

לוגיקת הפעולות:

`data_rdd.map(lambda row: [float(x) for x in row])` ממירה כל רכיב ב-`data_rdd` ל-`float`.

`data_rdd.map(assign_cluster)`: פעולה זו משמשת לסיווג כל נקודה למרכז הקרוב ביותר. היא מכילה פונקציה המסווגת כל נקודה למרכז המתאים לה.

`classification.filter(lambda obs: check(obs, j))`: פעולת הסיווג משמשת לסיווג הנקודות השייכות ל-`cluster-j`. היא מכילה את הפונקציה `check` הבודקת שייכות ל-`cluster`.

`cluster_j.reduce(sum_points)`: פעולה זו משמשת לחישוב סכום הנקודות בתוך `cluster-j`. היא מכילה את פונקציית `sum_points` על זוגות נקודות בתוך `cluster-j` ומחזירה נקודה בודדת המייצגת את סכום כל הנקודות באותו ה-`cluster-j`.

תיאור של מבני הנתונים:

Spark DataFrame: הנתונים נקראים בתחילה מקובץ CSV לתוך `Spark DataFrames`. `Spark DataFrame` הם אוספים מבוזרים של נתונים המאורגנים בעמודות עם שם. במקרה זה, ה-`DataFrame` משמש לקריאה ועיבוד מקדים של הנתונים לפני המרתם ל-`RDD` לצורך חישובים נוספים.

RDD: בפתרון זה, `RDDs` משמשים לביצוע פעולות הנדרשות על ידי אלגוריתם `k-means`. הם מאפשרים עיבוד מבוזר יעיל של הנתונים ומאפשרים פעולות כמו `map`, `filter` ו-`reduce`.

הסבר על יישום הפתרון שנבחר:

הפתרון משתמש ביכולות ה-`DataFrame` וה-`RDD` של `Spark` כדי לטפל ולעבד נתונים בקנה מידה גדול ביעילות בצורה מבוזרת. `Spark` מספק אופטימיזציות מובנות ויכולות עיבוד מקבילי, מה שהופך אותו למתאים לתרחישי ביג דאטה.

`DataFrames` משמשים בתחילה כדי לקרוא ולעבד מראש את הנתונים באמצעות `DataFrame API` של `Spark`, המציע מגוון רחב של טרנספורמציות ופעולות על נתונים מובנים.

לאחר הכנת הנתונים, הם מומרים ל-`RDD` כדי לבצע את האלגוריתם `k-means`.

הפתרון משתמש בפונקציות `lambda` כדי להגדיר פעולות ומספק דרך תמציתית וקריאה לבטא טרנספורמציות וחישובים.

בסך הכל, השילוב של `Spark DataFrames` ו-`RDDs` מאפשר עיבוד מבוזר של אלגוריתם `k-means`, מה שהופך אותו למתאים לטיפול במערכי נתונים גדולים בצורה מקבילה ויעילה.

Dataset name	The value of K	Average and std CH	Average and std ARI
Iris	2	353.37, 0.00	0.57, 0.00
Iris	3	302.55, 71.51	0.62, 0.14
Iris	4	260.41, 71.50	0.60, 0.11
Iris	5	241.13, 42.56	0.56, 0.10
Iris	6	212.57, 43.55	0.54, 0.07
glass	2	121.71, 47.55	0.16, 0.10
glass	3	101.10, 4.59	0.15, 0.04
glass	4	84.48, 11.19	0.16, 0.04
glass	5	71.45, 11.68	0.14, 0.03
glass	6	65.91, 9.94	0.15, 0.03
Parkinsons	2	84.22, 0.00	0.05, 0.00
Parkinsons	3	75.74, 1.38	0.07, 0.02
Parkinsons	4	60.87, 7.28	0.08, 0.04
Parkinsons	5	58.04, 6.30	0.07, 0.05
Parkinsons	6	55.54, 2.65	0.09, 0.04