N/A,. Cross area)
9. “Create case” where the user can enter/create their own case
10. Design new rules based on selection rectangle (DT generalization)
11. Output the decision tree with adjusted thresholds (store the result)
12. Fix zoom / pan (Coding)
13. Remove unnecessary buttons (Coding)
14. Make background density coloring change while adjusting thresholds (Coding)
15. While drawing a rectangle, leave markers behind to tell where clicked points are.
16. Make user-drawn rectangles draggable (activation point)
17. Fix button active states (GUI)
18. Put project on research assistant GitHub (nonexistent at the time of writing)
19. Make documentation
20. Experiments with new data
21. Support DTs with more than two branches
22. Random Forest: (generate logical rules that don’t have a shared root, ideas)
23. Make GitHub Organization
24. Make single attribute plots one-dimensional (no pairing) (coding)
    1. Make sure that vertical/horizontal spread is considered.
    2. Make sure swapping/inverting still work
25. Make zones object-oriented
26. Remove white zones on plots
27. Make Plots object oriented
28. Add ability to swap out attributes on each plot.
29. In presentation, distinguish between our decision trees:
    1. ML can’t add weights
       1. (3x1>5) ^ (4x2>7)
       2. (X1-**5/3** > 0) ^ (X2 – **7/4** > 0) 🡪 modify thresholds
       3. (**3**\*X1-5 > 0) ^ (**4**\*X2 – 7 > 0) 🡪 modify attributes. 🡪easier to understand
          1. 3 and 4 used for pre DT generation.
          2. Optimization problem.
          3. Though this will still probably generate the same tree
       4. Y1 = 3x1, y2 = 4x2
    2. Point A to terminal node -> then find branch where least cost
30. Explore the importance of attribute order.
    1. Sometimes attribute order in DT can be different from domain expert attribute importance.
       1. If user says “attribute 8” is most important, we can put it to the root.
    2. Standard DT generation algorithms do not allow the user to select the root node (or order)
       1. Random Forest does, but only because it generates multiple trees.
       2. If we want to put X2 to the root instead of X1, we compute X2 < 7 for the root. 350 instances with X2 < 7. 380 cases X2 >= 7. Build one tree with 350, then another 380 >= 7. Then build tree with only those cases in Tanagra / Scikit-Learn. Manual trick.
31. Altering DT tree:
    1. Implement Gini measurement, then DT generation.
32. Users need some sort of guidance to understand the benefit of our visualization vs traditional, especially for uses in ML.
33. Add a “hide data” button that hides all data.
    1. User can select what attributes / cases to show.
34. We could hardcode a dataset to show on startup (maybe)

IN PROGRESS

1. Rectangles (6, 7, 8)
2. Brainstorm automatic overlap minimization ideas
3. Resize window
4. Make zones object-oriented

TODO LIST

1. Fix sues with mushroom dataset

For 1 negate vertical coord.