A Deeper Dive into Unsupervised Learning

Data Boot Camp

Lesson 20.2

Class Objectives

By the end of this lesson, you will be able to:

- Create a machine learning pipeline that incorporates hyperparameters tuning.
- Give an overview of how t-SNE works, as well as use it to explore data visually.
- Use DBSCAN algorithm to cluster data, and explain its strengths and drawbacks.
- Use hierarchical clustering to cluster data and to generate dendrograms.

Activity: Warmup

In this activity, you will create a grid search cross validation instance and use it to evaluate the SVC parameters (hyperparameters tuning) on the wine dataset built into Scikit-learn.

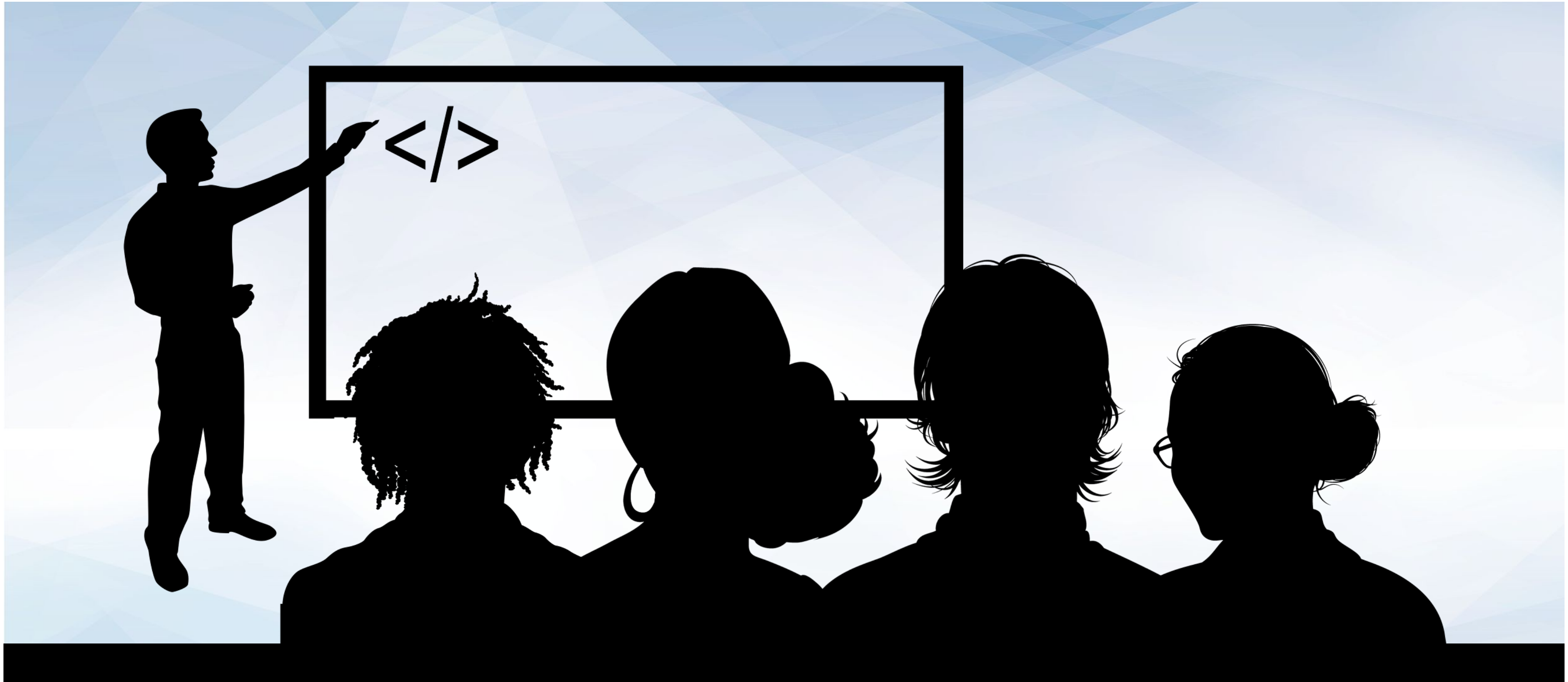
Suggested Time:
15 Minutes

Instructions:

Activity: Warmup

- The dataset you'll be working is the wine dataset built into Scikit-learn. For each sample, there is data on traits such as color intensity, malic acid content, and magnesium content. Also available is the target column, which lists the type of wine. There are 3 types of wine in the dataset.
- You will perform the following tasks.
  - Split the data into training and testing sets.
  - Scale the data. Check your slack for link for tips on avoiding data leakage.
  - Create a support vector machine model (support vector classifier) to classify a wine type based on its features.
  - Create a grid search cross validation instance and use it to evaluate the SVC parameters (hyperparameter tuning). Feel free to use your own parameters, or use the ones provided in the notebook.
  - Predict classifications of the testing dataset.
  - Assess the accuracy score of the predicted classifications.

Let's Review



Instructor Demonstration

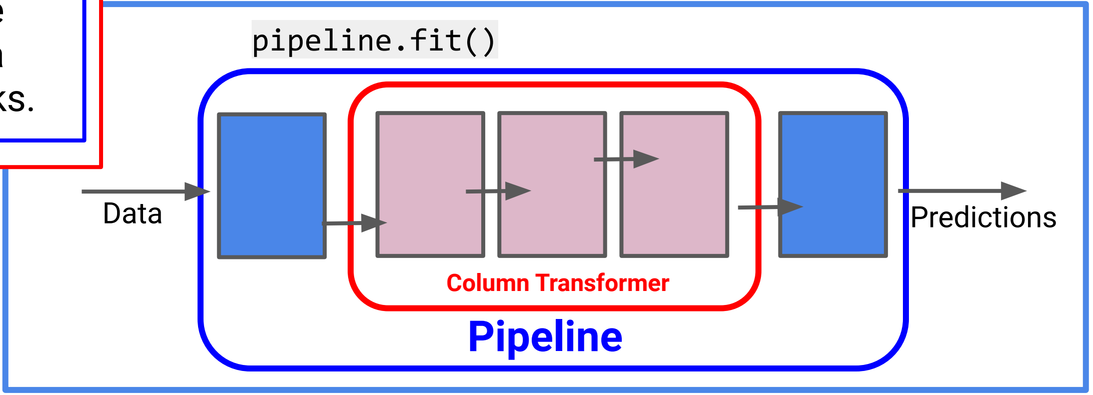
Machine Learning Pipeline

sklearn.pipeline.Pipeline

Instructor Do: Machine Learning Pipeline

➔ Pipelines

Allow us to streamline much of the routine processes, encapsulating a sequence of machine learning tasks.



```
pipeline.fit()
```

Activity: Credit Modeling

In this activity, you will create a classification model with logistic regression using a dataset collected in the1970's in Germany. Tune the model hyperparameters with grid search to evaluate the performance of the model, and the best parameters.

Suggested Time:

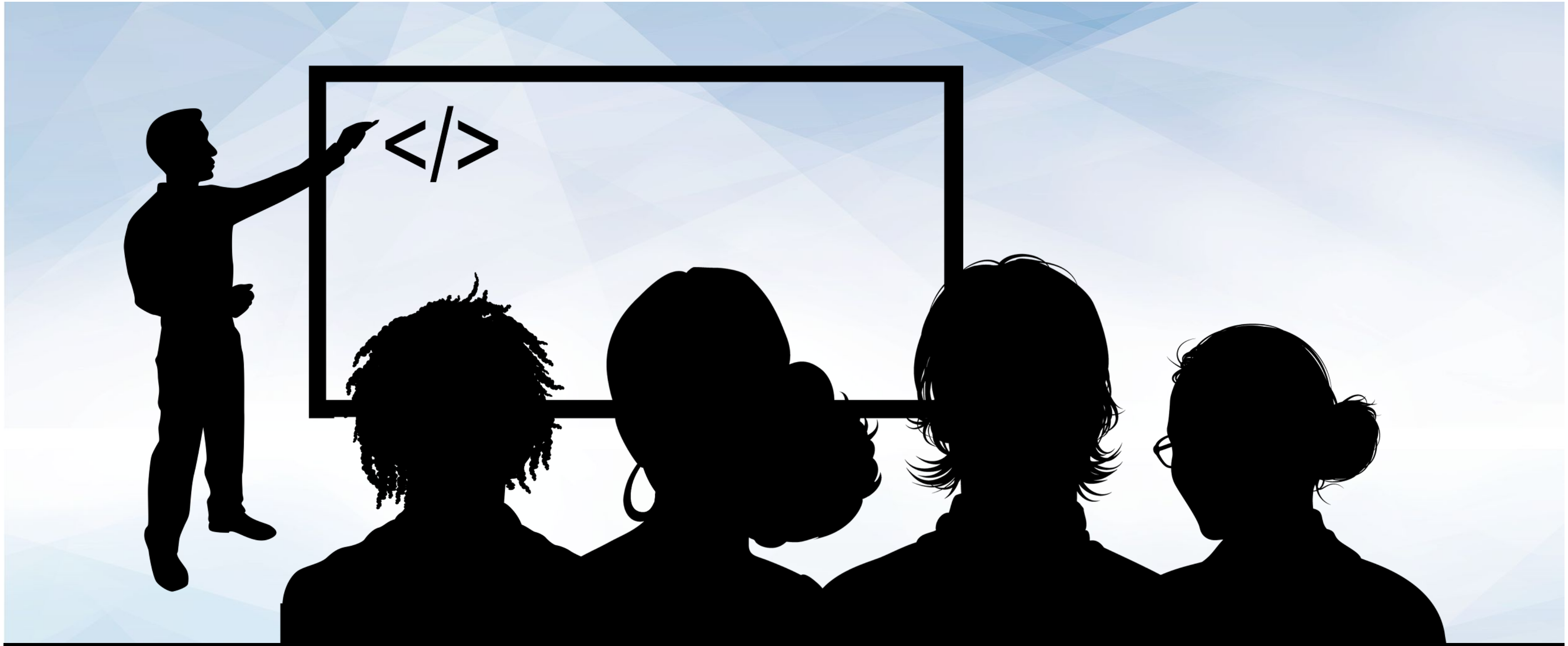
15 Minutes

Instructions:

Activity: Credit Modeling

- The dataset was collected in the 1970s in Germany. Each row contains information on a loan, such as the amount of the loan, as well as whether the loan was repaid. See https://archive.ics.uci.edu/ml/datasets/South+German+Credit+%28UPDATE%29 for more information.
- Create a classification model with logistic regression, completing the following tasks:
  - Read in the dataset and check to see if there are rows with null values.
  - Remove all rows with null values.
  - Split the dataset into data (X) and labels (y), then split them further into training and testing datasets. The `kredit` column should be the labels.
  - Create a pipeline with the following estimators: standardization of data, dimensionality reduction, logistic regression.
  - Tune the model hyperparameters with a grid search. Evaluate the performance of the model, and the best parameters.

Let’s Review



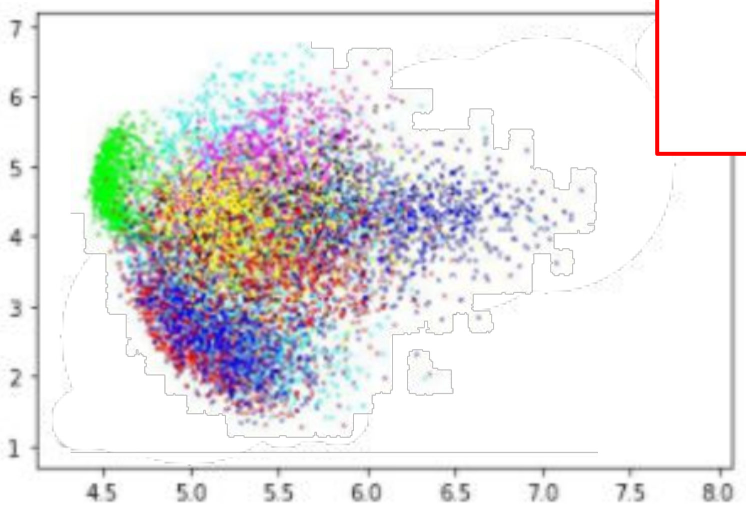
Instructor Demonstration

t-SNE

Instructor Do: t-SNE

- t-distributed Stochastic Neighbor Embedding
  - **t-distributed:** A probability curve is generated.
  - **Stochastic:** there is randomness involved (results will be different each time).
  - **Neighbor embedding:** similar data points become neighbors.

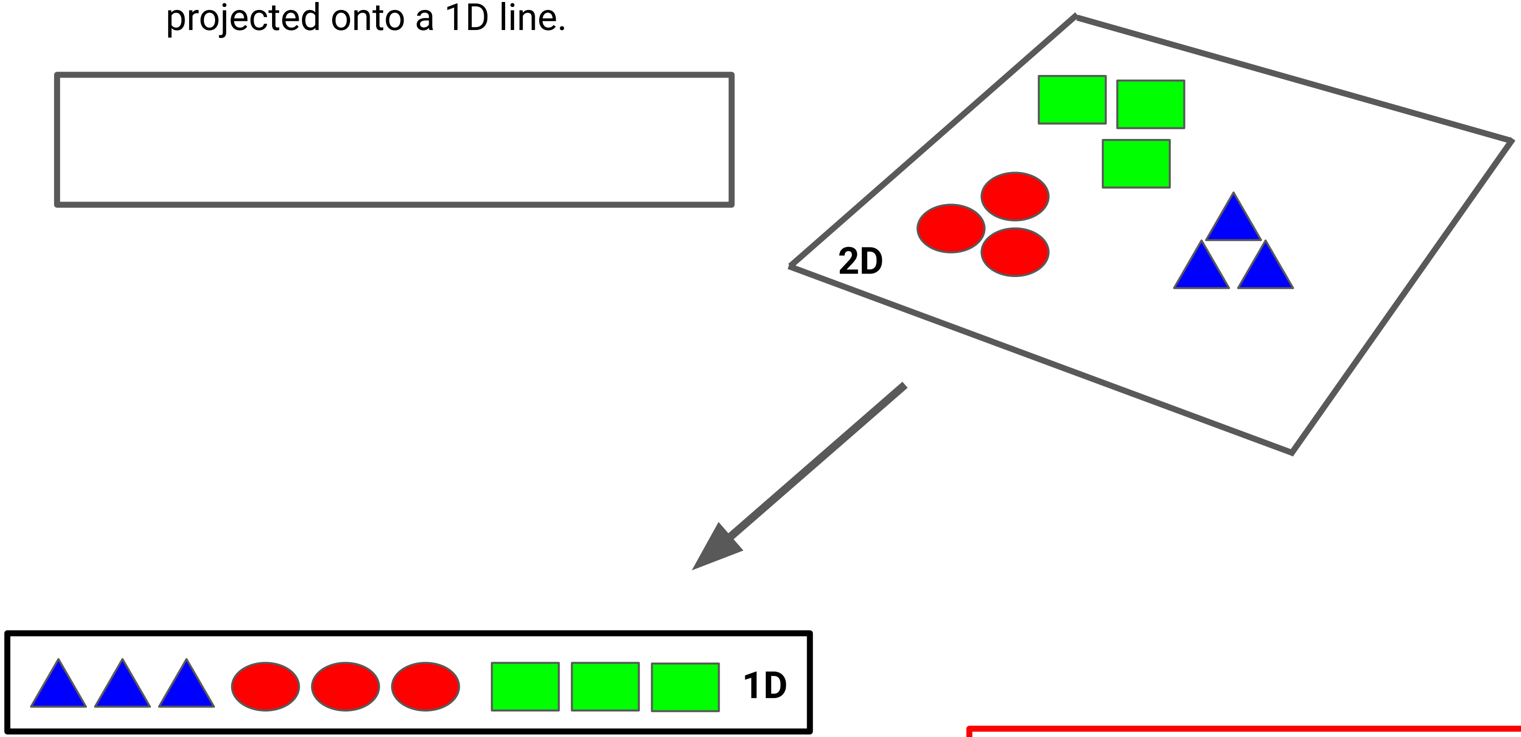
- Is an Unsupervised Learning Algorithm
  - **Like** PCA, t-SNE reduces a dataset's dimensions (to 2 or 3).
  - **Unlike** PCA, t-SNE sorts unlabeled data into clusters.
  - **Unlike** PCA, t-SNE is mainly used to **visualize** data.



t-SNE reduces dimensions

Instructor Do: t-SNE

- Here, shapes from 2D space are projected onto a 1D line.

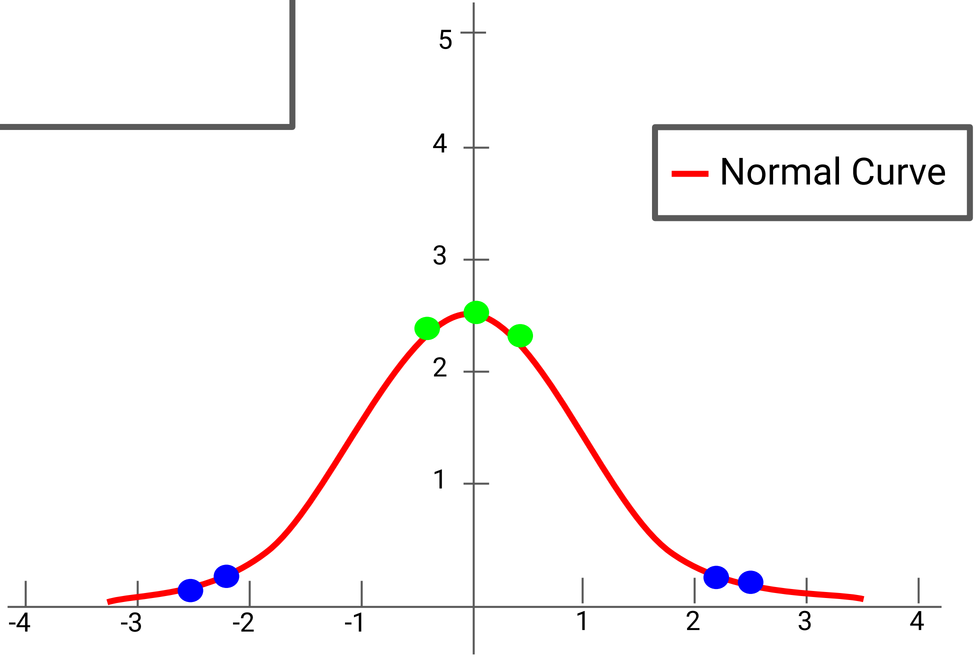


NOTE: This is NOT what t-SNE does.

t-SNE: finding neighbors

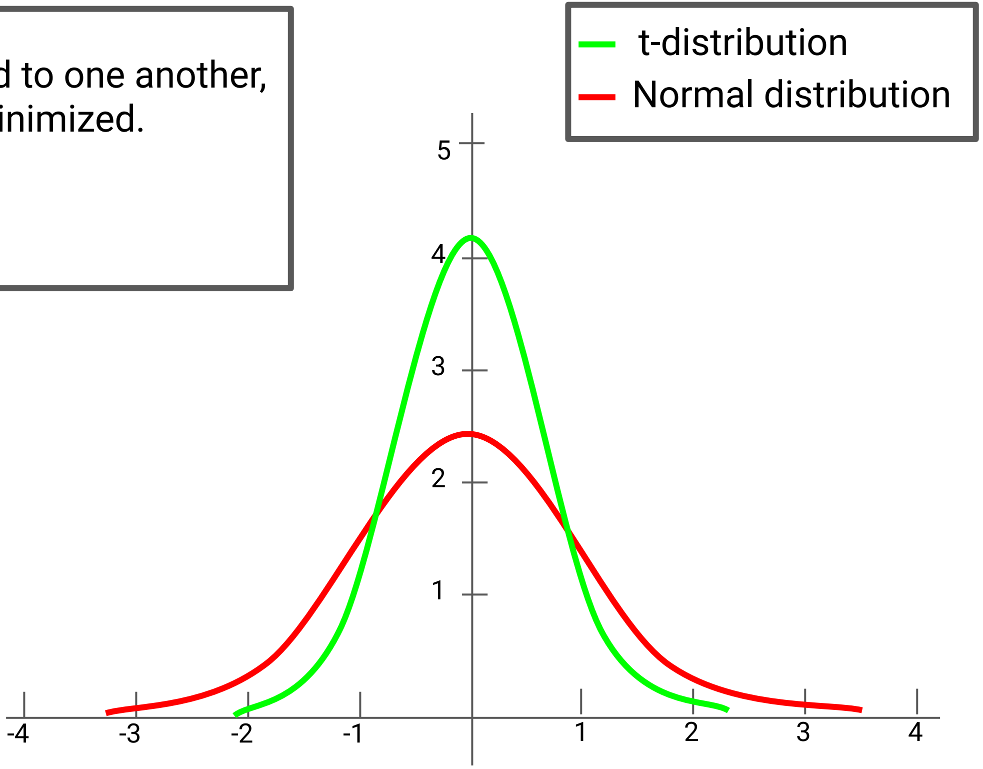
Instructor Do: t-SNE

- For a given point, a probability distribution curve is generated, with the point at the center.
- Similar data points are close to it on the curve (neighbors).
- Dissimilar points are farther apart.



Instructor Do: t-SNE

- Two probability curves are generated:
  - High-dimensional space.
  - Lower-dimensional space.
- The two curves are overlaid and compared to one another, and the difference between the two are minimized.

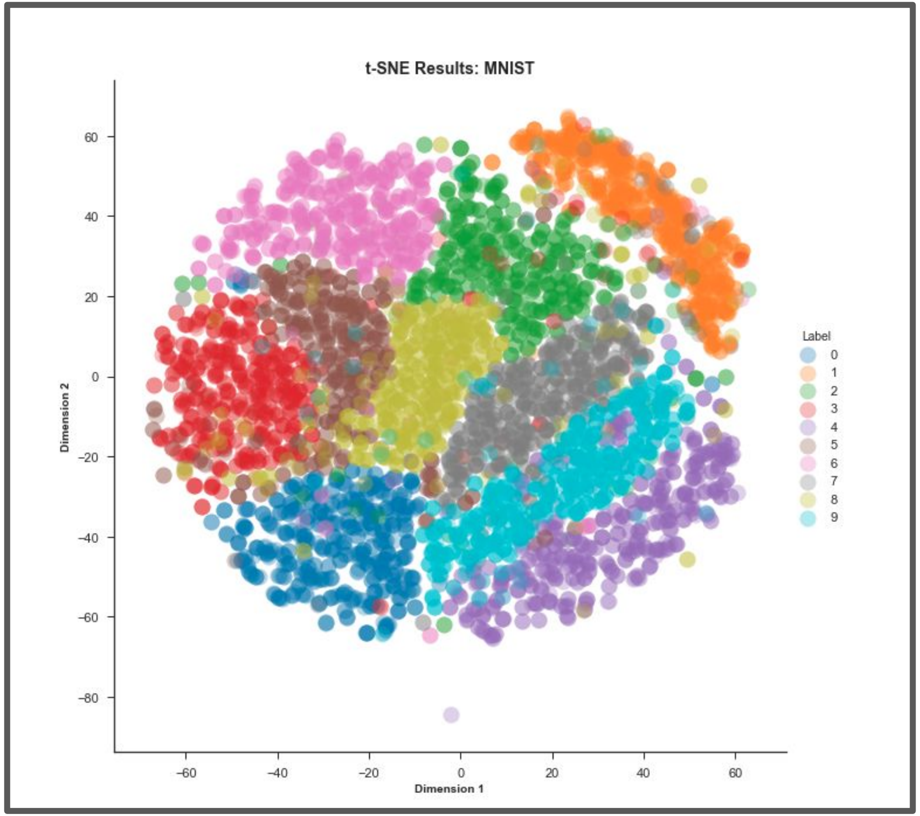


t-SNE:

Instructor Do: t-SNE

MNIST dataset: handwritten digits

(0 through 9)



Activity: Grape Clusters

In this activity, you will visually analyzing the wine dataset using t-SNE.

Suggested Time:
15 Minutes

Instructions:

Activity: Grape Clusters

- You will be working with the wine dataset, in which each sample belongs to one of three varieties of wines. Visually analyze the data using t-SNE. Follow these guidelines:
  - Standardize the data beforehand.
  - You may have to tweak the `learning_rate` parameter of your t-SNE model. The normal range is 10 to 1000. If you wish to adjust other parameters, consult the documentation.
  - After reducing dimensions with t-SNE, create a scatter plot of the transformed data. How many clusters do you detect?
- In this exercise, the target labels are available, as `labels = wine.target`. Use these labels to color your scatter plot. Does the result confirm or reject your findings?

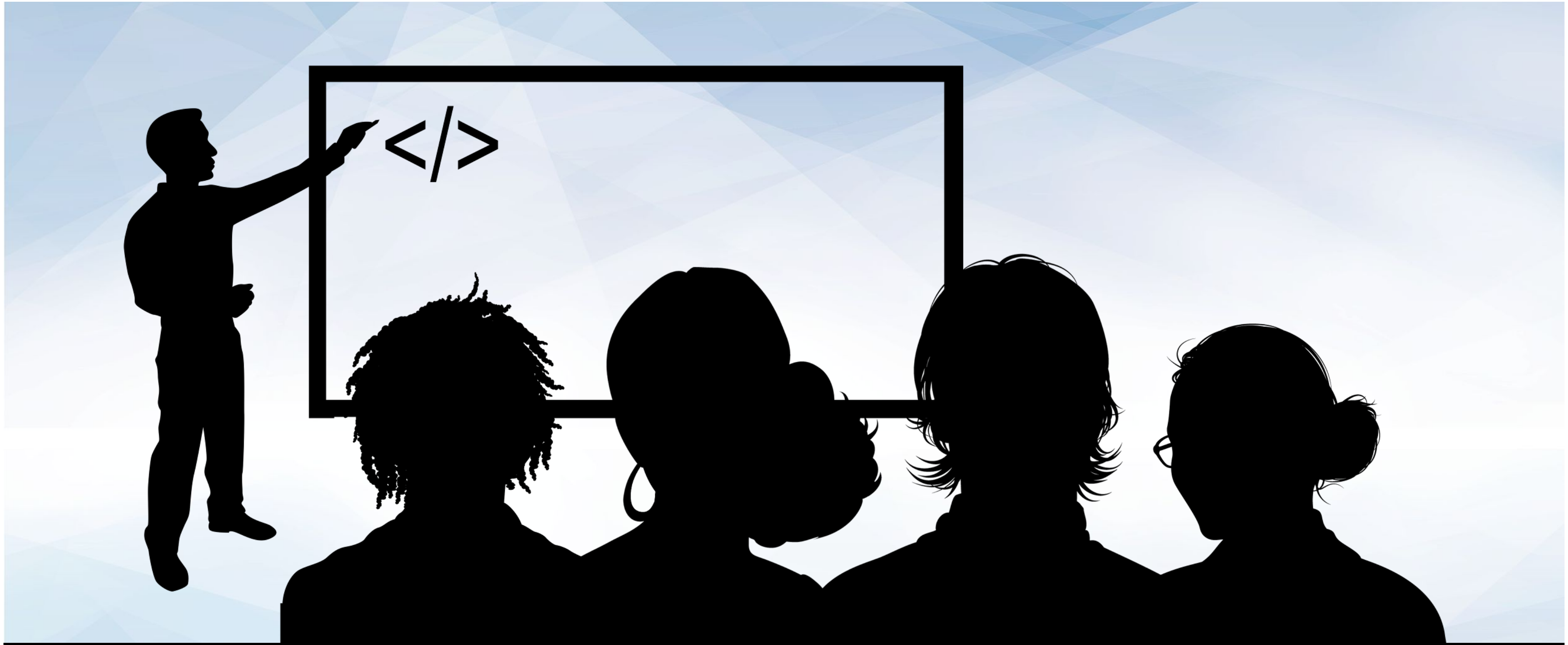
- **Bonus:**
  - Sometimes dimensions are reduced with PCA before they are further reduced with t-SNE.
  - With the supplied `breast_cancer.csv`, try performing PCA, then run t-SNE with the results. When running PCA, try setting `n_components` to 0.95, which selects the number of components that preserve 95% of the explained variance.

Let's Review

Countdown timer

15:00

(with alarm)



Instructor Demonstration

DBSCAN

Instructor Do: DBSCAN

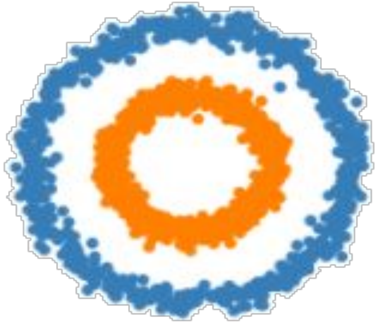
DBSCAN - Density-Based Spatial Clustering of Application with Noise

How it works?

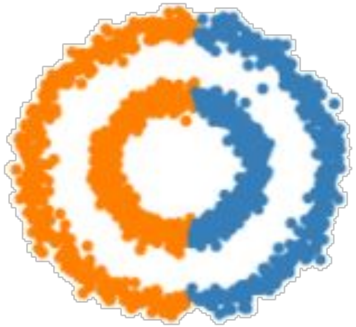
- A random data point is chosen
- If it meets a threshold for the number of neighbors, it is part of a cluster. (The same is performed for each neighbor).
- Another random data point is chosen, and the process is repeated.

DBSCAN vs k-means

Instructor Do: DBSCAN



DBSCAN



k-means

- k-means is best suited for ball-shaped clusters, while DBSCAN deals better with asymmetrical shapes.
- In k-means, the   of clusters must be determined by the user. DBSCAN figures this out on its own.
- k-means is sensitive to outliers, while DBSCAN has a more robust tolerance for outliers.
- k-means is relatively fast, while DBSCAN can slow down dramatically with larger datasets.

Activity: DBSCAN with Iris

In this activity, you will attempt to identify clusters in the iris dataset using DBSCAN.

Suggested Time:

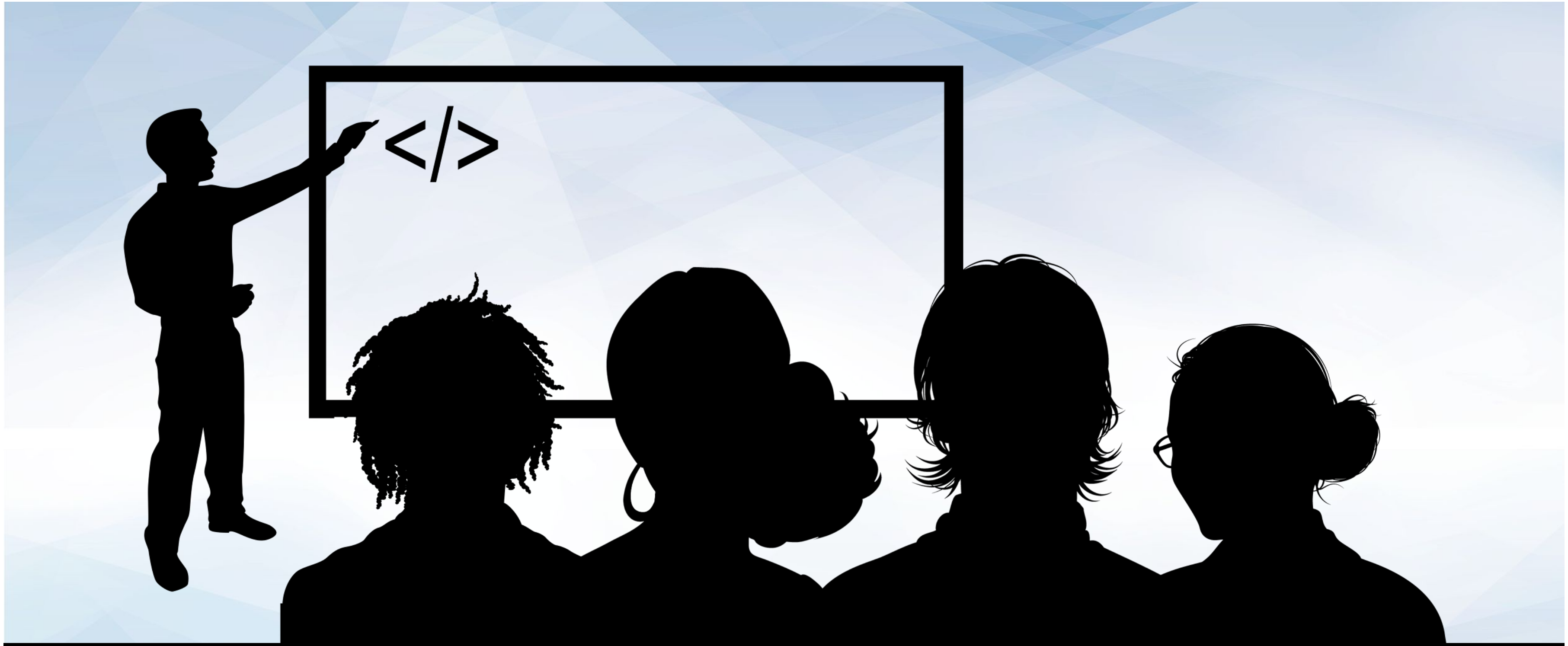
10 Minutes

Instructions:

Activity: DBSCAN with Iris

- In this activity, you will attempt to identify clusters in the iris dataset using DBSCAN.
- Scale the dataset, then perform DBSCAN for cluster analysis. You will have to adjust your `eps` and `min_samples` parameters.
- Use DBSCAN to generate target labels the dataset.
- Try creating a scatter plot of the dataset, coloring by cluster. The dataset has four features, but you will only be able to use two of them.
- Be ready to discuss the following:
  - How many clusters did DBSCAN identify?
  - Did you end up with the expected number of clusters? Explain your results.

Let’s Review



Instructor Demonstration

Hierarchical Clustering

Hierarchical Clustering Linkage Methods

Instructor Do: Hierarchical Clustering

How is distance calculated between two clusters?

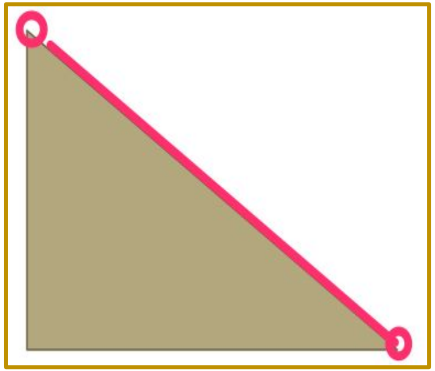
- Single: the difference between two clusters is defined by the closest distance between two clusters.
- Complete: the difference between two clusters is defined by the farthest distance between two clusters.
- Ward: this method is based on the squared euclidean distance between clusters. It's the method used in our example, and often used as a default.

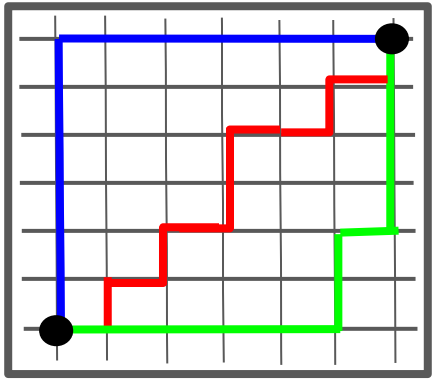
Hierarchical Clustering Affinity

Instructor Do: Hierarchical Clustering

How is distance calculated between two points?

- **Euclidean**: shortest distance between two points.





- **Manhattan**: sum of the absolute values of the difference between two points (looks like a city grid)

Activity: Customer Data

In this activity, you will perform clustering using hierarchical clustering to group and plot customer data.

Suggested Time:
15 Minutes

Instructions:

Activity: Customer Data

- The data comes from UCI's (University of California, Irvine) repository for machine learning datasets. In the dataset, each row represents a customer's region, as well as whether the customer is a retail customer or affiliated with catering (horeca). See here for a fuller description of the dataset.
- Use hierarchical clustering to perform clustering. Perform the following tasks:
  - After reading in the dataset, normalize it.
  - Perform hierarchical clustering on the normalized data, using the Ward method.
  - Create a dendrogram of the results.
  - Generate cluster labels with AgglomerativeClustering, using `euclidean` and `ward` for affinity and linkage, respectively.
  - Create a 2D scatter plot of the normalized dataframe. Use the cluster labels to color the data points. Since you're only using two features in the dataset, the plot will be a rough approximation. You will also have to select which two features to plot.

Let's Review

Questions?