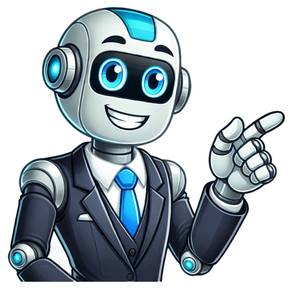
Continue



K means algorithm with example

Given article text here Browse through thousands of daily free articles to learn new things. The backup process in MS SQL Server is crucial for safeguarding data against system failures, hardware issues, and database corruption. MS SQL Server offers various types of backups, including full, differential, and transactional backups. A full backup allows you to restore the entire database from a previous point in time. We have previously discussed Creational, Structural, and Behavioral design patterns in Python applications. These fundamental patterns provide insights into crafting elegant and maintainable code. However, as we delve deeper into Python, unique language-specific design patterns emerge. These patterns are essential for building complex projects with Python. Python enables developers to create a wide range of applications, from simple scripts to machine learning models. Working on complex projects requires the use of or creating modules, which serve as the building blocks of these projects. In this article, we'll explore Python modules and their importance in project development. JavaScript is an asynchronous language that uses callbacks for task management. Although ES6 introduced promises, many early libraries still rely on callbacks. Comprehensive documentation is vital when working with Python code. Python places great emphasis on readability and straightforward documentation. One key aspect of Python documentation is the docstring, which serves as a crucial tool for programmers. GPUs have evolved significantly in recent years, particularly in the field of artificial intelligence. They play a pivotal role in enhancing computational speed and efficiency. In Natural Language Processing (NLP), parts of speech tagging is a fundamental task that involves assigning grammatical categories to words. This process plays a significant role in many NLP applications, including text analysis and information retrieval. Splitting strings is an essential string operation that allows developers to divide large texts into smaller components. While splitting using a single delimiter is common, there are alternative methods for specific use cases. In today's digital world, natural language processing techniques help process vast amounts of text data from various sources. This enables developers to manipulate, analyze, and interpret text data more efficiently. Data structures are fundamental in computer science, providing a foundation for organizing and accessing data. The list data structure is one of the most versatile and widely used data structures in programming. As developers work with APIs and web services, understanding API documentation is essential for successful project development. Given text: paraphrase this text: is sending data in JSON (JavaScript Object Notation) format to a server. Fortunately, Python provides us with a powerful requests library, which makes HTTP requests a breeze, including sending JSON data. In this article, we'll go Page 13You are part of a project that will use deep learning to try to identify what is in images - such as cars, ducks, mountains, sky, trees, etc. In this project, two things are important - the first one, is that the deep learning model trains quickly, with efficiency (because Page 14Email is an essential method of business communication that is very fast, cheap, and top-free organic. When building products, emails can greatly provide your business with an effective and efficient medium to transmit various kinds of data, electronically. In this article, we will learn how to build a form for Page 15When developing a large application, it is always a good idea to break it down into smaller components to make the code easier to read, structure, and maintain. Most Vue beginners understand what components are at least conceptually, but they might not understand what they can and cannot do fully.... Page 16In Python, strings are sequences of characters, which are effectively stored in memory as an object. Each object can be identified using the id() method, as you can see below. Python tries to re-use objects in memory that have the same value, which also makes comparing objects very fast in Page 17 In Python, functions are objects: they can be assigned to variables, can be returned from other functions, stored in lists or dicts and passed as parameters for other functions. Consider, for example, the map() built-in function. Its syntax is map(function, iterable) and it is used to handily apply function Page 18Python's json library has many utilities for encoding and decoding data in JSON format. In particular, the json.load() method decodes an JSON read as a file, and the json.loads() decode an JSON read as a string. In general, when decoding JSON files, the data is converted to Python Page 19Turning a collection of numbers (or objects whose fields you'd like to inspect) into a distribution of those numbers is a common statistical technique, and is employed in various contexts in reporting and data-driven applications. Given a collection: 1, 1, 1, 2, 1, 2, 3, 1, 4, 5, 1, 3 You Page 20During textual processing, whether you're searching for certain words and making pattern matching rules, counting the frequency of elements, etc. - punctuation can throw a wrench in your plans. Oftentimes, you'll want to remove stopwords, punctuation, digits or otherwise some category of characters, depending on what your end goal is....Page 21Data augmentation has, for a long while, been serving as a means of replacing a "static" dataset with transformed variants, bolstering the invariance of Convolutional Neural Networks (CNNs), and usually leading to robustness to input. Note: Invariance boils down to making models blind to certain perturbations, when making Page 22Python programmers intensively use arrays, lists, and dictionaries as serialized data structures. Storing these data structures persistently requires either a file or a database to properly work. In this article, we'll take a look at how to write a list to file, and how to read that list back into Page 23Missing values are common and occur either due to human error, instrument error, processing from another team, or otherwise a lack of data for a specific observation. Here, we'll explore how to handle missing values in a DataFrame if you choose to deal with them by... It's always beneficial to include visuals on a website or web application as they engage users, but when these images are too large and distort the page, it has the opposite effect. In this article, we will learn how to... When developing web applications using React, we typically want to incorporate visual elements to capture users' attention. These visual elements could be any type of image, including JPGs, PNGs, SVGs, GIFs, and many others. In this article, we will learn how to import images with React and explore... Axios is a promise-based HTTP client library that simplifies sending asynchronous HTTP requests (such as POST, GET, and DELETE) to REST endpoints, primarily APIs. In this article, we will learn how to send POST JSON requests with Axios and handle both serialized and unserialized data. K-means clustering algorithm computes the centroids and iterates until it finds an optimal centroid. It assumes that the number of clusters are already known. It is also called a flat clustering algorithm. The number of clusters identified from data by the algorithm is represented by 'K' in K-means. In this algorithm, data points are assigned to a cluster in such a manner that the sum of the squared distance between the data points and centroid would be minimum. The working of K-Means clustering algorithm can be understood with the following steps − First, specify the number of clusters, K, need to be generated by this algorithm. Next, randomly select K data points and assign each data point to a cluster. Now, compute the cluster centroids. Then, keep iterating until we find an optimal centroid which is the assignment of data points to the clusters that are not changing any more. K-means follows Expectation-Maximization approach to solve the problem. The algorithm requires standardizing data because it uses distance-based measurement to determine the similarity between data points. Due to the iterative nature of K-Means and random initialization of centroids, K-Means may stick in a local optimum and may not converge to global optimum. That is why it is recommended to use different initializations of centroids. The K-Means algorithm is straightforward and efficient, and it can... Given text: The K-Means Clustering Algorithm ========== Overview -------- K-Means clustering is a widely used unsupervised machine learning algorithm for segmenting data into clusters based on their similarities. It has numerous applications in data analysis, pattern recognition, and visualization. Limitations of K-Means Clustering ---------- While K-Means clustering is a powerful tool, it has some limitations: * **Sensitivity to initial centroids**: The results of K-Means clustering can be highly sensitive to the choice of initial centroids. * **Tendency to converge to local optima**: K-Means clustering may converge to local optima instead of the global optimum. * **Assumption of equal variance for all clusters**: K-Means clustering assumes that the variance of each cluster is equal, which may not be true in reality. Implementation of K-Means Algorithm in Python ---------- Python has several libraries that provide implementations of various machine learning algorithms, including K-Means clustering. In this section, we will explore how to implement the K-Means algorithm in Python using the scikit-learn library. Example 1: Simple Example ---------- This example demonstrates how to implement the K-Means algorithm in Python using the scikit-learn library. ### Step 1: Import Required Libraries ```python import numpy as np import matplotlib.pyplot as plt from sklearn.cluster import KMeans ``` ### Step 2: Generate Data ```python X = np.random.rand(300,2) plt.figure(figsize=(7.5, 3.5)) plt.scatter(X[:, 0], X[:, 1], s=20, cmap='summer') plt.show() ``` ### Step 3: Initialize K-Means Algorithm ```python kmeans = KMeans(n_clusters=3, max_iter=100) ``` ### Step 4: Train the Model ```python kmeans.fit(X) ``` ### Step 5: Visualize the Clusters ```python plt.figure(figsize=(7.5, 3.5)) plt.scatter(X[:,0], X[:,1], c=kmeans.labels_, s=20, cmap='summer') plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], marker='x', c='r', s=50, alpha=0.9) plt.show() ``` Example 2: 2D Dataset with Four Blobs ---------- In this example, we generate a 2D dataset containing four blobs and apply the K-Means algorithm to see the result. ### Step 1: Import Required Libraries ```python %matplotlib inline import matplotlib.pyplot as plt import seaborn as sns; sns.set() import numpy as np from sklearn.cluster import KMeans ``` ### Step 2: Generate Dataset ```python from sklearn.datasets import make_blobs X, y_true = make_blobs(n_samples=400, centers=4, cluster_std=0.60, random_state=0) ``` ### Step 3: Visualize Dataset ```python plt.scatter(X[:, 0], X[:, 1], s=20) plt.show() ``` ### Step 4: Initialize and Train K-Means Algorithm ```python kmeans = KMeans(n_clusters=4) kmeans.fit(X) y_kmeans = kmeans.predict(X) ``` ### Step 5: Plot and Visualize Clusters ```python plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=20, cmap='summer') ``` Given text: Looking forward to seeing everyone at the meeting tomorrow and discussing our strategies. Rewritten (SE): Lookin forward to seein everyone at the meeting tomorow and discussin our strategies. Rewritten (NNES): Looking forward to seeing everyone at the meeting tomorrow to discuss our strategy. Rewritten (IB): Looking forward to the meeting tomorrow, where we'll see everyone and discuss our strategies in detail. Given article text here The technique of K-means clustering has diverse applications across various domains such as customer retention, loyalty programs, targeted advertising, fraud detection, network intrusion detection, predictive maintenance, bioinformatics, and drug discovery among others. It can be utilized to identify data points that do not belong to any cluster by leveraging anomaly detection in a dataset. K-means clustering is also employed in genomic data analysis to analyze gene expression data and identify groups of genes that are co-regulated or co-expressed. This method is widely used in bioinformatics applications, such as disease diagnosis and personalized medicine. The K-means algorithm assigns each observation to exactly one cluster after specifying the desired number of clusters. It is a simple yet elegant approach for partitioning a dataset into distinct non-overlapping groups. The results obtained from performing K-means clustering can be seen in various simulated examples consisting of observations in two dimensions, using different values of K. Clustering is an essential concept in data science that enables grouping similar items and products, helping us understand our data in a unique way by forming clusters. In this article, we will delve into the world of k-means clustering comprehensively, covering its applications, importance, and deep diving into the algorithm itself Objects are grouped into clusters that share similarities, but not the same characteristics as objects in other clusters. A "K" value is a number that determines how many groups to create, for example K = 2 means two groups. Finding the best K value for specific data involves understanding the algorithm used. The K-means clustering procedure results from a mathematical problem where observations are assigned to non-overlapping clusters. Each observation belongs to one cluster, and no observation can belong to more than one group. The goal of K-means is to have small variation within each cluster by minimizing distances between observations in the same cluster. This involves defining a formula that measures how much observations differ from each other within a cluster, typically using squared Euclidean distance. An optimization problem is created to find an algorithm that minimizes this variation. Solving this problem precisely is challenging due to the large number of ways observations can be grouped into clusters, but a simple algorithm provides a local optimum solution. This algorithm randomly assigns initial cluster assignments and iterates until these assignments stop changing by computing centroid vectors for each cluster, then assigning observations based on which centroid is closest. Finding the optimal K value involves running K-means with different values and determining the best one using criteria such as the elbow or silhouette method. The elbow method is a technique used to determine the optimal number of clusters in unsupervised learning algorithms. It involves finding a metric that evaluates the quality of clustering for various values of K and identifying the "elbow point" where the relative improvement is not very high anymore. Two popular metrics used are the average within-cluster sum of squares and the percentage of variance explained. The former measures how cohesive the clusters are, while the latter normalizes with the total variance in the dataset. The Silhouette Method is another approach to determine the optimal number of clusters. It assumes that the data has already been clustered into k clusters using a clustering technique. For each data point, it calculates two values: b(i), which gives a measure of how well assigned the ith data point is to its cluster, and s(i), the silhouette coefficient. The average silhouette for each value of k is calculated, and the value of k with the maximum value of s(i) is considered the optimal number of clusters. For example, given a dataset with 8 points, the Silhouette Method can be applied by iterating values of k from 2 to 5. The tables show the average silhouette for each value of k, and it is concluded that the optimal number of clusters is 3. The K-Means algorithm works by iteratively updating the cluster assignments until convergence or a stopping criterion is met. The goal is to find clusters in the given input data. The initial step in the K-means process involves determining the optimal number of clusters, denoted as 'k'. Subsequently, we randomly select 'k' points from the dataset to serve as centroids. For instance, if we desire two clusters, 'k' would be equal to 2, and we would then randomly assign a centroid for each cluster. Here, the red and green circles represent the centroids of these two clusters. Following this initialization, all data points are assigned to their nearest cluster centroid. The next step involves recalculating the centroids of newly formed clusters after assigning each point to its closest cluster. This process is repeated iteratively until convergence, where the calculated centroids remain stable and no data points switch between clusters. K-means offers several advantages, including a step-by-step approach that simplifies understanding, independence from programming languages, logical sequence, and ease of debugging. However, it also has some limitations such as being time-consuming, difficulty in representing branching and looping, and applicability to large tasks. The K-means algorithm finds practical applications in scenarios where grouping similar data points is essential. Some examples include optimizing delivery routes using a combination of K-means and genetic algorithms, identifying crime-prone areas based on crime statistics, cyber-profiling, and analyzing player stats for fantasy leagues. KMeans clustering involves grouping observations into k clusters based on their proximity to a cluster centroid. The algorithm works by randomly selecting initial cluster centroids, calculating the Euclidean distance between each point and its nearest centroid, assigning points to clusters, recalculating centroids, and iterating until convergence. A key step is measuring distances between points and centroids by squaring differences in X and Y coordinates. The optimal number of clusters (k) often equals the number of classes, but can be determined using the Elbow method, which involves running KMeans clustering for various k values and selecting the one with the most stable Sum of Squares Errors (SSE). This process continues until the maximum number of iterations is reached or computational resources are reduced. Clusters involve more computation costs. When data is evenly distributed, the elbow method doesn't clearly define the optimal K value. Instead, one can choose the best K programmatically or by using the silhouette coefficient or gap statistics. KMeans inertia (or Sum of Squares Errors) calculates the sum of distances between all points within a cluster and its centroid. The sum of squares errors often has alternative names: Inertia (Scikit-learn), Distortion (SciPy). KMeans is best suited for unsupervised learning problems with numeric data, which doesn't work on categorical data. It's also better suited for linear datasets. The advantages include the ability to define clusters, but it has disadvantages such as handling outliers and failing on non-linear datasets. KMeans can perform poorly when cluster shapes are not round or equally sized/dense, especially with many outliers. In these cases, HDBSCAN clustering algorithm and document comparison embeddings can be used, which understand semantic meaning to help cluster documents. For this article, we will use KMeans clustering using Scikit-learn. We'll create dummy data for clustering, train the model using 5 clusters and 10 iterations with different centroid seeds. Then, we'll plot the clustered data and pick the best value for K using the Elbow method. K-means Algorithm Tutorial: Understanding Clustering and Its Applications ========== Technical SEO expert with a background in writing, Python, information retrieval, SEO, and machine learning. Published as a guest author on prominent platforms including SearchEngineJournal, SearchEngineLand, and OnCrawl.