

Machine Learning with Python Scikit-Learn: Part II

Research Computing Services
Information Services & Technology



After the tutorial



Navigate to <http://rcs.bu.edu/eval>



Tutorial feedback survey

Plan of the Presentation



Overview of Machine Learning projects



Classification

Decision Trees: Classify Iris Flowers

Random Forest: Classify Images



Dimensionality Reduction

Curse of Dimensionality

Principal Component Analysis



Clustering

K-means: Cluster Iris Flowers

K-Means: Cluster Images



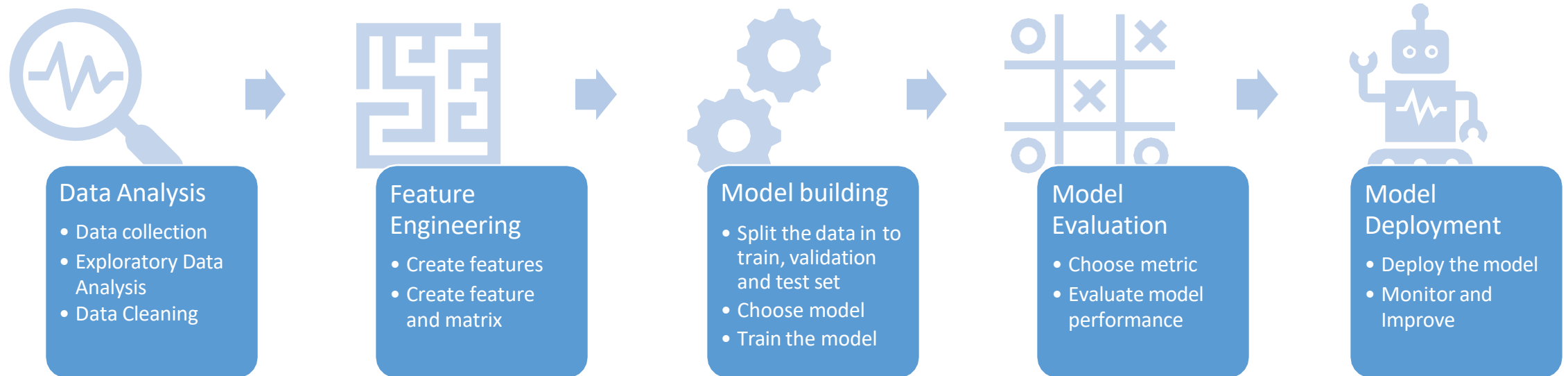
Where to go from here?



Conclusion

Overview of a Typical ML Project

The easiest way to think about data within Scikit-Learn is in terms of tables.



Classification

Classification: Decision Trees

Problem Statement

Build a Decision Tree Classifier to predict the species of an iris flower based on its features (sepal length, sepal width, petal length, and petal width).

Objectives

- Build the decision tree classifier as the ML model from scikit-learn
- Visualize the decision tree and highlight interpretability
- Evaluate model performance
- Make predictions

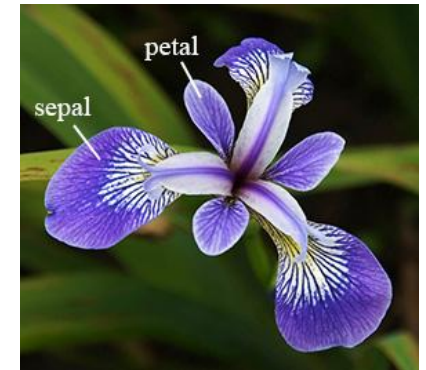
Setosa



Versicolor



Virginica



Classification: From Decision Trees to Random Forests



Decision tree is prone to overfitting



Random forest uses ensemble of estimators



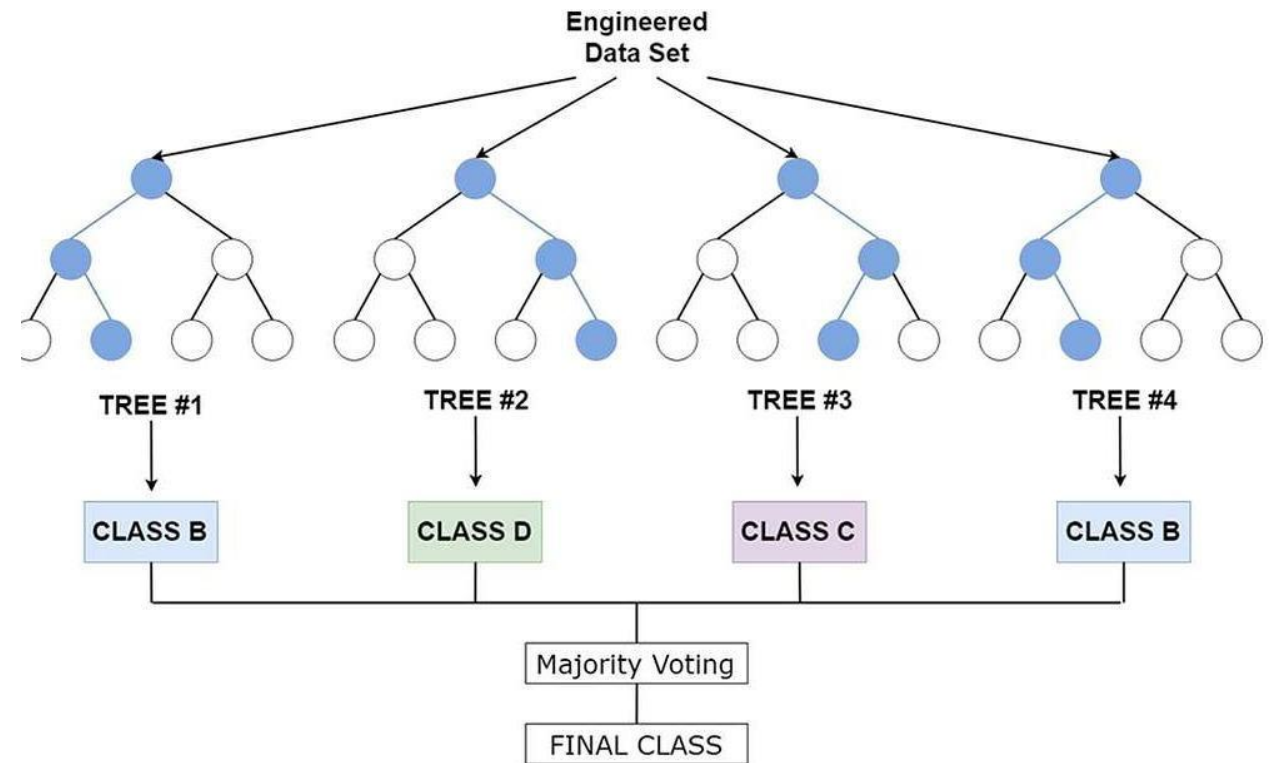
Each tree is trained on a random subset of the data and random subset of features.



This general technique is known as bagging



Randomness helps the model generalize and improves its performance on unseen data



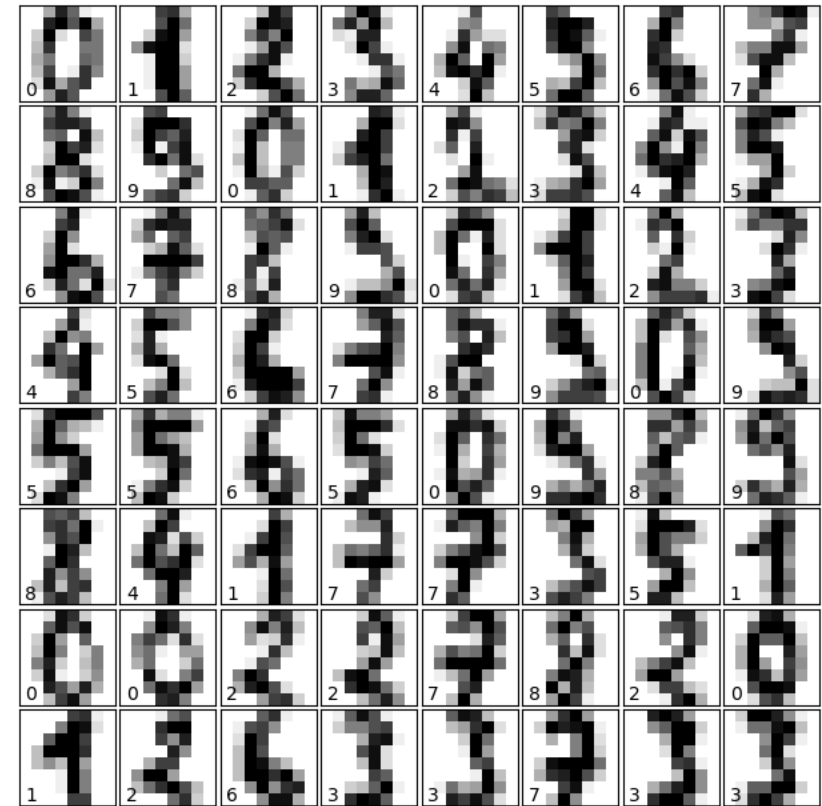
Classification: Random Forests

Problem Statement

Build a **Random Forest Classifier** to recognize handwritten digits, using the popular **Digits dataset**.

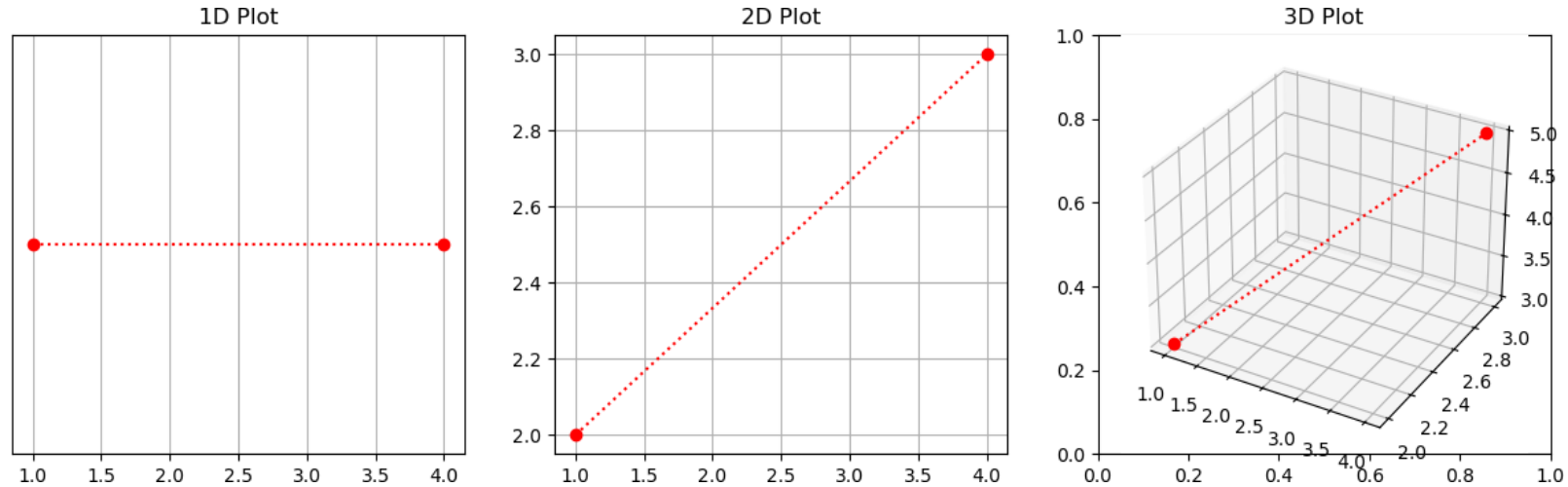
Objectives

- Build the Random Forest Classifier as the ML model from scikit-learn
- Evaluate model performance
- Make predictions
- Understand feature importance



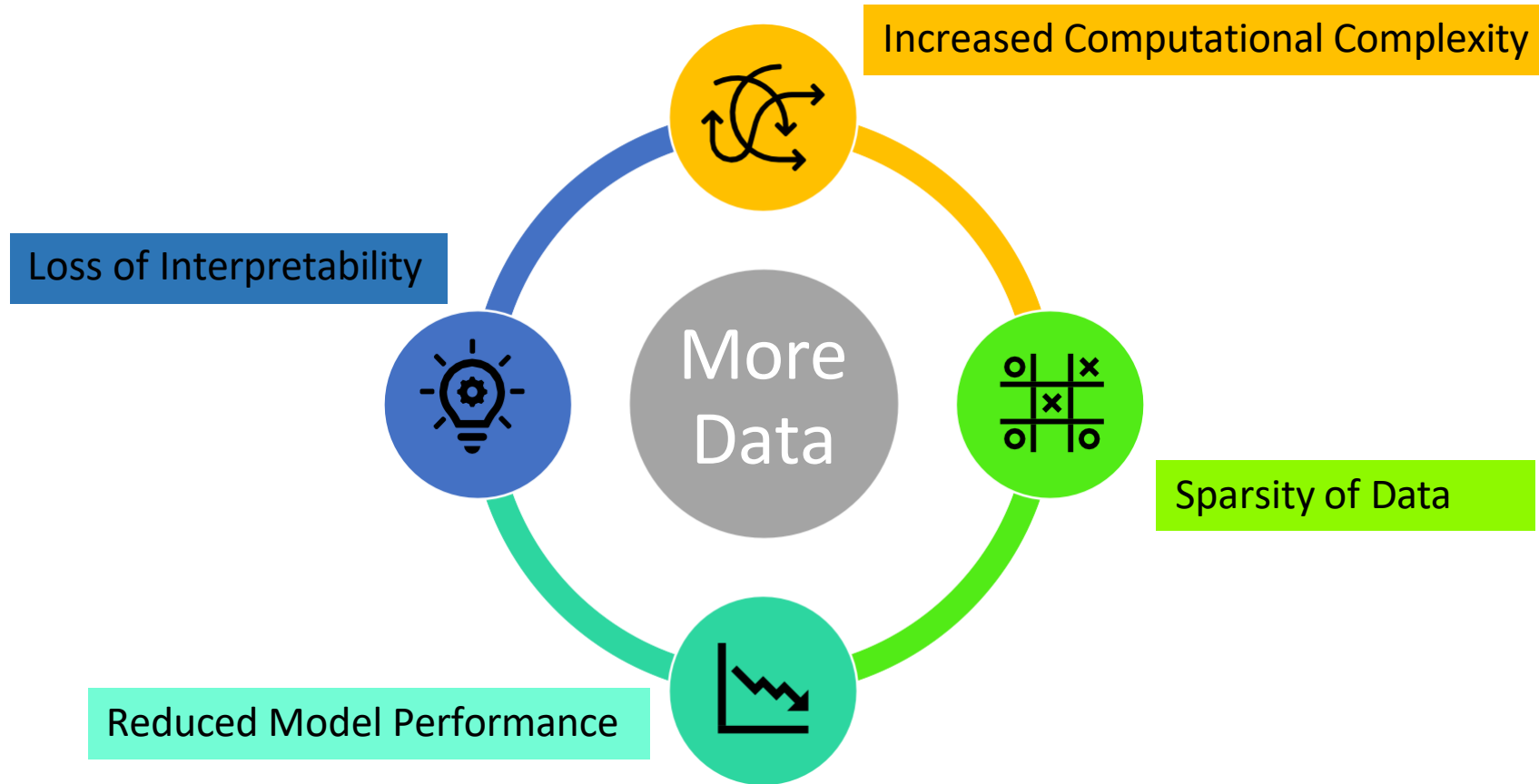
Dimensionality Reduction

The Curse of Dimensionality



- In modern data science you can collect more and more types of data related to the problem you're interested in.
- Your dataset can then become sparse (i.e. mostly zeroes) in many of these dimensions.
- Data organization and analysis strategies are inefficient when most of your data points are sparse and dissimilar.
- How can you tell what's important or interpret relationships between data points?

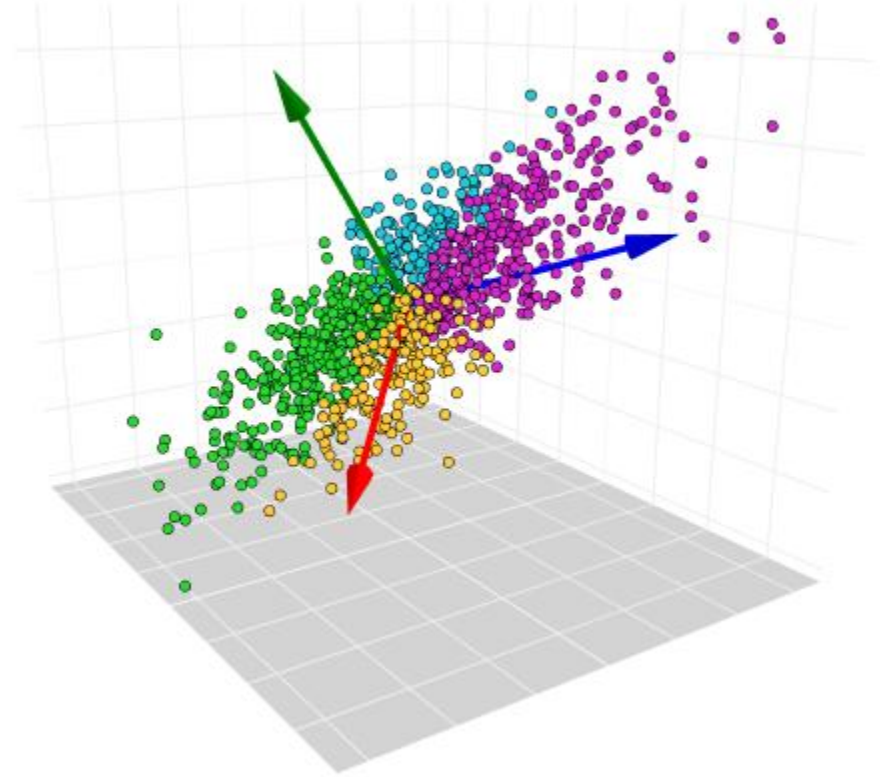
The Curse of Dimensionality



With hundreds or thousands of dimensions it's difficult to explain how a model is making a prediction.

Principal Component Analysis

- Derive a vector p_0 as a linear combination of the dataset variables that explains the most variance.
- Repeat after removing p_0 from the data to derive p_1 that matches most of the remaining variance.
- Keep repeating to get $p_0 \dots p_{k-1}$ (where k is the number of dimensions) are your principal components.
- If you'd like to see a summary of the mathematics check [Wikipedia](#).
- You can then select the most significant components for your model, thereby reducing the dimensions.



3 principal component axes for this set of points.
Taken from [this article](#) on PCA.

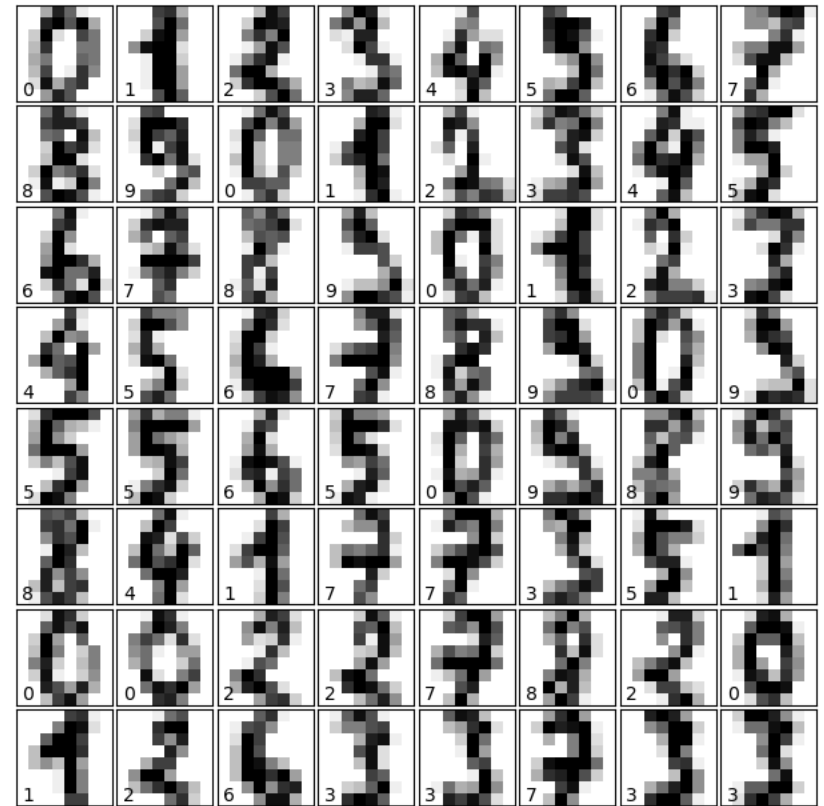
Dimensionality Reduction: Principal Component Analysis

Problem Statement

Use PCA to reduce the dimensionality of the data. Then build a **Random Forest Classifier** to recognize handwritten digits, using the popular **Digits dataset**.

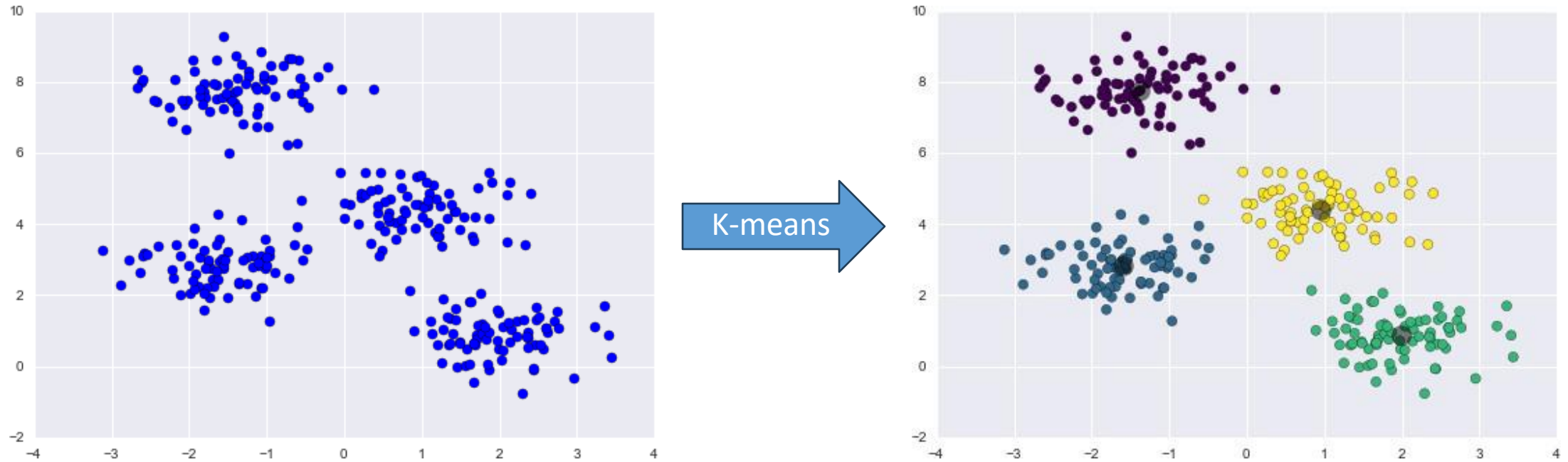
Objectives

- Apply PCA to the Digits dataset.
- Use PCA transformed data for image classification.
- Build the Random Forest Classifier as the ML model from scikit-learn
- Evaluate model performance
- Make predictions



Clustering

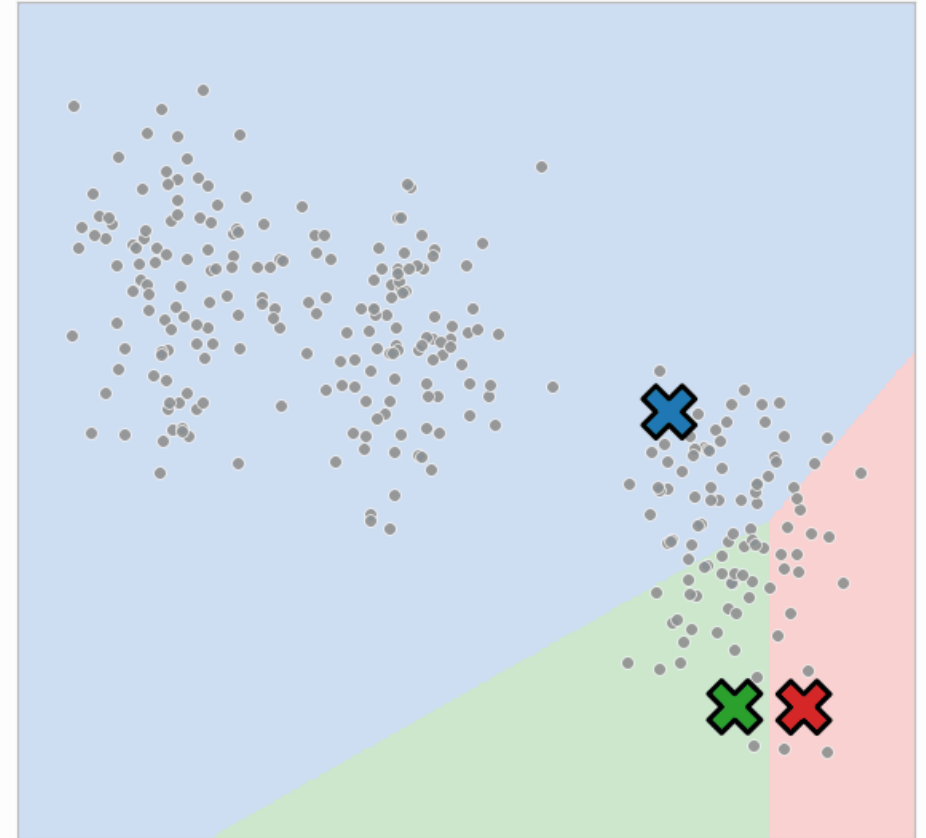
Clustering: K-Means



Break up your data into k clusters, compute the centroids of each cluster.
Then adjust: re-assign the points to the nearest cluster centroid, re-compute the centroids.
Repeat until the assignments stabilize.

K-means, animated

k-means clustering, $k = 3$
start: $k = 3$ centroids placed (deliberately badly)



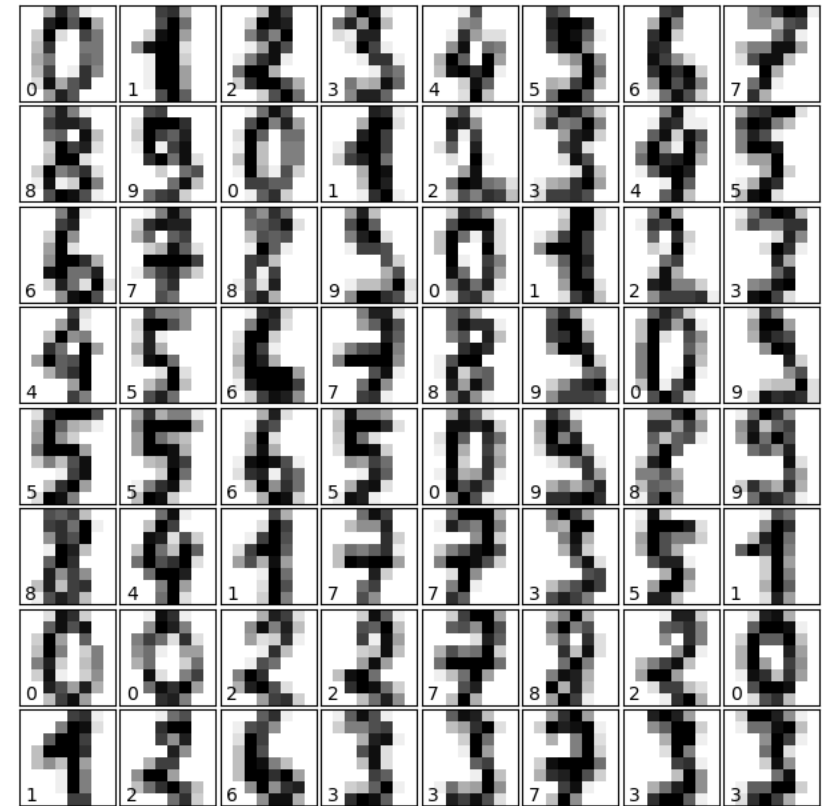
Clustering digits: K-means

Problem Statement

Use K-means algorithm to **group the data into K clusters** based on their positions in space.

Objectives

- Implement the K-means model from scikit-learn
- Visualize the clusters
- Evaluate the quality of clustering



Faces Detection using K-Means Clustering

Problem Statement

Use K-means algorithm to correctly **identify faces** in pictures.

Objectives

- Implement the K-means model from scikit-learn
- Apply PCA to the pictures
- Identify the faces in the pictures
- Evaluate the quality of clustering



Where to find more resources?



Contact RCS; help@scc.bu.edu



Python Data Science Handbook by Jake van der Plas:
<https://jakevdp.github.io/PythonDataScienceHandbook/index.html>



The Scikit-Learn website: <https://scikit-learn.org/stable/>



General Machine Learning Course by Andrew Ng: <https://www.coursera.org/learn/machine-learning>



KDnuggets, Fast.ai, and Google AI Blog