Satyam Kumar Follow

May 8, 2021 · 3 min read ★ · Listen

Clustering Algorithm for data with mixed Categorical and Numerical features

k-Modes and k-Prototype algorithm intuition and usage

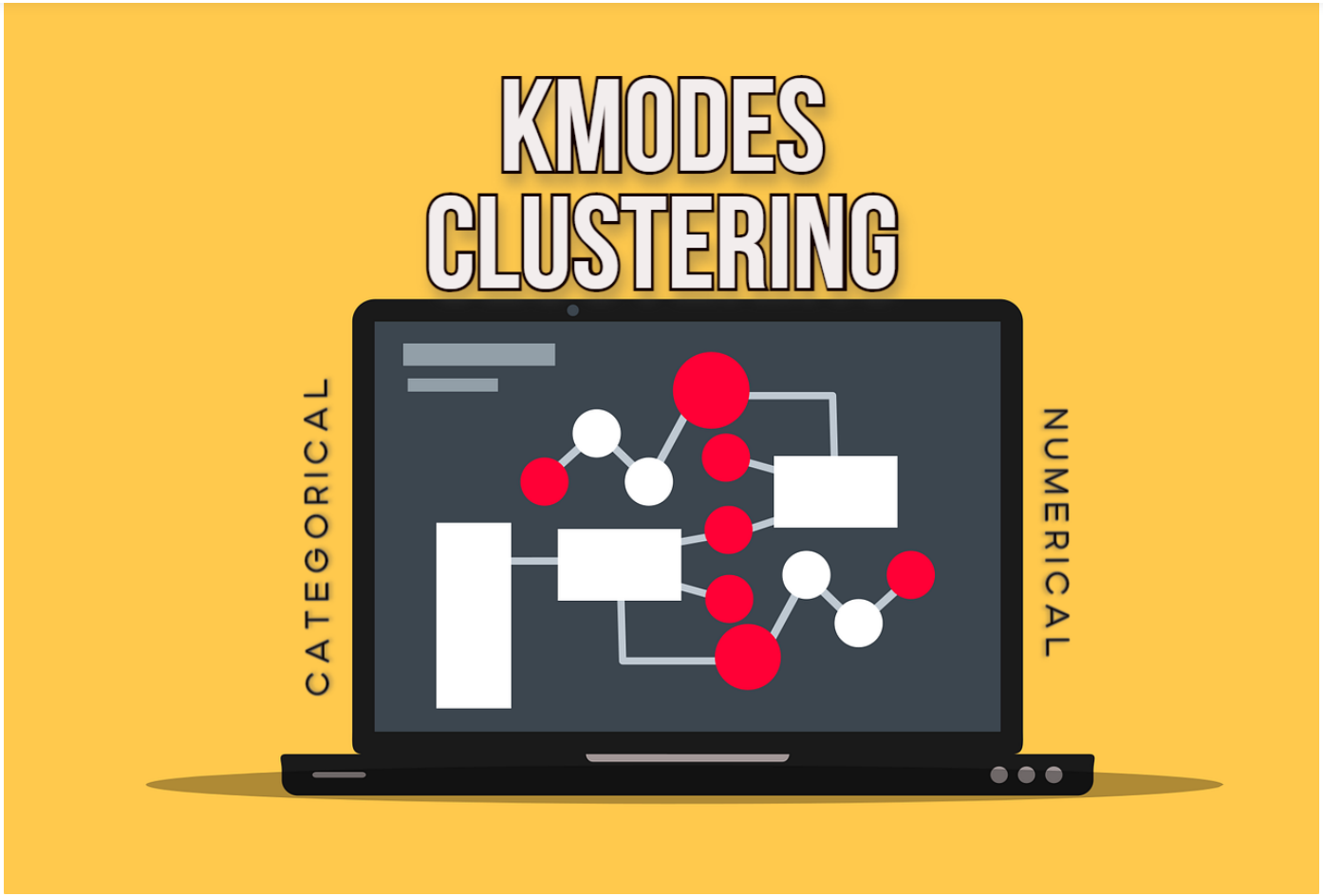

Image by Mohamed Hassan from Pixabay

Clustering is an unsupervised machine learning technique that devices the population into several clusters or groups in such a way that data points in a cluster are similar to each other, and data points in different clusters are

Why k-Means can't be used for Categorical features?

k-Means is a popular centroid-based clustering algorithm, that divides the data points of the entire population into k clusters each having an almost equal number of data points. The idea behind the k-Means clustering algorithm is to find k-centroid points and every point in the dataset will belong to either of the k-sets having minimum Euclidean distance.

The k-Means algorithm is not applicable to categorical data, as categorical variables are discrete and do not have any natural origin. So computing euclidean distance for such as space is not meaningful.

There is a certain variation to the k-Means algorithm, called k-Modes which is suitable for data with categorical features. k-Prototype is an extension of the k-Modes algorithm that works for mixed categorical and numerical features.

What is k-Modes and k-Prototype Algorithm:

k-Modes is an algorithm that is based on the k-Means algorithm paradigm and it is used for clustering categorical data. k-modes defines clusters based on matching categories between the data points. The k-Prototype algorithm is an



Installation:

k-modes and k-prototype algorithm can be implemented using an open-source library `kmodes` . kmodes library can be installed from PyPl using:

```
pip install kmodes
```

Usage:

As discussed earlier `kmodes` algorithm is used to cluster only the categorical variables. While one can use `KPrototypes()` function to cluster data with a mixed set of categorical and numerical features.

The dataset used for demonstrations contains both categorical and numerical features.

```
from kmodes.kprototypes import KPrototypes

kproto = KPrototypes(n_clusters=2, verbose=2, max_iter=20)
kproto.fit(df_array, categorical=cat_idx)
```



`KPrototypes` function is used to cluster the dataset into given `n_clusters` (number of clusters). The developers need to assign the indexes of the categorical features as a parameter while fitting the training dataset. While training `KPrototypes` algorithm will training the categorical features using the k-Modes algorithm and the remaining numerical features will be trained using the standard k-Means algorithm.

After training one can get the centroid of the clusters using `cluster_centroids_()` function.

```
print(kproto.cluster_centroids_)
```

Use the function `predict()` to predict the clusters.

```
clusters = kproto.predict(df_array, categorical=cat_idx)
```

Implementation



KPrototypes.ipynb hosted with ❤️ by GitHub view raw

(Code by Author)

Conclusion:

In this article, we have discussed how to apply clustering to the dataset having a

with mixed data types. The k-Modes algorithm can handle missing or NaN values, but it's suggested to impute the values for better performance.

The implementation of the kModes library is modeled after the clustering algorithms in `scikit-learn` and has the same API.

> *Read the below-mentioned article to get more understanding of k-Means, k-Means++, and k-Medoids algorithms.*

Understanding K-means, K-means++ and, K-medoids Clustering Algorithms

Understand an overview of K-means, K-means++ and, K-Medoids clustering algorithms, and their relations. This article...

towardsdatascience.com

References:

[1] KModes library GitHub repository: https://github.com/nicodv/kmodes