Model Selection

Predicting material properties using FESEM (Field Emission Scanning Electron Microscopy) images as input is a challenging and valuable application of machine learning and deep learning in materials science. Various algorithms can be considered for this task, and the choice depends on the nature of the material property to be predicted and the specific characteristics of the FESEM images. We here will be looking at k-means clustering

Implementation

K-Means clustering is typically used for unsupervised learning, where it groups data points into clusters based on their similarity. While it may not be used directly for predicting material properties, K-Means clustering can be valuable for exploratory data analysis and feature engineering when using FESEM (Field Emission Scanning Electron Microscopy) images as input for material property prediction. Here's how K-Means clustering can be applied in this context:

Data Preprocessing:

> Collect a dataset of FESEM images and their corresponding material property values. The material properties serve as labels for supervised learning.
>
> Preprocess the FESEM images, including resizing, normalization, and feature extraction. You can use techniques like Histograms of Oriented Gradients (HOG) or Local Binary Patterns (LBP) to represent each image as a feature vector.

Feature Extraction (if needed):

> Extract relevant features from the preprocessed FESEM images. This is necessary to represent each image as a feature vector for clustering. Different feature extraction techniques can be applied based on the nature of the images and the material properties of interest.

Data Preparation:

> Combine the feature vectors of the FESEM images with their material property values to create a dataset suitable for K-Means clustering. The feature vectors will serve as input data points, and the material properties will be used for analysis and interpretation.

K-Means Clustering:

> Apply the K-Means clustering algorithm to the feature vectors of the FESEM images. The algorithm will group similar images into clusters based on their feature similarities.

Cluster Analysis:

> Analyze the clusters created by K-Means to understand the structure of the data. This can help identify patterns or subgroups within the dataset, which may

correspond to different material properties or characteristics. By analyzing these clusters, you can gain insights into how the material properties may be related to the images.

Visualization:

Visualize the results of the clustering using techniques like scatter plots or t-SNE (t-distributed stochastic neighbor embedding) to see how data points (FESEM images) are distributed in feature space.

Feature Engineering:

Use the cluster assignments as new features to enrich your dataset. For each FESEM image, create a feature that represents the cluster it belongs to. This can be used as input for a subsequent supervised learning model for material property prediction.

Material Property Prediction:

After creating new features based on the cluster assignments, you can build a supervised machine learning model (e.g., regression or classification) to predict material properties using these features. The cluster assignments can serve as additional information that may improve predictive accuracy.

Model Evaluation:

Evaluate the predictive model's performance using appropriate metrics (e.g., Mean Absolute Error for regression or accuracy for classification) to assess how well it predicts material properties.

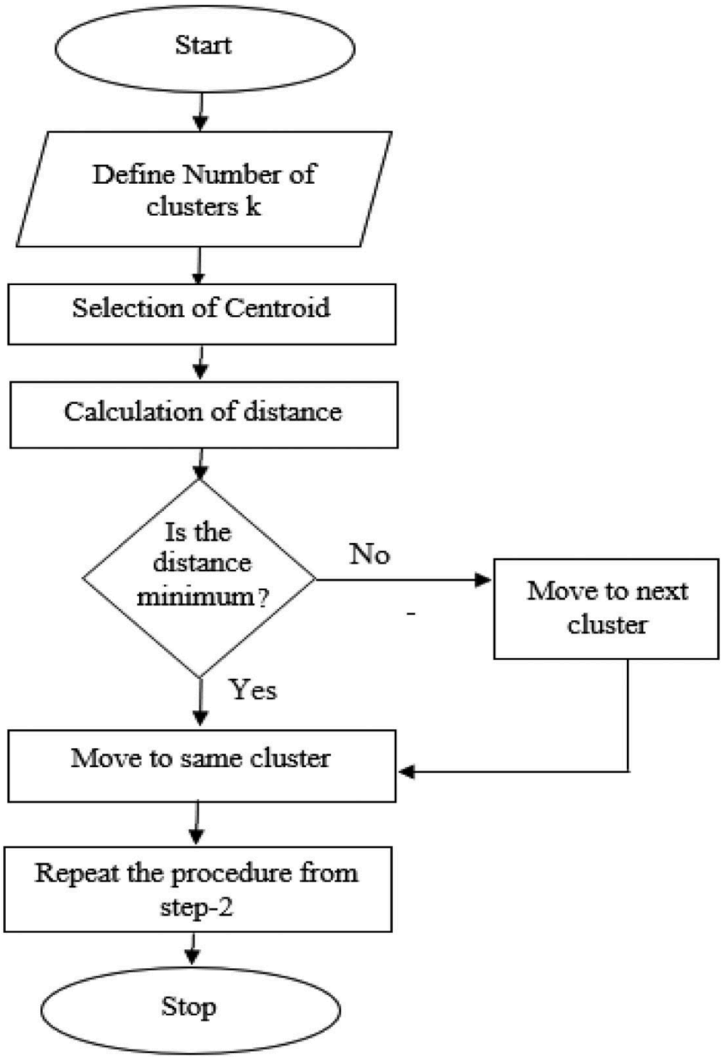
The k means clustering technique is a method based on the iterative technique applied for clustering a set of 'N' vector points into 'k' number of groups, refereed as clusters of points. Clustering is the procedure of separating a collection of data set points into the insignificant size of groups. In a general way, it can be understood for 'n' data points (X_i), where (i = 1, 2 ... n) that essentially be subdivided in 'k' number of clusters. The objective is to allocate a bunch against individual data points. The K- means algorithm is a clustering technique, used to determine the locations μ_i, where, (i = 1, 2 ... k) of the bunches that diminish the least distance from the data set points to the cluster or bunch. The K-means clustering solves

$$\arg.min \sum_{i=1}^{k} \sum_{x \epsilon c_i} d(x, \mu_i) = \arg.\min \sum_{i=1}^{k} \sum_{x \epsilon c_i} \|x - \mu_i\|_2^2$$

Here, c_i = set points that have their place to cluster 'i'. The partitioning in the K- means clustering technique follows the concept of Euclidean distance calculation based on square distance.

$$d(x, \mu_i) = \|x - \mu_i\|_2^2$$

Flowchart



Step-1: Begin the midpoint as the center of the cluster
μ_i = some value, i = 1, 2, …..k

Step-2: Feature the neighboring cluster against each data point
$c_i = \{ j : d(x_j,\mu_i) <= d(x_j,\mu_l)$, $l != i$, $j = 1,2,\ldots.n \}$

Step-3: Fix the location of an individual cluster for the means of all set data-points associated with that group.

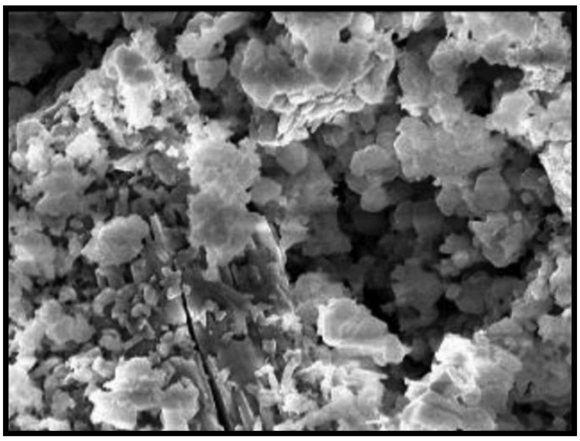
Step-4: Replicate the step-2 and step-3 until convergence is obtained
|c|=Count of present elements in c

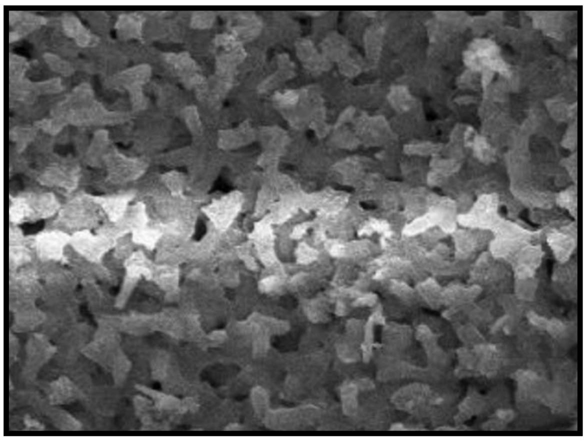
Code

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import adjusted_rand_score

# Loading FESEM image data and material labels
# Make sure 'X' contains your image data, and 'y' contains the corresponding material labels

fesemds = datasets.load_fesemds()
X = fesemds.data
# Features (FESEM images)
y = fesemds.target
# Material labels

# We can reduce the dimensionality of the data for visualization (PCA)
pca = PCA(n_components=2)
X_reduced = pca.fit_transform(X)

# Set the number of clusters
n_clusters = 3

# Create and fit the K-Means model
kmeans = KMeans(n_clusters=n_clusters, random_state=0)
kmeans.fit(X)

# Get cluster assignments for the data points
cluster_labels = kmeans.labels_

# Visualize the clustering results and original data (optional)
```

```
plt.figure(figsize=(12, 6))

# Original data (FESEM images)
plt.subplot(1, 2, 1)
plt.scatter(X_reduced[:, 0], X_reduced[:, 1], c=y, cmap='viridis')
plt.title("Original Data")

# K-Means clustering results
plt.subplot(1, 2, 2)
plt.scatter(X_reduced[:, 0], X_reduced[:, 1], c=cluster_labels, cmap='viridis')
plt.title("K-Means Clustering")

plt.show()

# Evaluate the clustering quality using the Adjusted Rand Index (ARI)
ari = adjusted_rand_score(y, cluster_labels)
print(f"Adjusted Rand Index: {ari}")
```

Sample Images





Field Emission Scanning Electron Microscopy Images

Future Improvements:

When applying K-Means clustering to FESEM images for material property prediction or exploratory data analysis, several future improvements and enhancements can be considered to enhance the effectiveness and robustness of the algorithm. Here are some potential areas for improvement:

Feature Engineering and Selection:

> Experiment with different feature extraction techniques to capture more meaningful information from FESEM images. Consider deep learning-based feature extraction or unsupervised feature learning methods to identify relevant patterns in the images.

Dimensionality Reduction:

> Employ dimensionality reduction techniques like Principal Component Analysis (PCA) or t-SNE to reduce the feature space's dimensionality while preserving important information. This can improve clustering performance.

Optimal Cluster Number Selection:

> Determine the optimal number of clusters (k) using techniques such as the Elbow Method, Silhouette Score, or Gap Statistics. This ensures that the number of clusters aligns with the inherent structure in the data.

Cluster Validation:

> Implement cluster validation techniques like silhouette analysis or Davies-Bouldin index to assess the quality and separation of clusters. This helps to ensure that the clusters are meaningful.

Outlier Detection:

> Incorporate outlier detection methods, such as the Local Outlier Factor (LOF) or Isolation Forest, to identify and handle outliers in your data effectively. Outliers can negatively impact the clustering results.