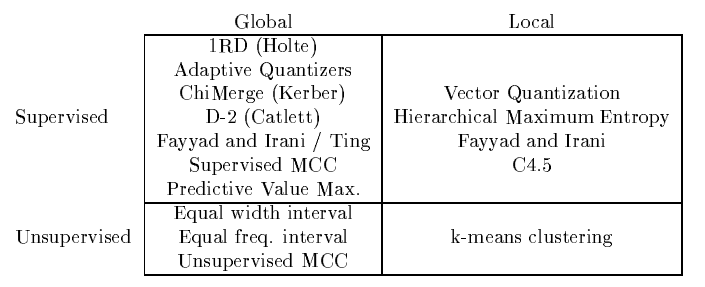
1. חוקי הקשר

א+ב)

ראשית נשים לב כי אנו נדרשים לבצע הבניה נוספת של קובץ הקלט המקורי (40119 שורות לאחר ניקוי) שכן אלגוריתמי חוקי-הקשר איתם נעבוד דורשים כי המידע יהיה דיסקרטי.

בפנינו מספר רב של אפשרויות לביצוע דיסקרטיזציה על המידע, קיימות אפשרויות של דיסקרטיזציה נאיבית (חלוקה לBINים לפי כמות קבועה, רוחב קבוע וכו') וכן דיסקרטיזציה מונחית.

דוגמא לשיטות שונות ניתן לראות בצד שמאל ([מקור](http://robotics.stanford.edu/users/sahami/papers-dir/disc.pdf)).

אני בחרתי לנקוט בשיטה משולבת. ראשית, נבצע דיסקרטיזציה ידנית בעלת משמעות.

במקרים שבהם קיימת משמעות שמובנת לנו בחלוקה מסוימת נעדיף לבצע אותה כך (לדוגמא, אנו יודעים כי ניתן לחלק את משתנה הגיל לטווחים קבועים ומוגדרים כגון : צעיר : 18-30, בינוני 30-50, מבוגר 50-65, זקן 65-100) בגישה זו נוכל להבין ביתר קלות את משמעות החוקים שיצאו אולם גם לשיטה זו יש חסרון שכן ייתכן וחוקי ההקשר דווקא קיימים בין טווחים, לדוגמא לקבוצת הגילאים 25-35 ישנה קורלציה גבוה למאפיין כלשהו אחר.

ולכן ראשית אחלק את המאפיינים שלדעתי יש משמעות לחלוקתם בצורה קבועה ולמאפיינים הנותרים נבצע דיסקרטיזציה בעזרת weka.filters.unsupervised.attribute.Discretize

* נשים לב כי בעולם האמיתי היינו מצפים מהלקוח לספק דגשים לגבי אופן חלוקה בעל משמעות עסקית עבורו במידה וישנה חלוקה שכזה.

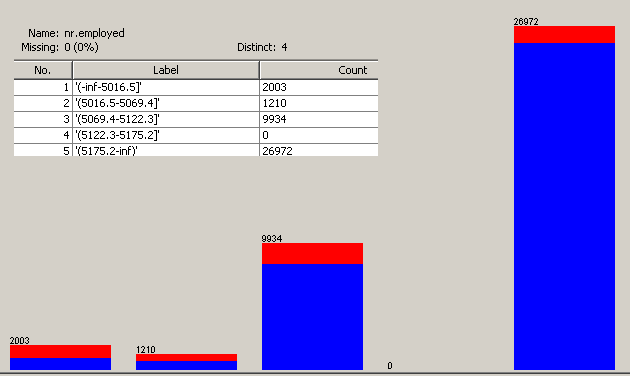
**המאפיינים הדורשים טיפול**

|  |  |  |  |
| --- | --- | --- | --- |
| **שם תכונה** | **תיאור התכונה** | **סוג** | **דיסקרטיזציה** |
| age | גיל | נומרי - שלם | 18-30 = צעיר | 30-50 = בינוני | 50-65 = מבוגר | 65-100 = זקן |
| duration | משך שיחה אחרונה | נומרי - שלם | 0-60 = קצר | 60-300 = בינוני | 300+ = ארוך |
| campaign | כמות התקשרויות בקמפיין הנוכחי | נומרי - שלם | 1-2 = מעט | 3-4 = בינוני | 4+ = ארוך |
| pdays | כמות ימים מאז התקשרות בקמפיין קודם | נומרי - שלם | 999 = no | אחר = yes  האם נוצר קשר בקמפיין קודם |
| previous | כמות התקשרויות לפני קמפיין זה | נומרי - שלם | 0 = no | אחר = yes  האם נוצר קשר בעבר |
| emp.var.rate | שינוי המועסקים במשק (רבעוני) | נומרי - רציף | >=0 = חיובי | <0 = שלילי |
| cons.price.idx | מדד המחירים לצרכן (חודשי) | נומרי - רציף | מקריאה על התכונות ומבחינת ההתפלגויות לא מצאתי הגיון בחלוקה כלשהי |
| cons.conf.idx | מדד הביטחון הכלכלי (חודשי) | נומרי - רציף |
| euribor3m | ריבית היוריבור ל3 חודשים (יומי) | נומרי - רציף | <1 = נמוך | 1-2 = בינוני | 2+ = גבוה |
| nr.employed | כמות המועסקים (רבעוני) | נומרי - רציף | מקריאה על התכונות ומבחינת ההתפלגויות לא מצאתי הגיון בחלוקה כלשהי |

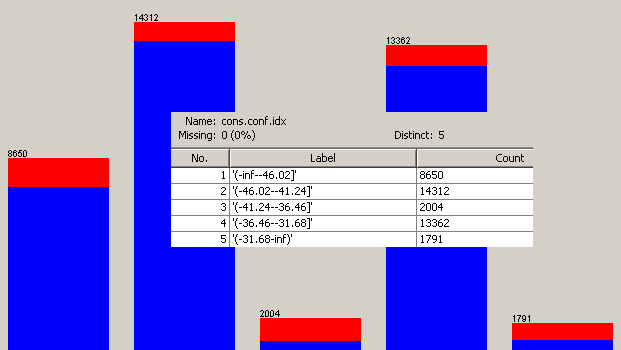
* החלטתי להשאיר את שדה duration הפעם שכן מטרת מציאת חוקי הקשר אינה קלסיפיקציה וחיזוי ולכן אם קיימים חוקי הקשר בין ערכי duration מסוימים למשתנה אחר אזי נתון זה עלול לעניין אותנו לצרכים אחרים (לדוגמא, הערכת משך שיחה עתידי עם לקוח)
* תכונות אשר שונו ידנית, התווסף לשמם "D" כך שהתכונה age הפכה לageD

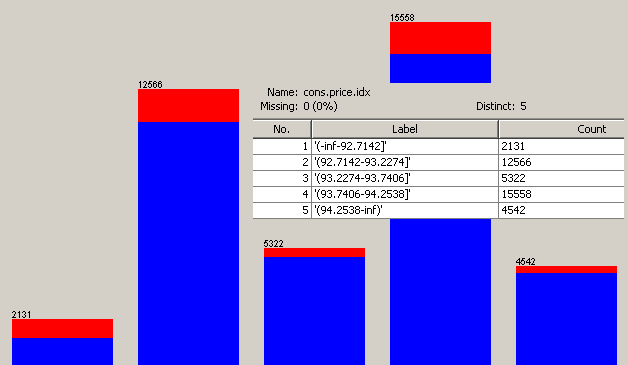
נשארנו עם 3 תכונות בלבד הדורשות דיסקרטיזציה אוטומטית.

תוצאות הדיסקרטיזציה



בחרתי לחלק ל5 BINים , התוצאות נראות לי סבירות, לגבי מדד כמות המועסקים ניתן לראות כי רוב ההתקשרויות היו בתקופה שבה הכמות הייתה גבוהה ולכן יש הבדל של פי ~22 בין הקבוצה הגדולה ביותר לקטנה ביותר אולם בכל זאת ייתכן כי ישנה אינפורמציה מעניינת בקבוצות הקטנות (כל עוד הן יותר גדולות מהMinSupport כמובן).  
אציין רק כי לא לגמרי ברור מה המשמעות של להגדיר טווח של מאפיין כמו מדד בטחון לדוגמא שכן למספר כשלעצמו אין משמעות להבנתי והמשמעות היחידה שלו היא יחסית למדד הקודם ולכן ייתכן שנוריד חלק ממאפיינים אלו בעתיד במידה ויפרעו ללמידה.





כעת יש לנו מידע מוכן ללמידה.

נתחיל בהגדרת Support ו Confidence.

Support(X=>Y) – אחוז הרשומות מתוך מסד נתונים אשר מכילות את האיבר\קבוצה X ו- Y.   
פורמאלית :

נכוון מדד זה על מנת לוודא כי חוקי ההקשר המתקבלים הינם בעלי משמעות ולא נכונים לכמות זניחה בלבד של הרשומות.

Confidence(X=>Y) – רמת הוודאות של חוק הקשר, קרי- אחוז הרשומות מתוך הרשומות המכילות את X אשר מכילות גם את Y. פורמאלית:

נכוון מדד זה על מנת לוודא כי חוק ההקשר המתקבל רלוונטי למציאות.

לצורך הגדרת confidence / support נשתמש בערכים הדיפולטים בweka (אלגוריתם אפריורי)

* נוודא כי car=false שכן אנו מעוניינים בקשרים שאינם בהכרח מובילים למשתנה המטרה מהממן הקודם.

upperBoundMinSupport – Upper bound for minimum support. Start iteratively decreasing minimum support from this value. (**1.0**)

numRules – Number of rules to find. (**10**)

delta -- Iteratively decrease support by this factor. Reduces support until min support is reached or required number of rules has been generated. (**0.05**)

lowerBoundMinSupport – Lower bound for minimum support. (**0.1**)

lowerBoundMinSupport זהו בעצם הSupport כפי שהוגדר לעיל.

metricType - Set the type of metric by which to rank rules. )**confidence**(

minMetric- Minimum metric score. Consider only rules with scores higher than this value. (**0.9**)

כאן אנו מגדירים את סוג מטריקת הדירוג (confidence) ואת ערכה המינימלי (0.9)

בהרצה הדיפולטית התקבלו 9 חוקי הקשר עם confidence = 1. אלו הם חוקים מובנים מאליהם אשר לא תורמים לנו כל אינפורמציה חדשה. כל החוקים הם קשרים בין 3 תכונות :

previousD=No

pdaysD=No

poutcome=nonexistent

כולם אומרים את אותו דבר, "לא נוצר קשר עם הלקוח בעבר". בנוסף, קיים רק עוד ערך אחד (yes) עבור שתי התכונות הראשונות והרי ברור שאם נוצר קשר בקמפיין הקודם (previousD=YES) אז הוא גם נוצר בעבר (pdaysD=YES) ולכן ניתן להוריד את התכונה pdaysD.

ראשית רציתי להשאיר את previousD שכן ייתכן שקיים קשר בין previousD=YES לבין תכונה אחרת וקשר זה לא יכול לבוא לידי ביטוי דרך poutcome שכן ישנן 2 אפשרויות במקרה שפנו ללקוח בעבר – כישלון \ הצלחה אולם לאחר מס' הרצות ראיתי כי רוב הקשרים מכילים שילוב כלשהו בין previousD לבין poutcome ויוצרים המון רעש וחוסר יכולת לנתח את הנתונים ולכן החלטתי להוריד גם את previousD.

כעת (numRules שונה), התקבלו קשרים אשר עדיין עם conf=1 אולם מעניינים במידה מה. לדוגמא :

1. nr.employed='(5175.2-inf)' 26972 ==> euribor3mD=High 26972 <conf:(1)> lift:(1.49) lev:(0.22) [8838] conv:(8838.73)

2. euribor3mD=High 26972 ==> nr.employed='(5175.2-inf)' 26972 <conf:(1)> lift:(1.49) lev:(0.22) [8838] conv:(8838.73)

ניתן לראות כי קיים קשר בין מספר המועסקים לבין הריבית במשק. זהו קשר אשר מצביע על נתון מקרו-כלכלי שייתכן ומעניין בפני עצמו. אולם פחות מעניין את מחלקת השיווק של הבנק לדעתי. מה שכן יהיה מעניין זה לראות קשרים בין נתונים מקרו-כלכליים אלו לבין תוצאות של קמפיינים – זאת נראה בהמשך.

בנוסף קבלתי שלל חוקים עם conf=1 אשר קושרים בין המשתנים המקרו-כלכליים לבין משתנים פרטיים של לקוחות. הרבה מהם חסרי כל הגיון ותוקפם סטטיסטי בלבד שהרי ברור שאין קשר בין גיל הלקוח של הבנק לבין הריבית במשק. לדוגמא :

9. ageD=Adult nr.employed='(5175.2-inf)' 24048 ==> euribor3mD=High 24048 <conf:(1)> lift:(1.49) lev:(0.2) [7880] conv:(7880.53)

10. ageD=Adult euribor3mD=High 24048 ==> nr.employed='(5175.2-inf)' 24048 <conf:(1)> lift:(1.49) lev:(0.2) [7880] conv:(7880.53)

11. ageD=Adult poutcome=nonexistent nr.employed='(5175.2-inf)' 23359 ==> euribor3mD=High 23359 <conf:(1)> lift:(1.49) lev:(0.19) [7654] conv:(7654.75)

12. ageD=Adult poutcome=nonexistent euribor3mD=High 23359 ==> nr.employed='(5175.2-inf)' 23359 <conf:(1)> lift:(1.49) lev:(0.19) [7654] conv:(7654.75)

ניתן לראות כי עבור חוקים 1\2 ממקודם, המתקיימים עבור 26972 מופעים, מתוכם, פשוט נבחרה קבוצה גדולה מספיק, במקרה זה ageD=Adult והוצג קשר עם conf=1.

אם כך, הבאנו מספר חוקים המקיימים את דרישות ה confidence / supportאולם ראינו כי הם אינם מעניינים.

לחלופין, נוכל לא להשתמש בconfidence כלל ולבחור במקומו בLift כדי לדרג ולסנן קשרים.

Lift, כפי שמוגדר בWeka הינו:

Confidence divided by the proportion of all examples that are covered by the consequence.

ולכן, בעזרת שימוש בLift נוכל לקבל לא רק קשרים נכונים סטטיסטית אלא גם בעלי משמעות כאשר המשמעות באה לידי ביטוי בכך שחוקי הקשר עם lift גבוה נבדלים משאר האוכלוסיה, מעניינים וחשובים יותר.

שילוב של פרמטרים הנדונו לעיל ותוצאותיהם יוצגו בסעיף ד'.

ג)

**אלגוריתם Apriori**

מסתמך על התכונה האפריורית, כלומר העיקרון של האלגוריתם הוא שבמידה וקיים סט נתונים שכיח בבסיס הנתונים אז כל קבוצה חלקית של אותה קבוצת פריטים שכיחה חייבת גם היא להיות שכיחה. זהו אלגוריתם איטרטיבי, כאשר במחזור k מוצאים את כל קבוצות הפריטים השכיחות בגודל k, כלומר אלה שמכילות k פריטים. בכל מחזור באלגוריתם האפריורי מתבצעים שלושה מעברים עיקריים:

1. יצירת מועמדים. 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. גיזום מועמדים שלא מקיימים את התכונה האפריורי, כלומר מתעלמים מהצירופים שאינם שכיחים
3. בדיקת התמיכה של קבוצות המועמדים שלא נגזמו בשלב 2 ע"י מניית מספר התנועות שבהן הם מופיעים בתוך בסיס הנתונים עצמו, כלומר בדיקה האם הקבוצות המועמדות באמת שכיחות.

**אלגוריתם HotSpot**

זהו אלגוריתם שלומד מתוך סט של חוקים מהו הערך המקסימלי/המינימאלי עבור משתנה יעד שנקבע מראש. כלומר הוא מאפשר לקבל ניתוח מגמות עבור ערך מסוים אותו אנחנו מחפשים. לדוגמא חיפוש בנתוני חברת הביטוח מהי קבוצת הסיכון הנפוצה ביותר ועל בסיס זה להציע פוליסה ייעודית או לוותר על קבוצת פילוח זו.

מבנה החוקים מוצג באלגוריתם בתצורה של עץ. האלגוריתם עובד בצורה מחזורית, ובכל פעם מוסיף לתוך עץ החוקים את הקישור הבא התואם להגדרות הבסיס ואילוצי התמיכה המינימאלית שהוגדרו על ידי המשתמש.

אלגוריתם הHotSpot מאשר שליטה מלאה במציאת חוקי הקשר לגבי משתנים ספציפיים המעניינים אותנו (בחרתי לבחון את המשתנים loan ו housing) כך נוכל לבחון האם יש קשר בין ערכים מסוימים של משנים אלו לבין משתנים אחרים. לעומתו, אלגוריתם Apriori מתרכז במציאת סטים שכיחים כלליים או עבור משתנה המטרה.

קיימת הצדקה לכל אחד מהם בפני עצמו והמטרות שלהם שונות ולכן מעניין יהיה לבחון את שניהם.

נימוק הבחירה באלגוריתמים הללו מופיע [בסיכום](#_לסיכום,_ניתן_לראות)

ד+ה+ו)

הרצת Apriori

ההרצות התמקדו במציאת קשרים בעלי Lift חיובי גבוה ככול שניתן, תוך שמירה על support מוגדר.

הורדתי את הsupport ל5%.

כאמור בסעיף ב' – צצה בעיה של "ערבוב" בין משתנים מקרו-כלכליים לבין משתנים פרטיים של לקוחות. לצורך הבדיקה, הורדתי כל המשתנים המקרו-כלכליים כדי לבדוק האם קיימים חוקי הקשר בין מאפיינים הלקוח השונים.

ראשית התחלתי מדרישת Lift גבוה של 2. כמצופה הריצה לקחה יותר זמן מהרגיל (שכן האלגוריתם בכל פעם מוריד את רמת הSupport הנדרשת בגודל Delta ומחפש חוקים אשר מספקים את סף המטריקה שהוגדר)

להלן התוצאות :

1. ageD=Adult contact=telephone 12729 ==> marital=married month=may poutcome=nonexistent 4894 conf:(0.38) < lift:(2.13)> lev:(0.06) [2601] conv:(1.33)

2. marital=married month=may poutcome=nonexistent 7225 ==> ageD=Adult contact=telephone 4894 conf:(0.68) < lift:(2.13)> lev:(0.06) [2601] conv:(2.12)

3. marital=married contact=telephone 9452 ==> ageD=Adult month=may poutcome=nonexistent 4894 conf:(0.52) < lift:(2.11)> lev:(0.06) [2570] conv:(1.56)

4. ageD=Adult month=may poutcome=nonexistent 9863 ==> marital=married contact=telephone 4894 conf:(0.5) < lift:(2.11)> lev:(0.06) [2570] conv:(1.52)

5. contact=telephone durationD=Med 8881 ==> month=may poutcome=nonexistent y=no 4968 conf:(0.56) < lift:(2.08)> lev:(0.06) [2577] conv:(1.66)

6. month=may poutcome=nonexistent y=no 10801 ==> contact=telephone durationD=Med 4968 conf:(0.46) < lift:(2.08)> lev:(0.06) [2577] conv:(1.44)

7. month=may durationD=Med poutcome=nonexistent 7010 ==> contact=telephone y=no 4968 conf:(0.71) < lift:(2.06)> lev:(0.06) [2553] conv:(2.25)

8. contact=telephone y=no 13817 ==> month=may durationD=Med poutcome=nonexistent 4968 conf:(0.36) < lift:(2.06)> lev:(0.06) [2553] conv:(1.29)

9. month=may poutcome=nonexistent 11430 ==> contact=telephone campaignD=Few y=no 5204 conf:(0.46) < lift:(2.04)> lev:(0.07) [2653] conv:(1.43)

10. contact=telephone campaignD=Few y=no 8952 ==> month=may poutcome=nonexistent 5204 conf:(0.58) < lift:(2.04)> lev:(0.07) [2653] conv:(1.71)

נשים לב כי בתוצאות די מגוונות הן בConfidence שלהן והן בחוקים עצמם.

לדוגמא, החוק הטוב ביותר מבחינת Lift – 2.13 עם Confidence של 0.68 טוען כי : אנשים נשואים, אשר נוצר איתם קשר טלפוני פעם אחרונה בחודש מאי ולא השתתפו בקמפיין הקודם, אז בהסתברות של 68% הם מבוגרים ואמצעי יצירת הקשר איתם הוא טלפון קווי.

נראה כי דרישה לLift גבוה מובילה לחוקים מורכבים אשר נכונים אם כי להבנתי לא מעניינים עסקית (אולם זאת ישפוט הלקוח).

לאחר הורדת דרישת הLift ל 1.1 קבלנו :

1. contact=telephone 14588 ==> poutcome=nonexistent y=no 13521 conf:(0.93) < lift:(1.18)> lev:(0.05) [2037] conv:(2.91)

2. poutcome=nonexistent y=no 31581 ==> contact=telephone 13521 conf:(0.43) < lift:(1.18)> lev:(0.05) [2037] conv:(1.11)

3. ageD=Adult durationD=Med 21206 ==> marital=married y=no 13395 conf:(0.63) < lift:(1.16)> lev:(0.05) [1836] conv:(1.23)

4. marital=married y=no 21868 ==> ageD=Adult durationD=Med 13395 conf:(0.61) < lift:(1.16)> lev:(0.05) [1836] conv:(1.22)

5. marital=married durationD=Med 15281 ==> ageD=Adult y=no 13395 conf:(0.88) < lift:(1.15)> lev:(0.04) [1727] conv:(1.92)

6. ageD=Adult y=no 30631 ==> marital=married durationD=Med 13395 conf:(0.44) < lift:(1.15)> lev:(0.04) [1727] conv:(1.1)

7. poutcome=nonexistent 34651 ==> contact=telephone y=no 13521 conf:(0.39) < lift:(1.13)> lev:(0.04) [1587] conv:(1.08)

8. contact=telephone y=no 13817 ==> poutcome=nonexistent 13521 conf:(0.98) < lift:(1.13)> lev:(0.04) [1587] conv:(6.34)

9. contact=telephone 14588 ==> poutcome=nonexistent 14194 conf:(0.97) < lift:(1.13)> lev:(0.04) [1594] conv:(5.03)

10. poutcome=nonexistent 34651 ==> contact=telephone 14194 conf:(0.41) < lift:(1.13)> lev:(0.04) [1594] conv:(1.08)

מדובר בחוקים ברורים הרבה יותר.

החוק הראשון לדוגמא יכול להיות מאוד מעניין. נראה כי לאנשים שדרך יצירת הקשר שלהם היא טלפון קווי, הסיכוי שהשיווק לא יצליח הוא 93% שיפור של 18% מהכלל.

דבר דומה ניתן גם לראות בחוק 5 עבור אנשים נשואים ששיחתם האחרונה הייתה במשך בינוני. (88% לא להצליח, שיפור של 15% מהכלל)

כעת נבחן בשנית את המידע על המשתנים המקרו-כלכליים בלבד ומשתנה המטרה שהרי המטרה שלנו אינה להבין את ההשפעות של המשתנים המקרו-כלכליים על עצמם אלא למצוא קשר למשתנים שמעניינים את הבנק, כמו משתנה המטרה.

עבור Lift של 2 מתקבל ודרישה ל50 חוקים, החוקים היחידים שמכלים את “y” הם :

37. cons.price.idx='(92.7142-93.2274]' 12566 ==> empvarrateD=Neg y=no 11057 conf:(0.88) < lift:(2.65)> lev:(0.17) [6882] conv:(5.56)

38. empvarrateD=Neg y=no 13327 ==> cons.price.idx='(92.7142-93.2274]' 11057 conf:(0.83) < lift:(2.65)> lev:(0.17) [6882] conv:(4.03)

41. empvarrateD=Neg 16766 ==> cons.price.idx='(92.7142-93.2274]' y=no 11057 conf:(0.66) < lift:(2.39)> lev:(0.16) [6436] conv:(2.13)

42. cons.price.idx='(92.7142-93.2274]' y=no 11057 ==> empvarrateD=Neg 11057 conf:(1) < lift:(2.39)> lev:(0.16) [6436] conv:(6436.21)

ניתן לראות כי החוק החזק ביותר מראה כי כאשר מדד המחירים לצרכן הינו בין 92.7142 לבין 93.2274 אזי יש סיכוי של 88% הלקוח לא יסכים לתנאי הפיקדון וכן שינוי המועסקים במשק יהיה שלילי - נתון זה גדול ב165% מהכלל.

בעזרת שינוי פרמטרים של האלגוריתם ובעזרת הוספה \ הורדה של מאפיינים ניתן להגיע לתוצאות משמעותיות בהרבה מביצוע נאיבי של האלגוריתם. כל החוקים שיצאו בסעיף ב' אינם באמת מעניינים אותנו והיינו צריכים לעשות שלל התאמות כדי לקבל חוקים שפוטנציאלית יכולים לעניין את הבנק.

הרצת HotSpot

הריצה לחיפוש חוקי הקשר עבור loan=yes :

Hot Spot

========

Total population: 40119 instances

Target attribute: loan

Target value: yes [value count in total population: 6236 instances (15.54%)]

Minimum value count for segments: 624 instances (10% of target value total population)

Maximum branching factor: 3

Maximum rule length: unbounded

Minimum improvement in target: 5%

loan=yes (15.54% [6236/40119])

housing = yes (17.09% [3681/21541])

| job = admin. (18.73% [1040/5552])

| campaignD = Avg (18.25% [772/4230])

| day\_of\_week = mon (18.14% [826/4553])

job = admin. (16.79% [1709/10181])

month = jul (16.74% [1175/7018])

ניתן לראות כי נמצאו מס' חוקי הקשר בין לקיחת הלוואה לבין מאפיינים אחרים.

שיעור התופעה באוכלוסייה של לקיחת הלוואות הוא 15.54%. אולם, אם בוחרים לשווק הלוואות רק לאוכלוסיית ה”admin”ים מקבלים Lift של ~1.08 (16.79%/15.54%).

הLift הטוב ביותר נובע מאוכלוסיית ה”admin”ים בעלי המשכנתא שם הLift הוא ~1.2 (18.73%/15.54%)

הריצה לחיפוש חוקי הקשר עבור housing=yes :

housing=yes (53.69% [21541/40119])

loan = yes (59.03% [3681/6236])

| contact = cellular (61.73% [2502/4053])

euribor3mD = Med (58.76% [5503/9365])

cons.conf.idx = '(-inf--46.02]' (58.71% [5078/8650])

cons.price.idx = '(92.7142-93.2274]' (58.67% [7372/12566])

| marital = single (59.5% [2371/3985])

גם במקרה זה נמצאו ממצאים מעניינים לדוגמא הLift הגבוה ביותר מתקבל באוכלוסיית בעלי ההלוואות האישיות אשר ניתן ליצור איתם קשר בסלולאר (Lift = ~1.15)

בנוסף נראה כי קיים קשר בין נתונים מקרו-כלכליים כגון הריבים במשק \ מדד המחירים לצרכן לאחוזי לקיחת משכנתאות (כמצופה בעולם האמיתי)

# לסיכום, ניתן לראות כי האלגוריתמים שונים אחד מהשני ומשרתים מטרות שונות, זאת גם אחת הסיבות שבחרתי להציג דווקא את שניהם, האלגוריתם האפריורי משמש אותנו לחיפוש חסר כיוון במידע, טוב מאוד לקבלת תמונה מלאה של חוקי הקשר, בין אם הם מובנים מאליהם ובין אם לאו. האלגוריתם יודע להציג שלל קשרים בין מאפיינים שונים הן בצד הגורר והן בצד הנובע, שיטה זו מומלצת עבור משתמשים\חוקרים עם מעט מידע על המידע ולפני גיבוש כוונה ברורה למציאת חוקי הקשר מסוימים. לעומתו, אלגוריתם הHotSpot הינו מוכוון מטרה מאוד קונקרטית, מציאת חוקי הקשר עבור ערך מסוים של מאפיין מסוים. כאשר מידע זה עלול לא לצוף בכלל באלגוריתם האפריורי שכן לעיתים מרוב עצים לא רואים את היער, כמות החוקים שניתנים להסקה במסד נתונים ממוצע היא ענקית ולכן כאשר השאלה ברורה וידועה מראש רצוי להשתמש בHotSpot אשר בנוסף גם מציג את המידע בצורה קלה ונוחה לעבודה כפי שניתן היה לראות.

בעזרת האלגוריתמים לעיל הראנו כי ניתן לקבל חוקים אשר יכולים לעזור למחלקת השיווק בעבודתה ונוגעים לשלל מאפיינים שאינם בהכרח משתנה המטרה המקורי.

חוקי אלו נותנים Liftים לא רעים – 20% ו15% בHotSpot ו 18% באפריורי בנתונים הפרטיים של הלקוח ועד 165% בקשרים בין הנתונים המקרו-כלכליים למשתנה מטרה.

בנוסף, ראינו כמה חושב להבין את המידע, את המשמעות שלו ואת ההשפעה של המשמעות הזו על האלגוריתמים. במקרה זה ביצעתי חלוקה בין משתנים פרטיים של הלקוח ומשתנים מקרו-כלכליים המשותפים לכלל הלקוחות בתקופה מסוימת וניסיתי לבצע חיפושים שונים של חוקי הקשר בין ובתוך הקבוצות הללו לקבלת תוצאות איכותיות יותר.

2) ניתוח אשכולות

א) אשכול הוא אוסף של אובייקטים אשר דומים אחד לשני ושונים מאובייקטים אחרים הנמצאים באשכול אחר.

ניתוח אשכולות היא פעולה של מציאת דמיון ביו מאפיינים שונים של אובייקטים וקיבוצם באשכול אחד

בעזרת ניתוח אשכולות ניתן לזהות קבוצות שונות במידע ולהסיק מסקנות לעולם הבעיה כגון : בפרסום – חלוקת הלקוחות לאשכולות ופיתוח תוכניות שיווק שונות לכל קבוצה.

ב) ככל שאשכול איכותי יותר כך הדמיון בתוך האשכול גדול יותר והדמיון בין אשכולות קטן יותר.

לדוגמא, בגישת הpartition נוכל להשתמש בקוטר האשכול כדי למדוד את טיבו – מדד זה מתייחס למידת הדמיון של האיברים **בתוך האשכול**

בנוסף, נוכל לבדוק את המרחק בין האשכולות השונים – מדד זה מתייחס למידת השוני **בין האשכולות השונים**.

מדד איכות נוסף הינו כללי ומתייחס ליכולת שלנו למצוא תבניות מוסתרות במידע. ככל שנמצא יותר תובנות כאלו אזי ניתוח האשכולות איכותי יותר.

ג) שיטות אשכול נתונים :

1. אשכול חלוקה (partitioning) – חלוקת המידע באופן כלשהו לאשכולות כך שכל רשומה שייכת לאשכול אחד ולאחר מכן תיקון סידור האיברים באשכולות עד לקבלת אשכולות אשר ממזערים את פונקציית המטרה. לדוגמא : k-means, k-medoids, CLARANS
2. אשכול היררכי (hierarchical) – יוצר סט אשכולות מקוננים המוצגים כעץ היררכי. כל רמה בהיררכיה היא בעלת סט נפרד של אשכולות, כאשר ברמה הנמוכה ביותר כל פריט הוא באשכול היחיד שלו, וברמה הגבוהה ביותר כל הפריטים שייכים לאותו אשכול. לדוגמא : Diana, Agnes, BIRCH, ROCK, CAMELEON
3. אשכול מבוסס צפיפות (density-based) – אשכולות מבוססי צפיפות הם אזורים אשר הצפיפות בהם גדולה יותר מאשר באזורים אחרים במידע. לדוגמא : DBSCAN, OPTICS, DenClue

בחרתי בגישת החלוקה (k-means) והצפיפות (DBSCAN)

K-Means

העיקרון העומד מאחורי ה- k-Means הוא שהמרכז בכל אשכול ואשכול מיוצג ע"י ערך ממוצע של האובייקטים בתוך אותו האשכול. להלן השלבים של האלגוריתם:

1. מחלקים את האובייקטים המצויים בבסיס הנתונים ל- K אשכולות בצורה אקראית
2. מחשבים ערך ממוצע "מוביל" לכל קבוצה K' על פי החלוקה לקבוצות שביצענו
3. מבצעים בדיקת שיוך של האובייקטים בכל אזור אל מול ערכי הממוצע של כל אשכול. האובייקטים עם המרחקים/ההפרשים הקטנים מול אותו ממוצע ישויכו לאשכול K' המתאים
4. במידה ושיוך האובייקטים בהתאם לערכי הממוצע השתנה חוזרים לסעיף 2
5. תהליך מחזורי זה יסתיים רק לאחר שלא יתקיימו יותר תזוזות של האובייקטים בין האשכולות בהתאם להגדרת ערכי הממוצע

בחרתי להשתמש באלגוריתם זה מתוך נוחות השימוש, כלומר האלגוריתם מאפשר גמישות החיפוש של הקבוצות המתאימות בבסיס נתונים גדול גם אם לא קיים שום פרמטר פילוח מקדים.

DBSCAN

האלגוריתם מאפשר לחלק תצפיות לאשכולות על סמך הצפיפות של הנקודות. הוא הופך אזורים עם צפיפות מספיק גבוהה לאשכולות, ומגלה אשכולות בעלי צורה בלתי הגיונית בבסיסי נתונים מרחביים המכילים גם רעש. האשכול, על פי ההגדרה של DBSCAN, זהו סט מקסימלי של נקודות שמחוברות בהתבסס על צפיפות. להלן שלבי האלגוריתם:

1. בוחרים, באופן אקראי, נקודה p בתוך הנתונים
2. מחפשים בבסיס הנתונים את כל הנקודות שניתן להשיג אותן בהתבסס על צפיפות מהנקודה p, בהינתן ערך מסוים של Eps וגם ערך מסוים של MinPts
3. אחרי הזיהוי של כל הנקודות שניתנות להשגה בהתבסס על צפיפות מהנקודה p, בודקים אם p מהווה נקודת ליבה (core).

* אם p היא אכן נקודת הליבה אז יוצרים אשכול
* אם התברר ש- p היא נקודת גבול, כלומר היא ניתנת להשגה בהתבסס על צפיפות מנקודות ליבה אחרות אבל אין לה מספיק נקודות בסביבה הקרובה שלה, ברדיוס Eps, אז מנסים לבדוק נקודה אחרת בתוך בסיס הנתונים וליצור אשכול מסביב לאותה נקודה

1. התהליך נמשך עד שכל נקודה בתוך בסיס הנתונים משויכת לאיזשהו אשכול או מוכרזת כ- outlier

בעזרת שימוש באלגוריתם מבוסס צפיפות ניתן לאתר גם אשכולות שאינם עגולים, אנו לא יודעים אילו אשכולות קיימים אצלנו ולכן זה יתרון. בנוסף, האלגוריתם עוצר כאשר הוא מסיים לשייך איברים לאשכולות או להגדירם כoutliers ולכן אין צורך לדעת מראש כמה אשכולות ידרשו.

לאור היתרונות לעיל בחרתי באלגוריתם זה כדי לבחון האם הוא נותן שיפור אל מול K-Means

ד+ה+ו)

במקרה זה השתמשתי בקובץ הקטן לאור זמני הריצה הארוכים, על הקובץ הקטן ביצעתי ניקוי של המידע כפי שהוסבר בממ"ן 21 אולם בעבור 2 האלגוריתמים לא היה צורך לבצע דיסקרטיזציה (שכן פונקציות מרחק לא דורשות זאת).

K-Means

בניתוח אשכולות ניתן לבחור להעריך את האשכולות על בסיס מאפיין כלשהו ובכך ראשית אפשר יהיה לבדוק את איכות חלוקת האשכול (% טעויות לפי המאפיין הנבחר)

שנית, במידע שלנו קיימים שלל נתונים מתחומים שונים כגון – נתונים אישיים על הלקוחות, נתונים סטטיסטיים על הקמפיינים, נתונים מקרו-כלכליים.

בשלב זה רציתי לבחון את 2 הקבוצות האחרונות, מה היא התפלגות הנתונים הסטטיסטים על הקמפיינים ומתוכם בחרתי את המאפיין poutcome (תוצאת הקמפיין הקודם) ואת התפלגות הנתונים המקרו-כלכליים (ריבית, שינוי המועסקים במשק)

באלגוריתם זה, נוכל לבחור את כמות האשכולות הנדרשת (K), את פונק' המרחק אשר משפיעה על בחירת האשכול לכל איבר בכל איטרציה וseed אשר משנה את הבחירה הראשונית.

הרצתי עם הערכים הדיפולטים, שיניתי את הseed ואת פונק' המרחק מ- Euclideanל-.Manhattan וכן שיחקתי עם K.

את סיכום הריצות אפשר לראות בטבלה זו :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Distance | S | K | Iterations | Distance/Error within Cluster | Assign Mistakes |
| Euc | 10 | 2 | 6 | 18139.10228 | 47.92% |
| Euc | 1 | 2 | 6 | 18387.4523 | 43.04% |
| Euc | 20 | 2 | 5 | 19064.16992 | 20.8546 |
| Man | 10 | 2 | 5 | 20878.33612 | 34.40% |
| Man | 1 | 2 | 4 | 21963.0511 | 37.34% |
| Man | 20 | 2 | 5 | 20926.15597 | 29.79% |
| Euc | 10 | 3 | 6 | 16716.98691 | 60.74% |
| Euc | 20 | 3 | 5 | 16927.416 | 58.19% |
| Euc | 20 | 5 | 5 | 15423.77755 | 69.12% |
| Euc | 20 | 10 | 5 | 13582.72479 | 83.54% |
| Euc | 20 | 100 | 5 | 8177.430984 | 95.75% |

נשים לב לכמה דברים מעניינים :

1. ככל שK עולה כך אחוז ההשמות השגויות עולה וכן הטעות יורדת, זה די ברור כשחושבים על כך שכן אם ניקח את מסקנה זו לקיצון וניצור אשכולות ככמות האיברים אזי השגיאה שלנו תהיה 0 ואחוז ההשמות הלא נכונות קרובה ל100 (שכן יש רק 2 אשכולות שיסווגו לYES \ NO)
2. פונקציית המרחק משחקת תפקיד חשוב ביצירת האשכולות וזאת ניתן לראות לפי אחוז ההשמות השגויות.
3. גם הSEED יכול להשפיע משמעותית על נתון זה ולכן חשוב לשחק איתו.

לאחר מחשבה נוספת, הבנתי כי ייתכן ובחירת המאפיינים שלי הייתה בעייתית. טענתי כי אני רוצה לבחון את החלוקה על פי נתונים מסוימים (סטטיסטיים\מקרו-כלכליים) אולם הכנסתי גם נתונים אישיים ביצירת האשכולות.

כעת אנסה להשאיר רק מאפיינים הקשורים למידע אותו אני מחפש.

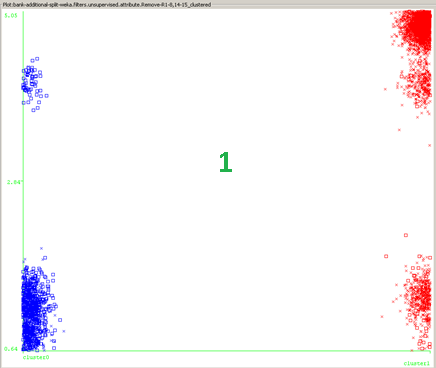
להלן התוצאות :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Distance | S | K | Iterations | Distance/Error within Cluster | Assign Mistakes |
| Euc | 1 | 2 | 7 | 7025.50922 | 48.07% |
| Euc | 20 | 2 | 3 | 7527.504146 | 21.17% |
| Man | 1 | 2 | 5 | 8819.356815 | 29.79% |
| Man | 20 | 2 | 8 | 8867.248835 | 29.42% |

ניתן לראות כי אשכול זה מחזיר אשכולות אשר גם הטעות הפנימית שלהם נמוכה יותר וגם אחוז ההשמות השגויות נמוך יותר.

אכן נראה כי המאפיינים האישיים הוסיפו רעש בחלוקה לאשכולות מסוג זה.

נבחן ויזואלית את תוצאות הריצה השנייה בטבלה לעיל כאשר :

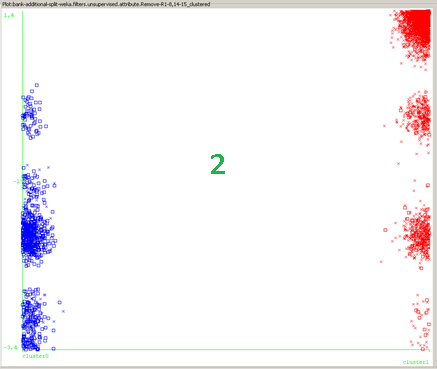
Cluster 0 <-- yes

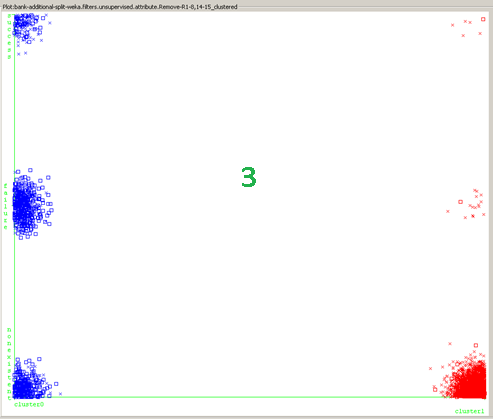
Cluster 1 <-- no

1 – פיזור הריבית בתוך האשכולות. ניתן לראות, כי באשכול 0 בו אנשים לקחו הלוואות, הריבית הייתה נמוכה מאוד בצורה גורפת. לעומת זאת, באשכול 1 רוב התצפיות הן בריבית גבוהה בהרבה. זאת התנהגות מצופה ומראה כי החלוקה לאשכולות היא בכיוון הנכון.

2 – פיזור שינוי המועסקים במשק. ניתן לראות כי באשכול 0, שינוי המועסקים במשק לאורך כל הציר היה שלילי ובאשכול 1 היה ברובו חיובי.

3 – פיזור תוצאות הקמפיין האחרון באשכולות. ניתן לראות כי באשכול 0, עבור אוכלוסייה גדולה מאוד הקמפיין הקודם היה כישלון או הצלחה – וזהו סימן לכך שבכלל היה קמפיין כלשהו לעומת אשכול 1 שבו כלל לא נוצר קשר על הלקוח לפני כן.





DBSCAN

במקרה זה, לאחר שלמדתי את הצורך בסינון מאפיינים מהבדיקה הקודמת התחלתי מראש עם המאפיינים שצוינו לעיל.

הפרמטרים איתם נשחק יהיו epsilon , מספר האיברים המינימלי באשכול ופונקציית המרחק.

לאחר הרצות שונות קבלתי את התוצאות הבאות :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Distance | Epsilon | minP | K | Assign Mistakes | Outliers |
| Euc | 0.9 | 6 | 71 | 87.28% | 339 |
| Euc | 1 | 100 | 12 | 38.75% | 2338 |
| Euc | 1.4 | 50 | 2 | 9.74 % | 30 |
| Euc | 1.5 | 100 | 1 | 10.95% | 0 |
| Euc | 2 | 100 | 1 | 10.95% | 0 |
| Man | 1.5 | 100 | 3 | 29.96% | 517 |
| Man | 2 | 100 | 2 | 23.31% | 365 |
| Man | 2.2 | 100 | 1 | 8.84% | 157 |

ראשית נשים לב שבאלגוריתם זה יש התמודדות עם Outliers וזה דבר חיובי, לא תמיד נרצה לכפות סיווג לקבוצה כלשהי ונרצה לדעת שיש רשומות שלא מתאימות לאף קבוצה (כמובן שמספר זה צריך להיות יחסית נמוך).

ניתן לראות כי בערכים הדיפולטים קבלנו המון אשכולות. בפועל אנחנו שואפים למספר של 2-3 אשכולות לכל היותר.

עם הגדלת Epsilon (מאפשר צפיפות נמוכה יותר) מצד אחד והגדלת מספר האיברים המינימלי באשכול מצד שני ניתן לראות את הירידה במספר האשכולות.

נשים לב כי יש הבדל משמעותי בין פונקציות המרחק השונות, עבור Epsilon מעל 1.5 אי אפשר היה למצוא יותר מאשכול אחד בשימוש במרחק אוקלידי. לעומת זאת, בשימוש במרחק מנהטן גם בערכי אפסילון של 1.5 וגם ב2 נמצאו אשכולות סבירים.

עוד נקודה חשובה שכדאי לשים אליהם לב היא שכמות האשכולות \ טעויות השמה וoutlier לא תמיד מספרים את הסיפור המלא.

לכאורה, בריצה 3, קבלנו 2 אשכולות עם מעט טעויות וoutliers אבל אם נבדוק את אופן החלוקה נקבל :

Classes to Clusters:

0 1 <-- assigned to cluster

3615 42 | no

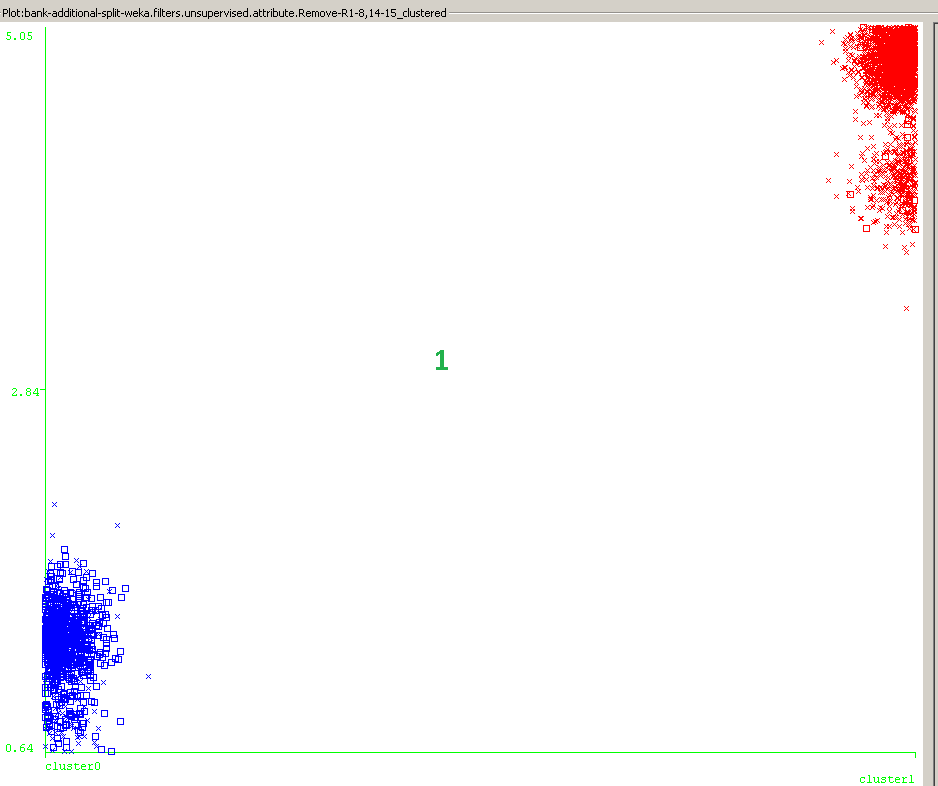
359 73 | yes

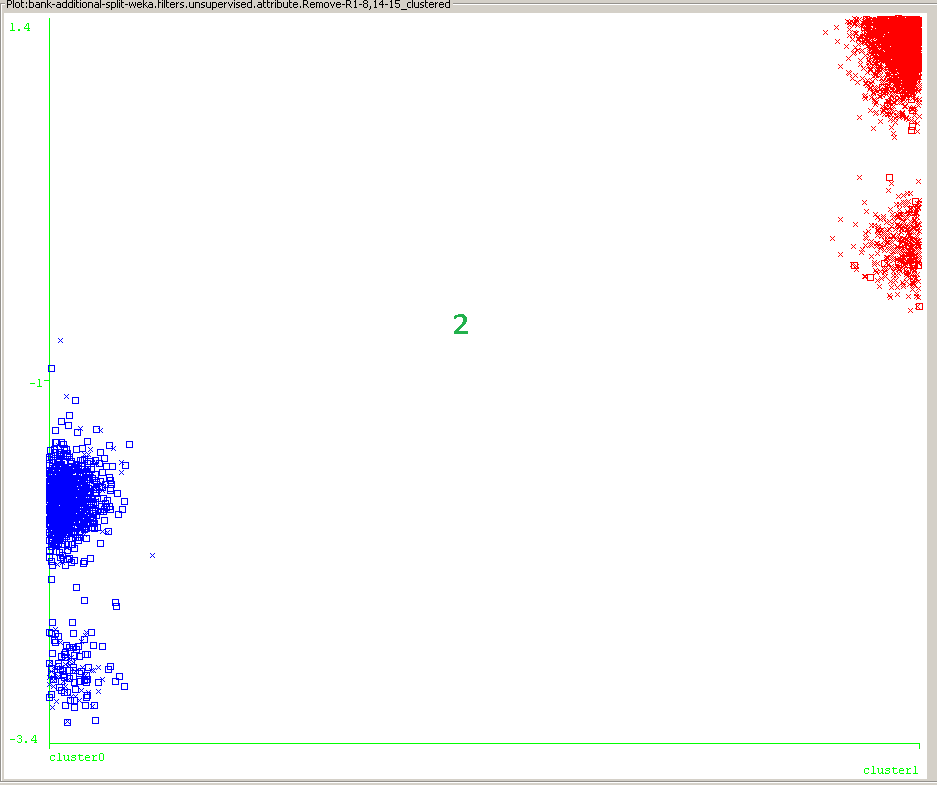
Cluster 0 <-- no

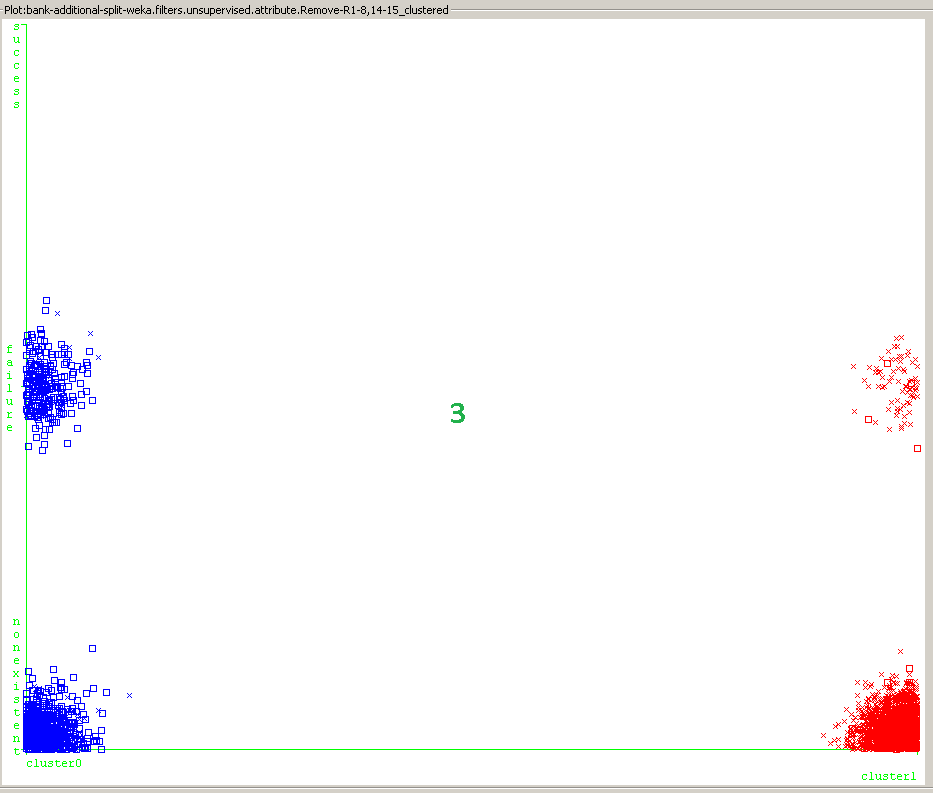
Cluster 1 <-- yes

כך שבפועל חולקה זו לא שונה בהרבה מאשכול בודד.

לבסוף, נבחן ויזואלית את ההתפלגויות באשכולות כי שעשינו לעיל בעבור הריצה הלפני אחרונה.







Cluster 0 <-- yes

Cluster 1 <-- no

1 – פיזור הריבית בתוך האשכולות. ניתן לראות, בדומה לk-means, כי באשכול 0 בו אנשים לקחו הלוואות, הריבית הייתה נמוכה מאוד בצורה גורפת. לעומת זאת, באשכול 1 רוב התצפיות הן בריבית גבוהה בהרבה ואכן גם הפעם זאת התנהגות מצופה ומראה כי החלוקה לאשכולות היא בכיוון הנכון.

2 – פיזור שינוי המועסקים במשק. ניתן לראות כי באשכול 0, שינוי המועסקים במשק לאורך כל הציר היה שלילי ובאשכול 1 היה ברובו חיובי.

3 – פיזור תוצאות הקמפיין האחרון באשכולות. ניתן לראות כי באשכול 0, קיימים יותר אנשים איתם יצרו קשר בעבר לעומת אשכול 1 אולם ההבדל הינו עצום.

אם נשווה בין האלגוריתמים נראה כי הם עובדים בדרכים שונות ואפשר להגיע לחלוקות שונות איתם.

בכל אלגוריתם אנו נדרשים לבחון שלל פרמטרים ולכוונן עד לקבלת תוצאות מדויקות יותר ויותר.

כפי שראינו לעיל DBSCAN הצליח ליצור קבוצות אשר מובחנות יותר על פני נתונים מקרו-כלכליים ולכן טוב בכדי להבין עולם זה, הקשרים שבו (על פי מאפיינים שקובצו יחד) ואופן ההשפעה של מאפיינים אלו על ההלוואות בבנק. מצד שני K-Means הבחין יותר טוב בנתונים הסטטיסטיים של הקמפיין ולכן יותר שימושי לתחום זה. אין לנו מנצח חד משמעי אבל הבנו יותר טוב איזו שיטה מתאימה לאיזה סוג של אשכול וזאת כנראה בגלל אופי הנתונים של הלקוח.

3) סיכום ומסקנות

בממ"ן הראשון נדרשנו לייצור מודל לחיזוי לקוח שיחתום על תנאי הפיקדון. לפני העבודה על המודל ניתחנו את שלבי הKDD השונים, את אופן הכנת המידע ועיבוד ראשוני שלו. לאחר מכן עברנו לבחינה של מודלי classification וצללנו לעומק לתחום זה, מצאנו כי מודל של C4.5 היה יותר מדויק וענה על התנאים שדרשנו בצורה טובה יותר. מודל זה הציג שיפור על פני majority vote בסיווג הרשומות ולכן יכול פרקטית לבוא לידי ביטוי בתהליכי השיווק בבנק.

בממ"ן זה ניסינו להסיק מסקנות על הנתונים עצמם ולאו דווקא על משתנה המטרה, חיפשנו חוקים וקשרים בין מאפיינים שונים וגילינו עובדות מעניינות על תחום השיווק בבנק ועל מאפייני הלקוחות, נתונים אלו יכולים לשפר את עבודתם של המכירות והמנהלים בבנק ויכולים לעזור לקבל החלטות חכמות יותר. משם המשכנו לתהליך של ניתוח אשכולות, ראינו כי באמצעות כלים שונים מתקבלות תוצאות שונות ולכל אחד יש את היתרונות והחסרונות שלו, צריך להכיר אותם ולהכיר היטב את הנתונים כדי לדעת באיזה פתרון לבחור לבעיה. בניתוח האשכולות הצלחנו לסווג את המידע לקבוצות שונות בעלות מאפיינים שונים בלי לדעת שום דבר מראש לכאורה.

באופן כללי, לאורך העבודה על הממנים הבנתי כי ישנם מספר דברים חשובים ביותר בעבודת כריית המידע וביניהם :

1. אין פתרון אחד שמתאים לכולם, צריך לבחון אלגוריתמים וגישות שונות, לכוון פרמטרים ולעשות עוד ועוד ניסויים. לעיתים נרצה להוריד מאפיינים ולעיתים נראה ליצור מאפיינים חדשים בעצמו על בסיס מאפיינים אחרים כדי "לעזור" למודל.
2. חשוב להבין בכל שלב לאן אנו רוצים להגיע ומה בדיוק המטרה, הבנה זו עוזרת לבחור כלים יותר טובים למדידת טיב התוצאות המתקבלות.
3. חשוב להבין את התמונה הגדולה, מה התחום שבו עוסקים, מי השחקנים בתחום, מי הולך להשתמש בתובנות שלנו, איך הוא מודד את התוצאות וכו'
4. צריך להבין את מגבלות כל מודל ושיטה ולהבין איך זה משפיע על המקרה שלנו. (שוב מתחבר עם היכרות עם המידע)