**מבוא ללמידה ממוכנת: שיעורי בית 1 –**

**מגישים:**   
איתי בר – 325839710  
ליאון גורין – 214511214

**שאלה 1:**

נרצה לדעת אם לקנות או לא לקנות ולשם כך נרצה לגלות מי יותר גדול: או . (החישוב נעשה בדומה להרצאה)

נרצה לחשב קודם את ה risk:

נחשב את שמתאר במקרה שלנו את את איכות התמונה של הטלוויזיה:

נחשב את לכל פרמוטציה של :

*כעת נוכל לחשב לכל פרמוטציה ולהגיד מה ההחלטה האופטימלית לכל מצב.*

זה אומר שיש סיכון גבוהה יותר לא לקנות, לכן אם התמונה של הטלוויזיה חדה נקנה!

זה אומר שיש סיכון גבוהה יותר לקנות, לכן אם התמונה של הטלוויזיה לא חדה לא נקנה!

ה risk הכולל יהיה:

**שאלה 2:**

1. נשתמש בחוק בייז בשביל לפתור את השאלה:

נתון:

נחשב את החלקים החסרים:

נסמן:

נציב במשוואה המקורית:

1. אם הייתי הולך להתחסן, יש לי סיכוי גבוהה יותר שאין לי את המחלה כי הסיכוי שיש לי את המחלה הוא 0.001 לעומת סיכוי שאין לי את המחלה שהוא 0.999. תוצאת הבדיקה הוא משתנה בלתי תלוי להסתברות שיש לי או אין לי את המחלה.
2. התשובה ל-(א) תהיה שונה כי MLE לא מתייחס ל posterior ולכן יחליט בא' בדיוק הפוך ממה שיצא לנו.

התשובה ל-(ב) לא תהיה שונה כי אין התייחסות לשום posterior בחישוב שביצענו ונקבל את אותה התשובה

**שאלה 4:**

הסבר על הקוד –

פונקצית def learn\_NB\_text():

1. הקוד קולט את המידע על ה training data עם readTrainData
2. הקוד סופר כמה משפטים יש מכל קטגוריה לתוך category\_occurrence\_dict ומחשב ומכניס לתוך לפי סדר ידוע (קבוע).
3. הקוד עובר על הקבצים וסופר מספר מופעמים של מילה מסוימת בקטגוריה ואת מספר המילים בקטגוריה.
4. לבסוף הוא עובר על כל המילים ומחשב את ההסתברות של מילה בהינתן קטגוריה (עם Laplacian smoothing)
5. מכניסים את כל ההסתברויות של מילים כתלות בקטגוריה כרשימה לתוך Pw ומקבלים מטריצה (השורה הראשונה של Pw הגדרתי להיות רשימת מילים לפי הסדר)

פונקצית def ClassifyNB\_text(Pw, P):

1. הקוד מקבל את המידע על ה test data עם readTrainData
2. הקוד עובר על כל הקבצים לפי הסדר ועבור כל קובץ הוא מחשב את ההסתברות שהקובץ בקטגוריה מסוימת
   1. אנחנו בעצם מחשבים את:
   2. אנחנו מאתחלים את ההסתברות עבור הקטגוריה הנוכחית ומוסיפים log של כל הסתברויות של מילה כתלות בקטגוריה (אם המילה קיימת).
   3. אחרי שעוברים על כל הקטגוריות בוחרים בקטגוריה שקיבלה את ההסתברות הכי גבוהה ובודקים אם צדקנו. אם כן מוסיפים 1 להצלחות.
3. מחזירים את מספר ההצלחות חלקי מספר הקבצים.

הקוד בסוף מדפיס את ה success rate שחושב.