אביחי בן לולו

ממן 22 כריית מידע

הנחות והכנת הנתונים לפני תחילת העבודה:

השתמשתי בקובץ של ממ"ן 21 לאחר ניקוי

1. ***Apriori:***

סיבה: בחרתי בו מאחר ובדיקות ביצועים הראו שהאלגוריתם יעיל.
הכנת המידע: על מנת להריץ את אלגוריתם זה היינו צריכים לעשות דיסקרטיזציה על הנתונים ביצעתי את הדיסקרטיזציה כך שהמידע יתחלק ל3 בינים לא בהכרח שווים.
האלגוריתם: אלגוריתם אפריורי מחפש תדירות של ITEMSETS לצורך ביצוע אסוציאציות בצורה בוליאנית. הוא מבוסס (על פי שמו Prior Knowledge) על ידע מוקדם של תדירות המאפייני הסטים.

האלגוריתם פועל בצורה איטראטיבית בצורת חיפוש ע"פ רמות שבו משתמשים ב K-סטים כדי לחקור K+1 סטים. בהתחלה מסננים את הסטים שלא עובדים בדרישת המינימום תמיכה -נסמן את התוצאה בתור L1. משם משתמשים בL1 לחיפוש בL2 - הסטים הכי שכיחים. הם משומשים למצוא עבור L3 וכן הלאה עד שאין יותר K סטים שכיחים למצוא. כדי למצוא עבור כל אחד מL_K נדרש מעבר שלם על בסיס הנתונים.

כל מחזור של האלגוריתם מחולק ל-2 שלבים עיקרים:

1) Join step - מציאת מועמדים בגודל k על בסיס קבוצות הפריטים השכיחות שנמצאו בשלב k – 1. בשלב זה מתבצע צירוף של שתי קבוצות שנמצאו בשלב קודם לצורך מציאת קבוצה חדשה.
2) The Prune - בשלב הגיזום מתעלמים מצירופים שאינם שכיחים שאינם לא עומדים בתכונה האפריורית. בשלב זה גם מבוצעת בדיקה מהירה (subset testing) של הסטים שלא נגזמו - האם הסטים אכן שכיחים. השכיחות מתבצעת ע"י ספירת המתמודדים בטבלת גיבוב (hash).

ישנם גם דרכים נוספות לשפר את היעילות שלו כמו לדוגמא לקחת תת-סט של המידע ולבצע את הניתוח עליו, משפרים מאוד את המהירות אך מאבדים מהדיוק.

שיטה נוספת היא להשתמש בספירה דינמית של הסטים - מועמדים חדשים יכולים להתווסף בכל שלב בניגוד לאלגוריתם האפריורי המקורי שסופר כבר בהתחלה את המועמדים ודורש סריקה שלמה של בסיס הנתונים.

2. *FPGrowth*:

סיבה: בחרתי בו מאחר הוא מכיל את שלמות המידע ואת התדירות, כמו כן הוא מאוד קומפקטי (פריטים שאינם שכיחים נעלמים). הוא נותן תוצאה מסודרת ע"פ שכיחות כך שהאלמנטים שכנראה מעניינים יותר נמצאים בהתחלה.

הכנת המידע: על מנת להריץ את אלגוריתם זה על הנתונים הייתי צריך להמיר את כל הערכים בעזרת הפילטר-NominalToBinary ולאחר מכן בעזרת הפילטר NumerictoBinary כשאופציית ה-CLASS מכוונת ל-TRUE על מנת שהאלגוריתם יוכל לעבוד.
למעשה מה שקרה למידע זה שהוא נהפך לאוסף של קבוצות בינאריות כשכל אחת מכילה את כמות השורות שתכונה מופיעה כ-1 וכמות המקומות שתכונה לא מופיעה כ-0.
האלגוריתם: פועל בשיטת הפרד-ומשול הוא דוחס את בסיס הנתונים אל תוך עץ שמציג את השכיחויות (FP-TREE) השומר את המידע ל itemset האסוציאטיבי. הוא מפרק את בסיס הנתונים המכווץ לסט של בסיסי נתונים מותנים ואז מבצע את הליך כריית המידע על כל אחד מהתת בסיסי נתונים בנפרד מה שמקטין משמעותית את כמות המידע שבה יש לחפש.
שלבים:

1) שלב מציאת המועמדים בדומה לאלגוריתם האפריורי ומייצרים רשימה שמכילה את הסטים שעומדים במינימום ואת מספר השיכחויות שלהם .

2) בניית עץ ה-FP-TREE בהתחלה, בונים את שורש העץ שמסומן ב-null. סריקה נוספת של בסיס הנתונים שבה מסדרים את הסטים בסדר יורד ע"פ כמות השכיחויות ומייצרים ענף עבור כל טרנזקציה. כל ענף מכיל PREFIX שידאג שאם מכניסים את "אותו האיבר" במקום לייצר ענף חדש לאותו האיבר, פשוט יתווסף 1 לcounter של השכיחויות.1

.ב.

נקבע את הבטחון והעזרה כמו שהתבקשנו בשאלה ונריץ זאת על שני האלגוריתמים הנבחרים.

נקבל:

אפריורי:

```
Apriori
=======

Minimum support: 0.95 (4459 instances)
Minimum metric <confidence>: 0.6
Number of cycles performed: 1

Generated sets of large itemsets:

Size of set of large itemsets L(1): 8

Size of set of large itemsets L(2): 16

Size of set of large itemsets L(3): 9

Size of set of large itemsets L(4): 1

Best rules found:

 1. free sulfur dioxide='(75.333333-inf)'_binarized=0 4634 ==> fixed acidity='(10.733333-inf)'_binarized=0 4632    <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.66)
 2. total sulfur dioxide='(232.666667-inf)'_binarized=0 4623 ==> fixed acidity='(10.733333-inf)'_binarized=0 4621    <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.66)
 3. residual sugar='(17.566667-inf)'_binarized=0 4588 ==> fixed acidity='(10.733333-inf)'_binarized=0 4586    <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.65)
 4. free sulfur dioxide='(75.333333-inf)'_binarized=0 total sulfur dioxide='(232.666667-inf)'_binarized=0 4569 ==> fixed acidity='(10.733333-inf)'_binarized=0 4567
 5. chlorides='(0.064333-inf)'_binarized=0 4549 ==> fixed acidity='(10.733333-inf)'_binarized=0 4547    <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.65)
 6. residual sugar='(17.566667-inf)'_binarized=0 free sulfur dioxide='(75.333333-inf)'_binarized=0 4528 ==> fixed acidity='(10.733333-inf)'_binarized=0 4526    <conf
 7. residual sugar='(17.566667-inf)'_binarized=0 total sulfur dioxide='(232.666667-inf)'_binarized=0 4526 ==> fixed acidity='(10.733333-inf)'_binarized=0 4524    <co
 8. volatile acidity='(0.46-inf)'_binarized=0 4524 ==> fixed acidity='(10.733333-inf)'_binarized=0 4522    <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.64)
 9. quality='(7-inf)'_binarized=0 4518 ==> fixed acidity='(10.733333-inf)'_binarized=0 4516    <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.64)
10. sulphates='(0.72-inf)'_binarized=0 4512 ==> fixed acidity='(10.733333-inf)'_binarized=0 4510    <conf:(1)> lift:(1) lev:(-0) [0] conv:(0.64)
```

הקבוצות התדירות:

ניתן לראות כי fixed acidity 10.7-inf חוזר על עצמו מספר פעמים, מחיקה של הערכים האלה – שהיו מעטים גם ככה – לא שינתה את התוצאות.

FPGrowth:

לאחר הכנת הנתונים כמתואר בסעיף א'

- הגדרתי את האלגוריתם לרוץ כך שהביטחון יהיה 0.6 והתמיכה 0.4 כנדרש

בנוסף הגדלתי את סך החוקים המתקבלים ל-100 וקיבלתי 62 חוקים המדורגים לפי רמתם.

```
1. [volatile acidity='(-inf-0.27]'_binarized=1]: 2661 ==> [citric acid='(0.246667-0.493333]'_binarized=1]: 2250   <conf:(0.85)> lift:(1.1) lev:(0.04) conv:(1.49)
2. [quality='(5-7]'_binarized=1]: 2991 ==> [citric acid='(0.246667-0.493333]'_binarized=1]: 2480   <conf:(0.83)> lift:(1.08) lev:(0.04) conv:(1.35)
3. [free sulfur dioxide='(-inf-38.666667]'_binarized=1]: 2903 ==> [residual sugar='(-inf-9.083333]'_binarized=1]: 2345   <conf:(0.81)> lift:(1.12) lev:(0.06) conv:(1.46)
4. [residual sugar='(-inf-9.083333]'_binarized=1, fixed acidity='(-inf-7.266667]'_binarized=1]: 2495 ==> [pH='(3.086667-3.453333]'_binarized=1]: 1983   <conf:(0.79)> lift:(1.13) le
5. [residual sugar='(-inf-9.083333]'_binarized=1, pH='(3.086667-3.453333]'_binarized=1]: 2508 ==> [fixed acidity='(-inf-7.266667]'_binarized=1]: 1983   <conf:(0.79)> lift:(1.1) lev
6. [residual sugar='(-inf-9.083333]'_binarized=1]: 3374 ==> [citric acid='(0.246667-0.493333]'_binarized=1]: 2654   <conf:(0.79)> lift:(1.02) lev:(0.01) conv:(1.08)
7. [residual sugar='(-inf-9.083333]'_binarized=1, pH='(3.086667-3.453333]'_binarized=1]: 2508 ==> [citric acid='(0.246667-0.493333]'_binarized=1]: 1972   <conf:(0.79)> lift:(1.02)
8. [pH='(3.086667-3.453333]'_binarized=1]: 3307 ==> [citric acid='(0.246667-0.493333]'_binarized=1]: 2577   <conf:(0.78)> lift:(1.01) lev:(0.01) conv:(1.04)
9. [free sulfur dioxide='(-inf-38.666667]'_binarized=1]: 2903 ==> [citric acid='(0.246667-0.493333]'_binarized=1]: 2261   <conf:(0.78)> lift:(1.01) lev:(0.01) conv:(1.04)
10. [fixed acidity='(-inf-7.266667]'_binarized=1, pH='(3.086667-3.453333]'_binarized=1]: 2555 ==> [residual sugar='(-inf-9.083333]'_binarized=1]: 1983   <conf:(0.78)> lift:(1.08) le
11. [citric acid='(0.246667-0.493333]'_binarized=1, fixed acidity='(-inf-7.266667]'_binarized=1]: 2483 ==> [pH='(3.086667-3.453333]'_binarized=1]: 1919   <conf:(0.77)> lift:(1.1) le
12. [pH='(3.086667-3.453333]'_binarized=1]: 3307 ==> [fixed acidity='(-inf-7.266667]'_binarized=1]: 2555   <conf:(0.77)> lift:(1.08) lev:(0.04) conv:(1.25)
13. [density='(0.99239-0.99767]'_binarized=1]: 2445 ==> [chlorides='(0.036667-0.064333]'_binarized=1]: 1882   <conf:(0.77)> lift:(1.14) lev:(0.05) conv:(1.41)
14. [total sulfur dioxide='(121.333333-232.666667]'_binarized=1]: 2827 ==> [chlorides='(0.036667-0.064333]'_binarized=1]: 2165   <conf:(0.77)> lift:(1.13) lev:(0.05) conv:(1.38)
15. [citric acid='(0.246667-0.493333]'_binarized=1, pH='(3.086667-3.453333]'_binarized=1]: 2577 ==> [residual sugar='(-inf-9.083333]'_binarized=1]: 1972   <conf:(0.77)> lift:(1.06)
16. [fixed acidity='(-inf-7.266667]'_binarized=1]: 3361 ==> [pH='(3.086667-3.453333]'_binarized=1]: 2555   <conf:(0.76)> lift:(1.08) lev:(0.04) conv:(1.23)
17. [pH='(3.086667-3.453333]'_binarized=1]: 3307 ==> [residual sugar='(-inf-9.083333]'_binarized=1]: 2508   <conf:(0.76)> lift:(1.06) lev:(0.03) conv:(1.16)
18. [fixed acidity='(-inf-7.266667]'_binarized=1, pH='(3.086667-3.453333]'_binarized=1]: 2555 ==> [citric acid='(0.246667-0.493333]'_binarized=1]: 1919   <conf:(0.75)> lift:(0.98) l
19. [total sulfur dioxide='(121.333333-232.666667]'_binarized=1]: 2827 ==> [citric acid='(0.246667-0.493333]'_binarized=1]: 2116   <conf:(0.75)> lift:(0.97) lev:(-0.01) conv:(0.92)
20. [chlorides='(0.036667-0.064333]'_binarized=1]: 3171 ==> [citric acid='(0.246667-0.493333]'_binarized=1]: 2367   <conf:(0.75)> lift:(0.97) lev:(-0.02) conv:(0.91)
21. [citric acid='(0.246667-0.493333]'_binarized=1, pH='(3.086667-3.453333]'_binarized=1]: 2577 ==> [fixed acidity='(-inf-7.266667]'_binarized=1]: 1919   <conf:(0.74)> lift:(1.04) l
22. [residual sugar='(-inf-9.083333]'_binarized=1]: 3374 ==> [pH='(3.086667-3.453333]'_binarized=1]: 2508   <conf:(0.74)> lift:(1.06) lev:(0.03) conv:(1.15)
23. [citric acid='(0.246667-0.493333]'_binarized=1, residual sugar='(-inf-9.083333]'_binarized=1]: 2654 ==> [pH='(3.086667-3.453333]'_binarized=1]: 1972   <conf:(0.74)> lift:(1.05)
24. [fixed acidity='(-inf-7.266667]'_binarized=1]: 3361 ==> [residual sugar='(-inf-9.083333]'_binarized=1]: 2495   <conf:(0.74)> lift:(1.03) lev:(0.02) conv:(1.09)
25. [quality='(5-7]' binarized=1]: 2991 ==> [residual sugar='(-inf-9.083333]' binarized=1]: 2218   <conf:(0.74)> lift:(1.03) lev:(0.01) conv:(1.09)
```

```
25. [quality='(5-7]'_binarized=1]: 2991 ==> [residual sugar='(-inf-9.083333]'_binarized=1]: 2218   <conf:(0.74)> lift:(1.03) lev:(0.01) conv:(1.09)
26. [residual sugar='(-inf-9.083333]'_binarized=1]: 3374 ==> [fixed acidity='(-inf-7.266667]'_binarized=1]: 2495   <conf:(0.74)> lift:(1.03) lev:(0.02) conv:(1.09)
27. [fixed acidity='(-inf-7.266667]'_binarized=1]: 3361 ==> [citric acid='(0.246667-0.493333]'_binarized=1]: 2483   <conf:(0.74)> lift:(0.96) lev:(-0.02) conv:(0.88)
28. [citric acid='(0.246667-0.493333]'_binarized=1]: 3611 ==> [residual sugar='(-inf-9.083333]'_binarized=1]: 2654   <conf:(0.73)> lift:(1.02) lev:(0.01) conv:(1.06)
29. [quality='(5-7]'_binarized=1]: 2991 ==> [fixed acidity='(-inf-7.266667]'_binarized=1]: 2193   <conf:(0.73)> lift:(1.02) lev:(0.01) conv:(1.06)
30. [total sulfur dioxide='(121.333333-232.666667]'_binarized=1]: 2827 ==> [pH='(3.086667-3.453333]'_binarized=1]: 2039   <conf:(0.72)> lift:(1.02) lev:(0.01) conv:(1.06)
31. [chlorides='(0.036667-0.064333]'_binarized=1]: 3171 ==> [pH='(3.086667-3.453333]'_binarized=1]: 2282   <conf:(0.72)> lift:(1.02) lev:(0.01) conv:(1.05)
32. [volatile acidity='(-inf-0.27]'_binarized=1]: 2661 ==> [residual sugar='(-inf-9.083333]'_binarized=1]: 1905   <conf:(0.72)> lift:(1) lev:(-0) conv:(0.99)
33. [citric acid='(0.246667-0.493333]'_binarized=1]: 3611 ==> [pH='(3.086667-3.453333]'_binarized=1]: 2577   <conf:(0.71)> lift:(1.01) lev:(0.01) conv:(1.03)
34. [volatile acidity='(-inf-0.27]'_binarized=1]: 2661 ==> [quality='(5-7]'_binarized=1]: 1895   <conf:(0.71)> lift:(1.12) lev:(0.04) conv:(1.26)
35. [volatile acidity='(-inf-0.27]'_binarized=1]: 2661 ==> [chlorides='(0.036667-0.064333]'_binarized=1]: 1882   <conf:(0.71)> lift:(1.05) lev:(0.02) conv:(1.11)
36. [free sulfur dioxide='(-inf-38.666667]'_binarized=1]: 2903 ==> [fixed acidity='(-inf-7.266667]'_binarized=1]: 2052   <conf:(0.71)> lift:(0.99) lev:(-0.01) conv:(0.97)
37. [volatile acidity='(-inf-0.27]'_binarized=1]: 2661 ==> [fixed acidity='(-inf-7.266667]'_binarized=1]: 1879   <conf:(0.71)> lift:(0.99) lev:(-0.01) conv:(0.97)
38. [quality='(5-7]'_binarized=1]: 2991 ==> [pH='(3.086667-3.453333]'_binarized=1]: 2103   <conf:(0.7)> lift:(1) lev:(-0) conv:(0.99)
39. [chlorides='(0.036667-0.064333]'_binarized=1]: 3171 ==> [fixed acidity='(-inf-7.266667]'_binarized=1]: 2226   <conf:(0.7)> lift:(0.98) lev:(-0.01) conv:(0.95)
40. [total sulfur dioxide='(121.333333-232.666667]'_binarized=1]: 2827 ==> [fixed acidity='(-inf-7.266667]'_binarized=1]: 1969   <conf:(0.7)> lift:(0.97) lev:(-0.01) conv:(0.93)
41. [free sulfur dioxide='(-inf-38.666667]'_binarized=1]: 2903 ==> [pH='(3.086667-3.453333]'_binarized=1]: 2018   <conf:(0.7)> lift:(0.99) lev:(-0.01) conv:(0.97)
42. [residual sugar='(-inf-9.083333]'_binarized=1]: 3374 ==> [free sulfur dioxide='(-inf-38.666667]'_binarized=1]: 2345   <conf:(0.7)> lift:(1.12) lev:(0.06) conv:(1.25)
43. [pH='(3.086667-3.453333]'_binarized=1]: 3307 ==> [chlorides='(0.036667-0.064333]'_binarized=1]: 2282   <conf:(0.69)> lift:(1.02) lev:(0.01) conv:(1.05)
44. [citric acid='(0.246667-0.493333]'_binarized=1]: 3611 ==> [fixed acidity='(-inf-7.266667]'_binarized=1]: 2483   <conf:(0.69)> lift:(0.96) lev:(-0.02) conv:(0.91)
45. [citric acid='(0.246667-0.493333]'_binarized=1]: 3611 ==> [quality='(5-7]'_binarized=1]: 2480   <conf:(0.69)> lift:(1.08) lev:(0.04) conv:(1.16)
46. [chlorides='(0.036667-0.064333]'_binarized=1]: 3171 ==> [total sulfur dioxide='(121.333333-232.666667]'_binarized=1]: 2165   <conf:(0.68)> lift:(1.13) lev:(0.05) conv:(1.25)
47. [fixed acidity='(-inf-7.266667]'_binarized=1]: 3361 ==> [chlorides='(0.036667-0.064333]'_binarized=1]: 2226   <conf:(0.66)> lift:(0.98) lev:(-0.01) conv:(0.96)
48. [chlorides='(0.036667-0.064333]'_binarized=1]: 3171 ==> [residual sugar='(-inf-9.083333]'_binarized=1]: 2100   <conf:(0.66)> lift:(0.92) lev:(-0.04) conv:(0.83)
49. [residual sugar='(-inf-9.083333]'_binarized=1]: 3374 ==> [quality='(5-7]'_binarized=1]: 2218   <conf:(0.66)> lift:(1.03) lev:(0.01) conv:(1.06)
50. [citric acid='(0.246667-0.493333]'_binarized=1]: 3611 ==> [chlorides='(0.036667-0.064333]'_binarized=1]: 2367   <conf:(0.66)> lift:(0.97) lev:(-0.02) conv:(0.94)
51. [fixed acidity='(-inf-7.266667]'_binarized=1]: 3361 ==> [quality='(5-7]'_binarized=1]: 2193   <conf:(0.65)> lift:(1.02) lev:(0.01) conv:(1.04)
52. [free sulfur dioxide='(-inf-38.666667]'_binarized=1]: 2903 ==> [quality='(5-7]'_binarized=1]: 1892   <conf:(0.65)> lift:(1.02) lev:(0.01) conv:(1.04)
53. [pH='(3.086667-3.453333]'_binarized=1]: 3307 ==> [quality='(5-7]'_binarized=1]: 2103   <conf:(0.64)> lift:(1) lev:(-0) conv:(1)
54. [quality='(5-7]'_binarized=1]: 2991 ==> [volatile acidity='(-inf-0.27]'_binarized=1]: 1895   <conf:(0.63)> lift:(1.12) lev:(0.04) conv:(1.18)
55. [quality='(5-7]'_binarized=1]: 2991 ==> [chlorides='(0.036667-0.064333]'_binarized=1]: 1892   <conf:(0.63)> lift:(0.94) lev:(-0.03) conv:(0.88)
56. [quality='(5-7]'_binarized=1]: 2991 ==> [free sulfur dioxide='(-inf-38.666667]'_binarized=1]: 1892   <conf:(0.63)> lift:(1.02) lev:(0.01) conv:(1.04)
57. [citric acid='(0.246667-0.493333]'_binarized=1]: 3611 ==> [free sulfur dioxide='(-inf-38.666667]'_binarized=1]: 2261   <conf:(0.63)> lift:(1.01) lev:(0.01) conv:(1.02)
58. [citric acid='(0.246667-0.493333]'_binarized=1]: 3611 ==> [volatile acidity='(-inf-0.27]'_binarized=1]: 2250   <conf:(0.62)> lift:(1.1) lev:(0.04) conv:(1.15)
59. [residual sugar='(-inf-9.083333]'_binarized=1]: 3374 ==> [chlorides='(0.036667-0.064333]'_binarized=1]: 2100   <conf:(0.62)> lift:(0.92) lev:(-0.04) conv:(0.86)
60. [pH='(3.086667-3.453333]'_binarized=1]: 3307 ==> [total sulfur dioxide='(121.333333-232.666667]'_binarized=1]: 2039   <conf:(0.62)> lift:(1.02) lev:(0.01) conv:(1.04)
61. [fixed acidity='(-inf-7.266667]'_binarized=1]: 3361 ==> [free sulfur dioxide='(-inf-38.666667]'_binarized=1]: 2052   <conf:(0.61)> lift:(0.99) lev:(-0.01) conv:(0.98)
62. [pH='(3.086667-3.453333]'_binarized=1]: 3307 ==> [free sulfur dioxide='(-inf-38.666667]'_binarized=1]: 2018   <conf:(0.61)> lift:(0.99) lev:(-0.01) conv:(0.98)
```

הקבוצות התדירות:

ב- FPGrowth:

ניתן לראות שהמשתנה residental sugare -inf-9.0 חוזר יחסית כמות גדול לשאר המשתנים.

ג. חוקי ההקשר:

אפשר לראות שרמת הבטחון מאפשרת62 חוקים אך התוכנה מאפשרת לסנן את הטובים ביותר, במעבר על החוקים לא מצאתי חוק שחוזר על עצמו מספר רב של פעמים כך שיש חשש שהוא משפיע לרעה על הנתונים.

לא מצאתי תכונה החוזרת על עצמה מספר רב של פעמים – מספיק כדי שהיא תהיה דומיננטית – בשביל להוציא אותה, ולכן לדעתי החוקים שקיבלתי הם נכונים וללא תכונות דומיננטיות מובהקות.

אם הייתה תכונה כזאת הייתי מוציא אותה ובודק את הדיוק ואת התוצאות שהיתי מקבל ובהתאם לזה בוחן האם המעשה היה נכון או לא.

בנוסף סיננתי את החוקים בהם קיבלתי רמת ביטחון 1

ד.בוצע בסעיפים ב' וג'.

ה. בעוד שהאלגוריתם האפריורי נתן לנו תוצאות רצויות לקח לו הרבה יותר זמן לרוץ מאשר לאלגוריתם הFPGROWTH. הסיבה העיקרית היא שהאלגוריתם האפריורי מבצע הרבה סריקות כל פעם מחדש על בסיס הנתונים בעוד ש-FPGROWTH עובד בתור עץ בינארי שמפצל את בסיס הנתונים למספר תת בסיסי נתונים ומגובה בטבלת גיבוב.
חשוב לציין שבFPGROWTH הCONF הנמוך ביותר היה 0.61 לעומת האפריורי שבו היה הCONF הנמוך ביותר 0.95.
בFPGROWTH קיבלנו משמעותית יותר תוצאות שאינן CONF 1, כלומר יותר חוקים שמעניינים אותנו.

בFPGROWTH קיבלנו למעשה יותר מידע בתוצאה מהאפריורי שנוכל לעבד בנוסף ולכן נעדיף אותו גם בקטגוריה זאת:
<u>בצד ימין של כל חוק באפריורי קיבלנו 2 נתונים(ללא LIFT):</u>
שכיחות,conf

<u>בצד ימין של כל חוק ב- FPGROWTH סיפק לנו 5 נתונים:</u>
שכיחות,conf,lift,lev,conv

בדוחות של האפריורי ניתן היה גם לקבל רשימה של סטים גדולים בכל אחת מהדרגות, מה המינימום תמיכה שהתקבל ומספר המחזורים, מה שיכול לסייע לנו להבין את התהליך יותר טוב ואולי אף לשפר אותו באמצעות שינוי הפרמטרים. בFPGROWTH קיבלנו רק את התוצאה הסופית.

<u>מסקנה:</u> בסיכומו של עניין שני האלגוריתמים עובדים בצורות שונות ובעוד שהאפריורי מספק יותר דוחות הביצועים של FPGROWTH מהירים פי כמה וכמה. אסייג את דברי ואומר שהביצועים מאוד תלויים בסוג וכמות המידע.
לצורך העניין הייתי בוחר בFPGROWTH אם הייתי צריך לעבוד באופן תכוף יותר עם המידע לעומת האפריורי שבו הייתי נותן לו לרוץ במשך זמן רב יותר.

.2

ניתוח אשכולות

אשכול – קבוצת רשומות – והמטרה כאשר מבצעים הליך זה היא ליצור חלוקה כך שהאיברים באשכולות דומים אבל האשכולות עצמם שונים מה שמאפשר לנו להסיק מסקנות המאפשרות ניתוח והבנה של מגמות מסויימות אשר לא ניתן היה להבין והסיק אותן ללא צורת חלוקה זו.

כמו כן ישנן 2 סוגי חלוקות לאשכול:

חלוקה קשה- בה כל איבר יכול להשתייך לאשכול אחד בלבד.

חלוקה רכה - בה כל איבר יכול להשתייך למספר אשכולות.

ב. בבואנו להעריך מדדי איכות של אשכולות נמדוד אותם לפי המדדים הנ"ל:

קריטריון החלוקה: מועיל במיוחד לתהליכי אשכול שטוחים (שאין להם היררכיה) ועוזר לנתח קבוצות שונות לדוגמא כשמחלקים את האיברים לקבוצות ומקנים לכל קבוצה מנהל.

הפרדה לאשכולות: כשרוצים לבצע הפרדות לקבוצות שונות רוצים לוודא שכל איבר שייך לקבוצה אחת ובמידה ויש חלוקה רכה שכל איבר שייך לפחות לקבוצה אחת. אם קיבלנו אשכולות עם איברים שלא עומדים בתנאים אז למעשה האיכות של האשכול היא פחות טובה.

מדד הדימיון: חלק מהחישובים לדימיון בין איברים נעשים ע"י חישוב מרחק בין איברים חישוב של מדד זה. ישנה גם פונקציית האיכות שבודקת עד כמה אשכול מסויים הוא טוב או עד כמה חלוקה מסויימת טובה - מאוד תלוי בכל מקרה.

שטח\חלל האשכול: אשכולים מפוזרים על גבי שטחים דו, תלת מימדים ואף יותר. אם קיבלנו אשכולות שמכסים את כל השטח כנראה שהאשכול הוא חסר משמעות.

מציאת תבניות: אם האלגוריתם של האשכול מצליח לזהות תבניות במידע נקבל תוצאות טובות יותר (דבר נפוץ בכלי גרפיקה של השלמת חלקים חסרים בתמונה).

ג. **קיימות שתי גישות עיקריות לאשכול:**

אשכול חלוקה:

נעשה שימוש במדידות מרחק. תחילה בוחרים מספר נקודות, אקראיות או קבועות מראש, ובהמשך מצרפים אליהן את האובייקטים הקרובים ביותר. עבור כל איטרציה מחושב מרכז כל אשכול מחדש, בהתחשב באובייקטים החדשים שצורפו אליו, ולפיו נמדד המרחק עבור אובייקטים חדשים המועמדים להצטרף לאשכול.

האלגוריתם העיקרי שעושה שימוש בשיטת החלוקה הוא K-Means. היתרון באלגוריתם זה הוא חוסר ההתערבות הסובייקטיבי ביצירת אשכולות.
האלגוריתם בוחר בעצמו את נקודות המרכז הראשוניות, בצורה אקראית. בצורה זו גם מובטח כי יתקבלו אשכולות, אפילו אם הנתונים ההתחלתיים אינם מפולגים בצורה ברורה. חסרונות האלגוריתם בכך שהוא ניתן לעבודה עם משתנים מספריים בלבד, ובכך שהתוצאה משתנה בין הרצה להרצה בהכרח, עקב נקודות ההתחלה השונות המתקבלות בתחילת כל הרצה. ניתן לראות גם את הצורך בבחירת מספר האשכולות על-ידי המשתמש כחסרון.

רעיון האלגוריתם:

- בחר מספר נקודות מתוך סט הנתונים, כמספר האשכולות שהוגדרו.
- חזור:
  - עבור כל אובייקט – הקצה אותו לאשכול בעל נקודת המרכז הכי קרובה אליו.
  - חשב מחדש את נקודת המרכז של כל אשכול.
- הפסק כאשר אין שינוי.

<u>אשכול היררכי:</u>

נוצרים אשכולות מקוננים בצורה של עץ היררכי – הרמה העליונה היא סט הנתונים כולו, המהווה אשכול אחד גדול, והרמה התחתונה היא החלוקה הגדולה ביותר של האובייקטים בסט הנתונים, כאשר בכל אשכול מספר אובייקטים קטן יחסית.

בחרתי באלגוריתם (EM (expectation maximization. הבחירה באלגוריתם זה נעשתה, בחלקה, עקב אילוצי ריצה וזמן. אלגוריתמים אחרים דוגמאת HierarchicalClusterer, Cobweb, DBSCAN, עפו בזמן הריצה, או נתקעו, או רצו למשך שעות ארוכות ללא תוצאה. גם EM רץ זמן רב, לכן החלטתי להריץ אותו ללא תכונות רבות, ולהשאיר רק את אלה, שכבר זוהו כבעלות משקל על-ידי K-Means. בנוסף, יש לציין כי התצוגה של האלגוריתם התקבלה בצורה גרפית בלבד ולא עצית.

EM הוא אלגוריתם היררכי-הסתברותי, האלגוריתם מקצה חלוקה הסתברותית עבור כל משתנה, לגבי סיכוייו להיכלל בכל אחד מן האשכולות. כך הוא מחליט אילו אובייקטים דומים מספיק כדי להיכלל באותו אשכול, ומהי ההיררכיה שבין האשכולות. מספר האשכולות יכול להיקבע על-ידי האלגוריתם, או על ידי המשתמש, מראש.

האלגוריתם משתמש במשתנה LogLikelihood, שמחשב את ההסתברות הממוצעת עבור כל סט הנתונים, בכל שלב בו נבדקים אובייקטים נוספים מהסט. עלייה בערך המשתנה גורמת ליצירת אשכול חדש.

היתרון של אלגוריתם היררכי הסתברותי על-פני אלגוריתמים היררכיים אחרים, הוא בעיקר בכך שאלגוריתמים היררכיים מחפשים אפשרויות אשכול בצורה מקומית-נקודתית, עבור כל שלב בו הם נמצאים, ואילו היררכי-הסתברותי בוחן עצמו תמיד ביחס גלובלי (ההסתברות הכללית).

ד+ה.

<u>K Means :</u>

קובץ הנתונים מכיל רשומות בעלות 11 מאפיינים (כולל quailty שאותו נרצה לנבא).

מאחר וישנם כל כך הרבה מאפיינים החלטתי בשלב הראשון להתחיל עם אשכול אחד ולקבל מושג כללי לגבי איך מפוזר המידע שלנו - אולי משם אוכל להסיק לכמה אשכולות לחלק.

באמת ראיתי קודם שכל האיברים שייכים לאותו הקלאסטר – כלומר 100% בקלאסטר 1.

כעט ניסיתי לשחק עם ה-SEED בכדי להגיע לפיזור אחיד יותר אבל הפיזור היה עדיין בעייתי בעיני לתוצאה הטובה ביותר הגעתי כך שהSEED = 10 ומס' הקלאסטרים הוא 3.

Clusterer output

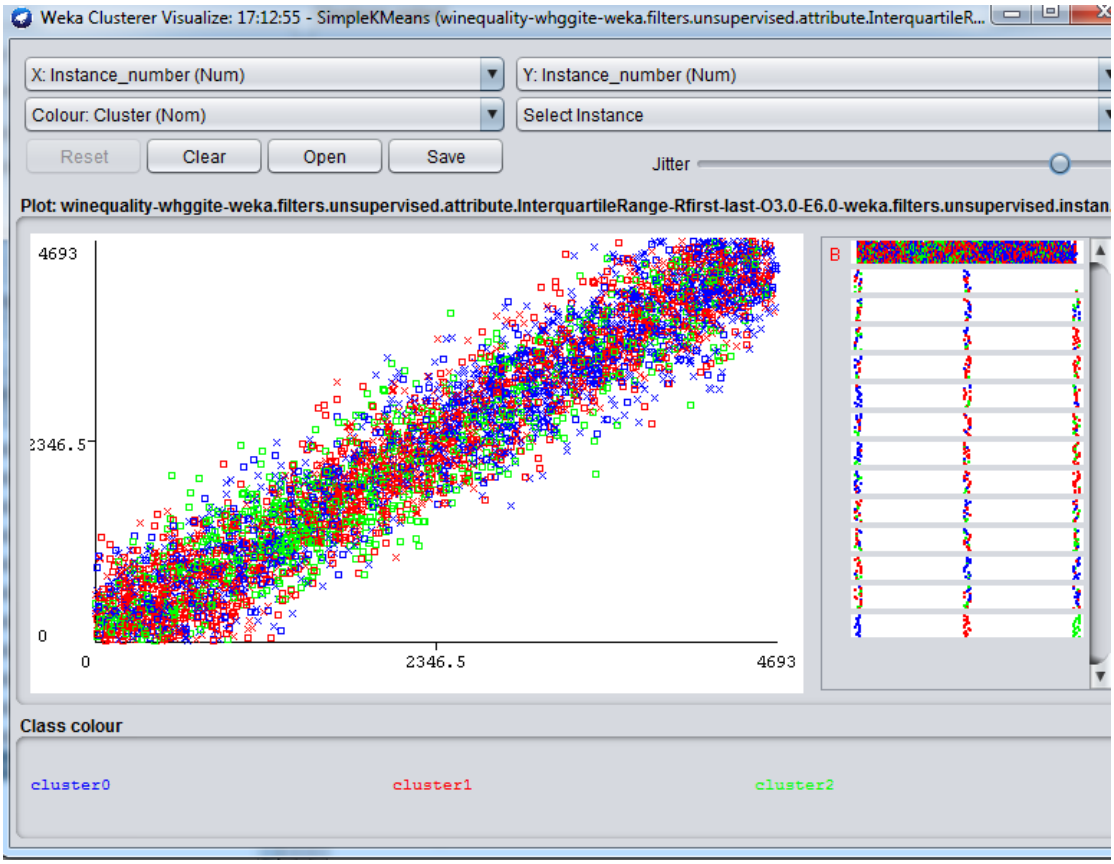
```
Final cluster centroids:
                                                                    Cluster#
Attribute                                   Full Data                      0                      1                      2
                                             (4694.0)                (1862.0)               (1853.0)                (979.0)
==================================================================================================================================
fixed acidity                        '(-inf-7.266667]'        '(-inf-7.266667]'        '(-inf-7.266667]'        '(-inf-7.266667]'
volatile acidity                         '(-inf-0.27]'            '(0.27-0.46]'            '(-inf-0.27]'            '(-inf-0.27]'
citric acid                     '(0.246667-0.493333]'    '(0.246667-0.493333]'    '(0.246667-0.493333]'    '(0.246667-0.493333]'
residual sugar                       '(-inf-9.083333]'        '(-inf-9.083333]'   '(9.083333-17.566667]'        '(-inf-9.083333]'
chlorides                       '(0.036667-0.064333]'        '(-inf-0.036667]'    '(0.036667-0.064333]'    '(0.036667-0.064333]'
free sulfur dioxide                 '(-inf-38.666667]'       '(-inf-38.666667]'  '(38.666667-75.333333]'       '(-inf-38.666667]'
total sulfur dioxide        '(121.333333-232.666667]'      '(-inf-121.333333]' '(121.333333-232.666667]' '(121.333333-232.666667]'
density                           '(0.99239-0.99767]'         '(-inf-0.99239]'      '(0.99239-0.99767]'      '(0.99239-0.99767]'
pH                              '(3.086667-3.453333]'    '(3.086667-3.453333]'    '(3.086667-3.453333]'    '(3.086667-3.453333]'
sulphates                            '(-inf-0.47]'            '(-inf-0.47]'            '(0.47-0.72]'            '(-inf-0.47]'
alcohol                       '(10.066667-12.133333]'  '(10.066667-12.133333]'       '(-inf-10.066667]'  '(10.066667-12.133333]'



Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      1862 ( 40%)
1      1853 ( 39%)
2       979 ( 21%)


Class attribute: quality
Classes to Clusters:

    0    1    2  <-- assigned to cluster
  393  799  335 | '(-inf-5]'
 1354 1018  619 | '(5-7]'
  115   36   25 | '(7-inf)'

Cluster 0 <-- '(5-7]'
Cluster 1 <-- '(-inf-5]'
Cluster 2 <-- '(7-inf)'

Incorrectly clustered instances :       2516.0   53.6003 %
```

בנוסף ניסיתי גם לחלק את הנתונים מראש ל-2 בינים ולראות את ההשפעה, התוצאה יוצאת יותר מדויקת ואולם אין פיזור טוב ולכן העדפתי להישאר עם 3 bin ולחלק כמו שביצעתי בסוף.

```
Final cluster centroids:
                                                Cluster#
Attribute                     Full Data                0                1                2
                               (4694.0)         (1442.0)         (1081.0)         (2171.0)
==========================================================================================
fixed acidity                '(-inf-9]'       '(-inf-9]'       '(-inf-9]'       '(-inf-9]'
volatile acidity         '(-inf-0.365]'   '(-inf-0.365]'   '(-inf-0.365]'   '(-inf-0.365]'
citric acid               '(-inf-0.37]'    '(-inf-0.37]'    '(-inf-0.37]'    '(-inf-0.37]'
residual sugar          '(-inf-13.325]'  '(-inf-13.325]'  '(-inf-13.325]'  '(-inf-13.325]'
chlorides               '(-inf-0.0505]'  '(-inf-0.0505]'   '(0.0505-inf)'  '(-inf-0.0505]'
free sulfur dioxide         '(-inf-57]'      '(-inf-57]'      '(-inf-57]'      '(-inf-57]'
total sulfur dioxide       '(-inf-177]'     '(-inf-177]'      '(177-inf)'     '(-inf-177]'
density                '(-inf-0.99503]' '(-inf-0.99503]'  '(0.99503-inf)' '(-inf-0.99503]'
pH                        '(-inf-3.27]'    '(-inf-3.27]'    '(-inf-3.27]'    '(-inf-3.27]'
sulphates                '(-inf-0.595]'   '(-inf-0.595]'   '(-inf-0.595]'   '(-inf-0.595]'
alcohol                   '(-inf-11.1]'     '(11.1-inf)'    '(-inf-11.1]'    '(-inf-11.1]'



Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       1442 ( 31%)
1       1081 ( 23%)
2       2171 ( 46%)


Class attribute: quality
Classes to Clusters:

    0    1    2  <-- assigned to cluster
  807  985 1852 | '(-inf-6]'
  635   96  319 | '(6-inf)'

Cluster 0 <-- '(6-inf)'
Cluster 1 <-- No class
Cluster 2 <-- '(-inf-6]'

Incorrectly clustered instances :       2207.0   47.0175 %
```

כמו שניתן לראות החלוקה לא מאוזנת וישנו קלאסטר שלם המוגדר כ--no class

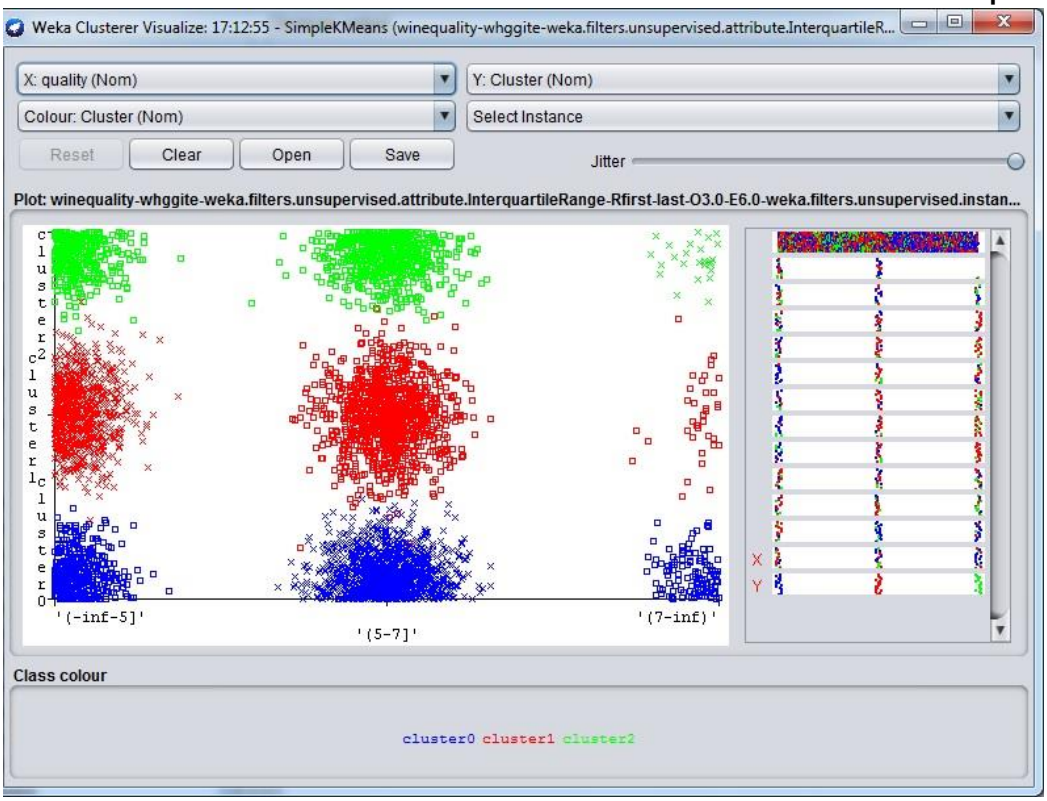
נבחן את הפיזור הכללי של התכונות לפי אשכולות:

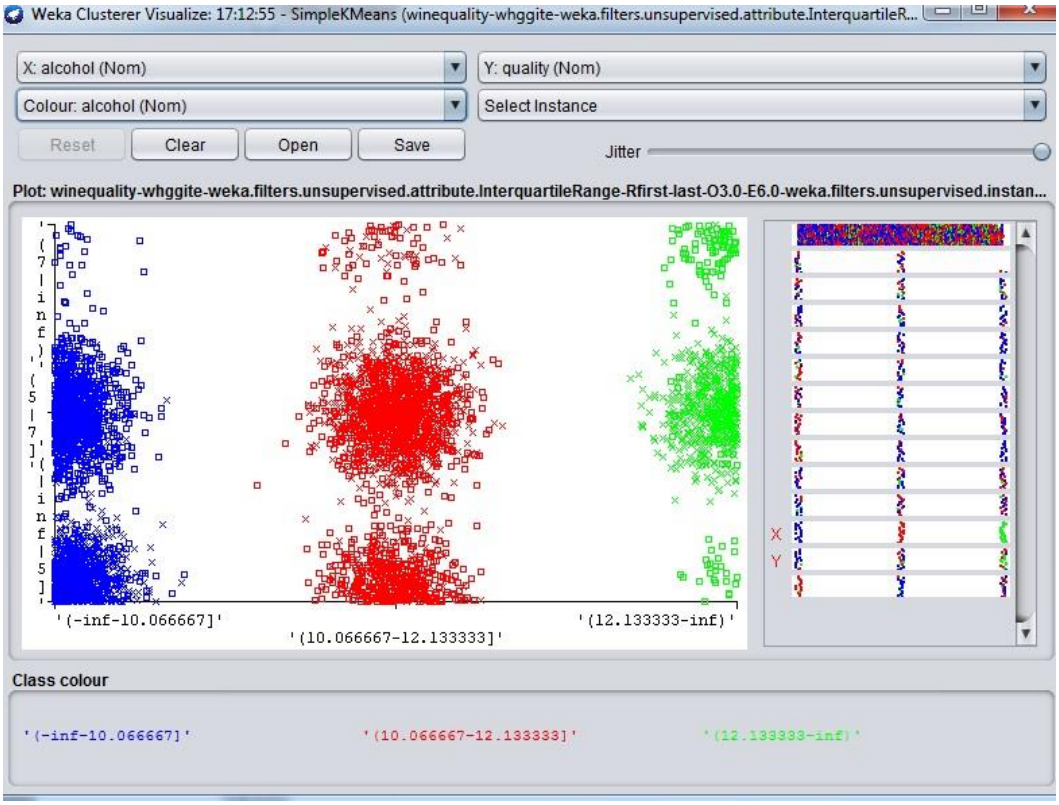


ניתן לראות כי הפיזור אחיד למדי על אף העובדה שיש עדיפות לאשכול הירוק בצד שמאל למטה אבל זה לא משהו מהותי בעיני ולכן החלטתי להישאר עם הפיזור וההגדרות הללו.

כעת התחלתי לבחון את הפיזור של האשכולות לפי התכונות, זה היה נראה לי מאוזן למדי
אתן כאן כמה דוגמאות מעניינות.

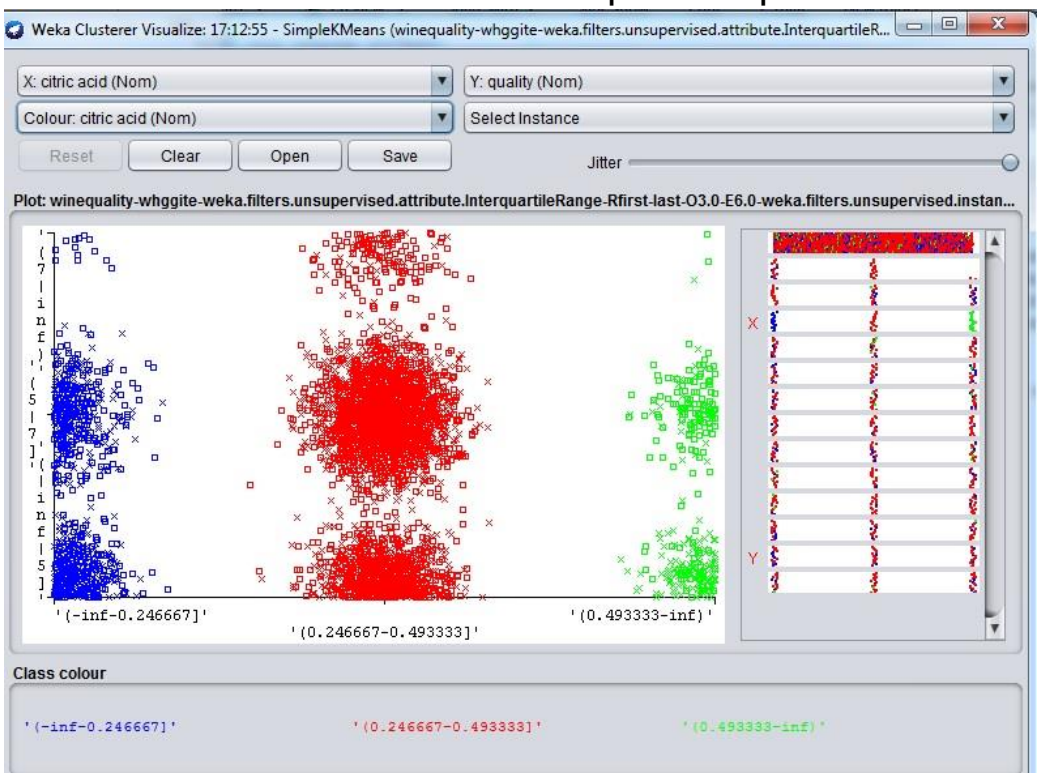
ניתן לראות בדוגמה הנ"ל





שיחסית הפיזור הוא די טוב ואמין לפי קלאסטרים. המשכתי להצליב בין הנתונים ולנסות להסיק מסקנות הקשורות לחיזוי האיכות.

ניתן להבחין כי ברמות האלכוהול הנמוכות איכות היין בינונית עד נמוכה, וברמות האלכוהול הגבוהות איכות היין בינונית עד גבוהה. כלומר קיים קשר בין רמת האלכוהול לבין איכות היין.



ניתן להבחין שכאשר כמות החומצה האצטרית נמצאת בטווח הבינוני, איכות היין עולה. כאשר כמות החומצה נמוכה מן הטווח הזה, או גבוהה ממנו- איכות היין נפגמת.

מסקנה:

לפי שילוב המסקנות ניתן להגיע למסקנה כך שנעדיף חומצה אצטרית בתחום הבינוני ורמת אלכוהול גבוהה כיוון שיש קשר בין תכונות אלו לבין איכות היין.

EM:

ניסיתי להריץ כמה וכמה פעמים את האלגוריתם כדי להגיע למצב אופטימלי כמו עם MEANS-K אבל לוקח לו זמן רב לכל פעם שמריצים אותו – ולכן ניסיתי 6 הרצות והגעתי למסקנה שההרצה
הכי אופטימלית שנותנת חלוקה מאוזנת יחסית היא ההגדרות ברירת המחדל בשינוי ל8 קלאסטרים במקסימום.

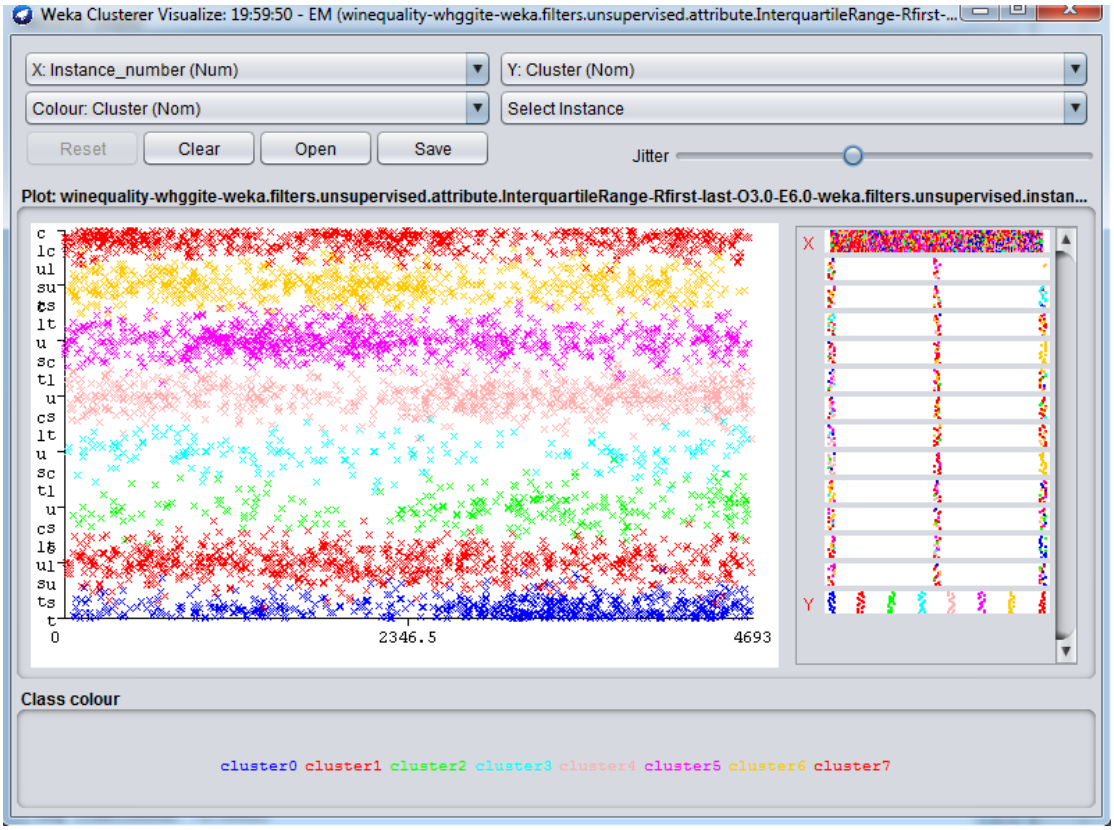
=== Model and evaluation on training set ===

Clustered Instances

0 578 (12%)

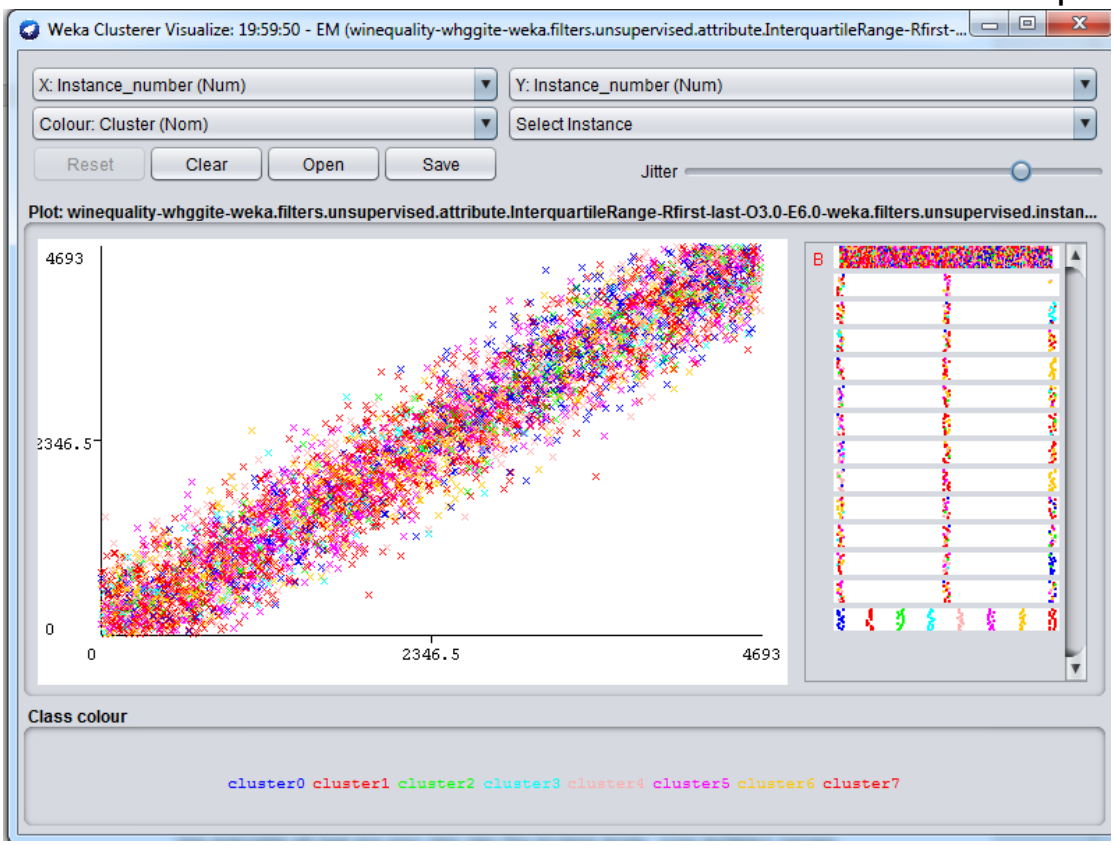
1	758	(16%)
2	239	(5%)
3	192	(4%)
4	771	(16%)
5	822	(18%)
6	663	(14%)
7	671	(14%)

בהרצה ה-6 הגעתי למצב הנ"ל



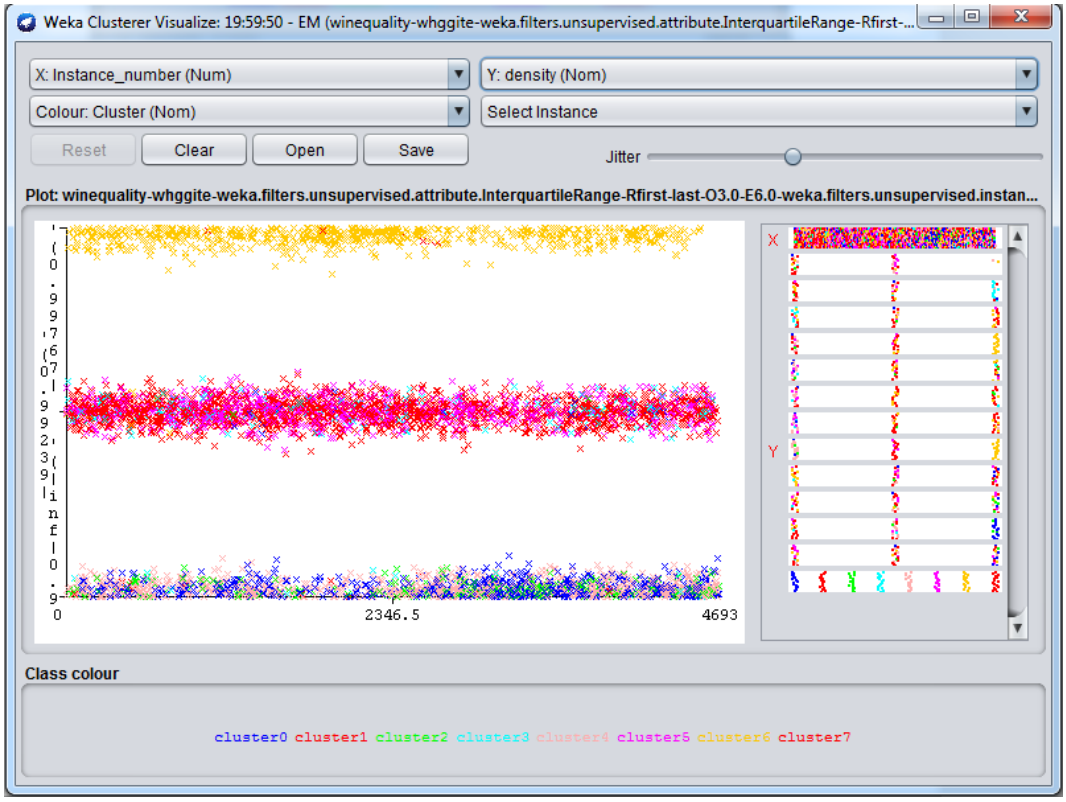
שמראה התפלגות טובה, יחסית מסודרת ובלי יותר מידי רעש ועם זאת לא מלאכותית מידי.

ניתן לראות כי בסידור הנתונים לפי פיזור המשתנים

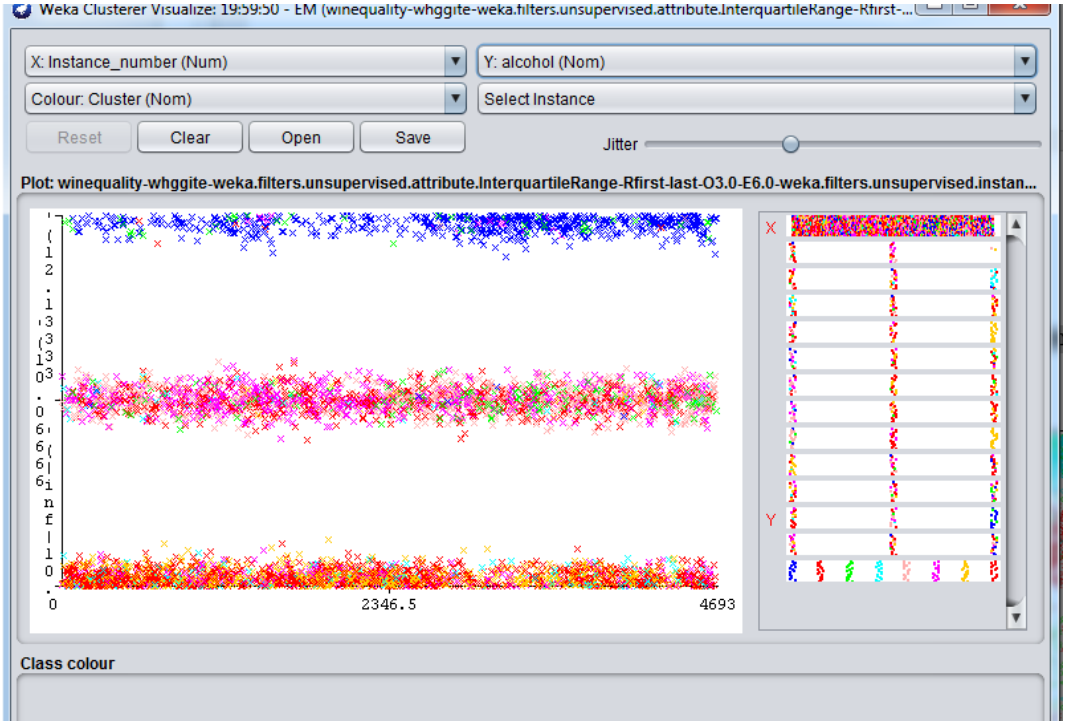


אפשר להבחין כי ישנה הצטברות של האשכול הכחול בחלק העליון של התמונה אבל לדעתי זה לא משפיע בצורה מהותית על פיזור אחיד של הנתונים.

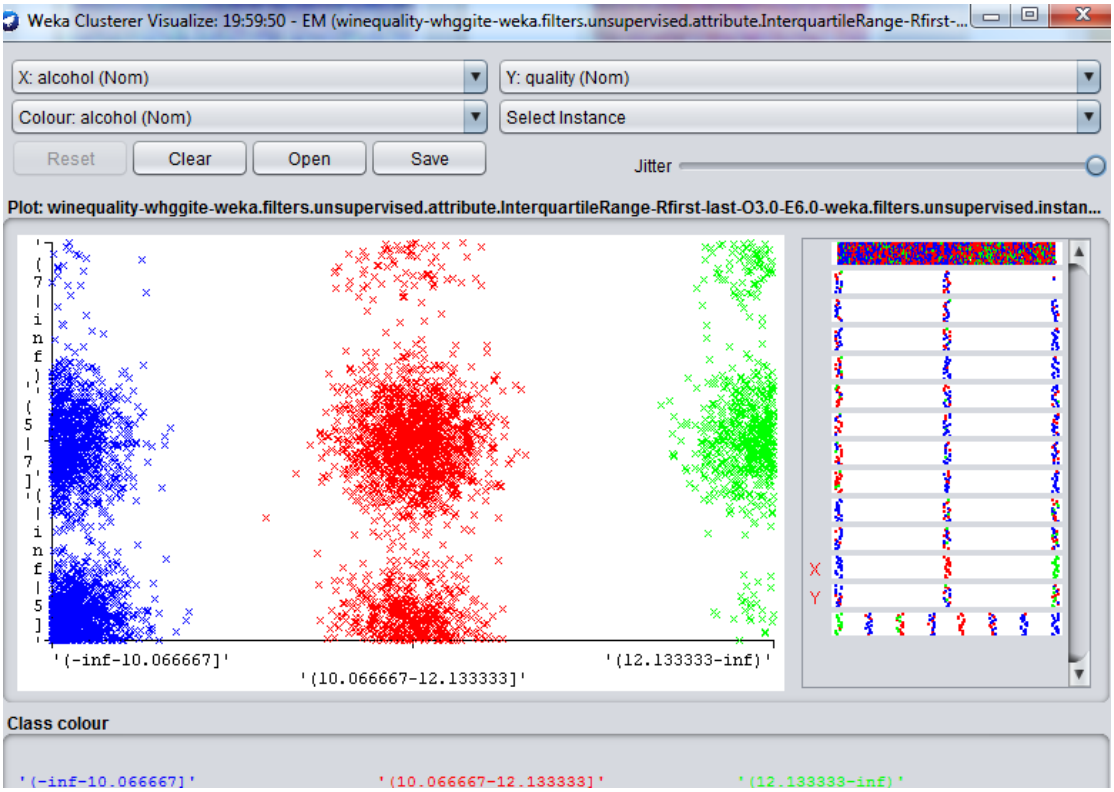
תוצאות ההרצה מראות על אשכול ברור למספר תכונות, גם עבור אלגוריתם ההיררכי, במיוחד עבור 2 התכונות הבאות:
צמיגות (density):



ועבור האלכוהול:



בדקתי את התכונות והקשרים שגיליתי באלגוריתם KMEANS נראה כי התוצאות דומות למדי
אם כי הפיזור שונה במעט.



אבל נראה שאת המסקנות שהסקתי מהאלגוריתם הקודם ניתן להסיק גם באלגוריתם הזה.

ניתן, אם כן, לסכם ולומר כי שני האלגוריתמים הניבו תוצאות דומות. אמנם EM חילק את הנתונים ל-8 אשכולות ו-K-Means ל-3, אך קשה למצוא הבדלים מובהקים המשתמעים מכך. 2 התכונות הדומיננטיות ביותר בשני האלגוריתמים הן density ו-alcohol. יתרון קל לטובת EM ניתן למצוא, כאמור, בכך שהנתונים מתפזרים אצלו בצורה יפה יותר. שאר התכונות, בשני האלגוריתמים, התפזרו בצורה פחות ברורה לעין. היתרון של K-Means מתבטא בכך שזמן הריצה של האלגוריתמים ההיררכיים ארוך בהרבה, דבר שגרם להסרת תכונות לשם קבלת תוצאות מהירות, כך שלא ניתן לדעת מה היו המסקנות במצב אחר.

3. **<u>סיכום ומסקנות</u>**

מטרת העל של שני הפרויקטים – ממ"ן 21 וממ"ן 22, הייתה כריית מידע והסקת מסקנות לגבי חיזוי איכות יין לבן מתוך סט תכונות נתונות, שהתפרסמו על ידי יקב נתון.

אך המטרות בממ"ן 21 היו שונות לגמרי מהמטרות בממ"ן 22. ממ"ן 21 התמקד בטיוב הנתונים והכנתם לכריית המידע ובסיווג הנתונים בעזרת אלגוריתמים שונים של עצי החלטה. ממ"ן 22 עסק בניתוח המידע והסקת מסקנות, על-ידי יצירה של חוקי הקשר ובניית אשכולות.

בסיום ממ"ן 21 נותרנו עם 11 תכונות ולאחר סינון של חלק מהמידע שהתקבל, בעיקר ערכים קיצוניים ולא רלוונטיים.

הדבר איפשר גישה קלה יותר לנתונים בממ"ן 22, ומתוך כך – סביבה נוחה להתמודדות עם המידע והוצאת התמצית מתוכו בדמות חוקי ההקשר ויצירת האשכולות.

המסקנות מממ"ן 21, מעבר לטיוב הנתונים, היו העדפת האלגוריתם J48 על-פני CART, לפחות לגבי סט הנתונים הנוכחי. J48 סיפק מספרים מעט טובים יותר, בכל הנוגע לדיוק הסיווג, ובנוסף זמן הריצה שלו נמוך לאין שיעור מזה של CART.

המסקנות בממ"ן 22, מתחלקות לשניים:

1. מבחינת הנתונים: מציאתן של שלוש תכונות עיקריות המראות על קשר מסוים בינן לבין מידת האיכות של יין, כמו שהראתי את הקשר בין חומצה ציטרית ואחוז האלכוהול לאיכות היין.
2. מבחינת כלי הכרייה:
   a. מבחינת תוצאות: העדפה קל של האלגוריתם ההיררכי-הסתברותי EM על פני האלגוריתם החלוקתי K-Means.
   b. מבחינת זמן ריצה וידידותיות למשתמש: העדפה ברורה של K-Means.