**דירוג טלפונים חכמים**

**באמצעות אלגוריתמים לדירוג**

****

**פרויקט סיום קורס**

**אלגוריתמים היוריסטיים ומקורבים ויישומים**

סמסטר ב', תשע"ט

**מגישים:**

לוי מלינדה 201310356

בארשבסקי צחי 311334544

לוקשין קוסטיה 310765821

**תוכן העניינים**

רקע 2

חלק א' – הפרמטרים הנבדקים3-7

מאפייני המכשיר3

הצרכים שלנו4

קריטריונים ודירוגם5-7

חלק ב' – אלגוריתמי הבדיקה8-16

שיטת BORDA8-9

אלגוריתם TOPSIS10-12

שיטת שולצה13-14

השוואה ובחירת האלגוריתם המתאים15-16

חלק ג' – יישום ומסקנות17-24

מימוש האלגוריתמים17-19

סימולציות והסברים20-22

מסקנות החישוב23

ציפייה לעומת מציאות24

נספח – תרגילי בית25

**רקע**

שוק הסמארטפונים כיום מספק לנו מגוון עצום של מכשירים מתקדמים שבאמצעותם נוכל לנהל את חיינו. הקדמה הטכנולוגית והחידושים הדיגיטליים מבטיחים שהסמארטפון שנרצה יהיה מהיר, איכותי ונוח.

מבדיקה כללית וסקר מקדים שביצענו במסגרת הפרויקט על מנת להבין את הבעיה המשתקפת מולנו לעומקה, מצאנו כי מרבית האנשים מכלל קבוצות האוכלוסייה מעדיפים לקנות מכשיר טלפון על סמך המלצות של קרוביהם, ואלו שבכל זאת מגיעים "מוכנים" לחנות הסלולר, לרוב מתמקדים בבדיקה שלכאורה נראית להם מקיפה, אך לא די בכדי להשיג, בסופו של דבר, את המכשיר האופטימלי עבורם.

במרבית המקרים, חיפוש הסמארטפון האידאלי מסתכם בבדיקת מספר יכולות מרכזיות, אך מאוד מצומצמות, של המכשיר:

* מחיר - ככל הנראה הפרמטר הכי חשוב למרבית האנשים. ניתן למצוא לא מעט מכשירים מצוינים במחיר שווה לכל כיס, כאלה שמספקים תמורה טובה מאוד מבחינת נוחות, אבזור, טכנולוגיה ועוצמה.
* איכות השמע – היכולת של המכשיר לנהל שיחה בנוחות, בעוצמה מספקת ונטולה קטיעות באמצעות הספיקר הפנימי, הרמקולים של הסמארטפון והספיקר החיצוני.
* מסך - הדגש המרכזי הוא בעיקר על הרזולוציה, טווח הצבעים, נוחות הקריאה וחוויה בזמן משחק.
* נוחות השימוש – עד כמה נוח וידידותי הסמארטפון למשתמש, החל ממידות המכשיר ועד לתפעול מערכת ההפעלה, ממהירות חיישן טביעת האצבע ועד לאיכות החומרים וההרכבה.
* תוכנה וביצועים – פרמטר המתייחס ליעילות המעבד ומהירות מערכת ההפעלה, ובפשטות: כולנו מעוניינים במכשיר שלא יגמגם ברגע האמת, לא יתבע מסכי טעינה רבים וארוכים מדי ולא יתסכל אותנו בתפעול השוטף כתוצאה מ"שיהוקי מערכת".
* איכות המצלמה – נמדדת על סמך איכות הצילומים באור יום, תחת תאורה מלאכותית וצילומי מאקרו בשעות הערב. כמו כן, איכות הווידאו נמדדת ברמת החדות והפירוט, וכמות הרעידות שתועדו בכל סרטון.
* סוללה – אחת התכונות החשובות ביותר לרוב משתמשי הסמארטפון. נקדים ונספר שנורא קשה למצוא סמארטפון שיידע לעבוד יותר מיום וחצי ולענות בצורה טובה על כל הקריטריונים שהוצגו קודם-לכן.

כידוע, Life time של סמארטפון נע סביב שלוש שנים. כל קונה במעמד הקנייה מוכן להשקיע סכום לא מבוטל של כסף, בהנחה שההשקעה "תסגור לו את הפינה" לפחות לשנתיים וחצי הקרובות, ושהוא יוכל להפיק ממנה את התועלת הרצויה.

**התוצאה בפועל:** רבים מכלל האוכלוסיה מחליפים את מכשיר הטלפון שברשותם לאחר שימוש של כשנה וחצי עד שנתיים בעקבות חוסר שביעות רצון.

**חלק א' – הפרמטרים הנבדקים**

על מנת לקבל את התמורה המיטבית להשקעה, החלטנו להקל על רוכשי הסלולר ולהציע דרך למציאת מכשיר הסמארטפון הנכון ביותר עבורם. הפתרון המוצע מתמקד להתמקד בשני פרמטרים מרכזיים: **מאפייני המכשיר והצרכים שלנו**.

**מאפייני המכשיר**

מאפייני המכשיר בהם בחרנו להתמקד במסגרת הפרויקט כוללים את הקריטריונים שהוצגו קודם-לכן וכן קריטריונים נוספים:

****

**הצרכים שלנו**

המילה "שלנו" מקבלת משמעות שונה כאשר מתבוננים בקבוצות אוכלוסייה שונות. במסגרת הפרויקט בחרנו להתמקד ב-3 קבוצות אוכלוסייה ולהבין את הצרכים שלהן לעומקם:

|  |  |  |
| --- | --- | --- |
| **קבוצת אוכלוסייה** | **עיקר הפעילות** | **הדרישה מהמכשיר** |
| **ילדים** | * הפעלת סרטוני יוטיוב ברצף * שיטוט באתרי תוכן כבדים * הפעלת משחקים כבדים | * חיי סוללה ארוכים * גודל מסך מירבי * טעינת סרטונים ומשחקים כבדים במהירות * יכולות וידאו גבוהות * תפעול אינטואיטיבי וקליל בעבור הילד הממוצע במינימום תסכול ובמקסימום הנאה |
| **אנשי הייטק** | * העברה שוטפת של מידע * הודעות טקסט * שמירה על בטיחות המידע * שיחות וידאו | * חיי סוללה ארוכים * מעבד חזק * ביצועי מערכת חזקים * נפח זיכרון גדול * מצלמות איכותיות (ראשית ומשנית) |
| **פנסיונרים** | * שיחות טלפון * הודעות טקסט * צילום תמונות | * עוצמת סאונד חזקה * איכות צליל * ביצועי מערכת חזקים * תפעול פשוט, מהיר וחלק * איכות אור התצוגה הכחול |

**קריטריונים לבדיקה ודירוגם**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **קריטריון** | **ילדים (8-12)** | **אנשי הייטק** | **פנסיונרים** |
| **1** | סוללה | 5 | 5 | 5 |
| **2** | שנת השקה | 4 | 5 | 2 |
| **3** | גובה | 3 | 3 | 4 |
| **4** | רוחב | 3 | 3 | 4 |
| **5** | משקל | 4 | 4 | 4 |
| **6** | מספר כרטיסי סים | 1 | 4 | 1 |
| **7** | סוג כרטיס סים | 1 | 3 | 1 |
| **8** | גודל מסך | 5 | 4 | 5 |
| **9** | רזולוציה | 5 | 5 | 3 |
| **10** | מערכת הפעלה | 5 | 4 | 1 |
| **11** | מעבד | 4 | 4 | 1 |
| **12** | מיקום לכרטיס זיכרון | 2 | 5 | 5 |
| **13** | גודל זיכרון חיצוני מירבי | 4 | 5 | 5 |
| **14** | זיכרון פנימי RAM | 4 | 5 | 1 |
| **15** | איכות מצלמה ראשית | 4 | 5 | 4 |
| **16** | וידאו מצלמה ראשית | 4 | 5 | 4 |
| **17** | איכות מצלמה משנית | 4 | 5 | 4 |
| **18** | וידאו מצלמה משנית | 4 | 5 | 4 |
| **19** | משדר אינפרא אדום | 2 | 2 | 4 |
| **20** | רדיו | 1 | 1 | 5 |
| **21** | סוג USB | 1 | 3 | 4 |
| **22** | NFC | 1 | 3 | 2 |
| **23** | טביעת אצבע | 2 | 4 | 1 |
| **24** | מחיר | 5 | 5 | 5 |
| **25** | בדיקת ביצועים basmark | 3 | 4 | 1 |
| **26** | עצמת רמקול | 5 | 1 | 5 |
| **27** | איכות שמע | 5 | 2 | 5 |
| **28** | זמן עבודה מירבי | 5 | 5 | 5 |
| **29** | חסינות למים | 4 | 4 | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **קריטריון** | **ילדים (8-12)** | **אנשי הייטק** | **פנסיונרים** |
| **1** | סוללה | ככל שיותר חזקה – יותר טוב | | |
| **2** | שנת השקה | ככל שיותר חדיש – יותר טוב | | |
| **3** | גובה | 150 - 140 mm – 5 | 155 - 145 mm - 5 | 180 - 170 mm - 5 |
| 150 - 155 / 140 - 135 mm - 4 | 155 - 160 / 145 - 140 mm - 4 | 180 - 185 / 170 - 165 mm – 4 |
| 155 - 160 / 135 - 130 mm - 3 | 160 - 165 / 140 - 135 mm - 3 | 185 - 190 / 165 - 160 mm - 3 |
| 160 < / 130 - 125 mm - 2 | 165 < / 135 - 130 mm - 2 | 190 < / 160 - 155 mm - 2 |
| 125 > mm - 1 | 130 > mm - 1 | 155 > mm - 1 |
| **4** | רוחב | 70 - 60 mm - 5 | 75 - 65 mm - 5 | 90 - 80 mm - 5 |
| 70 - 80 / 60 - 55 mm - 4 | 75 - 80 / 65 - 60 mm - 4 | 90 - 95 / 80 - 75 mm - 4 |
| 75 - 85 / 55 - 50 mm - 3 | 80 - 85 / 60 - 55 mm - 3 | 95 - 100 / 75 - 70 mm - 3 |
| 80 < / 50 - 45 mm - 2 | 85 < / 55 - 50 mm - 2 | 100 < / 70 - 65 mm - 2 |
| 45 > mm - 1 | 50 > mm - 1 | 65 > mm - 1 |
| **5** | משקל | ככל שיותר קל – יותר טוב | | |
| **6** | מספר כרטיסי סים | לא משנה | ככל שיותר גדול – יותר טוב | לא משנה |
| **7** | סוג כרטיס סים | לא משנה | nano - 4  micro - 3  mini - 2  Full - 1 | Full - 4  mini - 3  micro - 2  nano - 1 |
| **8** | גודל מסך | 4.5 - 5 inch - 5 | 5 - 5.5 inch - 5 | ככל שיותר גדול – יותר טוב |
| 4 - 4.5 / 5 - 5.5 inch - 4 | 4.5 -5 / 5.5 - 6 inch - 4 |
| 3.5 - 4 / 5.5 - 6 inch -3 | 4 - 4.5 / 6 - 6.5 inch -3 |
| 3 - 3.5 / 6 < inch - 2 | 3.5 - 4 / 6.5 < inch - 2 |
| 3 > inch - 1 | 3.5 > inch - 1 |
| **9** | רזולוציה | ככל שיותר גדול - יותר טוב | | |
| **10** | מערכת הפעלה | ככל שיותר חדישה - יותר טובה | | |
| **11** | מעבד | ככל שיותר גדול – יותר טוב | | |
| **12** | מיקום לכרטיס זיכרון | לא משנה | יש – 1  אין – 0 | יש – 1  אין – 0 |
| **13** | גודל זיכרון חיצוני מירבי | ככל שיותר גדול – יותר טוב | | |
| **14** | זיכרון פנימי RAM | ככל שיותר גדול – יותר טוב | | |
| **15** | איכות מצלמה ראשית | ככל שיותר גדול – יותר טוב | | |
| **16** | וידאו מצלמה ראשית | ככל שיותר גדול – יותר טוב | | |
| **17** | איכות מצלמה משנית | ככל שיותר גדול – יותר טוב | | |
| **18** | וידאו מצלמה משנית | ככל שיותר גדול – יותר טוב | | |
| **19** | משדר אינפרא אדום | לא משנה | יש – 1  אין – 0 | יש – 1  אין – 0 |
| **20** | רדיו | לא משנה | לא משנה | יש – 1  אין – 0 |
| **21** | סוג USB | IPHONE / Type-C - 3  Micro USB - 2  Mini USB - 1 | | |
| **22** | NFC | יש – 1  אין – 0 | יש – 1  אין – 0 | לא משנה |
| **23** | טביעת אצבע | יש – 0  אין – 1 | יש – 1  אין – 0 | יש – 0  אין – 1 |
| **24** | מחיר | ככל שיותר זול – יותר טוב | | |
| **25** | בדיקת ביצועים basmark | ככל שיותר גדול – יותר טוב | ככל שיותר גדול – יותר טוב | לא רלוונטי |
| **26** | עצמת רמקול | ככל שיותר גדול – יותר טוב | לא משנה | ככל שיותר גדול – יותר טוב |
| **27** | איכות שמע | ככל שיותר גדול – יותר טוב | | |
| **28** | זמן עבודה מירבי | ככל שיותר גדול – יותר טוב | | |
| **29** | חסינות למים | יש – 1  אין – 0 | | |

**חלק ב' – אלגוריתמי הבדיקה**

במהלך התמודדות עם הבעיה, בחנו מספר אלגוריתמים ושיטות שיכולים לסייע בפתרונה.

בחלק זה, נציג את האלגוריתמים והשיטות שבאמצעותם יהיה ניתן להציג פתרון יעיל לבעיה, בהתאם לפרמטרים השונים שהצגנו בחלק הקודם.

האלגוריתמים והשיטות הם:

* שיטת BORDA.
* אלגוריתם TOPSIS.
* שיטת שולצה.

כמו כן, נסקור את האלגוריתמים והשיטות הנ"ל על ידי הצגת סקירה כללית, אופן הפעולה, כיצד הם מסייעים בפתרון הבעיה וכן יתרונות וחסרונות.

**שיטת BORDA**

סקירה כללית

* שיטת BORDA היא שיטה לדירוג פרויקטים ומערכות, המבוצע על ידי הקצאת נקודות אשר מתקבלות על פי דעה של אנשי מקצוע. BORDA היא משפחה של שיטות בחירה של מנצח יחיד, שבהן הבוחרים מדרגים אופציות או מועמדים לפי סדר עדיפויות.
* שיטת BORDA קובעת את הזוכה בבחירות על ידי מתן פתק לכל מועמד, מספר נקודות המתאימות למספר המועמדים המדורגים נמוך יותר. לאחר שכל הקולות נספרים, האופציה או המועמד שקיבל את מרבית הנקודות – נבחר כמנצח.
* שיטה זו נועדה לבחור באופציות או במועמדים מקובלים, ולאו דווקא באלה העדיפים על פי הרוב, ולכן היא מתוארת לעיתים קרובות כמערכת הצבעה המבוססת על קונצנזוס ולא על רודנות.

אופן פעולה

* **נקודות BORDA:**
* נתונים פרויקטים וקיימים מומחים
* כל מומחה מדרג את הפרויקטים ונותן נקודות לכל פרויקט
* הפרויקט הטוב ביותר מקבל נקודות
* הפרויקט הגרוע ביותר מקבל 0 נקודות
* הפרויקט הזוכה הוא הפרויקט שמקבל את כמות הנקודות המקסימלית.
* **ספירת BORDA:**
* ספירת BORDA היא ספירה של סכום הנקודות שניתנות לכל פרויקט.
* החישוב מתבצע על יד הנוסחה הבאה:  
  , כאשר:

– מספר המומחים

– מספר הפרויקטים

– נקודות BORDA הניתנות לפרויקט שניתנו על ידי מומחה

הערות

* דירוג BORDA לפרויקט נתון הוא מספר הפרויקטים האחרים שהם קריטיים יותר.
* דירוג BORDA לפרויקט מראה על מספר הפרויקטים שהם טובים יותר מפרויקט זה.

שימוש באלגוריתם

על מנת לתת מענה לבעיה שהצגנו, נשתמש בשיטת BORDA באופן הבא:

* כל אחד מאיתנו ידרג את כל הסמארטפונים במסד הנתונים ויעניק לכל אחד נקודות לפי ראות עיניו בהתאם ל-28 הקריטריונים שמתארים כל סמארטפון.
* לאחר הדירוג נוכל לבצע ספירת BORDA לפי הנוסחה הנ"ל.
* הסמארטפון הנבחר הוא זה שיקבל את כמות הנקודות המקסימלית לאחר ביצוע ספירת BORDA.
* פעולה זו תבטיח את מציאת הסמארטפון הטוב ביותר בהינתן הקריטריונים שבחרנו ובהינתן הדירוג שביצענו.

יתרונות

* שיטה פשוטה להשוואה בין מערכות רבות ובין קריטריונים רבים.
* החישובים בשיטה פשוטים לביצוע.
* הבוחרים מסוגלים להביע מערכת העדפות מפורטת.

חסרונות

* אין התחשבות ביעילות ובאמינות (דיוק) החישוב.
* המערכת דורשת לפחות רמה מסוימת של נומריות כדי לעבוד. ייתכן שלבוחרים יהיה קשה להבין.
* רמת המידתיות ומספר הקולות המבוזבזים תלויים במידה רבה בגודל המחוזות, כלומר בכמות הבוחרים.

**אלגוריתם TOPSIS**

סקירה כללית

* אלגוריתם TOPSIS הינו שיטה של צירוף פיצוי אשר משווה סדרה של חלופות על ידי זיהוי משקל עבור כל קריטריון, נרמול התוצאות/הציונים עבור כל קריטריון וחישוב המרחק הגיאומטרי (אוקלידי) בין כל חלופה לבין האלטרנטיבה האידיאלית, שהיא התוצאה הטובה ביותר עבור כל קריטריון.
* ההנחה של TOPSIS היא שהקריטריונים עולים או יורדים באופן מונוטוני. הנורמליזציה נדרשת בדרך כלל כאשר הפרמטרים או הקריטריונים לעיתים קרובות במידות אשר לא מתאימות לבעיות של קריטריונים מרובים.
* שיטות פיצוי כמו TOPSIS מאפשרות תמורה בין קריטריונים, כאשר תוצאה גרועה בקריטריון אחד יכולה להישלל על ידי תוצאה טובה בקריטריון אחר.
* אופן פעולה זה מספק צורה מציאותית יותר של מודלים, לעומת שיטות שאינן מפצות, אשר כוללים / לא כוללים פתרונות חלופיים המבוססים על חתכים קשים.

אופן פעולה

1. יצירת מטריצה הערכות המכילה מ- אלטרנטיבות ו-קריטריונים,  
   עם חיתוך של כל אלטרנטיבה עם קריטריון המוגדר כ-*.*

*לפיכך, תתקבל מטריצה* .

1. נרמול המטריצה וקבלת מטריצת ההחלטה בעזרת הנוסחה .
2. חישוב את מטריצת ההחלטה המשוקללת ע"י הכפלה של כל תא במטריצה במשקל של כל קריטריון.
3. דירוג הערך הטוב ביותר והערך הגרוע ביותר עבור כל עמודה:

* - הערך **המקסימלי** בכל עמודה
* - הערך **המינימלי** בכל עמודה

1. חישוב המרחק הגיאומטרי (אוקלידי) בין כל קריטריון לבין כל חלופה על ידי הנוסחאות ; .
2. חישוב הדימיון למצב הגרוע ביותר על ידי הנוסחה .
3. דירוג סופי לפי הערך שהתקבל ב- מהגבוה לנמוך.

שימוש באלגוריתם

על מנת לתת מענה לבעיה שהצגנו, נשתמש בשיטת TOPSIS באופן הבא:

* ניצור מטריצת הערכות עבור כל הסמארטפונים במסד הנתונים לפי 28 הקריטריונים המתארים כל אחד.
* ננרמל את המטריצה ונקבל מטריצת החלטה, אותה נשקלל ונקבל מטריצת החלטה משוקללת.
* נמצא את הערך המינימלי ואת הערך המקסימלי בכל עמודה, ונחשב את המרחק האוקלידי בין כל קריטריון לבין כל סמארטפון וכן את הדימיון למצב הגרוע ביותר.
* לבסוף, נדרג באופן סופי את הערכים מהגבוה לנמוך ונבחר בערך הגבוה ביותר.
* פעולה זו תבטיח את מציאת הטלפון הכי טוב בהינתן הקריטריונים שבחרנו.

תרשים מסכם:

יתרונות

* פשוט, רציונלי ובעל קונספט קל להבנה.
* אינטואיטיבי ובעל לוגיקה ברורה שמייצגת את הרציונל מאחורי הבחירה האנושית.
* קל לחישוב ויעיל.

חסרונות

* כאמור, השיטה מחשבת מרחק על פי הנוסחה של מרחק אוקלידי. משמעות הדבר היא שלא לוקחים בחשבון את הקורלציה בין הקריטריונים.
* באופן יחסי, חישוב המשקל קשה.
* קשה לשמור על עקביות בשיפוט.

**שיטת שולצה**

סקירה כללית

* שיטת שולצה היא שיטת הצבעה אשר משקללת עמדות של מצביעים לגבי כמה מועמדים לכדי סדר עדיפויות משותף.
* בבסיס השיטה מונחת השוואה בין כל שני מועמדים (לצורך העניין נקרא להם מועמד א' ומועמד ב') ובחינה של מספר המעדיפים את מתמודד א' על פניו של מתמודד ב', ולהיפך. שיטה זו מקיימת את תנאי קונדורוסה, לפיו אם יש מועמד המנצח כל מועמד אחר בהתמודדות ראש בראש מולו, אז הוא המועמד הזוכה.
* אם לגבי מועמד א' לכל מועמד אחר ב' רוב הקבוצה מעדיפה את א' על ב', אז א' צריך לזכות בבחירות. אם אכן מכתירים את המועמד בעל תוצאה עקבית שכזאת, שיטת שולצה מאמצת את התוצאה הזאת ללא שום שינוי.
* עם זאת, התחכום בשיטה מתגלה כשהשוואת רוב פשוטה מביאה לתוצאות מעגליות, כגון העדפה של אפשרות א' על ב', ב' על ג' ו-ג' על א'. במקרים כאלה, הידועים כפרדוקס קונדורסה, שיטת שולץ פותרת את הקונפליקט באמצעות השוואת עצומת ההעדפה לאורך מסלולים שונים בגרף המועמדים.
* כמו בשיטות אחרות המשקללות סדרי עדיפויות, גם בשיטת שולצה רשאי כל מצביע לדרג את המועמדים, לפי סדר העדפותיו. המצביע אינו חייב לדרג את כל המועמדים. כל שנדרש ממנו הוא הצבעה עקבית (בלשון מתמטית, קביעת יחס סדר חלקי בין המועמדים).

אופן פעולה

1. מסמנים ב- את מספר הבוחרים שהעדיפו את מועמד על פני מועמד .
2. מסמנים
3. בונים גרף מכוון משוקלל, שקודקודיו הם המועמדים ועל הקשת ממועמד למועמד כתוב את הערך .
4. העוצמה של מסלול בגרף היא הערך הקטן ביותר המופיע על הקשתות במסלול. אם מסמנים ב-את העוצמה הגדולה ביותר על המסלולים מ-A ל-B, נקבל שהיחס הוא הוא יחס סדר חזק. בנוסף, מכיוון שמספר האפשרויות הוא סופי, יש בו איבר מקסימלי. איבר (מועמד) כזה הוא הזוכה בהצבעה לפי שיטת שולצה.

הערה: כדי למצוא את המסלולים הכבדים ביותר, ניתן להשתמש באלגוריתם פלויד-ווארשל למציאת המסלולים הקצרים ביותר בין כל שני זוגות צמתים.

שימוש בשיטה

על מנת לתת מענה לבעיה שהצגנו, נשתמש בשיטת שולצה באופן הבא:

* נדרג את הסמארטפונים על מנת לסווג אותם מהגרוע אל הטוב ביותר.
* נבנה גרף מכוון משוקלל בהתאם לדירוגים הנ"ל.
* נפעיל את אלגוריתם פלויד-ווארשל על הגרף שהתקבל כדי למצוא את המסלול הכבד ביותר.

יתרונות

* עם קולות המחוז שלהם, הבוחרים לא רק בוחרים את הנציגים של המחוז שלהם, אלא גם מחליטים על נציגים נוספים.
* קלה לקידוד.

חסרונות

* בבחירת המנצח בשיטה זו, סט של זוגות עשוי להיות גדול וחישוב של רכיבי קשירות חזק בגרף עושה שימוש רק בחלק מהמידע הנתון. כלומר, הגרף דומיננטי ובכך מתעלם ממשקל הקצוות).
* כאמור, בשיטה זו משתמשים באלגוריתם פלויד-ווארשל ויעילותו היא , כאשר הוא מספר הצמתים בגרף.

**השוואה בין האלגוריתמים ובחירת האלגוריתם המתאים**

|  |  |  |  |
| --- | --- | --- | --- |
| **היבט** | **BORDA** | **TOPSIS** | **שולצה** |
| **עבודה מקדימה** | כן.  מתן משקל לקריטריונים | כן.  יש ליצור מטריצת הערכות ולנרמל אותה טרם תחילת העבודה | כן.  בניית גרף מכוון ומשוקלל בהתאם לתוצאות ההצבעה |
| **סיבוכיות** | לא פולינומיאלי | פולינומיאלי | פולינומיאלי |
| **מבנה נתונים** | מטריצה | מטריצה | גרף מכוון ממושקל |
| **חישוב מרחק** | --- | מרחק אוקלידי | --- |
| **חשיבות למשקל** | אין | יש | יש |

על מנת לבחור את האלגוריתם המתאים, נבחן מספר היבטים:

* קלות קידוד האלגוריתם.
* סיבוכיות זמן ריצה.
* יעילות האלגוריתם בהתאם לבעיה.

קלות קידוד האלגוריתם

שיטת שולצה עושה שימוש בגרפים. אמנם גרף הוא מבנה נתונים מופשט וקל לקידוד, אך במקרה של הבעיה שלנו, הוא איננו עונה על הצרכים שלנו ולכן שיטת שולצה ירדה מן הפרק.

סיבוכיות זמן ריצה

מבחינת סיבוכיות זמן ריצה, שיטת BORDA נותנת פתרון בצורה מהירה יותר כמעט פי שניים יותר מאלגוריתם TOPSIS.

יעילות האלגוריתם

במקרה זה, בהתאם לבעיה הספציפית בה אנחנו דנים, שיטת BORDA תיתן תוצאה פחות טובה מאלגוריתם TOPSIS, מהסיבה שבשיטה זו אין משקל לקריטריונים, כלומר הם שווי משקל. למשל, לבחירת ראש ממשלה, שיטת BORDA תתאים יותר מכיוון שכל קול של אזרח שווה ואין חשיבות למשקל.

מסקנות מההשוואה

בהתאם לצרכים שלנו ובהתאם לפרמטרים וההיבטים השונים, נבחר להשתמש באלגוריתם TOPSIS, שכן אנחנו מחפשים פתרון יעיל ומהיר. בנוסף, בחרנו לממש גם את שיטת BORDA.

**חלק ג' – יישום ומסקנות**

בחלק זה נציג את מימוש הבעיה באמצעות תכנה שנכתבה בשפת פייתון, וכן את המסקנות שהגענו אליהן לאחר קבלת התוצאות הרצויות מריצות שונות של התכנה שפיתחנו.

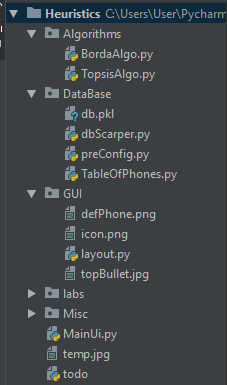
**מימוש האלגוריתמים**

סקירה כללית

* המימוש בוצע בשפת Python 3.7 בסביבת העבודה PyCharm 2019.
* לטובת הממשק הגרפי, עשינו שימוש בספריות wxPython.
* הפרויקט מחולק לשלושה חלקים, והם:

1. בסיס הנתונים.
2. אלגוריתמים לדירוג.
3. ממשק משתמש.

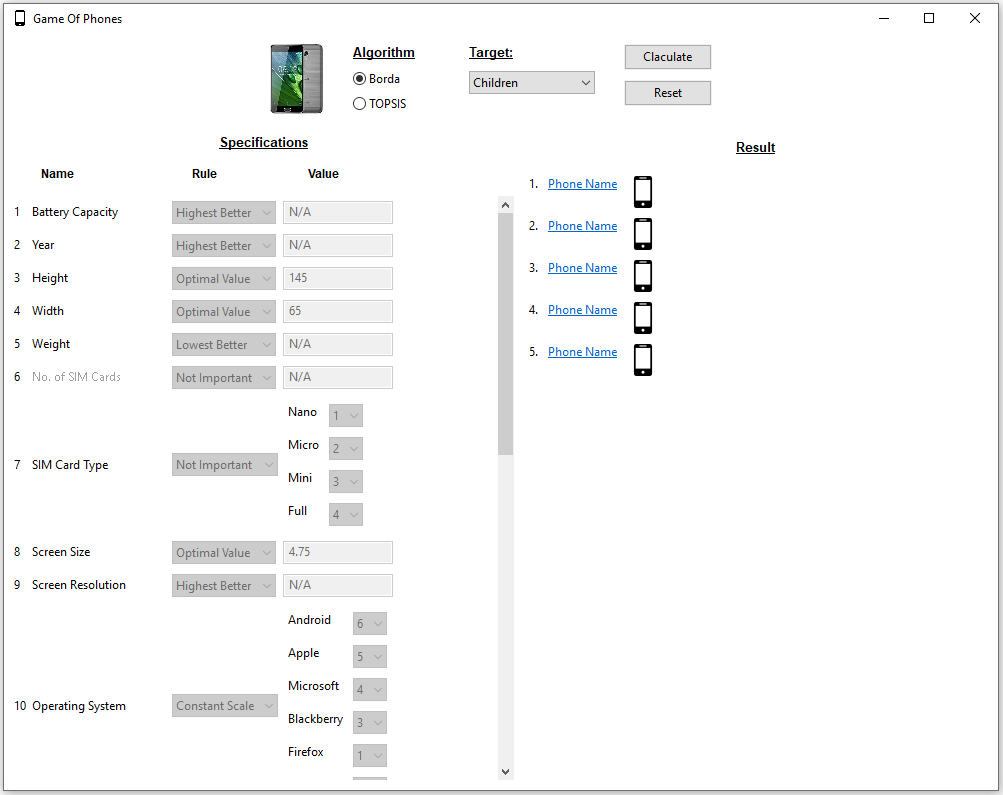
מבנה החלקים בפרויקט



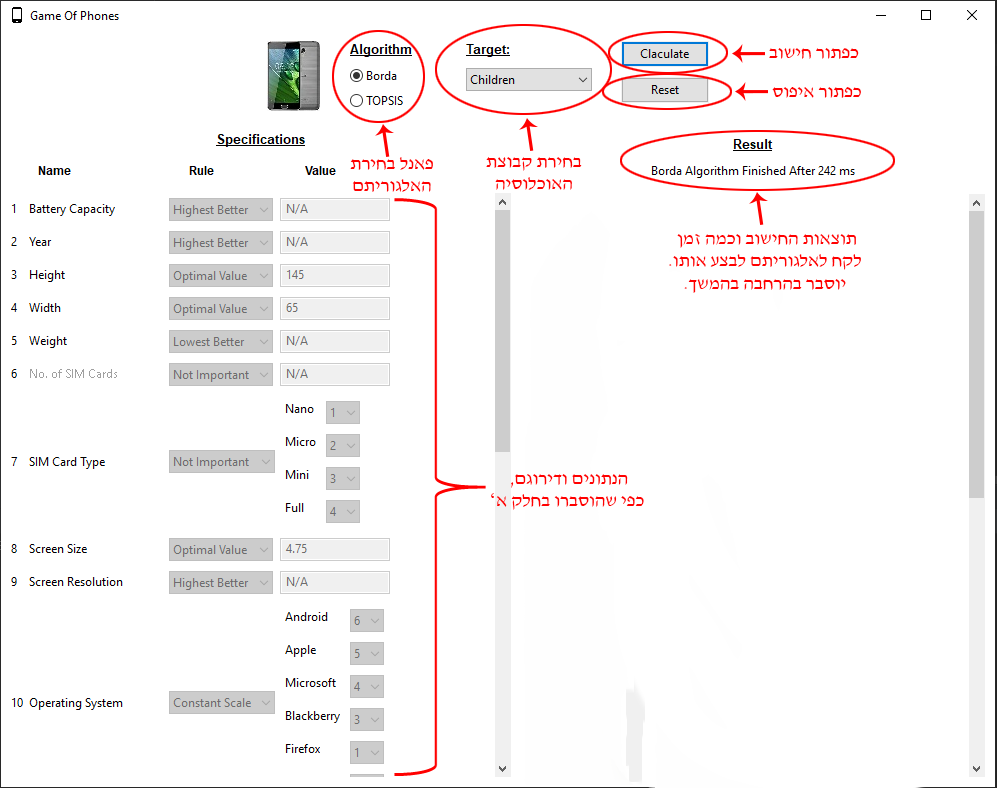
1. בסיס הנתונים:
   1. **dbScraper** - סקריפט אשר ניגש לאתר gsmarena.com ומבצע איסוף שיטתי של הספסיפיקציות של כל הטלפונים אשר קיימת באתר, בחלוקה ליצרנים ודגמים.
   2. **TableOfPhones** - מחלקה אשר מקבלת את בסיס הנתונים וההגדרות של הקריטריונים לפי קבוצת היעד ובונה טבלה, בה כל טלפון מקבל ציון לכל קריטריון שלו על פי הכללים שהוגדרו לקבוצת היעד הרלוונטית.
   3. **preConfig** - מילון אשר מכיל את ההגדרות ברירת מחדל של קבוצות היעד שאיתן עבדנו (ילדים, עובדי הייטק, פנסיונרים).
   4. **db** - בסיס הנתונים עשר נוצר ע"י הרצת הסקריפט dbScraper איתו אנו עובדים באפליקציה.
2. אלגוריתמים לדירוג:
   1. **BordaAlgo** - מחלקה אשר מקבלת טבלה, בה כל טלפון מקבל ציון לכל קריטריון שלו על פי הכללים שהוגדרו לקבוצת היעד הרלוונטית ומפעילה עליה את שיטת BORDA. המחלקה מחזירה מערך של טלפונים כאשר לכל טלפון ציון BORDA.
   2. **TopsisAlgo** - מחלקה אשר מקבלת טבלה, בה כל טלפון מקבל ציון לכל קריטריון שלו על פי הכללים ומערך משקלים לכל קריטריון שהוגדרו לקבוצת היעד הרלוונטית ומפעילה את אלגוריתם TOPSIS. המחלקה מחזירה מערך של טלפונים כאשר לכל טלפון ציון על פי האלגוריתם.
3. ממשק משתמש:
   1. **layout** - מחלקה אשר מגדירה ויזואלית את המסך הראשי של האפליקציה.
   2. **MainUi** - הקוד הראשי של האפליקציה, מוגדרות בו EventHanler לממשק משתמש וההפעלה הראשית של האפליקציה.

עיקרון הפעולה של התכנה

1. כאשר האפליקציה מופעלת לראשונה, נטען בסיס הנתונים אשר הוכן מראש (כאמור, דרך הסקריפט dbScraper ששואב את הנתונים הרלוונטיים מהאתר gsmarena.com.
2. על המשתמש לבחור את האלגוריתם שאיתו ירצה להשתמש על מנת לבצע את הדירוג. אם יבחר בTOPSIS, תתווסף עמודה של משקלים למסך.
3. המשתמש יבחר את אוכלוסיית היעד אליה הוא רוצה להתאים טלפון, בהתאם לאוכלוסייה יטענו ההגדרות והכללים למתן ציון לכל קריטריון.
4. ניתן להגדיר את הכללים למתן ציון לכל קריטריון בצורה ידנית ומותאמת אישית.
5. כעת ניתן להריץ את האלגוריתם.
6. בסיום הריצה יופיע במסך התוצאה הזמן שלקח לאלגוריתם לרוץ (זמן זה לא כולל את זמן המידע מהבסיס נתונים והזמן להצגת התוצאה).
7. התוצאה היא חמשת הטלפונים שדורגו במקומות הראשונים, בתוספת תמונה להמחשה של כל טלפון וקישור לעמוד של כל טלפון באתר gsmarena.com.



מסך הפתיחה של התכנה עם ערכי ברירת המחדל

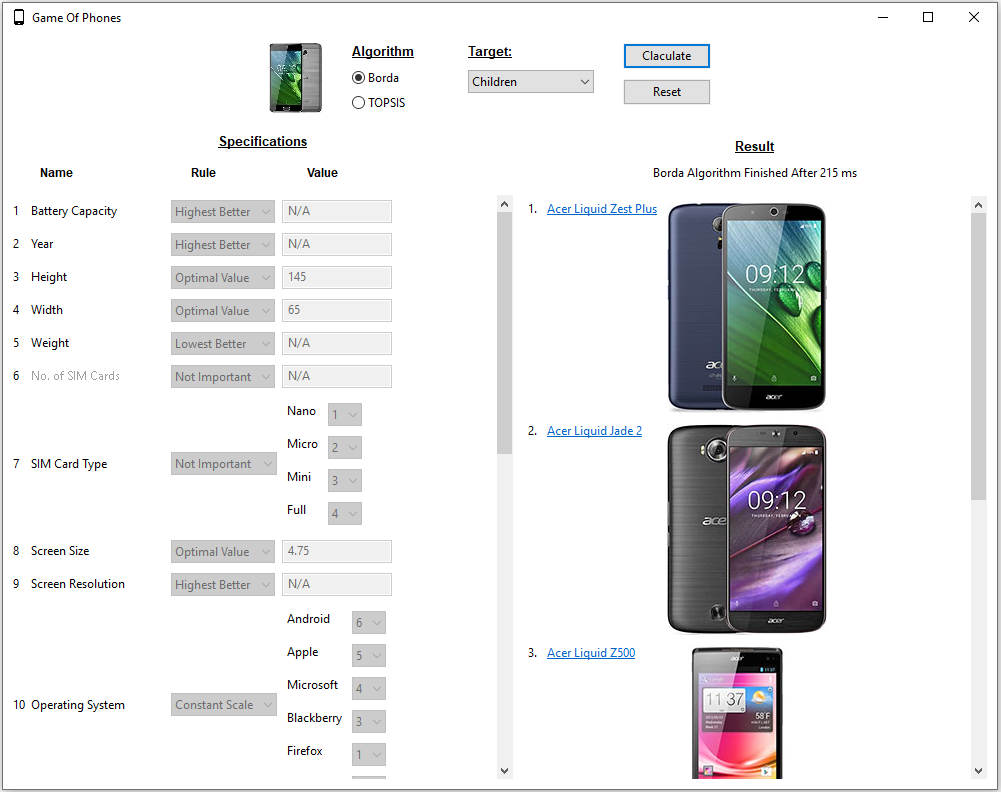


הסבר כללי על חלקי התכנה

**סימולציות**

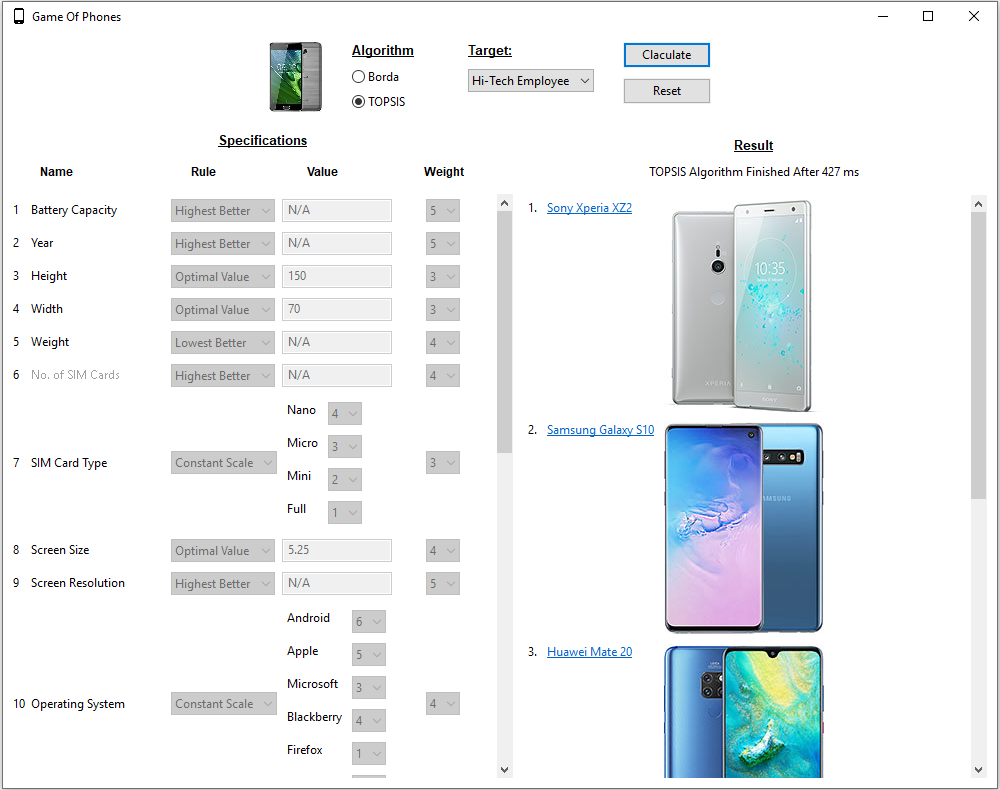
כעת נציג מספר סימולציות שונות של ריצת התכנה ונסביר אותן, על מנת להראות ולהדגים את פעולת התכנה.

סימולציה 1



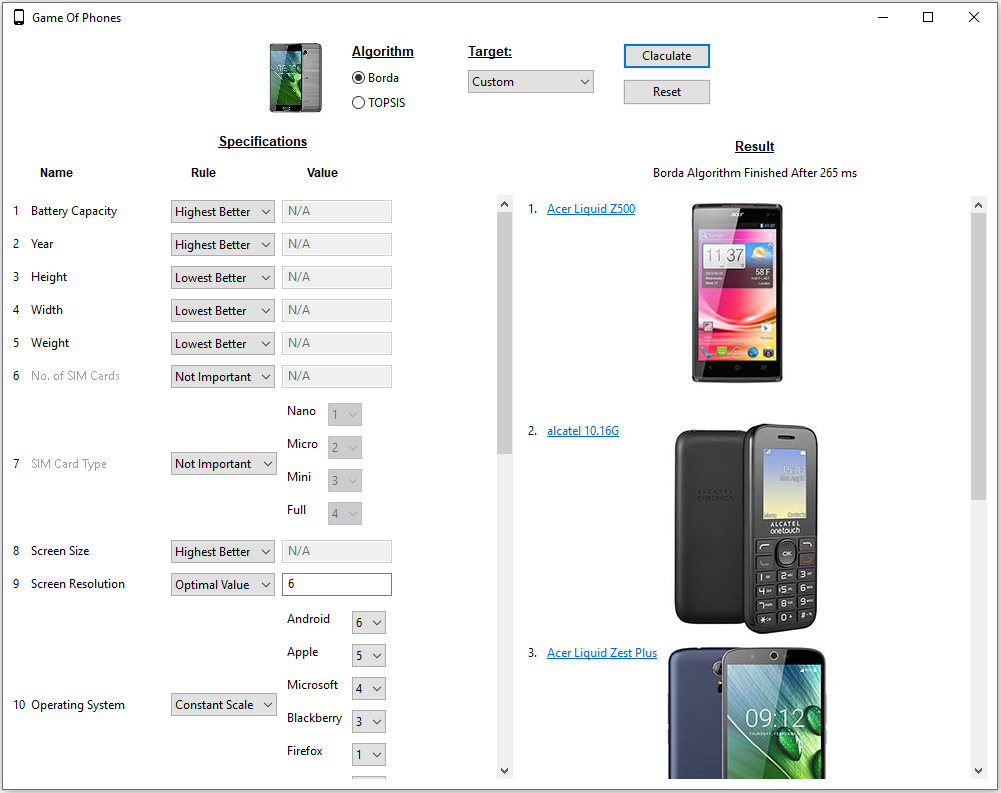
בסימולציה זו, הרצנו את התכנה על קבוצת האוכלוסייה ילדים (8-12) וקיבלנו את התוצאות באמצעות שיטת BORDA. ניתן לראות כי מרגע תחילת חישוב ועד הצגת התוצאות, האלגוריתם פעל 215 מילי-שניות. בנוסף, מוצגים חמשת הפלאפונים שקיבלו את הדירוג הכי גבוה מבין כולם עם אפשרות לכניסה לאתר gsmarena.com שמציג את כל המידע הרלוונטי על הפלאפון הנבחר.

סימולציה 2



בסימולציה זו, הרצנו את התכנה על קבוצת האוכלוסייה אנשי הייטק וקיבלנו את התוצאות באמצעות אלגוריתם TOPSIS. במקרה זה, בגלל השימוש ב-TOPSIS, נוספה גם עמודה של משקלים לכל קריטריון. ניתן לראות כי מרגע תחילת חישוב ועד הצגת התוצאות, האלגוריתם פעל 427 מילי-שניות (מעט יותר מ-BORDA). גם כאן, מוצגים חמשת הפלאפונים שקיבלו את הדירוג הכי גבוה מבין כולם עם אפשרות לכניסה לאתר gsmarena.com שמציג את כל המידע הרלוונטי על הפלאפון הנבחר.

סימולציה 3



בסימולציה זו, התאמנו בצורה ידנית ומותאמת אישית את הקריטריונים (כלומר, אין שייכות לאף אחת מקבוצות האוכלוסייה שבדקנו) וקיבלנו את התוצאות באמצעות שיטת BORDA. ניתן לראות כי מרגע תחילת חישוב ועד הצגת התוצאות, האלגוריתם פעל 265 מילי-שניות. גם כאן, מוצגים חמשת הפלאפונים שקיבלו את הדירוג הכי גבוה מבין כולם עם אפשרות לכניסה לאתר gsmarena.com שמציג את כל המידע הרלוונטי על הפלאפון הנבחר.

**מסקנות מהחישוב**

לסיכום, נבחן את ביצועי התכנה ונסיק מסקנות מהחישובים שהתבצעו מבחינת כמה היבטים שונים.

1. **מה דומה ומה שונה בין קבוצות האוכלוסייה (באופן חלקי)?**באופן יחסי, נקודות הדימיון בין קבוצות האוכלוסייה הנתונות הן הדברים שכל משתמש סמארטפון באשר הוא מחפש, ללא השתייכות לאחת מקבוצות האוכלוסייה שבדקנו. למשל, כל אחד רוצה סוללה טובה שתחזיק מעמד כמה שיותר זמן כי לא תמיד אנחנו נמצאים ליד נקודת חשמל קרובה. בנוסף, כל אחד רוצה את המכשיר החדיש ביותר, שכן, סמארטפונים שהושקו רק לפני מספר חודשים כבר לא מכילים את אותם פיצ'רים שסמארטפונים שיצאו לאחרונה מכילים.   
   לעומת זאת, ישנם מספר קריטריונים שמבדילים בין קבוצות האוכלוסייה, מכיוון שקריטריון X לא בהכרח רלוונטי לקבוצה אחת אבל כן רלוונטי לקבוצה שנייה, ובניגוד לכך, קריטריון Y כן מתאים לקבוצה אחת ולא רלוונטי לקבוצה השנייה.

לדוגמה, מספר כרטיסי SIM: הקריטריון רלוונטי אך ורק לקבוצת אנשי ההייטק, כיוון שהם אנשים שמשתמשים רבות בסמארטפון שלהם ולעיתים נדרשים לבצע שיחות רבות. לעומת זאת, הקריטריון רדיו חשוב יותר לפנסיונרים ואילו לקבוצות האחרות לא חשוב כלל.

לסיכום, באופן חלקי ויחסי, לשלושת הקבוצות יש מספר מאפיינים משותפים אבל גם מספר מאפיינים שונים לחלוטין, מה שמחלק את האוכלוסייה לשלוש קבוצות.

1. **מה התקבל כתוצאה משימוש בשיטת BORDA ומה כתוצאה משימוש באלגוריתם TOPSIS?**כאמור, באלגוריתם TOPSIS יש חשיבות לכל קריטריון ולכן נותנים לכל אחד משקל בהתאם לחשיבותו. לכן, השימוש באלגוריתם הניב תוצאות שיותר קרובות למציאות ולמה שמצפים ביחס לאותה קבוצת אוכלוסייה. אמנם בבעיה זאת BORDA הייתה פחות מתאימה כי המשקל של קריטריון היה שווה (בחיים האמיתיים כמובן שההפך הוא הנכון – בכל קבוצת אוכלוסין יש לתת לכל קריטריון משקל רלוונטי), אבל עדיין קיבלנו תוצאות שקרובות למציאות ולמה שציפינו שיצא.
2. **מדוע אלגוריתם TOPSIS עדיף על שיטת BORDA?**

כאשר עשינו שימוש בשיטת BORDA, לאחר כמה וכמה ריצות על קבוצות האוכלוסייה השונות וקלט בהתאמה אישית, וכאשר עשינו שימוש באלגוריתם TOPSIS על אותן קבוצות אוכלוסייה ואותם קלטים בהתאמה אישית, ניתן להסיק כי BORDA מהיר יותר בכמעט פי שניים (אם כי שניהם מסיימים את החישוב בפחות משנייה בממוצע) אבל אינה נותנת תוצאה טובה כפי שאנחנו מחפשים מכיוון שאין משקל לקריטריונים, כלומר לכל קריטריון משקל שווה וזה פחות טוב לבעיה הנוכחית. לעומת זאת, כמובן TOPSIS מעט איטי יותר, אבל בגלל שיש משקל לכל קריטריונים הוא נותן תוצאה מדויקת ואמיתית יותר; לכן TOPSIS עדיף.

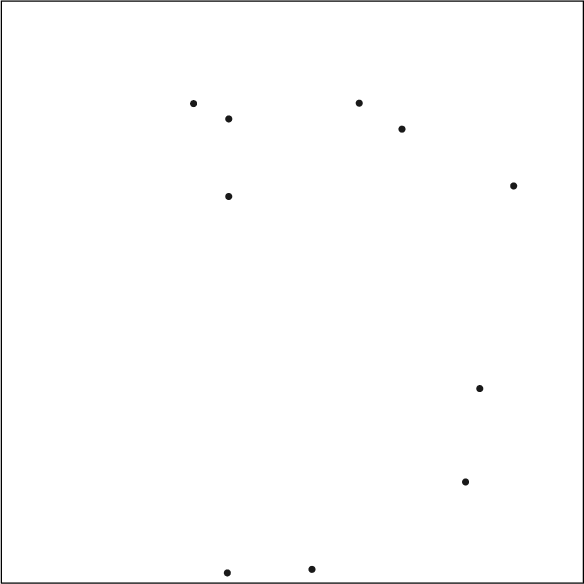
**ציפייה מהחישוב לעומת המציאות**

בהתאם לפרמטרים שבדקנו בסקר המוקדם ולקריטריונים שהצבנו להשוואה, הציפייה שלנו הייתה ששני האלגוריתמים יפיקו תוצאות אידיאליות וידרגו את הסמארטפונים בצורה הטובה והנכונה ביותר, כדי שיתאימו לכל אדם המשתייך לקבוצת האוכלוסייה שלו את הסמארטפון המתאים ביותר.

במציאות, לאחר הרצות של כמה סימולציות שונות ובחינת התוצאות שלהן לעומק בעזרת הקריטריונים הנבדקים עבור אותה קבוצת אוכלוסייה, הסקנו שהאלגוריתמים אכן נתנו את התוצאות המתאימות ביותר ולכן המציאות עמדה על הציפיות שלנו.

**נספח – תרגילי בית**

**תרגיל 1**



9

8

7

6

5

4

3

2

1

**N = 10**

**תרגיל 2**



נמצא את המסלולים הקצרים ביותר בגרף באמצעות אלגוריתם Dijkstra:

**~~Null~~ ~~A~~ C  
 ~~6~~ 5**

**~~Null~~ x  
 1**



**~~Null~~ ~~E~~ ~~D~~ B  
 ~~11~~ ~~9~~ 6**

**~~Null~~ X  
 3**

**~~Null~~ E  
 4**

**~~Null~~ X  
 3**

**Null  
0**

לכן, כל המסלולים הקצרים ביותר בגרף הם:

כעת, נמצא את המסלולים הקצרים ביותר באמצעות אלגוריתם \*A:

ראשית, נעניק את ההיוריסטיקה לכל צומת בגרף:  


**7**

**4**

**3**

**2**

**6**

**4**

**1**

נמצא את המסלולים הקצרים ביותר על פי הנוסחה , כאשר  
 – המרחק בין שני צמתים ו-– ההיוריסטיקה של הצומת:

מ-X ל-B המרחק הוא לפי החישוב הבא:

X

A

מ-X ל-D המרחק הוא לפי החישוב הבא:

X

E

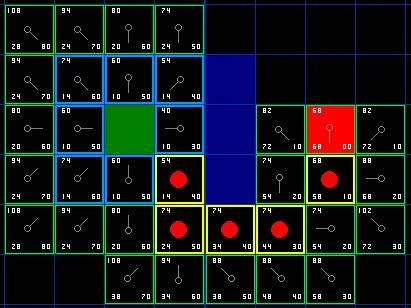
מ-X ל-Y המרחק הוא לפי החישוב הבא:

X

A

B

**תרגיל 3**



הסבר:

* המסלול חושב ע"י הנוסחה , כאשר:
  + – מצוין בכל ריבוע בצד שמאל למטה ומגדיר את המרחק מנקודת ההתחלה.
  + – מצוין בכל ריבוע בד ימין למטה ומגדיר הערכת מרחק בין הצומת הנוכחי לבין צומת היעד.
* תזוזה ימינה/שמאלה/למטה/למעלה = 10.
* תזוזה באלכסון = 14.
* במציאת פונקציה היוריסטיקה לא ניתן לנוע באלכסון.
* המסלול הוא הריבועים שתחומים בצהוב ובעלי עיגול אדום בתוכם.

**תרגיל 4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **מספר פרויקט** | **עוצמה** |  | **הסתברות** |  | **Borda Count**  **///**  **Borda Rank** |
| 1 | C | 1 | 0.8 |  |  |
| 2 | S | 2 | 1 |  |  |
| 3 | Mo | 3 | 0.3 |  |  |
| 4 | Mi | 4 | 0.6 |  |  |
| 5 | N | 5 | 0.4 |  |  |

נדרג באופן שרירותי את :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **מספר פרויקט** | **עוצמה** |  | **הסתברות** |  | **Borda Count**  **///**  **Borda Rank** |
| 1 | C | 1 | 0.8 | 3 |  |
| 2 | S | 2 | 1 | 1 |  |
| 3 | Mo | 3 | 0.3 | 2 |  |
| 4 | Mi | 4 | 0.6 | 4 |  |
| 5 | N | 5 | 0.4 | 5 |  |

נחשב את Borda Count ואת Borda Rank:

לכן, הטבלה הסופית שנקבל היא:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **מספר פרויקט** | **עוצמה** |  | **הסתברות** |  | **Borda Count**  **///**  **Borda Rank** |
| 1 | C | 1 | 0.8 | 3 | 1 /// 6 |
| 2 | S | 2 | 1 | 1 | 0 /// 7 |
| 3 | Mo | 3 | 0.3 | 2 | 2 /// 5 |
| 4 | Mi | 4 | 0.6 | 4 | 3 /// 2 |
| 5 | N | 5 | 0.4 | 5 | 4 /// 0 |

**תרגיל 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Weight** | **0.1** | **0.4** | **0.3** | **0.2** |
|  | **Style** | **Reliability** | **Fuel eco** | **Cost** |
| **Honda** | 5 | 3 | 9 | 8 |
| **Ford** | 8 | 7 | 8 | 6 |
| **Mazda** | 9 | 6 | 2 | 9 |
| **Subaru** | 6 | 7 | 8 | 6 |

נגדיר:

* – דגמי הרכבים.
* – קריטריונים.
* – ניקוד של אופציה עבור קריטריון .

נחשב את :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Style** | **Reliability** | **Fuel eco** | **Cost** |
| **Honda** | 25 | 9 | 81 | 64 |
| **Ford** | 64 | 49 | 64 | 36 |
| **Mazda** | 81 | 36 | 4 | 81 |
| **Subaru** | 36 | 49 | 64 | 36 |
|  | 206 | 143 | 213 | 217 |
|  | 14.35 | 11.95 | 14.59 | 14.73 |

נחלק ב- על מנת לקבל את לפי הנוסחה :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Style** | **Reliability** | **Fuel eco** | **Cost** |
| **Honda** | 0.35 | 0.25 | 0.62 | 0.54 |
| **Ford** | 0.55 | 0.58 | 0.55 | 0.41 |
| **Mazda** | 0.63 | 0.5 | 0.14 | 0.61 |
| **Subaru** | 0.42 | 0.58 | 0.55 | 0.41 |

נכפיל כל עמודה במשקל הנתון על מנת לקבל את :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Style** | **Reliability** | **Fuel eco** | **Cost** |
| **Honda** | 0.035 | 0.1 | 0.186 | 0.108 |
| **Ford** | 0.055 | 0.232 | 0.165 | 0.082 |
| **Mazda** | 0.063 | 0.2 | 0.042 | 0.122 |
| **Subaru** | 0.042 | 0.232 | 0.165 | 0.082 |

נקבע את הפתרון האידיאלי ואת הפתרון האידיאלי השלילי :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Style** | **Reliability** | **Fuel eco** | **Cost** |
| **Honda** | 0.035 | 0.1 | 0.186 | 0.108 |
| **Ford** | 0.055 | 0.232 | 0.165 | 0.082 |
| **Mazda** | 0.063 | 0.2 | 0.042 | 0.122 |
| **Subaru** | 0.042 | 0.232 | 0.165 | 0.082 |

נחשב את המרחק מהפתרון האידיאלי עבור כל שורה:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Style** | **Reliability** | **Fuel eco** | **Cost** |
| **Honda** | 46/62500 | 0.017424 | 0 | 169/250000 |
| **Ford** | 1/15625 | 0 | 0.020736 | 0 |
| **Mazda** | 0 | 0.0361 | 0.020736 | 1/625 |
| **Subaru** | 0.000441 | 0 | 0.000441 | 0 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Honda** | 0.018884 | 0.137 |
| **Ford** | 101/200000 | 0.022 |
| **Mazda** | 0.058436 | 0.242 |
| **Subaru** | 441/500000 | 0.029 |

נחשב את המרחק מהפתרון האידיאלי השלילי עבור כל שורה:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Style** | **Reliability** | **Fuel eco** | **Cost** |
| **Honda** | 0 | 0 | 0.144 | 169/250000 |
| **Ford** | 0.265225 | 0.017424 | 0.123 | 0 |
| **Mazda** | 49/62500 | 0.01 | 0 | 1/625 |
| **Subaru** | 49/1000000 | 0.17424 | 0.123 | 0 |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Honda** | 0.144676 | 0.380 |
| **Ford** | 0.405649 | 0.637 |
| **Mazda** | 0.012384 | 0.111 |
| **Subaru** | 0.140473 | 0.375 |

נחשב את הקרבה היחסית לפתרון האידיאלית ע"י הנוסחה :

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Honda** | 0.380/0.517 | 0.73 |
| **Ford** | 0.637/0.659 | 0.966 |
| **Mazda** | 0.111/0.353 | 0.315 |
| **Subaru** | 0.375/0.404 | 0.928 |

התוצאה הטובה ביותר היא 0.966. לכן נקבל את הפתרון הסופי:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Weight** | **0.3** | **0.4** | **0.3** |  |
|  | **Style** | **Reliability** | **Fuel eco** | **Score** |
| **Honda** | 5 | 3 | 9 | **5.4** |
| **Ford** | 8 | 7 | 8 | **7.6** |
| **Mazda** | 9 | 6 | 2 | **5.7** |
| **Subaru** | 6 | 7 | 8 | **7.0** |

**תרגיל 6**

כתבנו קטע קוד בפייתון שמממש את בעיית התרמיל (Knapsack problem):

profit\_input = [20, 18, 17, 15, 15, 10, 5, 3, 0, 0]  
weight\_input = [30, 25, 20, 18, 17, 11, 5, 2, 0, 0]  
num\_of\_items = 10  
  
b = [40, 45, 50, 55, 75, 95, 100]  
  
def knapsack(W, n, profit, weight):  
 B = [[0 for i in range(0, W + 1)] for i in range(0, n + 1)]  
 for i in range(1, n + 1):  
 for w in range(1, W + 1):  
 if weight[i-1] <= w:  
 if profit[i-1] + B[i - 1][w - weight[i-1]] > B[i - 1][w]:  
 B[i][w] = profit[i-1] + B[i - 1][w - weight[i-1]]  
 else:  
 B[i][w] = B[i - 1][w]  
 else:  
 B[i][w] = B[i - 1][w]  
  
 for i in range(0, n + 1):  
 print(B[i])  
  
 print("Max profit for b = " + str(W) + " is: " + str(B[n][W]))  
 print()  
  
 with open("Max profit for b = " + str(W) + ".csv", 'w') as f:  
 printout = "slot\\Weight"  
 for w in range(0, W+1):  
 printout += "," + str(w)  
 f.write(printout + "\n")  
  
 for i in range(0, i+1):  
 printout = str(i)  
 for w in range(0, W+1):  
 printout += "," + str(B[i][w])  
 f.write(printout + "\n")  
  
for test in b:  
 knapsack(test, num\_of\_items, profit\_input, weight\_input)

הפלטים עבור , כאשר מצורפים בקובץ ZIP של הפרויקט בקבצי CSV.