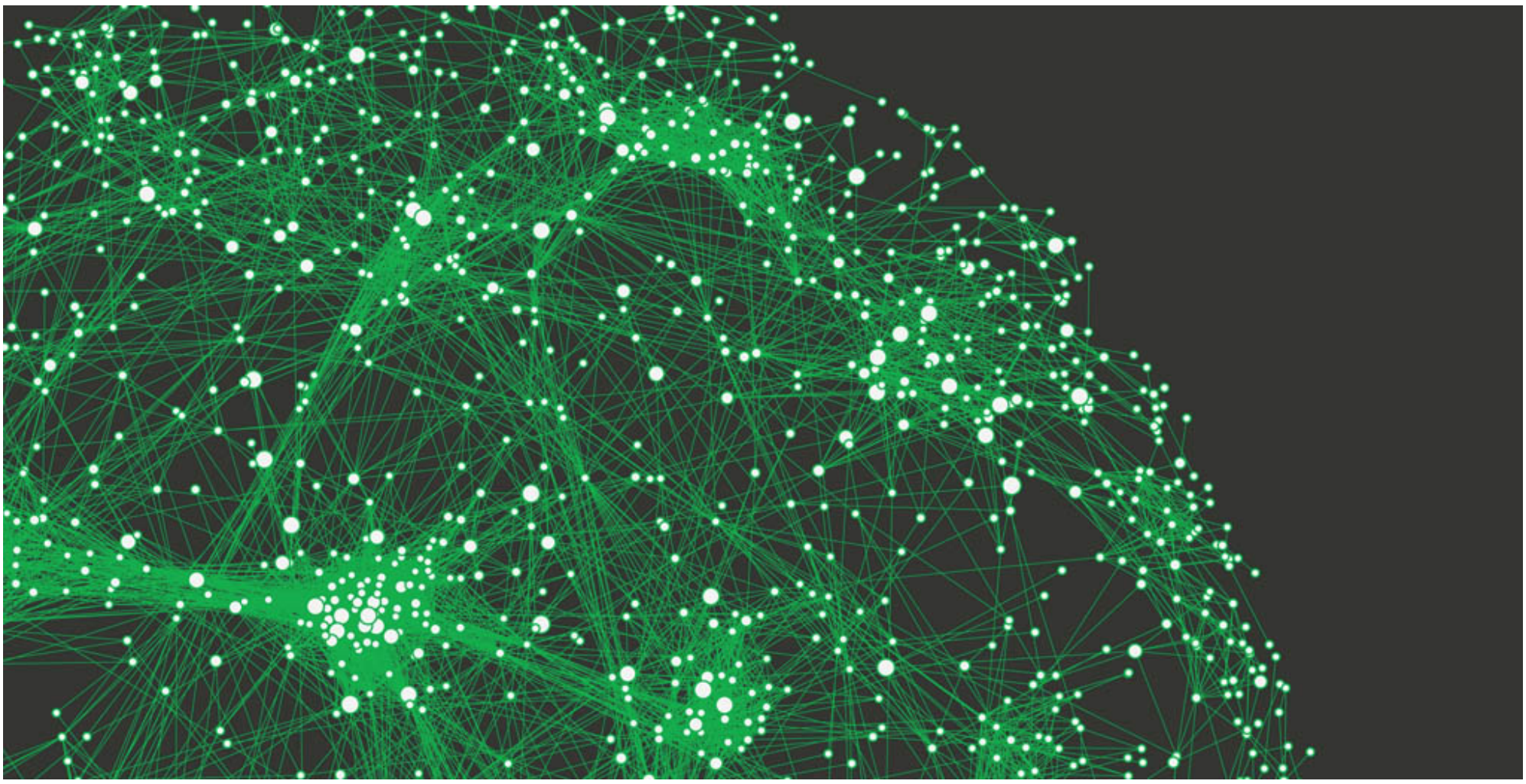
In-depth Intuition of K-Means Clustering Algorithm in Machine Learning

Basil Saji — Published On January 20, 2021 and Last Modified On June 23rd, 2022

Beginner Classification Clustering Machine Learning Python Structured Data Technique Unsupervised

This article was published as a part of the Data Science Blogathon.

Introduction



Clustering is an unsupervised machine learning technique. It is the process of division of the dataset into groups in which the members in the same group possess similarities in features. The commonly used clustering algorithms are K-Means clustering, Hierarchical clustering, Density-based clustering, Model-based clustering, etc. In this article, we are going to discuss K-Means clustering in detail.

K-Means Clustering

It is the simplest and commonly used iterative type unsupervised learning algorithm. In this, we randomly initialize the **K** number of centroids in the data (the number of k is found using the **Elbow** method which will be discussed later in this article) and iterates these centroids until no change happens to the position of the centroid. Let's go through the steps involved in K means clustering for a better understanding.

1) Select the number of clusters for the dataset (K)

2) Select K number of centroids

In-depth Intuition of K-Means Clustering Algorithm in Machine Learning

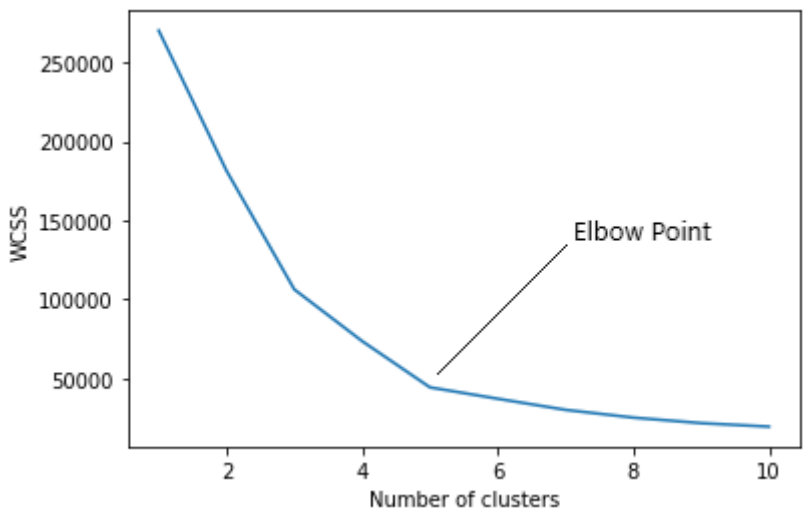
4) Now find the original centroid in each group

5) Again reassign the whole data point based on this new centroid, then repeat step 4 until the position of the centroid doesn't change.

Finding the optimal number of clusters is an important part of this algorithm. A commonly used method for finding optimal K value is **Elbow Method.**

Elbow Method

In the Elbow method, we are actually varying the number of clusters (K) from 1 – 10. For each value of K, we are calculating WCSS (Within-Cluster Sum of Square). WCSS is the sum of squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K value, the plot looks like an Elbow. As the number of clusters increases, the WCSS value will start to decrease. WCSS value is largest when K = 1. When we analyze the graph we can see that the graph will rapidly change at a point and thus creating an elbow shape. From this point, the graph starts to move almost parallel to the X-axis. The K value corresponding to this point is the optimal K value or an optimal number of clusters.



Now let's implement K-Means clustering using python.

Implementation

About Dataset – Dataset we are using here is the Mall Customers data (Download here). It's unlabeled data that contains the details of customers in a mall (features like genre, age, annual income(k$), and spending score). Our aim is to cluster the customers based on the relevant features annual income and spending score.

Index	ustomerI(	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72
10	11	Male	67	19	14

In-depth Intuition of K-Means Clustering Algorithm in Machine Learning

```
import matplotlib.pyplot as plt
import pandas as pd
import sklearn
```

Now let's import the dataset and slice the important features

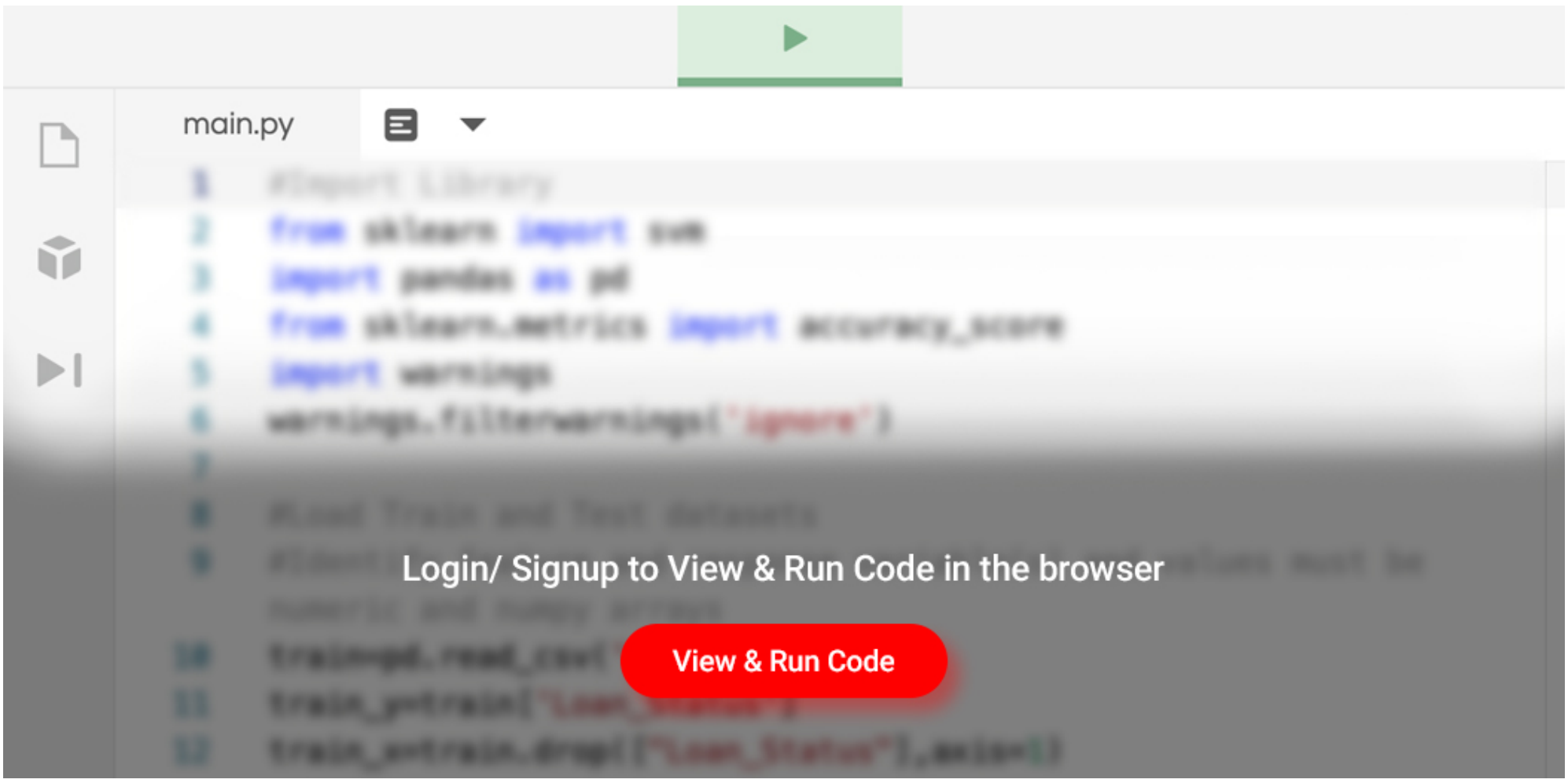
```
dataset = pd.read_csv('Mall_Customers.csv')
X = dataset.iloc[:, [3, 4]].values
```

We have to find the optimal K value for clustering the data. Now we are using the Elbow method to find the optimal K value.

```
from sklearn.cluster import KMeans
wcss = [] for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
```

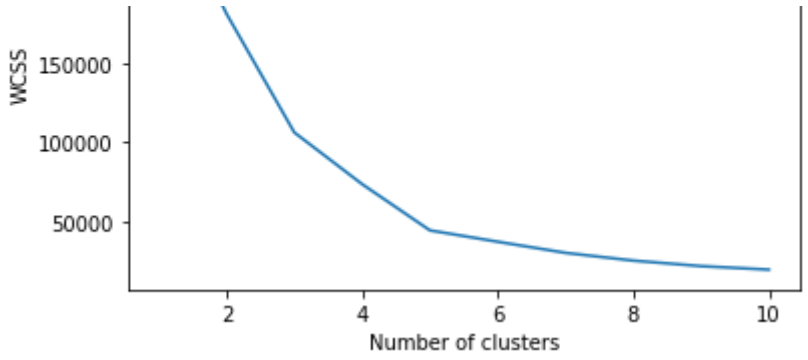
"init" argument is the method for initializing the centroid. We calculated the WCSS value for each K value. Now we have to plot the WCSS with K value

Python Code:



Graph will be-

In-depth Intuition of K-Means Clustering Algorithm in Machine Learning



The point at which the elbow shape is created is 5, that is, our K value or an optimal number of clusters is 5. Now let's train the model on the dataset with a number of clusters 5.

```
kmeans = KMeans(n_clusters = 5, init = "k-means++", random_state = 42)
y_kmeans = kmeans.fit_predict(X)
```

y_kmeans will be:

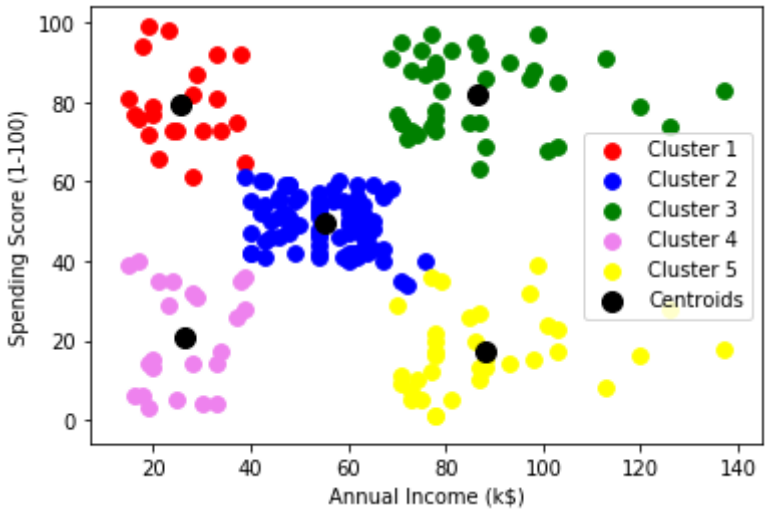
```
array([3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0,
       3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 1,
       3, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 2, 4, 2, 4, 2, 4, 2, 4, 2, 1, 2, 4, 2, 4, 2, 4, 2, 4, 2,
       4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2,
       4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2,
       4, 2])
```

y_kmeans give us different clusters corresponding to X. Now let's plot all the clusters using matplotlib.

```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 60, c = 'red', label = 'Cluster1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 60, c = 'blue', label = 'Cluster2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 60, c = 'green', label = 'Cluster3)
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 60, c = 'violet', label = 'Cluster4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 60, c = 'yellow', label = 'Cluster5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 100, c = 'black', label =
'Centroids')
plt.xlabel('Annual Income (k$)') plt.ylabel('Spending Score (1-100)') plt.legend()

plt.show()
```

Graph:



As you can see there are 5 clusters in total which are visualized in different colors and the centroid of each cluster is visualized in black color.

Full code

In-depth Intuition of K-Means Clustering Algorithm in Machine Learning

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd # Importing the dataset

X = dataset.iloc[:, [3, 4]].values
dataset = pd.read_csv('Mall_Customers.csv')

from sklearn.cluster import KMeans

# Using the elbow method to find the optimal number of clusters wcss = [] for i in range(1, 11):
 wcss.append(kmeans.inertia_)
 kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42) kmeans.fit(X) plt.plot(range(1, 11),
wcss) plt.xlabel('Number of clusters')

y_kmeans = kmeans.fit_predict(X)

plt.ylabel('WCSS') plt.show() # Training the K-Means model on the dataset kmeans = KMeans(n_clusters = 5, init
= 'k-means++', random_state = 42) y_kmeans = kmeans.fit_predict(X)

# Visualising the clusters
plt.scatter( X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 60, c = 'blue', label = 'Cluster2')
plt.scatter( X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 60, c = 'red', label = 'Cluster1') plt.scatter(
X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 60, c = 'green', label = 'Cluster3')
plt.scatter( kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 100, c = 'black', label =
'Centroids')
plt.scatter( X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 60, c = 'violet', label = 'Cluster4') plt.scatter(
X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 60, c = 'yellow', label = 'Cluster5') plt.xlabel('Annual Income
(k$)') plt.ylabel('Spending Score (1-100)') plt.legend()

plt.show()
```

Conclusion

This is all about the basic concept of the K-Means Clustering algorithm in Machine Learning. In the upcoming articles, we can learn more about different ML Algorithms.

The media shown in this article are not owned by Analytics Vidhya and is used at the Author's discretion.

blogathon K-means Clustering

In-depth Intuition of K-Means Clustering Algorithm in Machine Learning

About the Author

Basil Saji

Our Top Authors

view more

Download

Analytics Vidhya App for the Latest blog/Article

Previous Post

A Quick Guide to Setting up a Virtual Environment for Machine Learning and Deep Learning on macOS

Next Post

Implementation of Attention Mechanism for Caption Generation on Transformers using TensorFlow

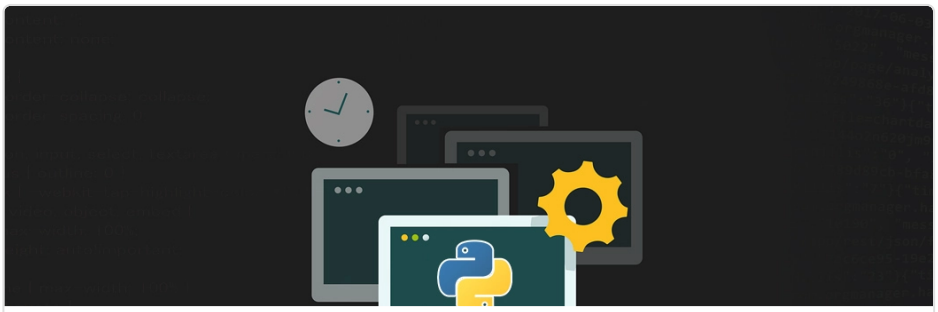
Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

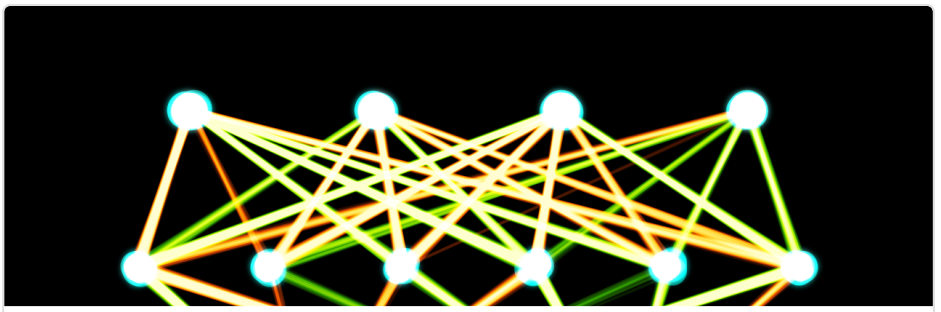
In-depth Intuition of K-Means Clustering Algorithm in Machine Learning

Top Resources



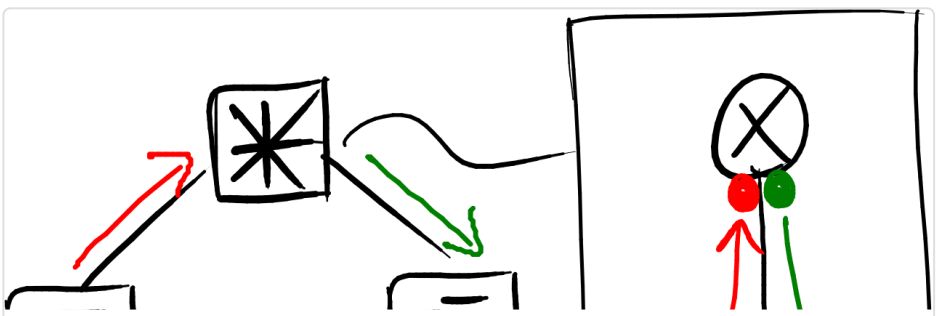
Python Tutorial: Working with CSV file for Data Science

Harika Bonthu - AUG 21, 2021



Boost Model Accuracy of Imbalanced COVID-19 Mortality Prediction Using GAN-based..

Bala Gangadhar Thilak Adiboina - OCT 07, 2020



Introductory guide on Linear Programming for (aspiring) data scientists

avcontentteam - FEB 28, 2017

Understanding Random Forest

Sruthi E R - JUN 17, 2021

In-depth Intuition of K-Means Clustering Algorithm in Machine Learning