Machine Learning For Design

Lecture 9 - Designing And Develop Machine Learning Models / Part 3

Alessandro Bozzon
XX/03/2022

mlfd-io@tudelft.nl
www.ml4design.com

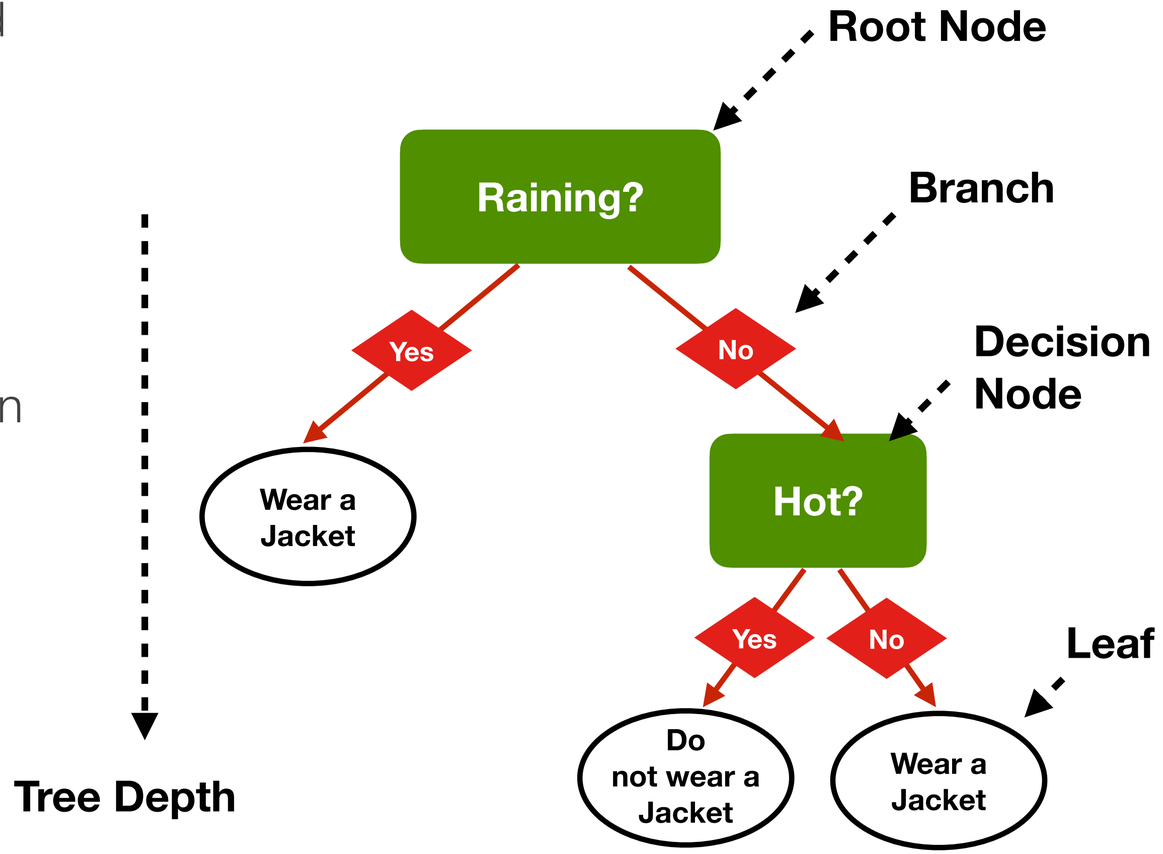
Decision Trees

Decision Trees

- Machine learning models used both for **classification** and **regression**
- Trained with labelled data (**supervised learning**)
  - classes —> classification
  - values —> regression

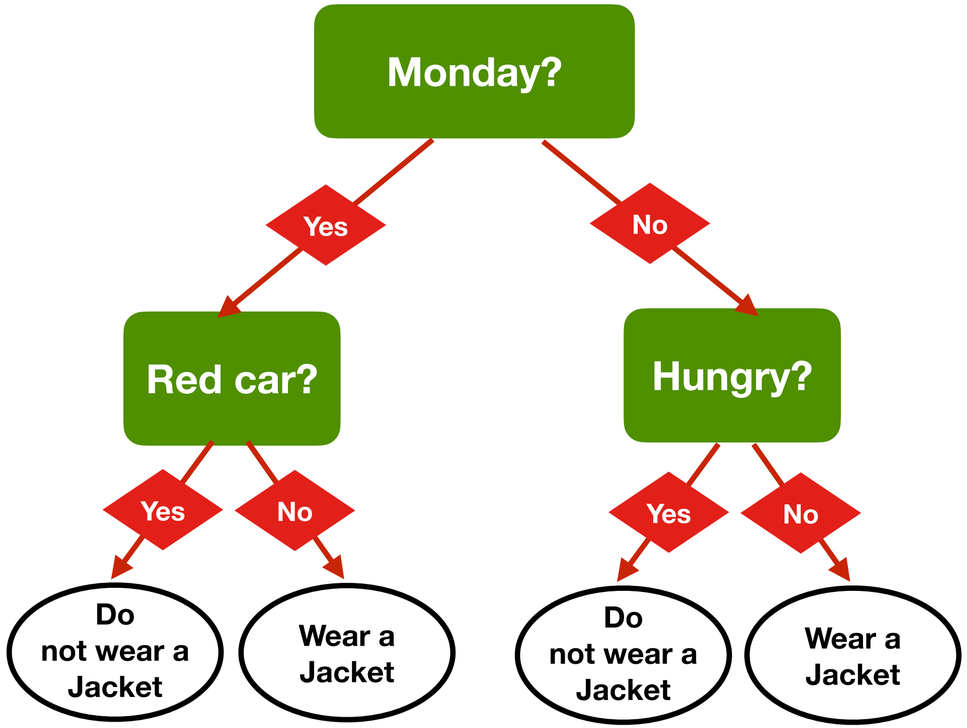
- A very simple model that resembles human reasoning when making predictions:
  - Answering a lot of yes/no questions based on feature values

- Problems:
  - Which questions to answer?
  - How many questions? (Tree depth)
  - In which order?



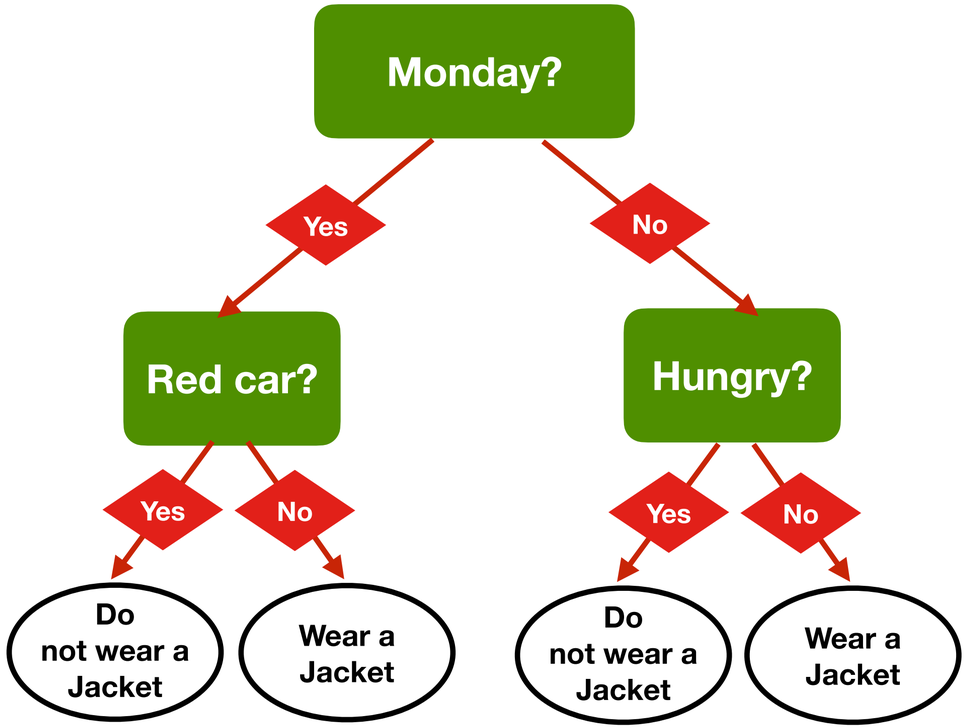
Same Problem, Multiple Trees

- Feature space
  - Am I hungry?
  - Is there a red car outside?
  - Is it Monday?
  - Is it raining?
  - Is it cold outside?



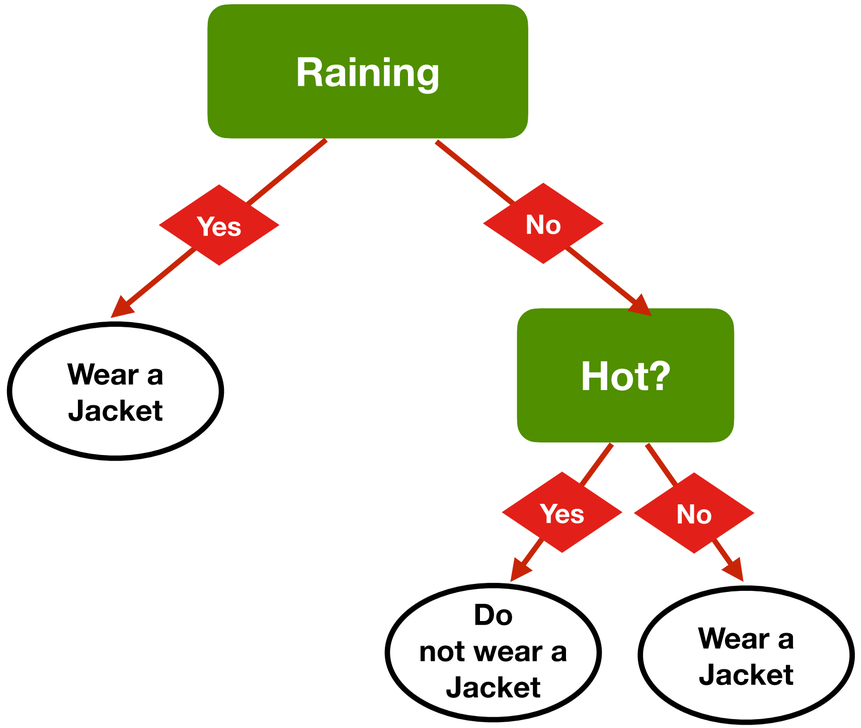
Same Problem, Multiple Trees

- Feature space
  - ~~Am I hungry?~~
  - ~~Is there a red car outside?~~
  - ~~Is it Monday?~~
  - Is it raining?
  - Is it cold outside?

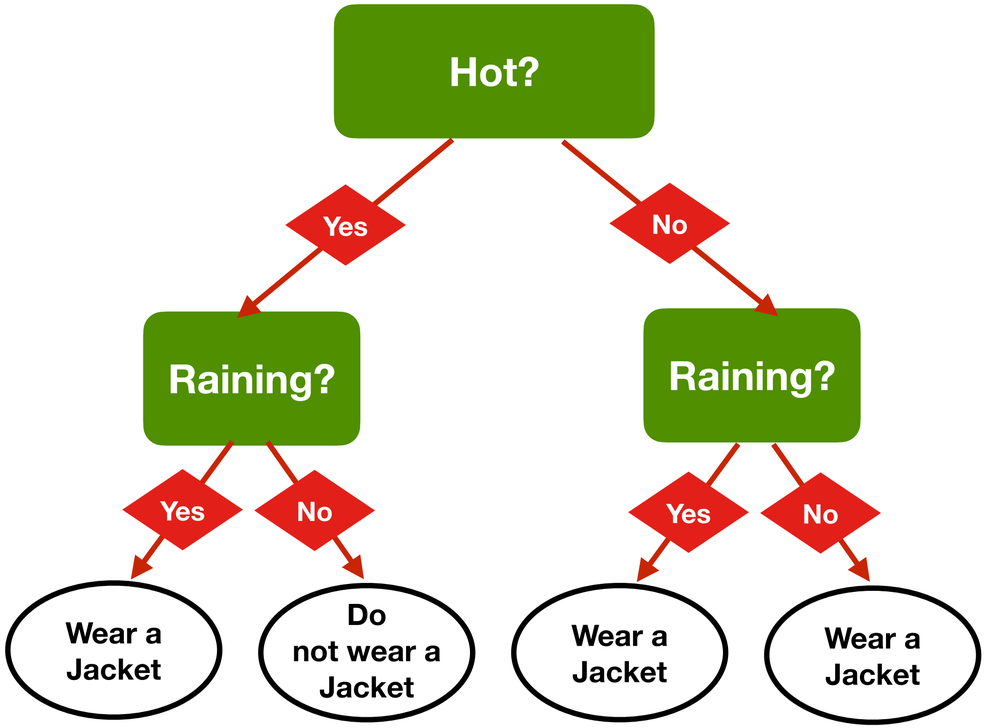


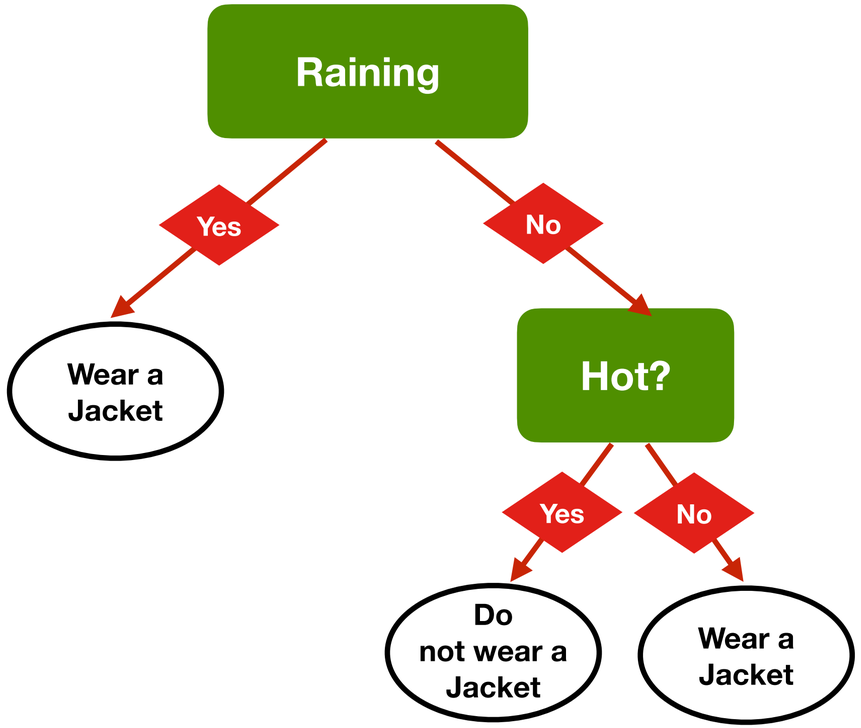
Same Problem, Multiple Trees

- Feature space
  - ~~Am I hungry?~~
  - ~~Is there a red car outside?~~
  - ~~Is it Monday?~~
  - Is it raining?
  - Is it cold outside?

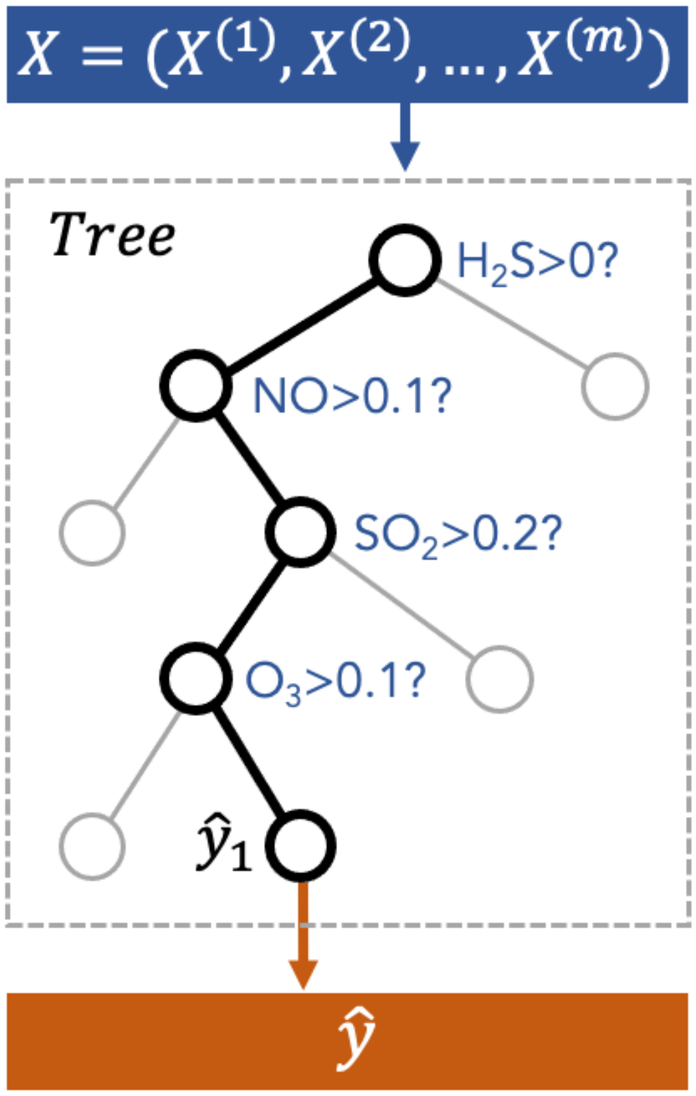


Same decision, different trees





Tutorial 3



PM, SO_2, CO, NO, NO_2, O_3, H_2S, and wind information

Prediction of bad smell (yes/no)

How to decide the best question to ask?

- **3 metrics**
  - **Accuracy**
    - Which question helps me be **correct** more often?
  - **Gini Impurity Index**
    - A measure of *diversity* in a dataset —> diversity of classes in a given leaf node
      - *index = 0* means that all the items in a leaf node have the same class
    - Which question helps me obtain the lowest average **Gini impurity Index**?
  - **Entropy**
    - Another measure of *diversity* linked to information theory
    - Which question helps me obtain the lowest average **entropy**?

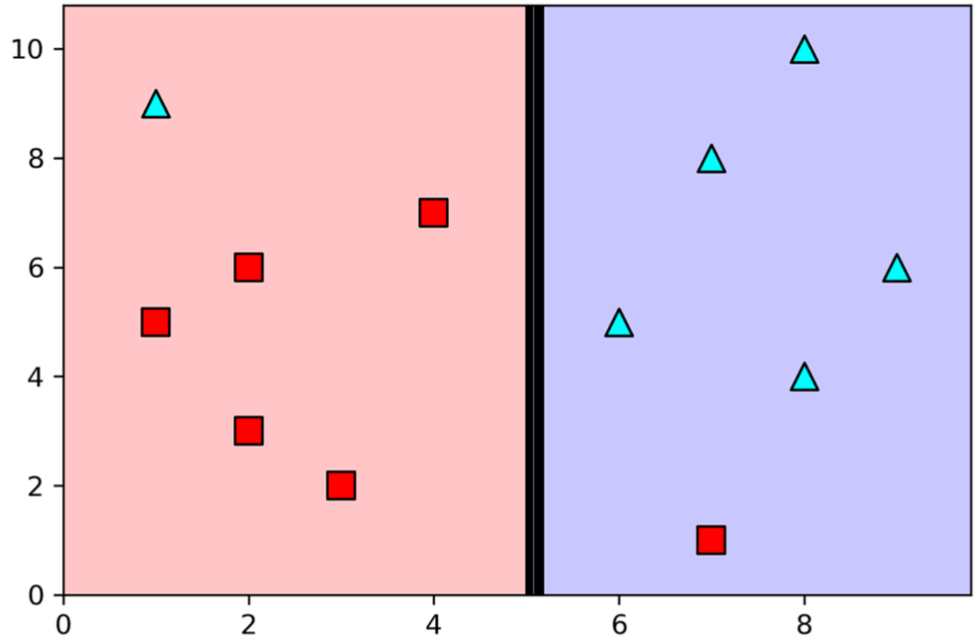
Building the tree (pseudo-code)

- **Add a root node, and associate it with the entire dataset**
  - This node has level 0. Call it a leaf node

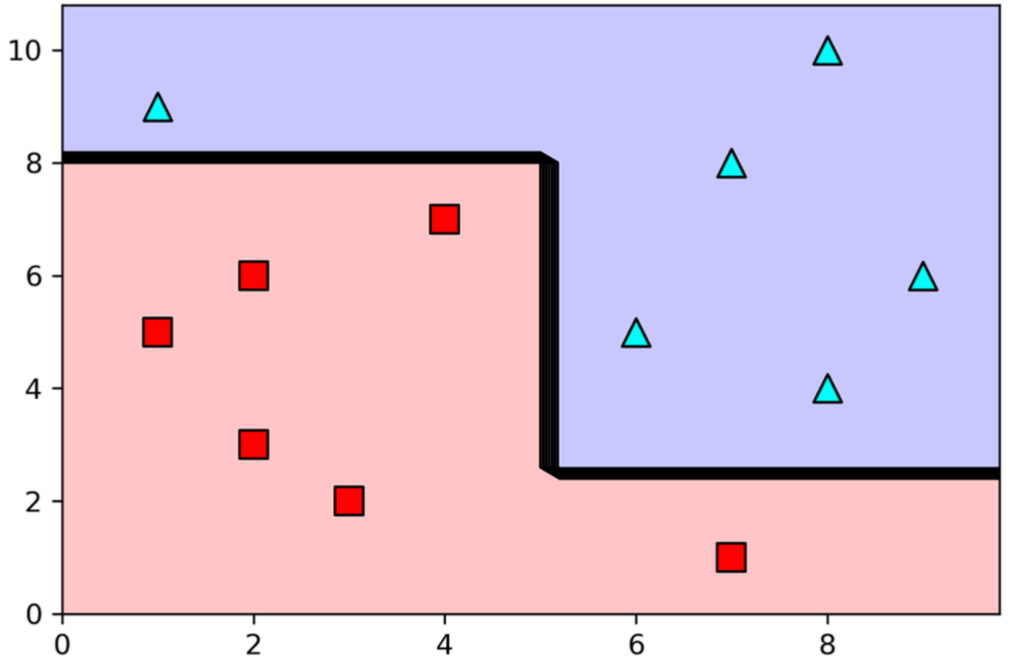
Hyperparameter: tree depth
Stopping condition

- **Repeat until the stopping conditions are met at every leaf node**
  - Pick one of the leaf nodes at the highest level
  - Go through all the features, and select the one that splits the samples corresponding to that node in an optimal way, according to the selected metric.
    - Associate that feature to the node
  - This feature splits the dataset into two branches
    - Create two new leaf nodes, one for each branch
    - Associate the corresponding samples to each of the nodes
  - If the stopping conditions allow a split, turn the node into a decision node, and add two new leaf nodes underneath it
    - If the level of the node is i, the two new leaf nodes are at level i + 1
  - If the stopping conditions don't allow a split, the node becomes a leaf node
    - Associate the most common label among its samples
    - That label is the prediction at the leaf

A geometrical perspective



- Step 1 - Select the first question
- **X >= 5**
  - Best possible prediction accuracy with one feature



- Step 2 - Iterate
- **x < 5 & y< 8; x >=5 & y>=2**
  - Perfect split of the feature space

Decision Trees: Pros and Cons

- **PROs**
  - Simple to understand and to interpret. Trees can be visualised
  - Requires little data preparation. Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed
  - Able to handle both numerical and categorical data

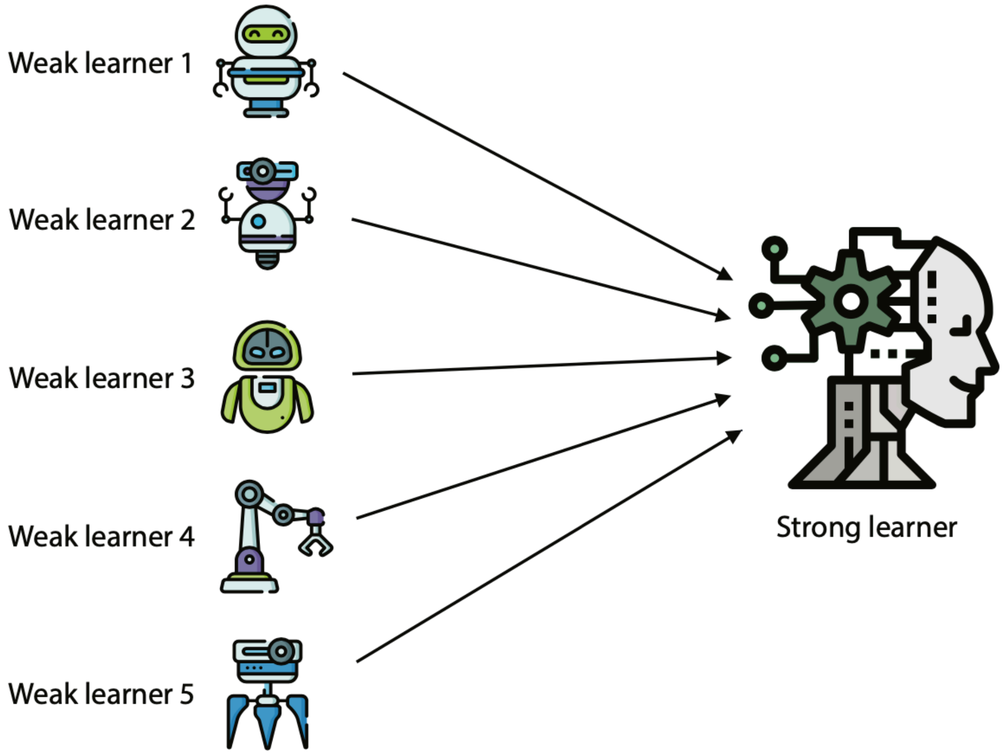
- **Cons**
  - Possible to create over-complex trees that do not generalise the data well
    - overfitting
  - Unstable —> small variations in the data might result in a completely different tree being generated
  - Biased trees if some classes dominate

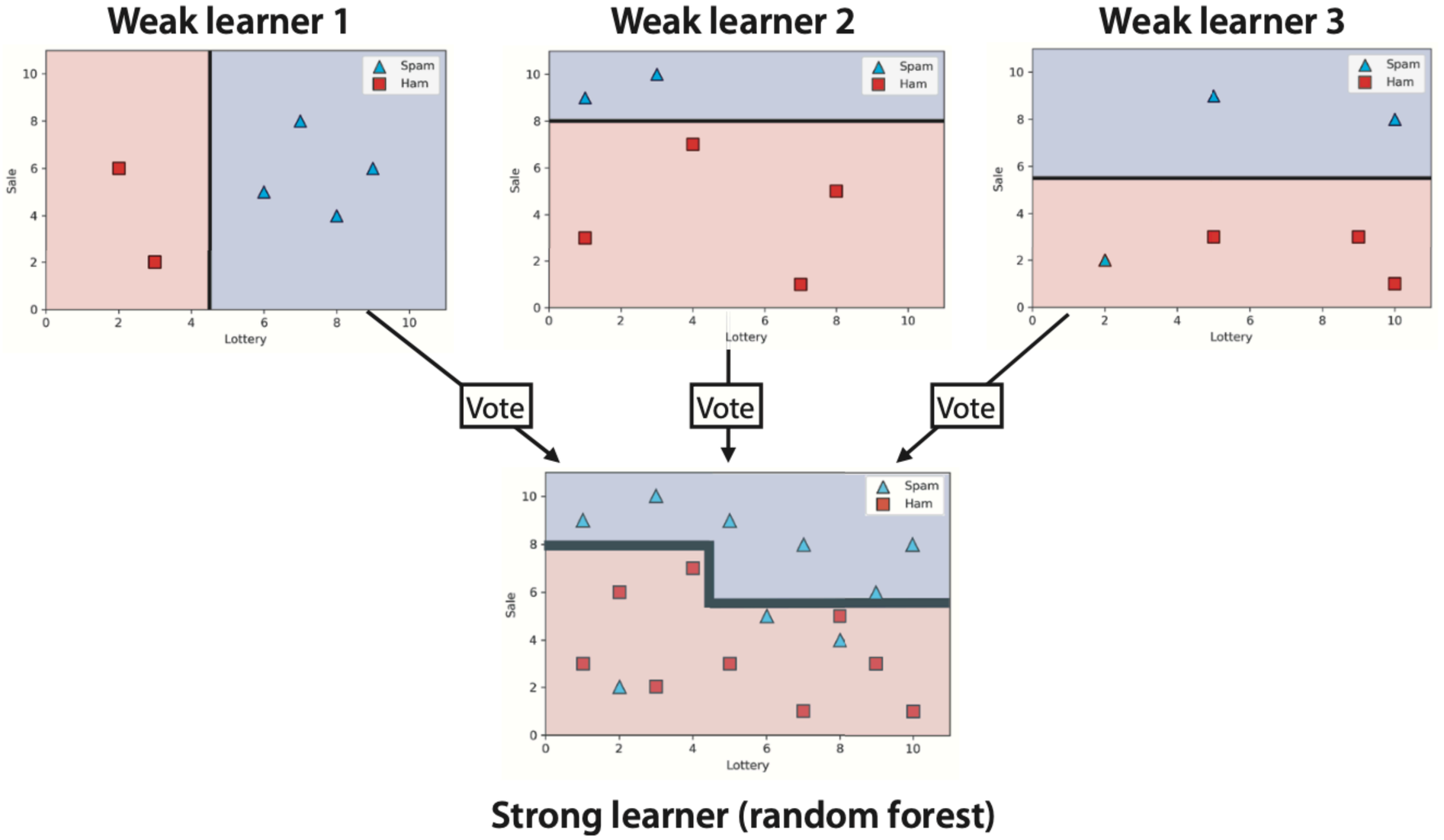
https://scikit-learn.org/stable/modules/tree.html

Ensemble learning: Random Forest

- Idea: combine several “weak” learners to build a strong learner
  - Build random training sets from the dataset
  - Train a different model on each of the sets
    - weak learners
  - Combination the weak models by voting (if it is a classification model) or averaging the predictions (if it is a regression model)
    - For any input, each of the weak learners predicts a value
    - The most common output (or the average) is the output of the strong learner

- Random Forest
  - Weak learners are **decision trees**




Weak learner 1
Weak learner 2
Weak learner 3
Spam
Ham
Sale
Lottery
Vote
Vote
Vote
Strong learner (random forest)

Clustering

What is clustering?

- Grouping items that “belong together” (i.e. have similar features)
- **Unsupervised learning**: we only use data features, not the labels
- We can detect patterns
  - Group emails or search results
  - Customer shopping patterns
  - Regions of images
- Useful when don’t know what you’re looking for
  - But: can get gibberish
- If the goal is classification, we can later ask a human to label each group (cluster)

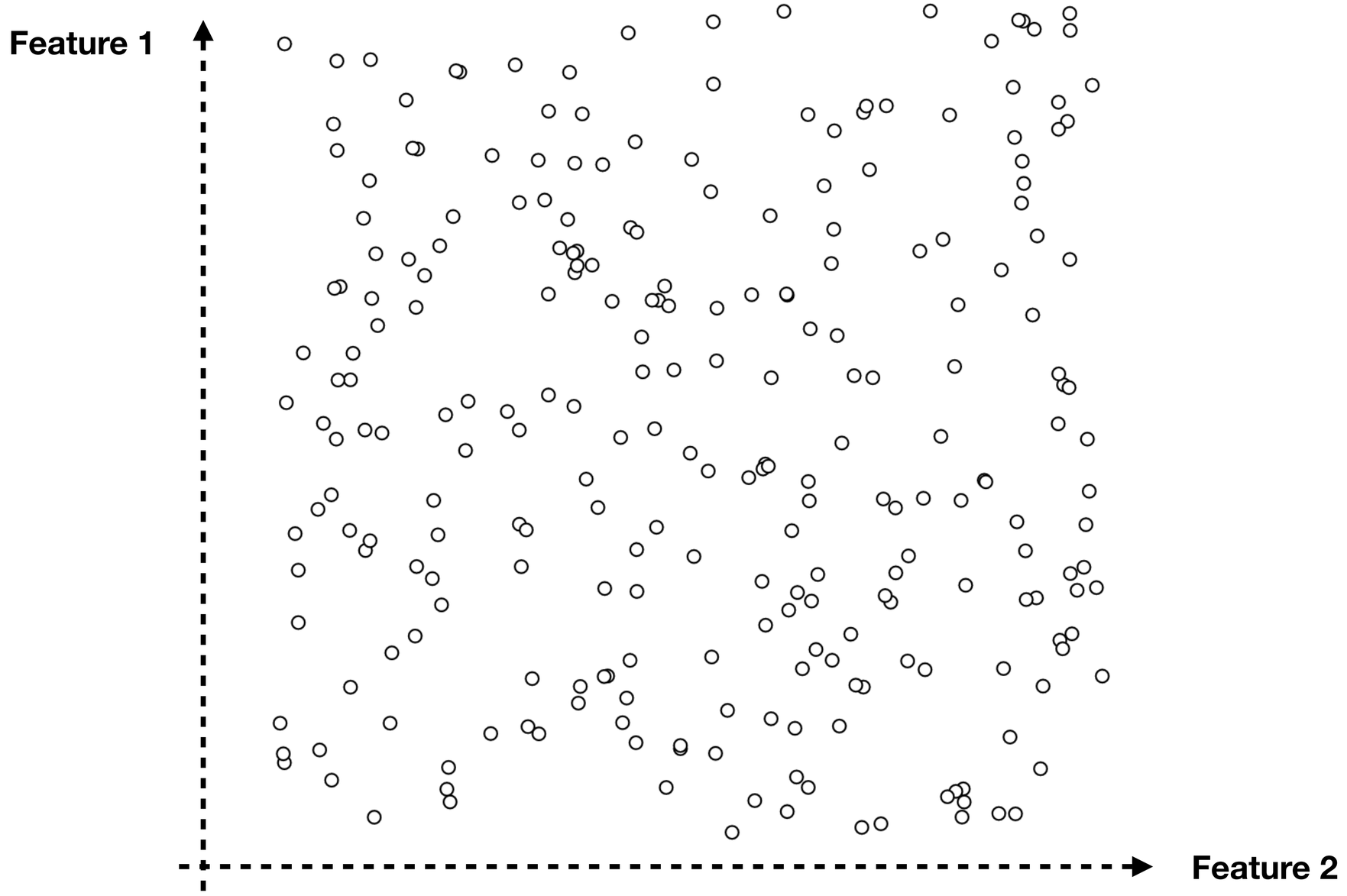
Why do we cluster?

- Summarizing data
  - Look at large amounts of data
  - Represent a large continuous vector with the cluster number
- Counting
  - Computing feature histograms
- Prediction
  - Images in the same cluster may have the same labels
- Segmentation
  - Separate the image into different regions

K-Means

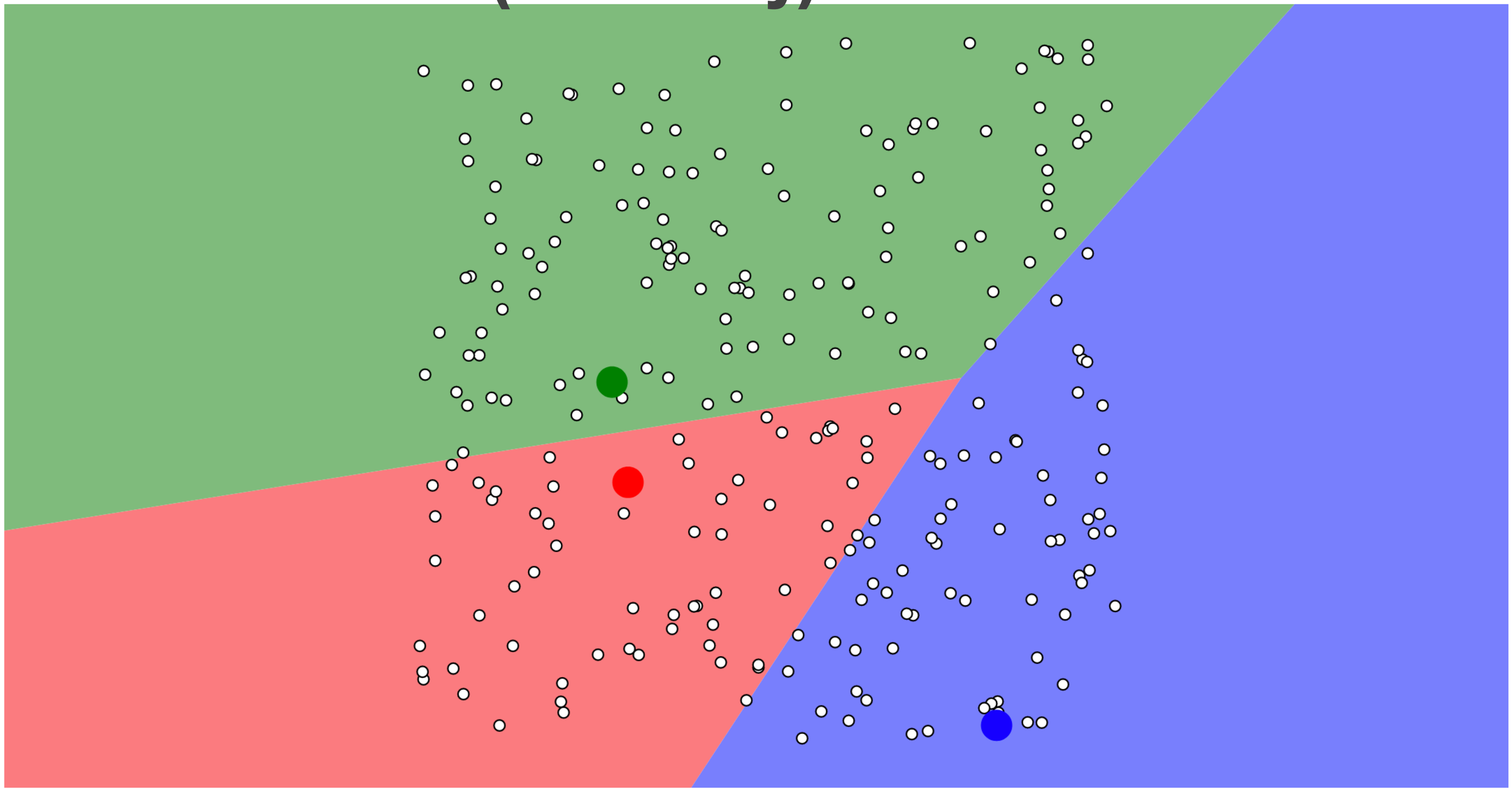
- An iterative clustering algorithm
  - **Initialize**: Pick K random points as cluster centres
  - **Alternate**:
    - Assign data points to the closest cluster centre
    - Change the cluster centre to the average of its assigned points
  - **Stop** when no points' assignments change

Data items distribution

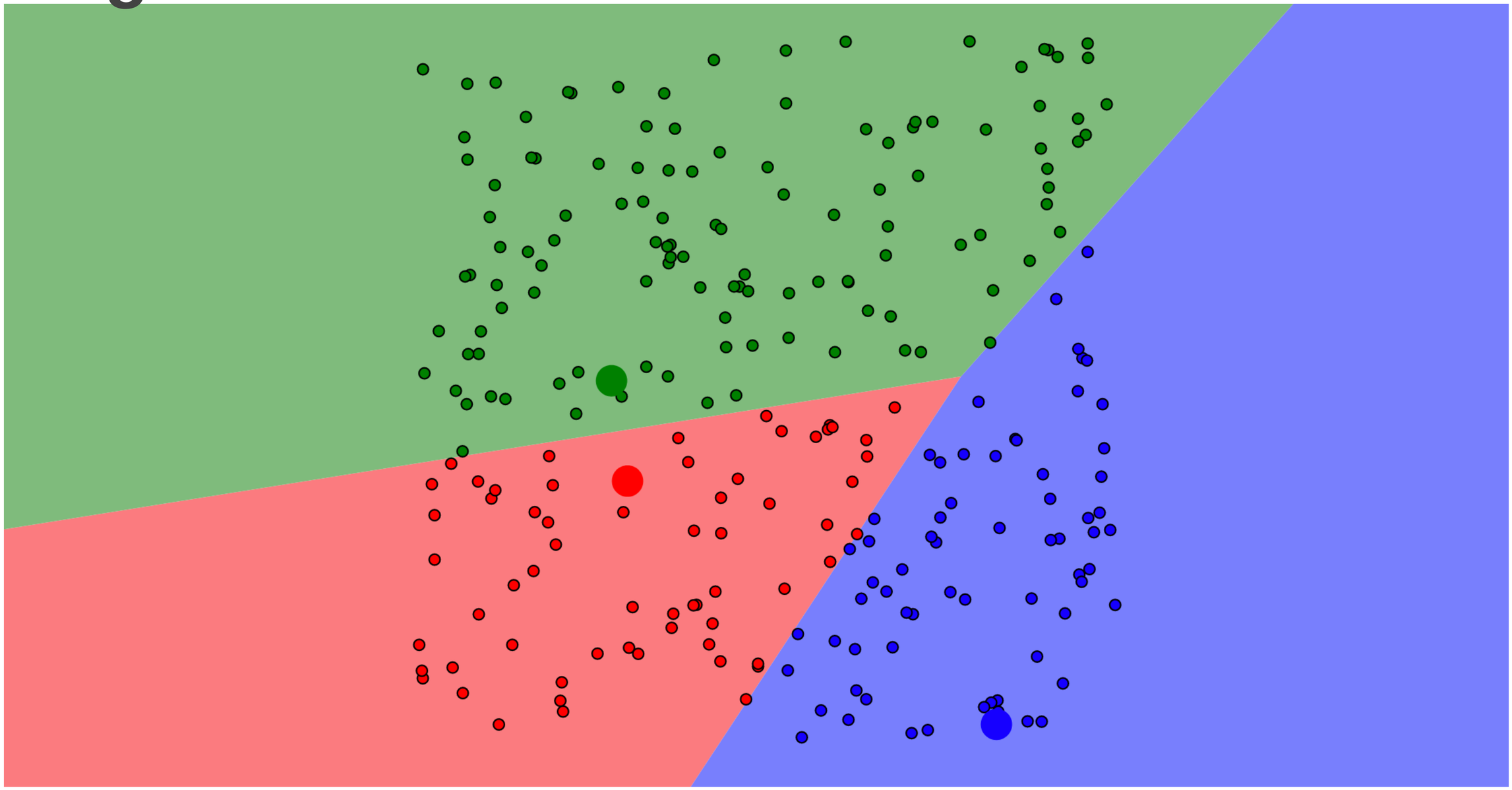


https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

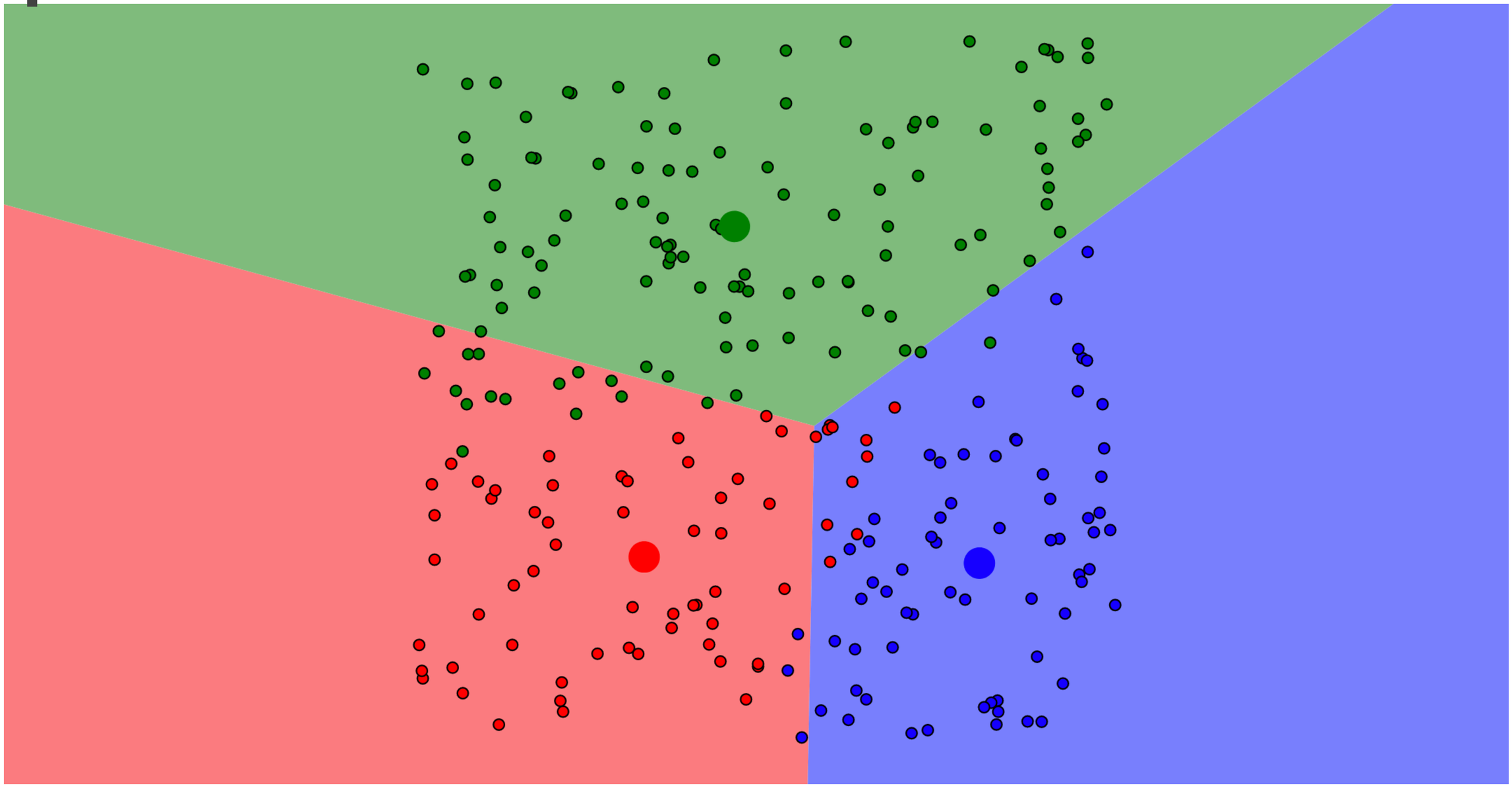
Add 3 Centroids (randomly)



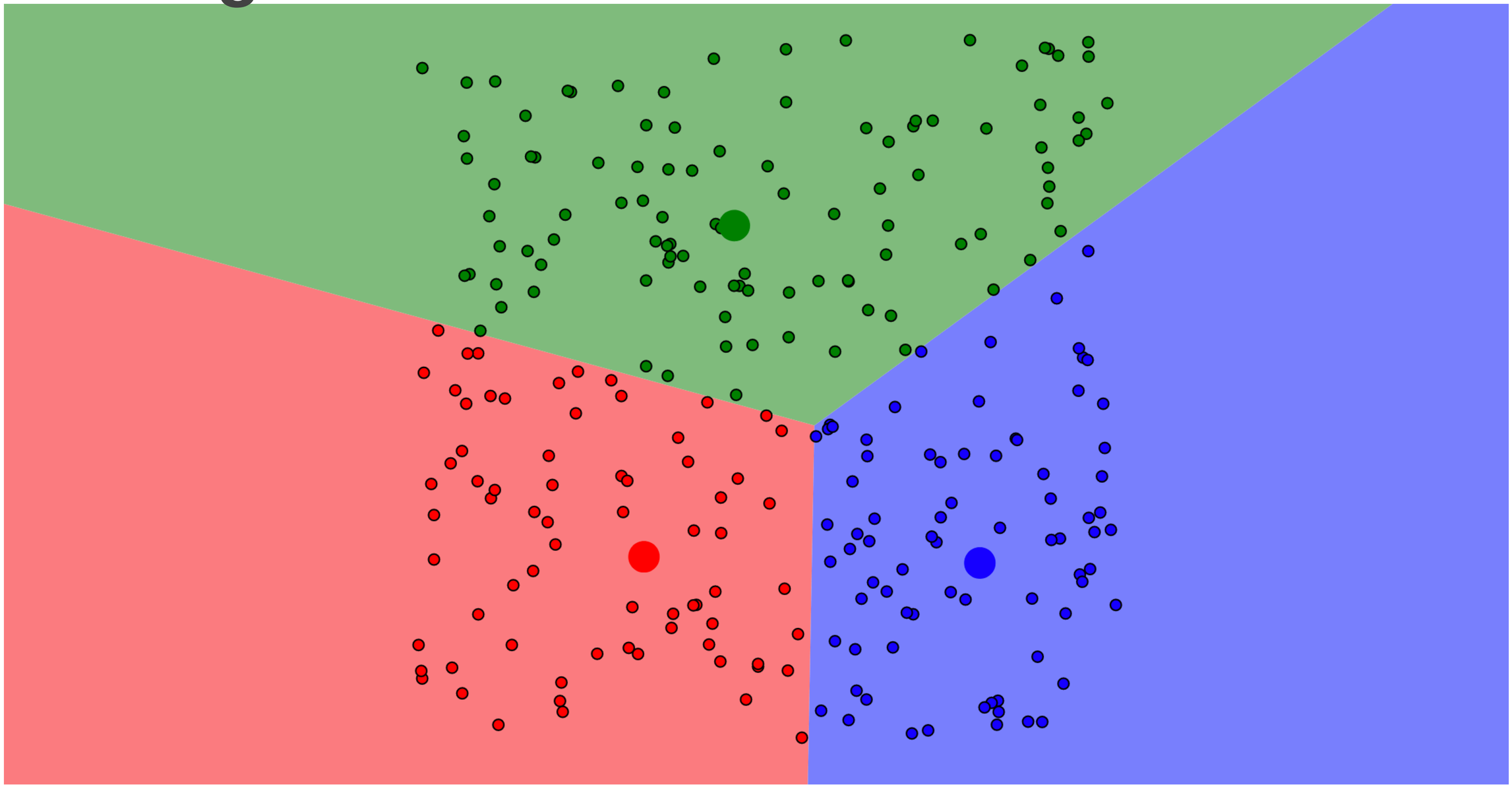
Assign Data Points



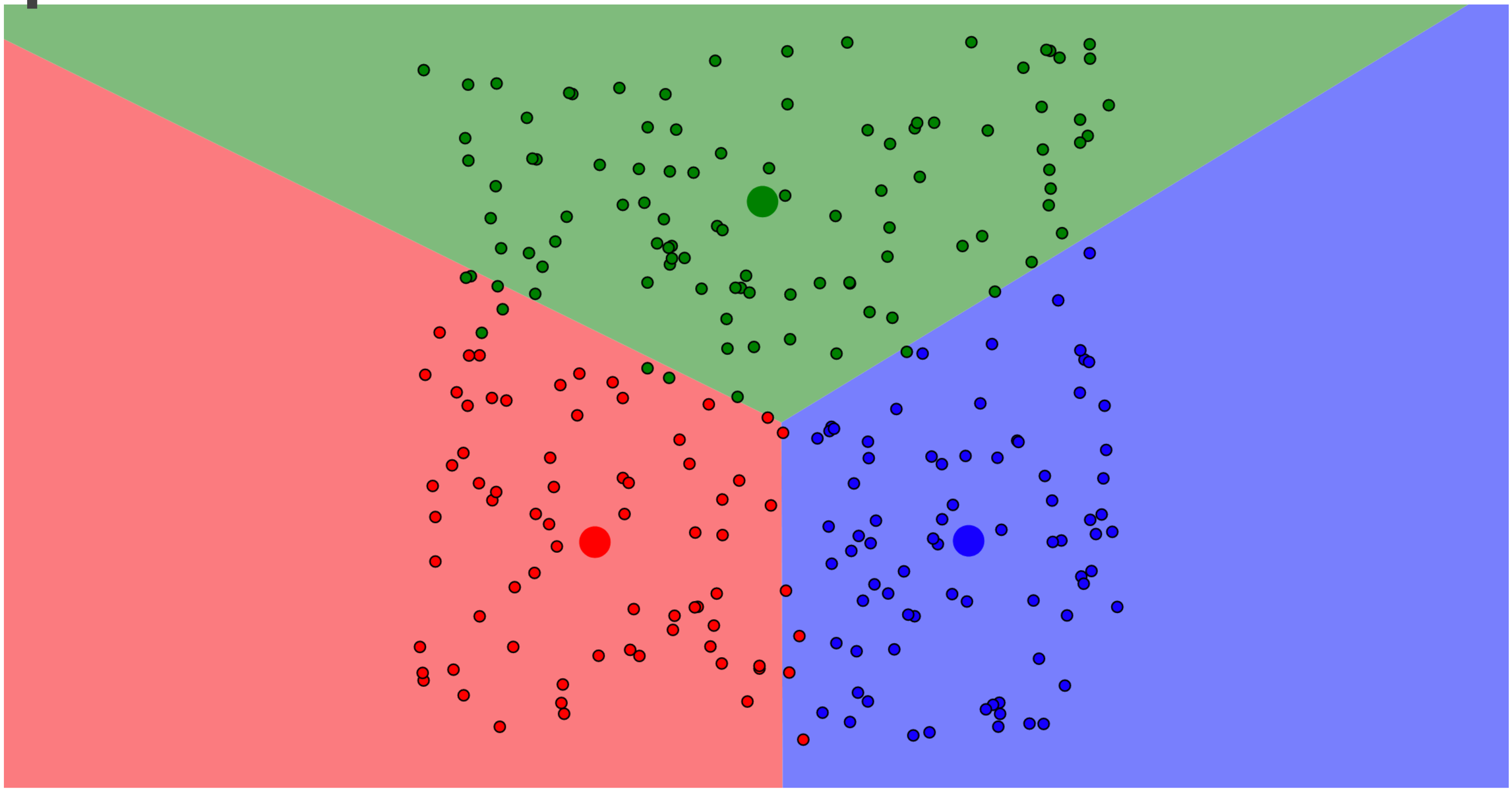
Update Centroids



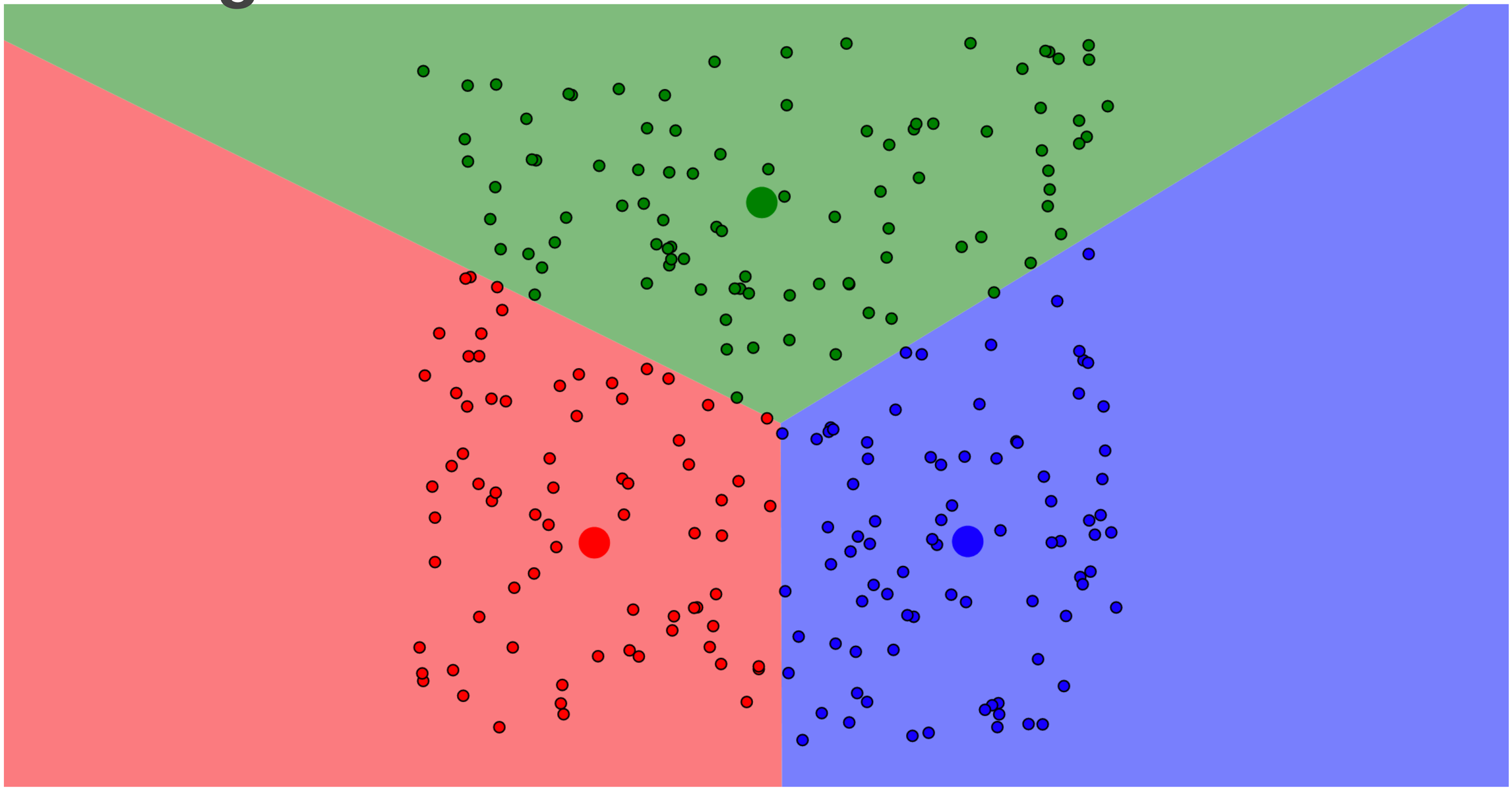
Re-Assign Data Points



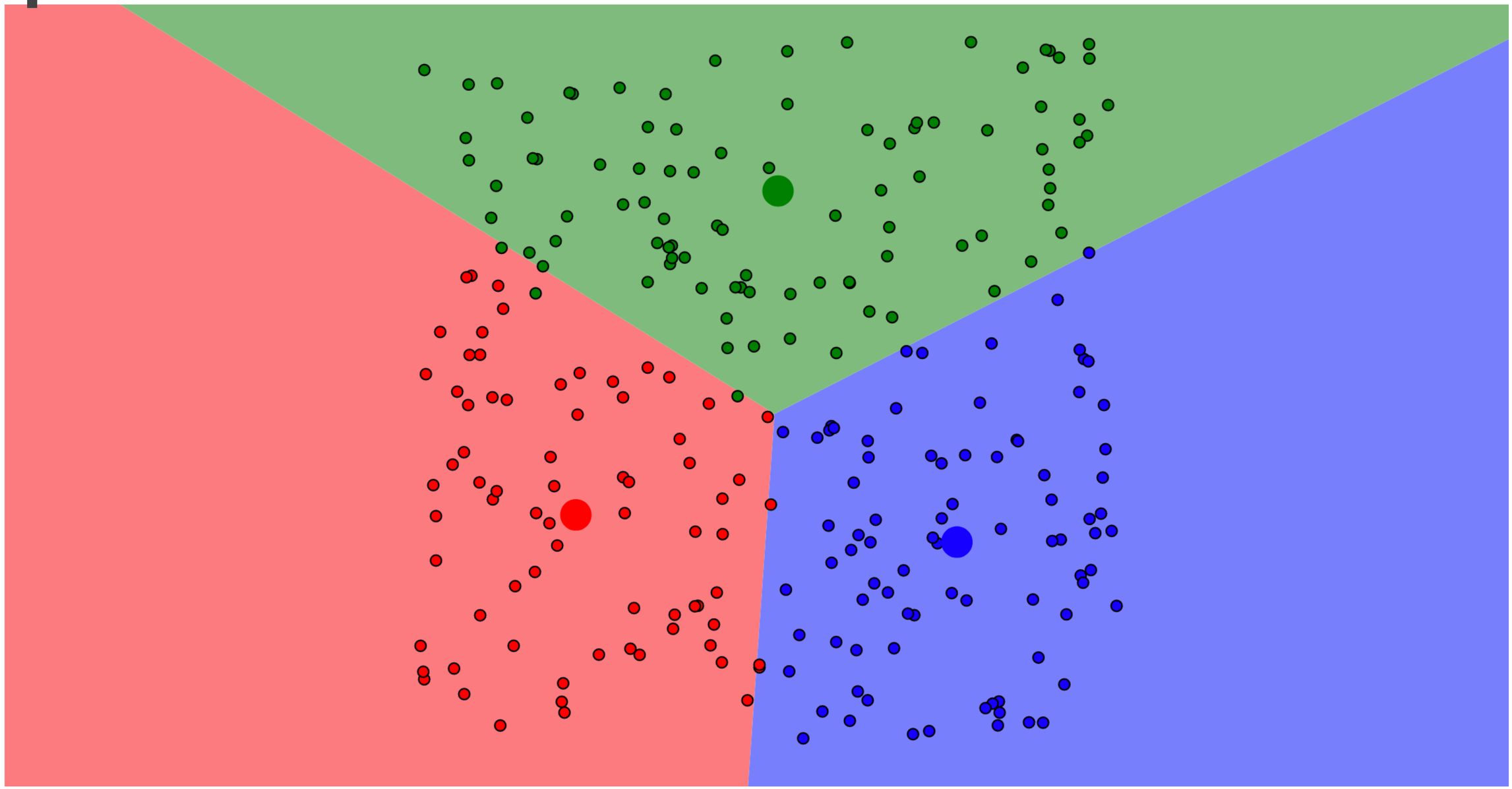
Update Centroids



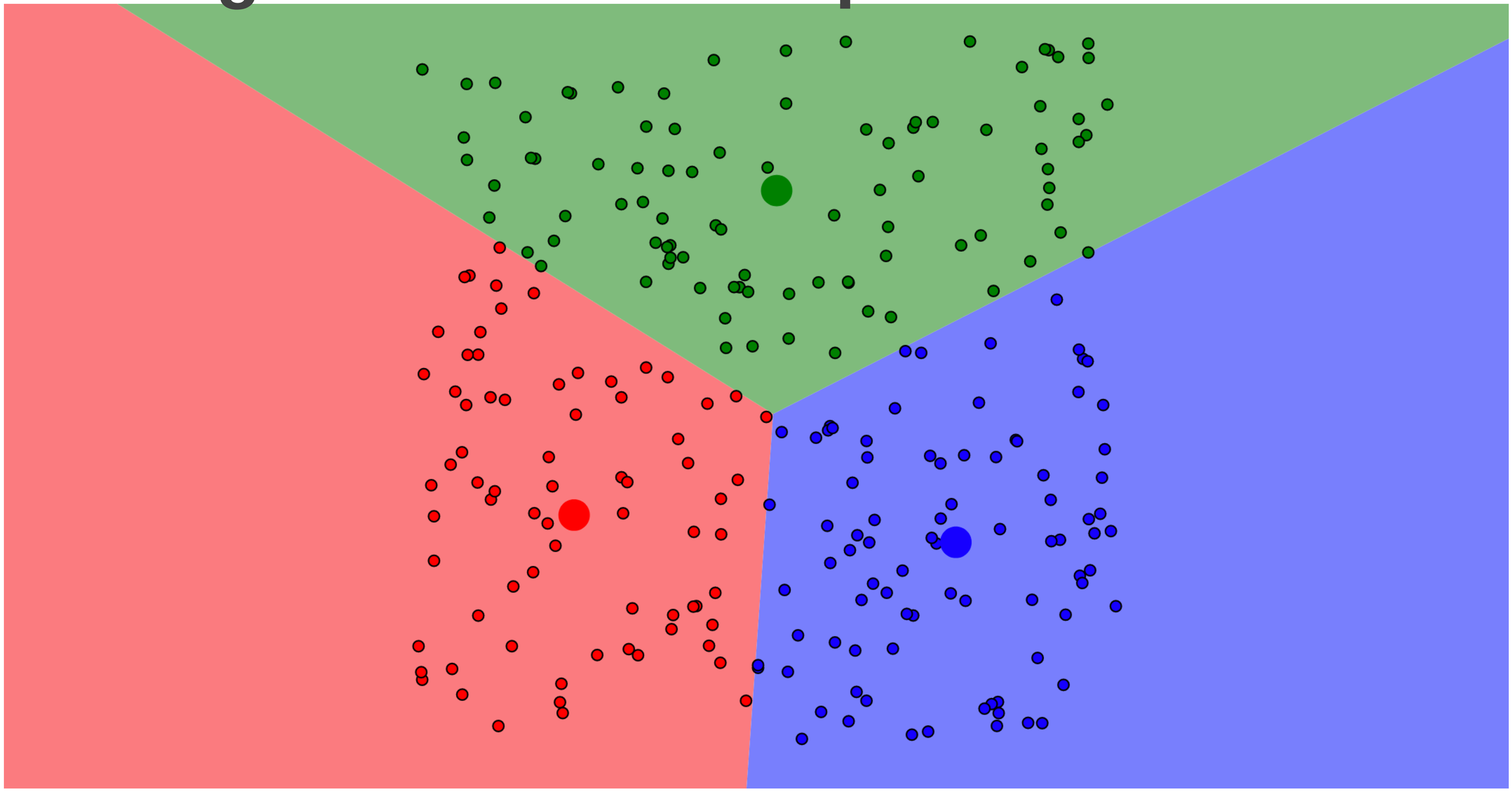
Re-Assign Data Points



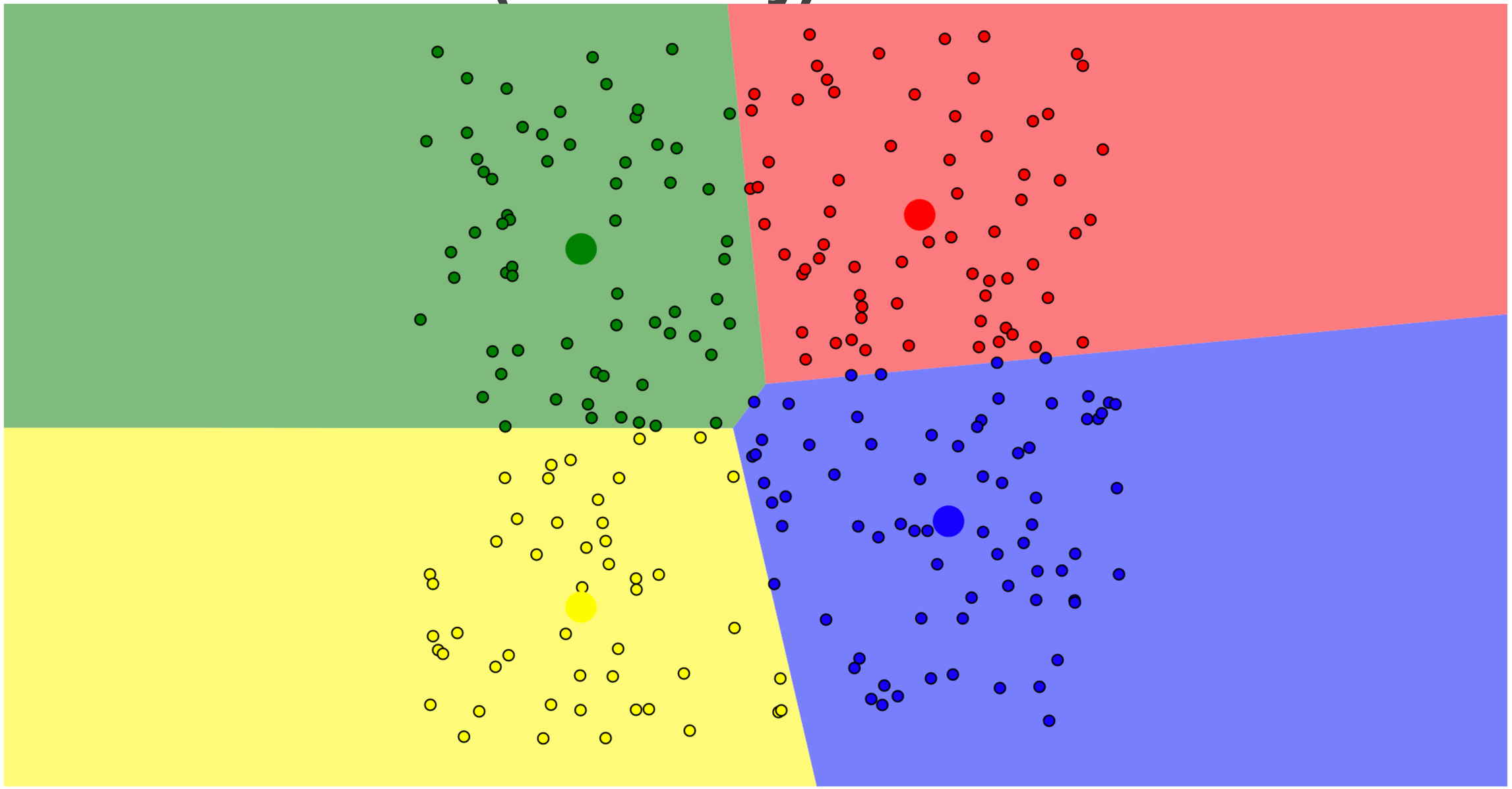
Update Centroids



Re-Assign Data Points - Stop

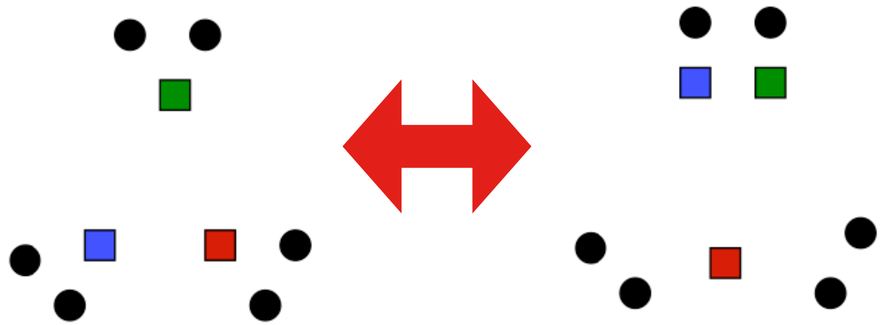


Add 4 Centroids (randomly)

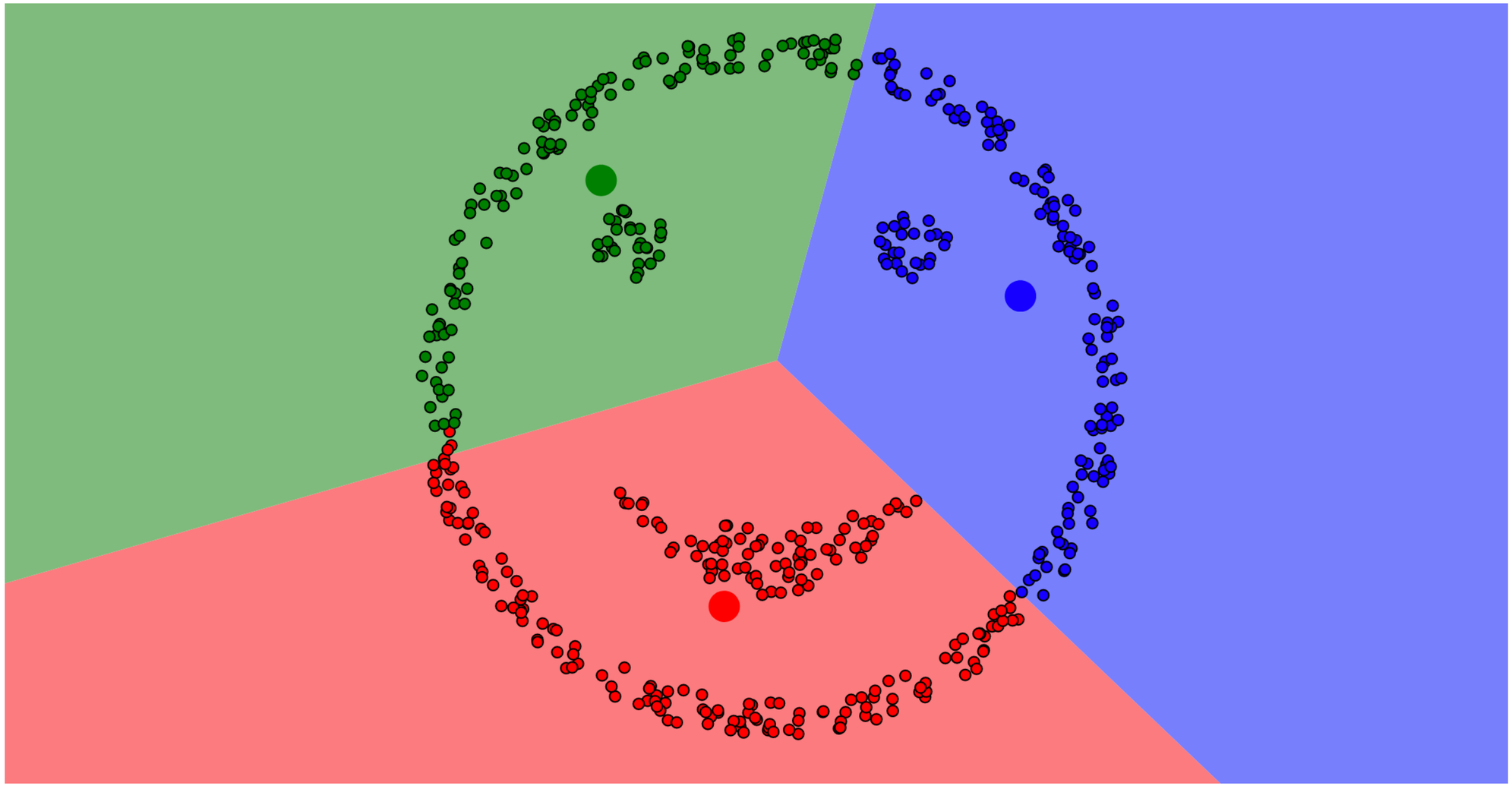


K-Means Pros and Cons

- **Pros**
  - Simple, fast to compute
  - Guaranteed to converge in a finite number of iterations
- **Cons/issues**
  - Setting *k*?
    - One way: silhouette coefficient
  - K-means algorithm is a heuristic
    - It does matter what random points you pick!
  - Sensitive to outliers
  - Detects spherical clusters



K-means not able to properly cluster



Machine Learning For Design

Lecture 9 - Designing And Develop Machine Learning Models / Part 3

Alessandro Bozzon
XX/03/2022

mlfd-io@tudelft.nl
www.ml4design.com

Credits

- Grokking Machine Learning. Luis G. Serrano. Manning, 2021
- https://scikit-learn.org/stable/modules/tree.html
- CIS 419/519 Applied Machine Learning. Eric Eaton, Dinesh Jayaraman. https://www.seas.upenn.edu/~cis519/spring2020/