מסמך אפיון -K-Means

1. פירוט לוגיקות

- rdd = spark.sparkContext.parallelize(MinMaxScaler().fit_transform(rdd.collect()))
פונקציה שמטרתה לנרמל את ה-rdd, בעזרת המודול MinMaxScaler של sklearn.

- def mapper(point, centroids: list)
פונקציה שאנחנו הגדרנו. הפונקציה מקבלת נקודה ומספר מרכזים, ומשייכת לנקודה את האינדקס של המרכז הקרוב ביותר. הפונקציה מחזירה (center_index, point) כערכי (key, value)לטובת פונקציית ה-reduce בהמשך.

- mapped_points = rdd.map(lambda x: mapper(x, centroids))
פעולת ה-map מחזירה מיפוי של נקודה (הערך) לאינדקס של המרכז אליו היא משתייכת (מפתח), ע"י שימוש בפונקציה mapper אותה הגדרנו.

- combined = mapped_points.mapValues(lambda x: [x])
- .reduceByKey(lambda x,y: x+y)
  - פעולת ה-mapValues מכניסה כל ערך (נקודה) לתוך רשימה רק של עצמה.
  - פעולת ה-reduceByKey משרשרת את כל הנקודות השייכות לאותו מפתח. כך אנחנו נשארים למעשה עם המיפוי של אינדקס קלאסטר, וכל הנקודות המשתייכות אליו.

- new_centroids = combined.mapValues(lambda x: np.mean(np.array(x), axis=0))
פעולת ה-mapValues ממצעת את כל הערכים המשתייכים לאותו קלאסטר. כך מתקבלים מרכזי הקלאסטרים החדשים.

2. <u>תיאור מבני נתונים</u>

אנחנו בחרנו להשתמש בשני סוגי אובייקטים מרכזיים – tuple, numpy array.
ה-tuple נועד לשמר את יחס הסדר ב-key וה-value.
ב-numpy array השתמשנו על מנת לבצע פעולות מתמטיות על ווקטורים (מערכים) בצורה נכונה ומהירה.
בנוסף השתמשנו ברשימות על מנת לשמור את תוצאות ההרצות השונות.

3. <u>תוצאות ההרצה</u>

Dataset name	The value of K	Average and std CH	Average and std ARI
iris	2	(353.36740323251195, 5.684341886080802e-14)	(0.5681159420289854, 1.1102230246251565e-16)
iris	3	(312.1038017499549, 70.97403944321789)	(0.6301248138958002, 0.13169909930361712)
iris	5	(269.07042812312284, 23.82207586275842)	(0.5389121222666937, 0.0647568131922123)
iris	7	(218.92616867013697, 16.929909486169343)	(0.4673120339822643, 0.08692269208869911)
iris	10	(210.76390285672443, 17.390417810601985)	(0.36470739330559765, 0.03408712333382743)
glass	2	(121.68024778360714, 45.100295998708106)	(0.15987500229633006, 0.09210876893396633)
glass	3	(98.09025954956306, 7.602075399354372)	(0.193907678915817, 0.04429019523320951)
glass	5	(83.66391494066158, 8.160298442088703)	(0.18304530169520888, 0.03782704876190398)
glass	7	(66.01445557194869, 11.38898258359348)	(0.15343057136429755, 0.03041039690888826)
glass	10	(54.26315021057818, 7.882818806398969)	(0.16986801568702, 0.029894389143650592)
parkinsons	2	(84.21490248528623, 0.002449632386740886)	(0.04933655916084388, 0.003206967915889237)
parkinsons	3	(76.61249843876948, 1.184844228530847)	(0.08013314885131571, 0.011975702889107619)
parkinsons	5	(63.368316397458486, 3.8847190609035063)	(0.09579923410619967, 0.051438566522630624)
parkinsons	7	(53.528801563112026, 3.37718441463822)	(0.06300099208888192, 0.027645007118145506)
parkinsons	10	(45.37961410094438, 1.7048506736637143)	(0.04175655431959015, 0.0166473188524759)