DFS –

נתקדם אל האות הקטנה מבין הצמתים שניתן להגיע אליהם בקו ישיר מהצומת הנוכחית ובכל פעם שנגיע לצומת חדשה, ניצור OPENLIST חדש שמכיל רק את הצמתים שבחץ מהצומת הנוכחית. מבחינתנו, G1 כלומר – goal היא האות הגדולה ביותר ב א"ב.

ב iterative DFS –

נכתוב את כל החיפושים של כל הרמות, עד שבאחת מהרמות הגענו אל GOAL, כלומר:

S | SAB | SACDBEG2

ב HillClimbing –

נסתכל על היוריסטיקה במרחק אל היעד,

התחלנו מ S שבו היוריסטיקה היא 10, נמשיך עם האותיות מהקטנה ביותר בא"ב כל פעם מהצומת הנוכחית שאנו נמצאים בה, כל עוד היוריסטיקה ממשיכה לשפר אותנו בדרך. לדוגמא:

S 10

A 5

D 2

G1 0

A\* -

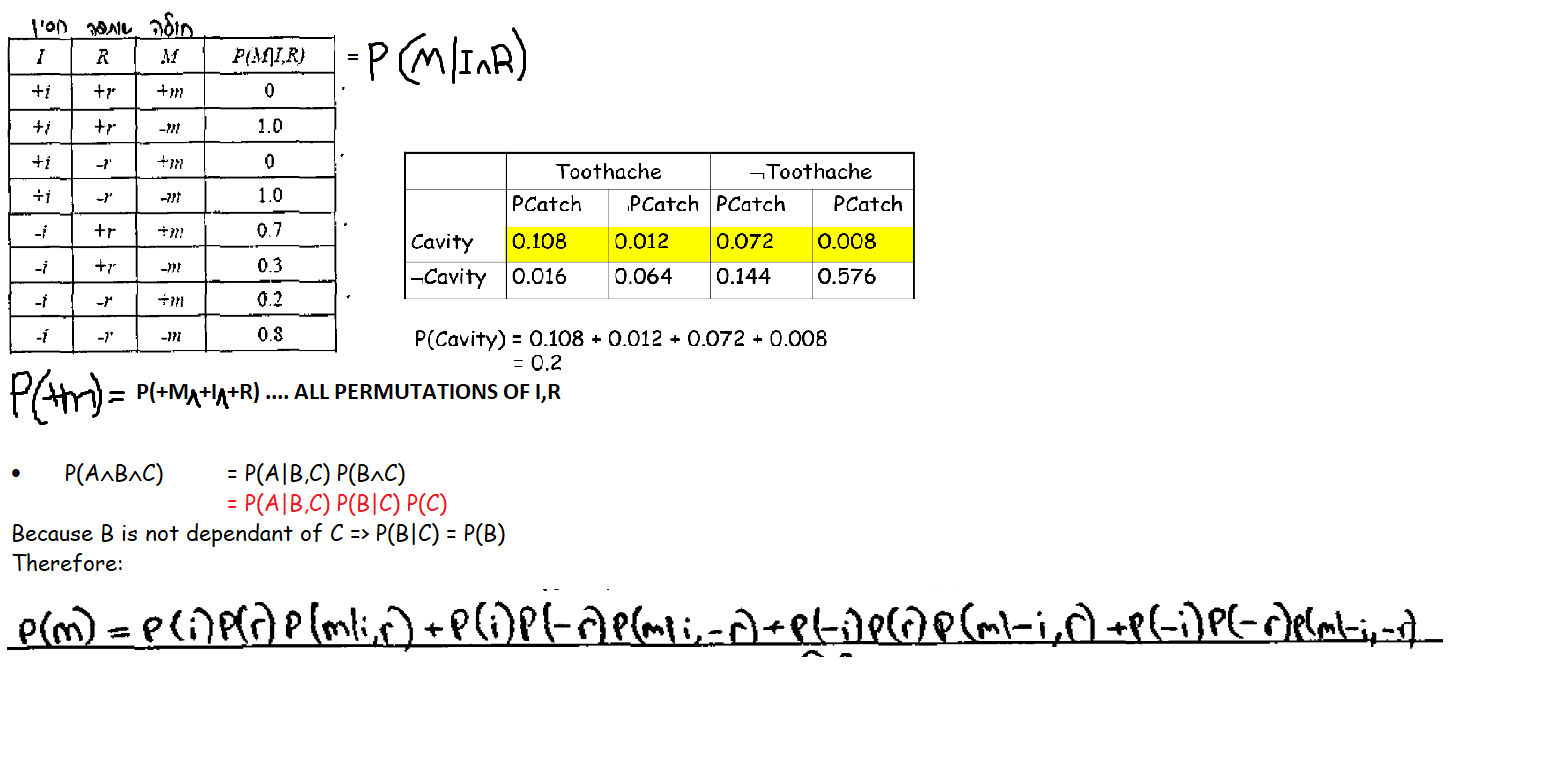
בכל צומת, אנו מראים את המסלול אל הצמתים שיוצאים ממנה והעלות שלהם היא העלות של היוריסטיקה + סכום הקשתות אליה מההתחלה. בכל שלב, נפתח את הצומת בעלת העלות הנמוכה ביותר שהגענו אליה. אם מצאנו צומת שביקרנו בה כבר ויש אליה מסלול קצר יותר, נשנה את הערך שלה ב VISITED.

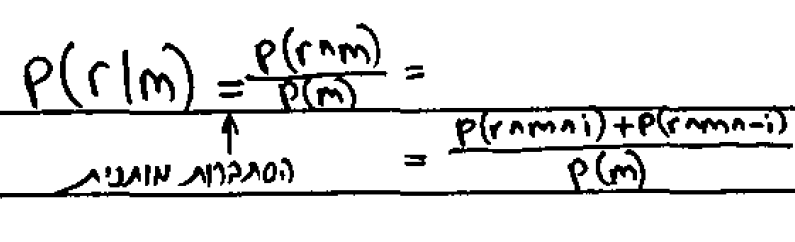
ברגע שפיתחנו צומת והגענו לצומת כזו שוב, כשנרצה לפתח אותה, לא נפתח ונשים סוגר.

כאשר אנו מפתחים והגענו אל צומת יעד, נבחר שזוהי צומת היעד הסופית רק אם בשלב הפיתוח, הגענו למצב שבו זהו המסלול הבא לפיתוח ולא נשאר מסלול עם ערך נמוך יותר.

התשובה שנציג היא ה Visited list שלנו לפי הסדר כי אלה הצמתים שעברנו עליהם ופיתחנו.

a-b pruning  
נבצע pruning רק במצב בו a וגם b אינם ריקם. במצב שהגענו לצומת ובדקנו את הצד הקשת השמאלית, כדי לדעת האם ניתן לגזום את הקשת הימנית צריך לבדוק האם: a>=b, אם כן, נעלה למעלה ונגזום את החלק הלא שמיש.

הוספנו את +i ואת -i מפני שלא היה ידוע אם הערך שלו הוא חיובי או שלילי ו m היה תלוי בו, אז אנו לוקחים בחשבון את שתי האופציות ומחברים אותן.



בA\* - במצב בו רוצים שניצור פונקציית היוריסטיקה

תחילה, נחשב עבור כל קודקוד מהסוף להתחלה, את המסלול הכי קצר בפועל דרך הקשתות לפי משקלי הקשתות מהקודקוד אל היעד ונסמן בכל קודקוד.

לאחר מכן, נתחיל מההתחלה ונוסיף ערך היוריסטי לכל קודקוד כך שיקיים:

ערך היוריסטיקה של הקודקוד קטן מהמסלול הכי קצר בפועל – אדמיסביליות

ערך היוריסטיקה של הקודקוד נמצא בין לבין הקודקודים מלפניו ואחריו במסלולים האפשריים - קונסיסטנטי

שיטת Naïve Bayes:

הנה שיטה שבה אנו מקבלים דוגמאות שונות ועבור כל דוגמא, אנו מקבלים את ה classification שלה – זהו שלב האימון.

לאחר מכן, נקבל דוגמא כלשהי ונרצה לבצע חיזוי מה יהיה ה classification של הדוגמא.

על מנת לבצע זאת, ניקח את כל הדוגמאות הקודמות שקיבלנו באימון התואמות לדוגמא הנוכחית שנאו רוצים לחזות

ונבצע עבור כל classification, חישוב בצורה הבאה:

נבדוק מה ההסתברות ל classification A בהנתן הדוגמא הנתונה, לדוגמא: x=T, y=T

נבצע את אותו הדבר עבור classification B בהנתן אותה הדוגמא

ולאחר מכן, נבחר את התוצאה הגבוהה מבין השניים, כלומר – זו שיש לה את ההסתברות הגבוהה היותר לקבל את ה classification שלה.

כלומר:

נחשב:

P(C-A | x=T ^ y=T)

P(C-B | x=T ^ y=T)

משתמשים בנוסחאת Naive בייס והופכים.

בד"כ משאירים את המכנה כי הוא קבוע ומצטמצם עם ההשוואה לclassification השני

ואז במונה, מצד ימין – נגיד ההסתברות ש P(classification=yes) רואים כמה פעמים היה Yes מתוך הכל

ואת השמאלי, נגיד: P(X=T^Y=T) מפצלים לכל אחד מהם בהנתן C, נגיד:

P(x=T|C=1)

ופשוט רואים מתוך כל הפעמים ש C=YES כמה פעמים X=T

המטרה שלנו בתרגילי Naïve Bayes היא פשוט לחזות את ה classification בהתאם להסתברויות של הדוגמאות שקיבלנו באימון.

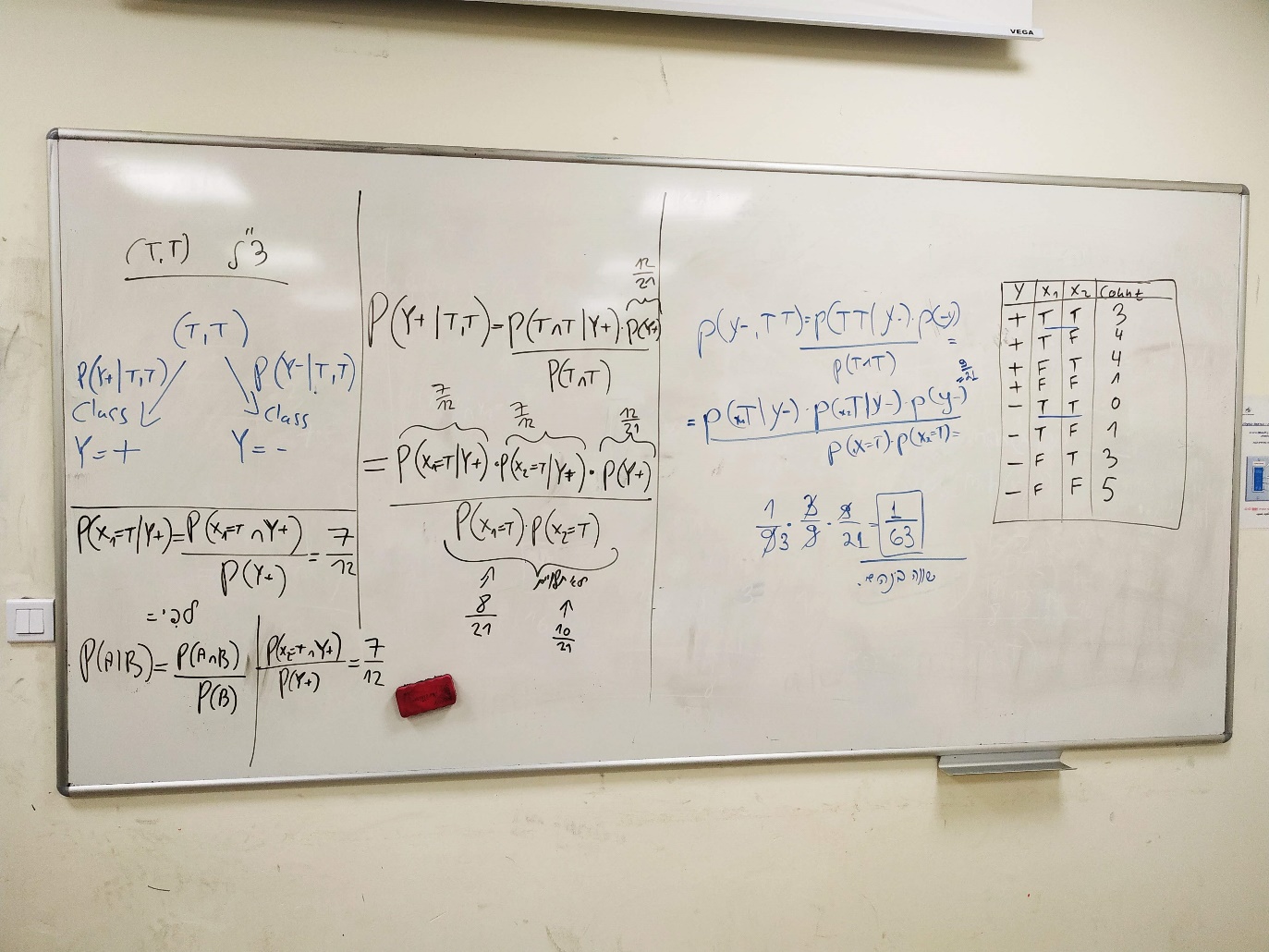
יצירת צביר Cluster בדרך – k-means

1. בחר רנדומלית בנקודת האמצע
2. עבור כל איבר – הוסף אותו לצביר הכי קרוב אליו
3. חשב את הממוצע של כל צביר
4. בחר עבור כל איבר מחדש, את הצביר הכי קור אליו
5. חזור על 3 ו-4 עד שאין שינוי

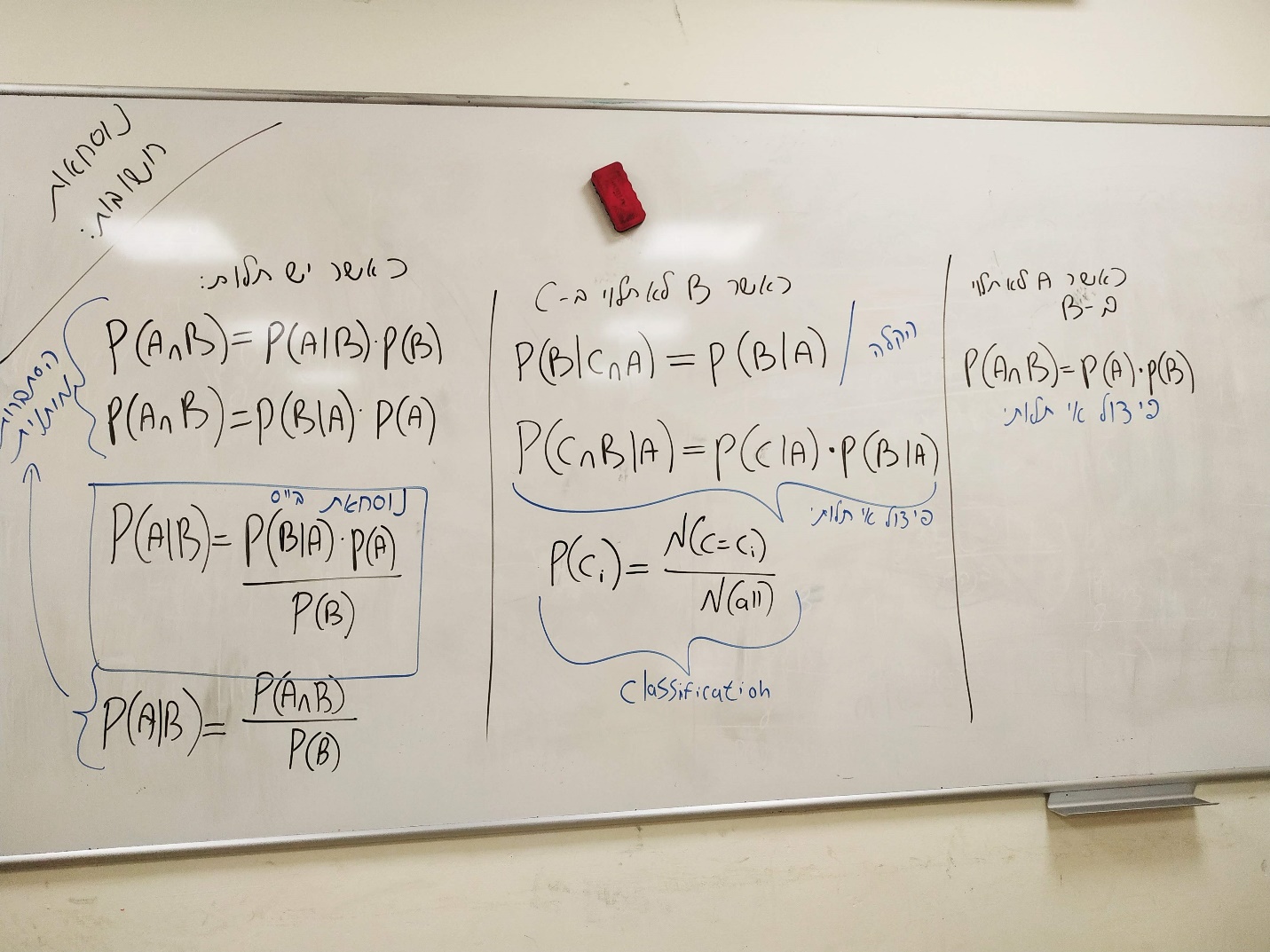
יצירת פונקציה היוריסטית –

1. קונסיסטנטי – נדאג שבכל מסלול, ערך היוריסטיקה של איבר קטן מהאיבר לפניו וגדול מזה שאחריו
2. אדמיסבילי – נדאג שלכל איבר, ערך היוריסטיקה יהיה קטן יותר מהמרחק הכי קצר ממנו ליעד.

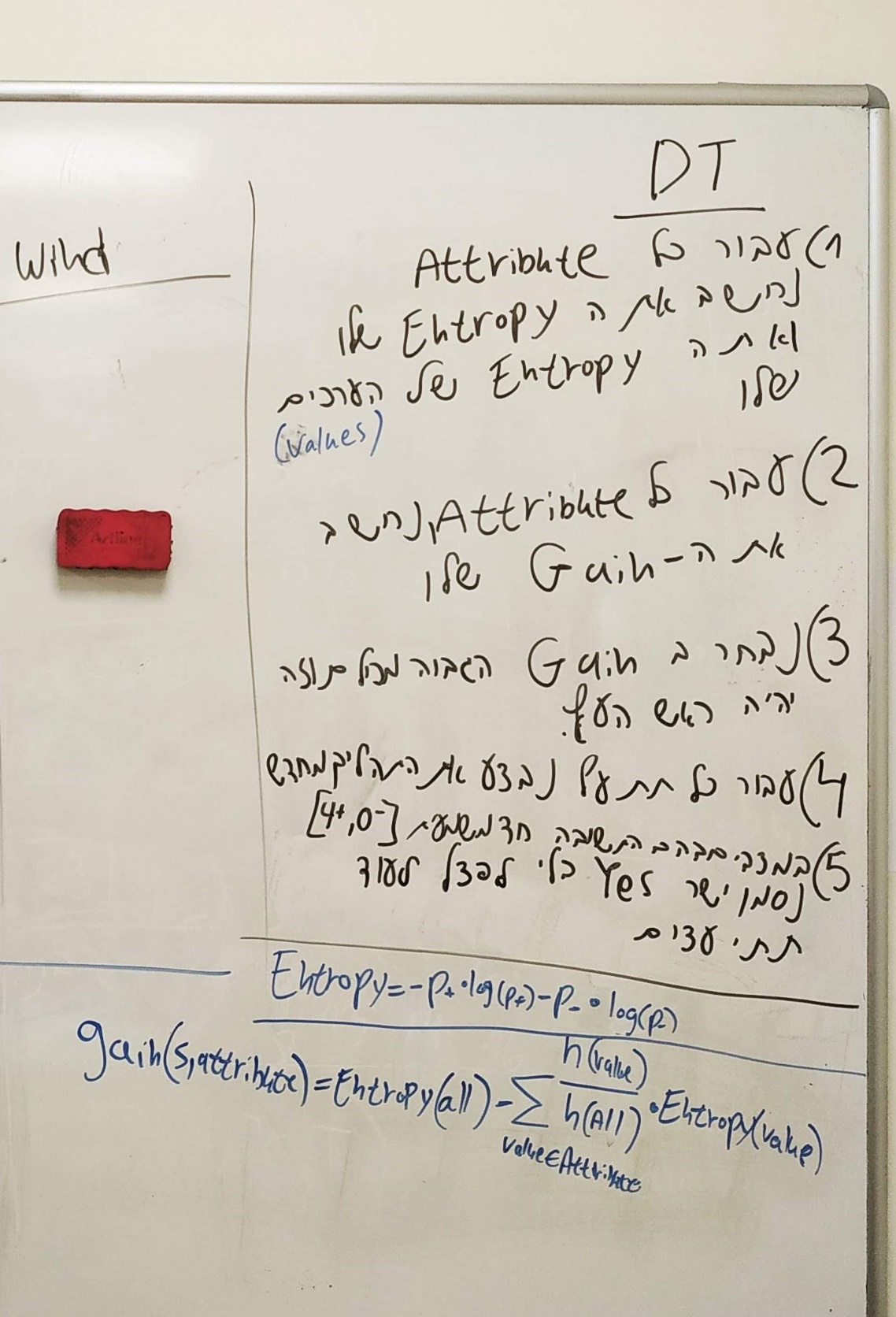
חישוב classification עבור דוגמא חדשה, בשיטת naïve bayes.



נוסחאות הסתברות חשובות:



יצירת עץ החלטות:



בד"כ ברשת בייסיאנית: פיצול הסתברויות כשיש שילוב של תלות ואי תלות.

לדוגמא, נתון:

B,D אינם תלויים באף אחד

A תלוי ב B

C תלוי ב B,D

נראה פיצול של הסתברות משולבת, נכפיל את כל ההסתברויות, נשים בהתחלה את הבלתי תלויים ואז נכפיל בהסתברות של התלויים בהנתן חיתוך של אלה שהם תלויים בהם:

P(A^B^C^D) = P(B)\*P(D)\*P(A|B)\*P(C|B^D)

בנוסף, כאשר יש לנו תלות במשתנה אחד או שניים ונרצה לדעת הסתברות מסויימת של משתנה, לדוגמא:

P(C=false)

לא מספיק רק לחבר את כל האופציות, נגיד לחבר אחד מכל ארבעת ה:

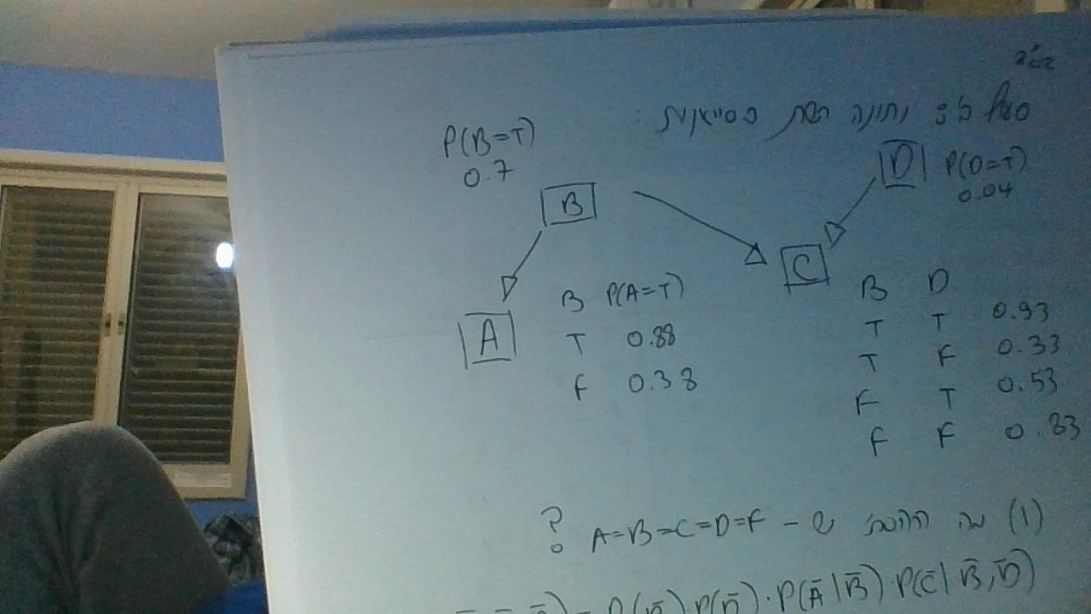
P(C=false | B = false ^ D = true)

בנוסף צריך להתחשב בסיכוי שיקרה מצב שבו: B = false ^ D = true

לכן נחבר את כל ארבעת המצבים האפשריים כך שבכל אחד מהמצבים נחשב:

P(C=false | B = false ^ D = true) \* P(B = false) \* P(D = true)

דוגמא להצגת רשת בייסיאנית:



ב Naïve bays כשיש צורך לבצע classification, נבדוק את שני ה classification, כל אחת את ההסתברות שלה בהנתן אותה הדוגמא שיש לנו ונבחר את ההסתברות הגדולה יותר.

בעצי החלטה, מי שיש לו remainder נמוך יותר, הוא משפיע יותר ונבחר אותו.

חישוב remainder עבור משתנה x:

Remainder(x) = P(x=1)\*E(F|X=1) + P(x=0)\*E(F|X=0)

את ה entropy מחשבים לפי הדף נוסחאות, עם ה I.

