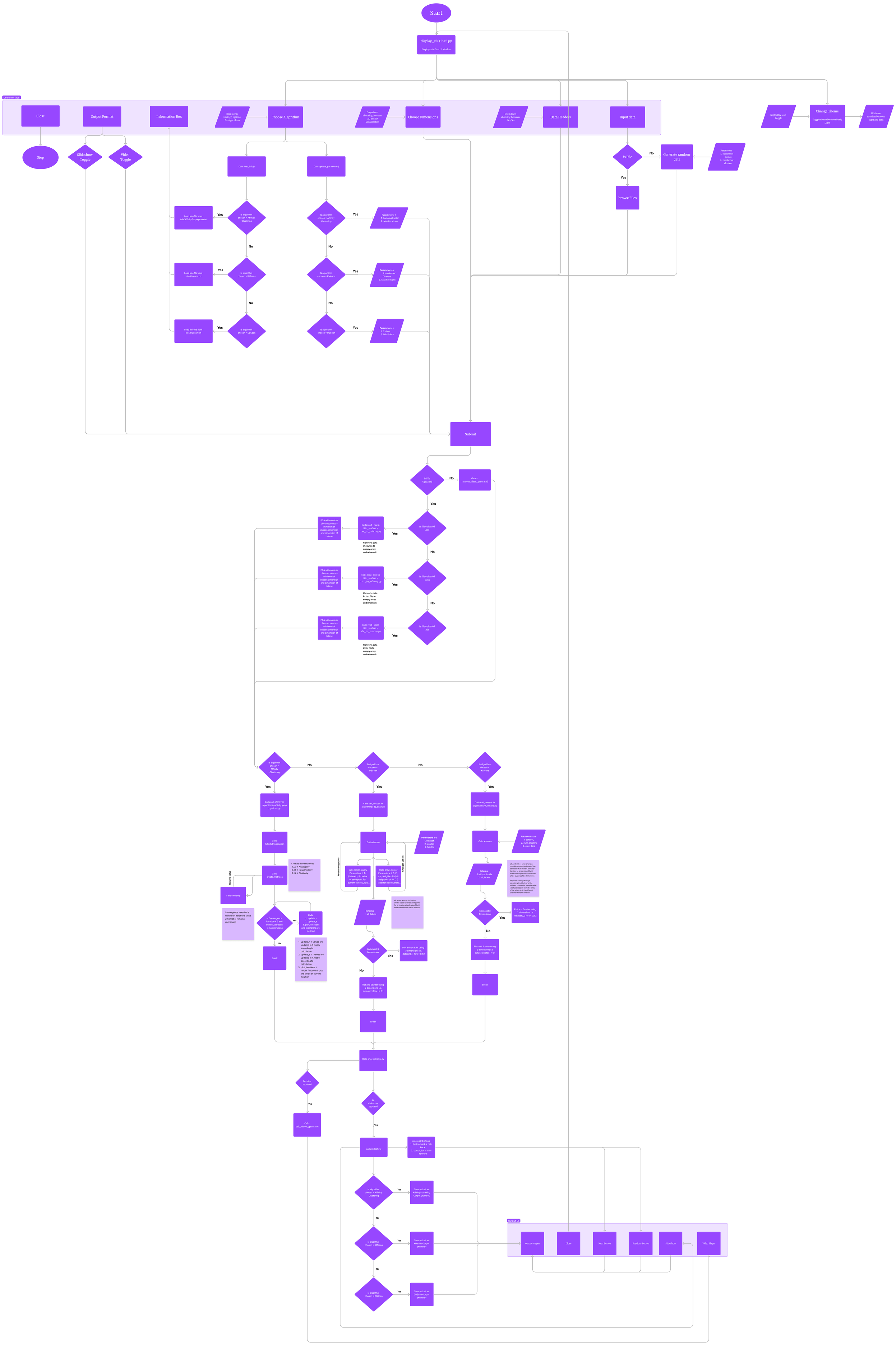
Start
display_ui() in ui.py
Displays the first UI window
User Interface
Close
Output Format
Information Box
Drop down having 3 options for algorithms
Choose Algorithm
Drop down choosing between 2D and 3D Visualization
Choose Dimensions
Drop down choosing between Yes/No
Data Headers
Input data
Night/Day Icon Toggle
Change Theme
Toggle theme between Dark/Light
UI theme switches between light and dark
Stop
Slideshow Toggle
Video Toggle
Calls load_info()
Calls update_parameter()
is File
No
Generate random data
Parameters
1. number of points
2. number of clusters
Yes
browseFiles
Load info file from info/AffinityPropogation.txt
Is algorithm chosen = Affinity Clustering
Parameters →
1. Damping Factor
2. Max Iterations
Load info file from info/Kmeans.txt
Is algorithm chosen = KMeans
Parameters →
1. Number of Clusters
2. Max Iterations
Load info file from info/DBscan.txt
Is algorithm chosen = DBScan
Parameters →
1. Epsilon
2. Min Points
Submit
Is File Uploaded
data = random_data_generated
Is file uploaded .csv
Calls read_csv in file_readers > csv_to_ndarray.py
Converts data in csv file to numpy array and returns it
PCA with number of components = minimum of chosen dimension and dimension of dataset
Is file uploaded .xlsx
Calls read_xlsx in file_readers > xlsx_to_ndarray.py
Converts data in xlsx file to numpy array and returns it
Is file uploaded .xls
Calls read_xls in file_readers > xls_to_ndarray.py
Converts data in xls file to numpy array and returns it
Is algorithm chosen = Affinity Clustering
Is algorithm chosen = DBScan
Is algorithm chosen = KMeans
Calls call_affinity in algorithms>affinity_propogations.py
Calls call_dbscan in algorithms>db_scan.py
Calls call_kmeans in algorithms>k_means.py
Calls AffinityPropagation
Calls dbscan
Parameters are
1. dataset,
2. epsilon
3. MinPts
Calls kmeans
Parameters are
1. dataset,
2. num_clusters
3. max_iters
Calls create_matrices
Creates three matrices
1. A → Availability
2. R → Responsibility
3. S → Similarity
Returns value
Calls similarity
Calls region_query
Parameters → D (dataset), P (Index of seed point for current cluster), eps
Calls grow_cluster
Parameters → D, P, eps, NeighborPts(all neighbors of P), C (label for new cluster)
Returns neighbors
Changes Labels
Returns
1. all_centroids
2. all_labels
all_centroids → array of arrays containing the co-ordinates of the centroids of all clusters for every iteration i.e all_centroids[i] will store the array of the co-ordinates of the clusters of the ith iteration
all_labels → array of arrays containing the labels of all the different clusters for every iteration i.e all_labels[i] will store the array of the labels of all the different clusters of the ith iteration
Convergence iteration is number of iterations since which label remains unchanged
Is Convergence iteration < 5 and current_iteration < max iterations
Calls
1. update_r
2. update_a
3. plot_iterations
and exemplars are defined
1. update_r → values are updated in R matrix according to calculation
2. update_a → values are updated in A matrix according to calculation
3. plot_iterations → helper function to plot the labels of current iteration
Break
all_labels → array storing the cluster labels for all dataset points for all iterations i.e all_labels[i] will store the labels for the ith iteration
Is dataset 3 Dimensional
Plot and Scatter using 3 dimensions i.e. dataset[:,i] for i = 0,1,2
Plot and Scatter using 2 dimensions i.e. dataset[:,i] for i = 0,1
Calls after_ui() in ui.py
Is video required
Calls call_video_generator
Is slideshow required
calls slideshow
creates 2 buttons
1. button_back → calls back
2. button_for → calls forward
Save output as AffinityClustering Output (number)
Save output as KMeans Output (number)
Save output as DBScan Output (number)
Output UI
Output Images
Close
Next Button
Previous Button
Slideshow
Video Player