DSA 610 Redesign, Lecture 10 Outline

Lecture Outline: Parallel Processing, Hadoop, Spark, and Online Analytical Processing (OLAP)
Duration: 50 minutes

---

1. Introduction to Parallel Processing (5 minutes)

- **Objective:** Understand the concept and importance of parallel processing in data analysis.
- **Content:**
  - **Definition:** Parallel processing involves performing multiple computations simultaneously to speed up data processing tasks.
  - **Benefits:**
    - **Increased Speed:** Reduced processing time for large datasets.
    - **Efficiency:** Better utilization of computing resources.

---

2. Hadoop (12 minutes)

- **Objective:** Explore Hadoop and its role in distributed data processing.
- **Content:**
  - **1. Overview:**
    - **Definition:** An open-source framework for processing large datasets across clusters of computers using simple programming models.
    - **Components:**
      - **Hadoop Distributed File System (HDFS):** Stores data across multiple machines.
      - **MapReduce:** A programming model for processing large datasets in parallel.
  - **2. How It Works:**
    - **Data Storage:** Data is divided into blocks and distributed across a cluster.
    - **Data Processing:** MapReduce jobs process data in parallel, with a "map" phase to distribute work and a "reduce" phase to aggregate results.
  - **3. Example Use Case:**
    - **Log Analysis:** Processing and analyzing large web server logs.
  - **4. Advantages and Disadvantages:**
    - **Advantages:**
      - **Scalability:** Handles large volumes of data.
      - **Fault Tolerance:** Redundant data storage ensures reliability.
    - **Disadvantages:**
      - **Complexity:** Requires significant setup and configuration.
      - **Performance:** Can be slower for some types of queries compared to other systems.

---

3. Apache Spark (12 minutes)

- **Objective:** Understand Spark and its advantages over Hadoop for data processing.
- **Content:**
  - **1. Overview:**
    - **Definition:** An open-source, distributed computing system designed for fast data processing and analytics.
    - **Components:**
      - **Spark Core:** The underlying engine for large-scale data processing.

            - **Spark SQL:** For querying data using SQL.
            - **Spark Streaming:** For processing real-time data streams.
            - **MLlib:** Machine learning library.
            - **GraphX:** For graph processing.
    - **2. How It Works:**
        - **In-Memory Processing:** Spark stores intermediate data in memory, speeding up processing.
        - **Resilient Distributed Datasets (RDDs):** Fault-tolerant data structures for distributed computing.
    - **3. Example Use Case:**
        - **Real-Time Analytics:** Processing and analyzing streaming data from social media or IoT devices.
    - **4. Advantages and Disadvantages:**
        - **Advantages:**
            - **Speed:** Faster processing due to in-memory operations.
            - **Flexibility:** Supports various data processing tasks.
        - **Disadvantages:**
            - **Memory Usage:** High memory requirements for in-memory processing.
            - **Complexity:** Requires knowledge of Spark's APIs and architecture.

---

4. Online Analytical Processing (OLAP) (10 minutes)

- **Objective:** Learn about OLAP and its role in data analysis and reporting.
- **Content:**
    - **1. Overview:**
        - **Definition:** OLAP systems allow users to interactively analyze multidimensional data from multiple perspectives.
        - **Types:**
            - **MOLAP (Multidimensional OLAP):** Uses multidimensional data cubes.
            - **ROLAP (Relational OLAP):** Uses relational databases to provide OLAP capabilities.
    - **2. Key Features:**
        - **Multidimensional Analysis:** Enables slicing, dicing, and drilling down into data.
        - **Aggregations:** Pre-computed summaries for fast querying.
    - **3. Example Use Case:**
        - **Sales Analysis:** Analyzing sales data across different dimensions such as time, geography, and product categories.
    - **4. Advantages and Disadvantages:**
        - **Advantages:**
            - **Ease of Use:** User-friendly interfaces for complex queries.
            - **Performance:** Fast query response times for analytical queries.
        - **Disadvantages:**
            - **Cost:** High setup and maintenance costs.
            - **Scalability:** Can be limited by the size of pre-computed data cubes.

---

5. Comparative Overview (5 minutes)

- **Objective:** Compare and contrast Hadoop, Spark, and OLAP systems in terms of use cases and performance.
- **Content:**

- **Hadoop vs. Spark:**
  - **Hadoop:** Best for batch processing and large-scale data storage.
  - **Spark:** Ideal for fast, iterative processing and real-time analytics.
- **OLAP vs. Hadoop/Spark:**
  - **OLAP:** Focuses on interactive data analysis and reporting.
  - **Hadoop/Spark:** Focus on large-scale data processing and analytics.

---

6. Q&A and Discussion (6 minutes)

- **Objective:** Address questions and discuss practical applications of parallel processing, Hadoop, Spark, and OLAP.
- **Content:**
  - **Q&A Session:** Open the floor for student questions.
  - **Discussion:** Explore scenarios where each technology might be used and the trade-offs involved.

Key Takeaways

- **Parallel Processing:** Essential for speeding up data processing tasks.
- **Hadoop:** A framework for distributed data processing using HDFS and MapReduce.
- **Spark:** A fast, in-memory processing engine with a wide range of capabilities.
- **OLAP:** Provides multidimensional analysis and interactive reporting.

Resources:
Parallel Processing: https://www.spiceworks.com/tech/iot/articles/what-is-parallel-processing/
Hadoop & HDFS: https://www.geeksforgeeks.org/how-does-hadoop-handle-parallel-processing-of-large-datasets-across-a-distributed-cluster/
MapReduce: https://courses.cs.washington.edu/courses/cse490h/07wi/readings/IntroductionToParallelProgrammingAndMapReduce.pdf
Apache Spark: https://www.databricks.com/glossary/what-is-apache-spark
Pyspark: https://www.datacamp.com/tutorial/pyspark-tutorial-getting-started-with-pyspark
Online Analytic Processing (OLAP): https://aws.amazon.com/what-is/olap/
OLAP, ROLAP, MOLAP, HOLAP: https://www.sisense.com/glossary/olap/

1. Introduction to Machine Learning Models (5 minutes)

- **Objective:** Understand the broad categories of machine learning models and their purposes.
- **Content:**
  - **Definition:**
    - **Machine Learning Models:** Algorithms and techniques used to analyze and interpret data, make predictions, and automate decision-making.
  - **Categories:**
    - **Supervised Learning**
    - **Unsupervised Learning**
    - **Semi-Supervised Learning**
    - **Reinforcement Learning**
    - **Natural Language Processing (NLP)**
    - **Neural Networks**
    - **Graph-Based Approaches**
    - **Image Processing**
    - **Spatial Analysis**

2. Supervised Learning (10 minutes)

- **Objective:** Understand the purpose and methods of supervised learning.
- **Content:**
  - **Definition:**
    - **Supervised Learning:** Models trained on labeled data where the outcome is known.
  - **Purpose:**
    - **Prediction:** Forecasting future values (regression) or classifying data into categories (classification).
  - **Common Algorithms:**
    - **Linear Regression**
    - **Logistic Regression**
    - **Support Vector Machines (SVMs)**
    - **Decision Trees and Random Forests**
  - **Pre-Processing:**
    - **Label Encoding**
    - **Feature Scaling**
    - **Handling Missing Values**

---

3. Unsupervised Learning (10 minutes)

- **Objective:** Explore the goals and methods of unsupervised learning.
- **Content:**
  - **Definition:**
    - **Unsupervised Learning:** Models trained on unlabeled data to identify hidden patterns or groupings.
  - **Purpose:**
    - **Clustering:** Grouping similar data points (e.g., K-Means, Hierarchical Clustering).
    - **Dimensionality Reduction:** Reducing the number of features while retaining important information (e.g., PCA).
  - **Common Algorithms:**
    - **K-Means Clustering**
    - **Principal Component Analysis (PCA)**
    - **Hierarchical Clustering**
  - **Pre-Processing:**
    - **Feature Scaling**
    - **Normalization**
    - **Handling Missing Values**

---

4. Semi-Supervised Learning (5 minutes)

- **Objective:** Understand how semi-supervised learning combines labeled and unlabeled data.
- **Content:**
  - **Definition:**
    - **Semi-Supervised Learning:** Uses a small amount of labeled data and a large amount of unlabeled data.
  - **Purpose:**
    - **Improving Model Accuracy:** Enhancing performance when labeled data is scarce.

    - **Common Techniques:**
        - **Self-Training**
        - **Co-Training**
    - **Pre-Processing:**
        - **Similar to supervised learning, with an emphasis on handling large amounts of unlabeled data.**

---

5. Reinforcement Learning (5 minutes)

- **Objective:** Explore the concepts and methods of reinforcement learning.
- **Content:**
    - **Definition:**
        - **Reinforcement Learning:** Models learn to make decisions by receiving rewards or penalties.
    - **Purpose:**
        - **Decision Making:** Optimizing actions in sequential environments (e.g., game playing, robotics).
    - **Common Algorithms:**
        - **Q-Learning**
        - **Deep Q-Networks (DQN)**
        - **Policy Gradient Methods**
    - **Pre-Processing:**
        - **Reward Shaping**
        - **Feature Engineering for Environment States**

---

6. Natural Language Processing (NLP) (5 minutes)

- **Objective:** Understand the goals and methods of NLP.
- **Content:**
    - **Definition:**
        - **NLP:** Techniques for processing and analyzing human language data.
    - **Purpose:**
        - **Text Classification:** Categorizing text data (e.g., sentiment analysis, spam detection).
        - **Named Entity Recognition (NER):** Identifying entities in text.
    - **Common Techniques:**
        - **Bag of Words (BoW)**
        - **TF-IDF**
        - **Word Embeddings (Word2Vec, GloVe)**
    - **Pre-Processing:**
        - **Tokenization**
        - **Stop-word Removal**
        - **Text Normalization (stemming, lemmatization)**

---

7. Neural Networks (5 minutes)

- **Objective:** Provide a broad overview of neural networks and their applications.
- **Content:**
    - **Definition:**
        - **Neural Networks:** Models inspired by the human brain, composed of interconnected layers of nodes (neurons).

- **Purpose:**
    - **Pattern Recognition:** Learning complex patterns in data.
- **Types:**
    - **Feedforward Neural Networks**
    - **Convolutional Neural Networks (CNNs)**
    - **Recurrent Neural Networks (RNNs)**
- **Pre-Processing:**
    - **Feature Scaling**
    - **Normalization**

---

8. Graph-Based Approaches (5 minutes)

- **Objective:** Understand the use of graph-based methods in machine learning.
- **Content:**
    - **Definition:**
        - **Graph-Based Approaches:** Models that use graph structures to represent data.
    - **Purpose:**
        - **Network Analysis:** Understanding relationships and interactions (e.g., social networks, web graphs).
    - **Common Techniques:**
        - **Graph Neural Networks (GNNs)**
        - **PageRank Algorithm**
    - **Pre-Processing:**
        - **Graph Construction**
        - **Feature Extraction from Graphs**

---

9. Image Processing (5 minutes)

- **Objective:** Explore image processing techniques and their applications.
- **Content:**
    - **Definition:**
        - **Image Processing:** Techniques for analyzing and interpreting visual data.
    - **Purpose:**
        - **Object Detection and Classification:** Identifying and labeling objects in images.
    - **Common Techniques:**
        - **Convolutional Neural Networks (CNNs)**
        - **Image Segmentation**
    - **Pre-Processing:**
        - **Image Normalization**
        - **Data Augmentation**

---

10. Spatial Analysis (5 minutes)

- **Objective:** Understand the applications and methods for spatial data analysis.
- **Content:**
    - **Definition:**
        - **Spatial Analysis:** Analyzing data with a spatial component (e.g., geographic data).
    - **Purpose:**
        - **Geospatial Analysis:** Understanding spatial patterns and relationships (e.g., heatmaps, spatial clustering).

- **Common Techniques:**
  - **Geographic Information Systems (GIS)**
  - **Spatial Autocorrelation**
- **Pre-Processing:**
  - **Geocoding**
  - **Spatial Data Cleaning**

---

11. Q&A and Discussion (5 minutes)

- **Objective:** Address questions and discuss practical applications of different model types.
- **Content:**
  - **Q&A Session:** Open the floor for student questions.
  - **Discussion:** Explore real-world applications and challenges associated with various models.

Key Takeaways

- **Model Categories:** Understanding the different types of machine learning models and their purposes.
- **Pre-Processing:** Overview of the necessary data preparation for various model types.
- **Applications:** Insight into how different models operate on different types of data and their practical uses.

Here's a guide to various machine learning models and the contexts in which they are most appropriate:

1. Supervised Learning

a. Linear Regression

- **Context:** Predicting a continuous value based on one or more features.
- **Examples:**
  - **Real Estate:** Predicting house prices based on features like square footage, number of bedrooms, etc.
  - **Finance:** Forecasting future stock prices based on historical data.

b. Logistic Regression

- **Context:** Binary classification problems where the output is a probability that can be mapped to two classes.
- **Examples:**
  - **Healthcare:** Predicting whether a patient has a disease (e.g., diabetes) based on medical test results.
  - **Marketing:** Classifying emails as spam or not spam.

c. Decision Trees

- **Context:** Classification or regression tasks with interpretable results.
- **Examples:**
  - **Customer Segmentation:** Classifying customers into different segments based on purchasing behavior.
  - **Credit Scoring:** Evaluating creditworthiness based on applicant features.

d. Random Forest

- **Context:** Classification or regression tasks with high accuracy and robustness to overfitting.
- **Examples:**
  - **Fraud Detection:** Identifying fraudulent transactions by combining results from multiple decision trees.
  - **Medical Diagnosis:** Predicting disease presence with complex datasets.

e. Support Vector Machines (SVM)

- **Context:** Classification tasks with a clear margin of separation between classes.
- **Examples:**
    - **Image Classification:** Recognizing handwritten digits (e.g., MNIST dataset).
    - **Text Classification:** Classifying news articles into topics.

f. K-Nearest Neighbors (KNN)

- **Context:** Classification or regression where relationships between instances are based on similarity.
- **Examples:**
    - **Recommendation Systems:** Recommending products based on the similarity to other users' preferences.
    - **Pattern Recognition:** Identifying patterns in image data.

g. Neural Networks

- **Context:** Complex tasks requiring learning from large amounts of data with non-linear relationships.
- **Examples:**
    - **Image Recognition:** Identifying objects within images (e.g., facial recognition).
    - **Natural Language Processing:** Machine translation, sentiment analysis.

2. Unsupervised Learning

a. Clustering (e.g., K-Means)

- **Context:** Grouping similar data points together without prior labels.
- **Examples:**
    - **Market Segmentation:** Grouping customers based on purchasing behavior.
    - **Anomaly Detection:** Identifying unusual patterns in network traffic.

b. Principal Component Analysis (PCA)

- **Context:** Dimensionality reduction for visualizing and interpreting high-dimensional data.
- **Examples:**
    - **Data Visualization:** Reducing dimensions of gene expression data for visualization.
    - **Feature Extraction:** Simplifying data for further analysis.

c. Association Rule Learning (e.g., Apriori)

- **Context:** Finding relationships between variables in large datasets.
- **Examples:**
    - **Market Basket Analysis:** Discovering associations between items purchased together.
    - **Recommender Systems:** Identifying products frequently bought together.

3. Reinforcement Learning

a. Q-Learning

- **Context:** Learning optimal actions in a given environment to maximize cumulative reward.
- **Examples:**
    - **Game Playing:** Training agents to play games like chess or Go.
    - **Robotics:** Teaching robots to navigate and perform tasks in dynamic environments.

b. Deep Q-Networks (DQN)

- **Context:** Handling complex environments with high-dimensional state spaces.
- **Examples:**
    - **Autonomous Vehicles:** Learning to drive by interacting with simulated environments.
    - **Complex Game Strategies:** Improving performance in complex games with large state spaces.

4. Semi-Supervised Learning

a. Self-Training

- **Context:** Using a small amount of labeled data and a large amount of unlabeled data to improve model performance.
- **Examples:**
    - **Text Classification:** Enhancing performance with limited labeled text data.
    - **Image Classification:** Improving accuracy with few labeled images and many unlabeled ones.

5. Natural Language Processing (NLP)

a. Recurrent Neural Networks (RNNs)

- **Context:** Processing sequences of data where context from previous steps is important.
- **Examples:**
    - **Language Modeling:** Predicting the next word in a sentence.
    - **Speech Recognition:** Converting spoken language into text.

b. Transformers (e.g., BERT, GPT)

- **Context:** Handling tasks requiring understanding of context and long-range dependencies.
- **Examples:**
    - **Text Summarization:** Generating concise summaries of long documents.
    - **Question Answering:** Answering questions based on context from a document.

6. Graph-Based Approaches

a. Graph Neural Networks (GNNs)

- **Context:** Learning from data represented as graphs with nodes and edges.
- **Examples:**
    - **Social Network Analysis:** Understanding relationships and influence within social networks.
    - **Knowledge Graphs:** Enhancing search and recommendation systems with entity relationships.

7. Image Processing

a. Convolutional Neural Networks (CNNs)

- **Context:** Handling image data and learning spatial hierarchies.
- **Examples:**
    - **Object Detection:** Identifying and locating objects within images.
    - **Facial Recognition:** Recognizing and verifying faces in images.

8. Spatial Analysis

a. Spatial Data Mining

- **Context:** Analyzing spatial and geographic data to uncover patterns.
- **Examples:**
    - **Urban Planning:** Analyzing geographic data to plan city infrastructure.
    - **Environmental Monitoring:** Studying environmental changes and patterns.

Summary

- **Supervised Learning:** Suitable for predictive tasks with labeled data.
- **Unsupervised Learning:** Useful for discovering patterns and relationships in unlabeled data.
- **Reinforcement Learning:** Ideal for decision-making problems in dynamic environments.
- **NLP:** Focuses on tasks involving language and text data.
- **Graph-Based Approaches:** Useful for data represented as networks or graphs.
- **Image Processing:** Specialized techniques for analyzing and interpreting image data.
- **Spatial Analysis:** Analyzes geographic and spatial data for insights.

Resources:

PCA: https://builtin.com/data-science/step-step-explanation-principal-component-analysis
Factor Analysis: https://www.datamation.com/big-data/what-is-factor-analysis/
Association Rules: https://www.geeksforgeeks.org/association-rule/
Neural Network Models: https://www.seldon.io/neural-network-models-explained/
Recurrent Neural Network (RNN): https://www.ibm.com/think/topics/recurrent-neural-networks
Feed Forward Neural Network: https://www.geeksforgeeks.org/feedforward-neural-network/
Deep Neural Network: https://www.sciencedirect.com/topics/computer-science/deep-neural-network
Convolutional Neural Networks: https://www.geeksforgeeks.org/introduction-convolution-neural-network/
Generative Adversarial Neural Network: https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/
Transformer Neural Networks: https://builtin.com/artificial-intelligence/transformer-neural-network
LSTM networks: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/
Graph Neural Networks: https://distill.pub/2021/gnn-intro/

Lecture Outline: Clustering Methods and Rescaling in Python
Duration: 50 minutes

---

1. Introduction to Clustering Methods (5 minutes)

- **Objective:** Understand the concept of clustering and its applications.
- **Content:**
    - **Definition:** Clustering is an unsupervised learning technique used to group similar data points into clusters.
    - **Applications:** Market segmentation, social network analysis, image compression.

---

2. K-Means Clustering (10 minutes)

- **Objective:** Implement and understand K-Means clustering in Python.
- **Content:**
    - **Using scikit-learn:**

```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
import matplotlib.pyplot as plt

# Generate synthetic data
X, _ = make_blobs(n_samples=300, centers=4, cluster_std=0.60, random_state=0)

# Fit K-Means
kmeans = KMeans(n_clusters=4)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)

# Plot
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='red', s=200, alpha=0.75)
```

```
plt.title('K-Means Clustering')
plt.show()
```

-
  - **Discussion:**
    - **Choosing K:** Use methods like the Elbow Method to determine the number of clusters.
    - **Pitfalls:** K-Means assumes spherical clusters and can be sensitive to initial cluster centers.

---

3. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) (10 minutes)

- **Objective:** Implement and understand DBSCAN in Python.
- **Content:**
  - **Using scikit-learn:**

```
from sklearn.cluster import DBSCAN

# Fit DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=5)
y_dbscan = dbscan.fit_predict(X)

# Plot
plt.scatter(X[:, 0], X[:, 1], c=y_dbscan, s=50, cmap='viridis')
plt.title('DBSCAN Clustering')
plt.show()
```

-
  - **Discussion:**
    - **Advantages:** Can find arbitrarily shaped clusters and is robust to noise.
    - **Pitfalls:** Requires careful selection of the eps parameter and min_samples.

---

4. Hierarchical Clustering (10 minutes)

- **Objective:** Implement and understand Hierarchical Clustering in Python.
- **Content:**
  - **Using scipy:**

```
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster

# Perform hierarchical clustering
Z = linkage(X, 'ward')
plt.figure(figsize=(10, 7))
dendrogram(Z)
plt.title('Dendrogram')
plt.show()

# Cut the dendrogram to form clusters
clusters = fcluster(Z, t=4, criterion='maxclust')
```

```
plt.scatter(X[:, 0], X[:, 1], c=clusters, s=50, cmap='viridis')
plt.title('Hierarchical Clustering')
plt.show()
```

-
    - **Discussion:**
        - **Advantages:** Does not require a predefined number of clusters.
        - **Pitfalls:** Computationally expensive for large datasets.

---

5. Clustering with HDBSCAN (Hierarchical DBSCAN) (8 minutes)

- **Objective:** Use HDBSCAN for clustering with better performance on varying densities.
- **Content:**
    - **Using hdbscan:**

```
import hdbscan

# Fit HDBSCAN
hdbscan_model = hdbscan.HDBSCAN(min_cluster_size=10)
y_hdbscan = hdbscan_model.fit_predict(X)

# Plot
plt.scatter(X[:, 0], X[:, 1], c=y_hdbscan, s=50, cmap='viridis')
plt.title('HDBSCAN Clustering')
plt.show()
```

-
    - **Discussion:**
        - **Advantages:** Effective at handling varying densities and shapes.
        - **Pitfalls:** Requires tuning of min_cluster_size and other parameters.

---

6. Rescaling Methods (7 minutes)

- **Objective:** Understand and apply rescaling methods in data preprocessing.
- **Content:**
    - **Standardization (Z-score Normalization):**

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Min-Max Scaling:

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
X_minmax = scaler.fit_transform(X)
```

-

- **Discussion:**
    - **Standardization:** Centers data around zero with unit variance.
    - **Min-Max Scaling:** Scales data to a specific range, usually [0, 1].

---

7. Pitfalls of Rescaling Before vs. After Train-Test Split (5 minutes)

- **Objective:** Understand the impact of rescaling on model evaluation.
- **Content:**
  - **Rescaling Before Train-Test Split:**
    - **Pitfall:** Data leakage, as information from the test set can influence scaling parameters.
    - **Solution:** Always perform rescaling after splitting the data.
  - **Rescaling After Train-Test Split:**
    - **Correct Approach:** Fit the scaler on the training data and apply the same transformation to the test data.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

-
  - **Discussion:** Ensures that test data remains unseen and unbiased during training.

---

8. Q&A and Discussion (5 minutes)

- **Objective:** Address questions and discuss practical considerations for clustering and rescaling.
- **Content:**
  - **Q&A Session:** Open the floor for student questions.
  - **Discussion:** Explore scenarios and best practices for clustering and rescaling in various data contexts.

Key Takeaways

- **Clustering Methods:** Overview of K-Means, DBSCAN, Hierarchical Clustering, and HDBSCAN with practical examples.
- **Rescaling:** Importance of rescaling in preprocessing and the correct approach to avoid data leakage.
- **Python Libraries:** Practical implementations using scikit-learn, scipy, and hdbscan.

Resources:
Which rescaling method should I use?: https://medium.com/@hhuseyincosgun/which-data-scaling-technique-should-i-use-a1615292061e
Clustering in Machine Learning: https://www.geeksforgeeks.org/clustering-in-machine-learning/
K-Means: https://www.geeksforgeeks.org/k-means-clustering-introduction/
DBSCAN: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html
Hierarchical Clustering: https://www.datacamp.com/tutorial/introduction-hierarchical-clustering-python
Hierarchical DBSCAN: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.HDBSCAN.html
When to rescale?: https://dev.to/gervaisamoah/why-feature-scaling-should-be-done-after-splitting-your-dataset-into-training-and-test-sets-14ia#

Spectral Clustering: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html
Fuzzy Clustering: https://www.geeksforgeeks.org/ml-fuzzy-clustering/
Mean Shift Clustering: https://www.geeksforgeeks.org/ml-fuzzy-clustering/
Affinity Propagation: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html
OPTICS: https://www.geeksforgeeks.org/ml-optics-clustering-implementing-using-sklearn/
BIRCH: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html