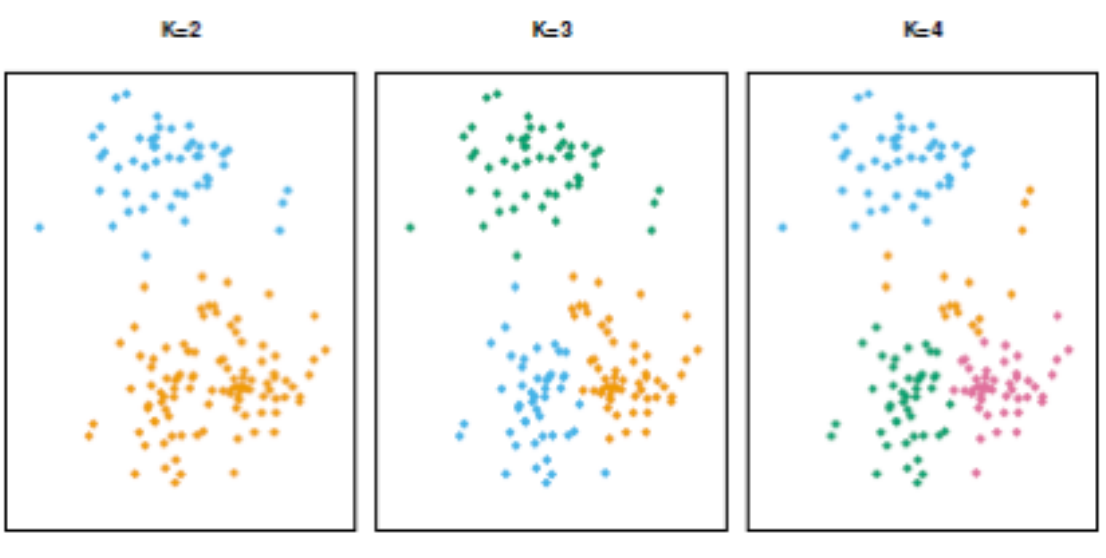

H2O.ai

K-Means Clustering

- K-Means clustering groups observations based on numeric features
  - Assumes clusters are roughly the same sized hyperspheres
  - Minimize Euclidean distance between observations and cluster centers
- Number of methods for choosing the number of clusters, k
  - Choose several and evaluate performance
  - Use business rules


K=2
K=3
K=4

H2O K-Means Clustering

```
h2o.kmeans(training_frame, x, model_id = NULL, validation_frame = NULL,
           nfolds = 0, keep_cross_validation_predictions = FALSE,
           keep_cross_validation_fold_assignment = FALSE,
           fold_assignment = c("AUTO", "Random", "Modulo", "Stratified"),
           fold_column = NULL, ignore_const_cols = TRUE,
           score_each_iteration = FALSE, k = 1, estimate_k = FALSE,
           user_points = NULL, max_iterations = 10, standardize = TRUE,
           seed = -1, init = c("Random", "PlusPlus", "Furthest", "User"),
           max_runtime_secs = 0, categorical_encoding = c("AUTO", "Enum",
           "OneHotInternal", "OneHotExplicit", "Binary", "Eigen",
           "LabelEncoder", "SortByResponse", "EnumLimited"))
```

```
from h2o.estimators.kmeans import H2OKMeansEstimator
clusters = H2OKMeansEstimator(…)
clusters.train(x = x, training_frame = data)
```

K-Means Clustering

- K-Means clustering groups observations based on numeric features
  - Assumes clusters are roughly the same sized hyperspheres
  - Minimize Euclidean distance between observations and cluster centers
- Number of methods for choosing the number of clusters, k
  - Choose several and evaluate performance
  - Use business rules