**תרגיל 3 - results wizards**

**ת.ז: 316350768, 318185956, 208983189, 208608448.**

שאלה 1:

להלן שני משפטים, המשפט הראשון תחת קטגוריה של טכנולגיה והמשפט השני תחת קטגוריה של ספורט:

sentence A(Technology): "The smartphone has an advanced processor."

sentence B(Sport): "The football team won the championship."

המשפט הבא שייך לקטגוריה של טכנולוגיה אבל נראה כי הוא מסווג לא נכון לקטגוריה של ספורט

"The advanced team used AI to win."

נחשב TF לכל המילים עבור כל המשפטים:

|  |  |  |  |
| --- | --- | --- | --- |
| Sentence C | Sentence B | Sentence A | term |
| 0 | 0 | 0.333 | smartphone |
| 0.2 | 0 | 0.333 | advanced |
| 0 | 0 | 0.333 | processor |
| 0 | 0.25 | 0 | football |
| 0.2 | 0.25 | 0 | team |
| 0.2 | 0.25 | 0 | win |
| 0 | 0.25 | 0 | championship |
| 0.2 | 0 | 0 | AI |
| 0.2 | 0 | 0 | use |

​= [⅓ ,⅓ ,⅓ ,0 ,0 ,0 ,0 ,0 ,0]

= [0,0,0,¼ ,¼ ,¼ ,¼ ,0,0]

= [0,⅕ ,0,0,⅕ ,⅕ ,0,⅕ ,⅕ ]

נניח ש - :

נחשב את הדמיון/המרחק בין המשפט החדש למשפטים עליו המסווג התאמן:

ניתן לראות כי הדמיון למשפט B גדול יותר ולכן לפי החישוב הדמיון לנושא הספורט גדול יותר אך לעומת זאת בפועל המשפט מדבר על נושא הטכנולוגיה ולכן הסיווג של Rocchio כאן הוא שגוי.

שאלה 2:

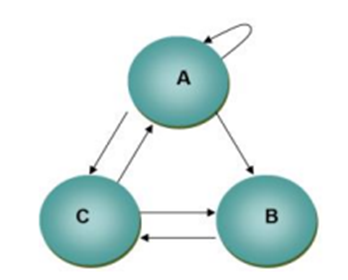
### **Authority Scores (In-degree)**

* **Node A**: 1 (from D)
* **Node B**: 1 (from A)
* **Node C**: 2 (from A and B)
* **Node D**: 1 (from C)

### **Hub Scores (Out-degree)**

* **Node A**: 2 (to B and C)
* **Node B**: 1 (to C)
* **Node C**: 1 (to D)
* **Node D**: 1 (to A)

**סעיף ג':**



Iteration 1:

**Initialize:**  **Update authority scores:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Authority |  | Authority | Hub | init |
| 2 |  | 0 | 1 | A |
| 2 | ⇒ | 0 | 1 | B |
| 2 |  | 0 | 1 | C |

**Normalization for authority and hub:**

**Update hub scores:**

|  |
| --- |
| Hub |
|  |
|  |
|  |

Iteration 2:

**Initialize:**  **Update authority scores:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Authority |  | Authority | Hub | init |
|  |  |  |  | A |
|  | ⇒ |  |  | B |
|  |  |  |  | C |

**Update hub scores:**

|  |
| --- |
| Hub |
|  |
|  |
|  |

Iteration 3:

**Initialize:**  **Update authority scores:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Authority |  | Authority | Hub | init |
|  |  |  |  | A |
|  | ⇒ |  |  | B |
|  |  |  |  | C |

**Update hub scores:**

|  |
| --- |
| Hub |
|  |
| 0.424 |
|  |

ניתן לראות שלאחר 3 איטרציות הגענו לציוני authority,hub מכונסים ואלו הם הציונים הסופיים שמצאנו עבור הגרף הנתון:

**שאלה 3:**

1. בתחילת הפרויקט, ניסינו להבין איך העברת דפים של סרטים פועלת באתר. בדקנו לפני כתובת הurl ומשתנה pages שעברנו עליו באיטרציות בכתובת והרצנו את הקוד. שמנו לב שהעמודים לא עוברים את עמוד 5 וראינו כי האתר חוסם את הגישה בדרך זו לעמודים הגדולים מ5. לאחר מכן, עברנו **לSelenium** על מנת להשתמש בלחיצה על כפתור לטעינת עמודים נוספים. לאחר כמה זמן, גוגל קולאב לא אפשר את הגישה הזו. לבסוף החלטנו לעבוד עם **pyppeteer** שמאפשר לנו ללחוץ על כפתור טעינת הדפים ללא בעיות. השתמשנו בנוסף בbeautiful soup וrequest על מנת לקרוא את הhtml של האתר ולקחת את הנתונים הרלוונטים.

על מנת לשמור את האינדקס (אשר יצא גדול כצפוי) היינו צריכים לשמור אותו באקסל בצ'אנקים מכיוון שזכרון הRAM של גוגל קולאב מוגבל.

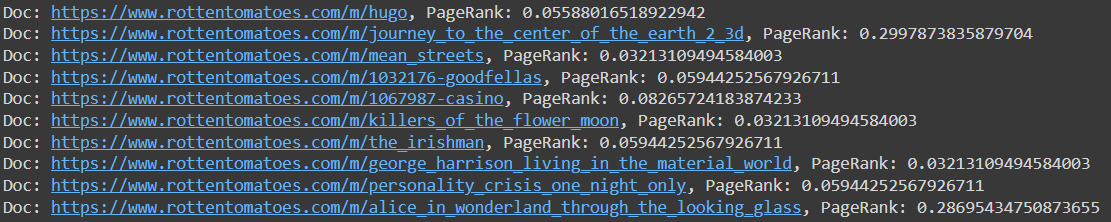
1. השאילתות שלנו רצו כ3 שעות וחישוב האינדקס ארך גם כ3 שעות.

זמן זה תלוי בכמות הדפים שרצנו עליהם ועל כמות הדפים שיש בכל עמוד מכיוון שכל לחיצה על טעינת עמוד סרטים נוסף משתמשת בpyppeteer על ידי לחיצה על כפתור טען עוד. לכל לחיצה על כפתור יש דילאי בגלל השימוש בscript של הכפתור בדף.

בנוסף, כאשר עברנו על התגובות של כל סרט, היה צורך בטעינת תגובות נוספות שגם משתמש בסקריפט של העמוד.

לדעתנו, ניתן לשפר את זמן הריצה בכך שאפשר להתחיל את הזחלן ישירות מעמוד 5, לדוגמא: <https://www.rottentomatoes.com/browse/movies_at_home/genres:action?page=5>. כך למעשה נתחיל ישירות מעמוד 5 בקטגורית אקשן ולא נצטרך שימוש בpyppeteer עד דף 5 (כלומר נצטרך מדף 5 והלאה).  
דבר זה יכול לחסוך בזמן ריצה מכיוון שלחיצה על כפתור לוקחת זמן מאשר שימוש בHTML של הדף בלבד.

1. בדפי שהוחזרו קיימים hubs.  
    ניתן לראות זאת כך שבכל דף של סרט, ישנם את הסרטים המומלצים לצפייה, כך שבכל סרט יש גישה לאותם סרטים הכי מומלצים – מה שגורם לכך שניתן לגשת לדפים אלו (המומלצים) מכל דף של סרט בrotten tomatoes. דוגמא למה שמופיע בכל דף סרט (קישור לסרטים מומלצים אחרים שהם קבועים): לכן, ניתן לראות כי קיימים גם authorities שהם דפי הסרטים המומלצים (בתמונה הנ"ל). נראה כי לכל סרט מומלץ מהרשימה אפשר לגשת על ידי דפי סרטים אחרים מכיוון שהסרטים המומלצים מופיעים בכל דף סרט.
2. לקחנו עשרה דפים של סרטים שחזרו מהשאילתה: הסרטים של הבמאים הטובים ביותר.

תוצאות הpagerank:  
  


ניתן לראות כי הpagerank הגבוה ביותר שקיבלנו הוא של הסרט:



# **Journey 2: The Mysterious Island**

(journey to the center of the earth 2)

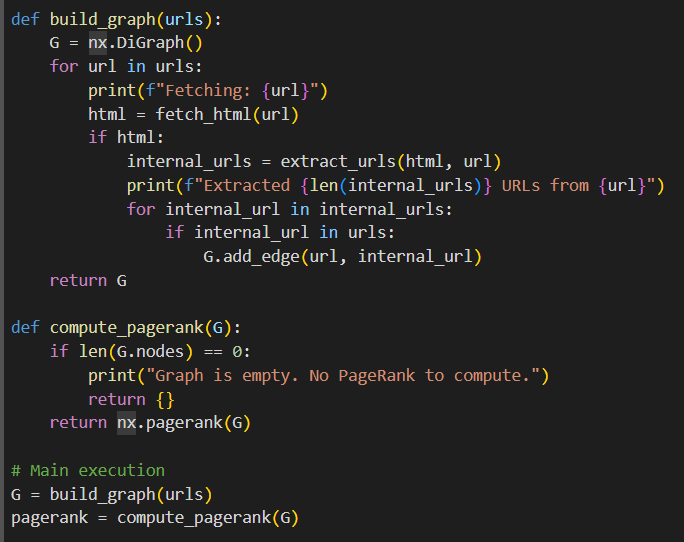
החישובים נעשו בקולאב, כך שהקישורים לסרטים נלקחים מהתגית של הסרטים הדומים ביותר לאותו הסרט בו אנחנו נמצאים. לכן, ניתר לומר כי הסרט עם הpagerank הגבוה ביותר הוא המקושר ביותר לשאר הסרטים.

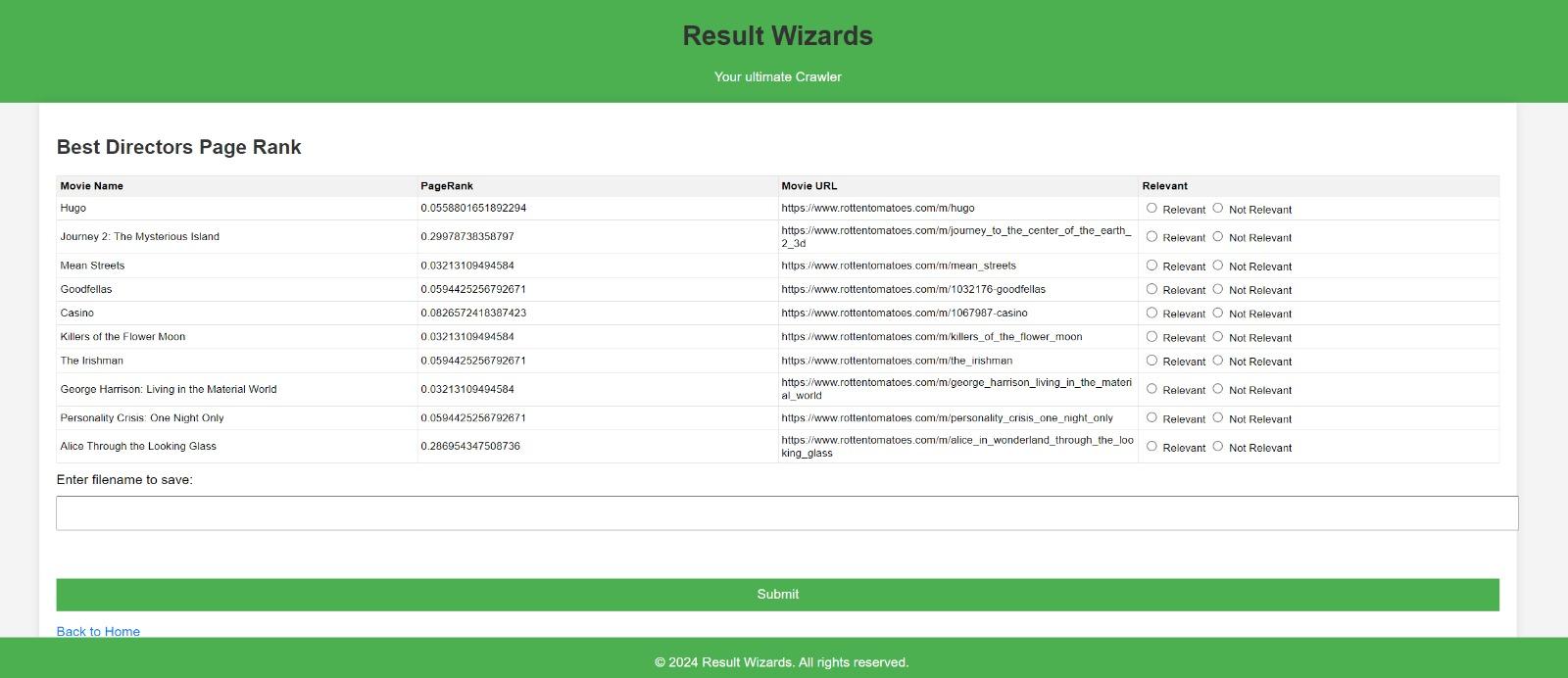
(הערה: מכיוון שהסרטים המומלצים שמופיעים בכל דף אינם חלק מהHTML של האתר, חישוב הpageRank אינו מכליל אותם).

השתמשנו ב networkx ליצירת הגרף וחישוב pageRank באופן הבא:  
nx.pagerank(G) כאשר G הינו הגרף שנבנה באמצעות nx.DiGraph().

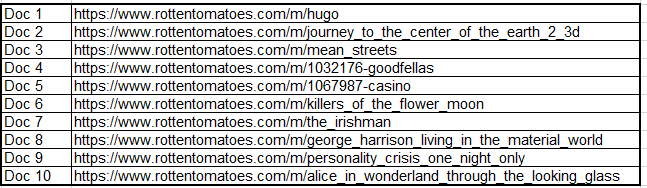






1.  הצגת הpageRank למשתמש עם עשרת הסרטים הרלוונטים:

תוצאות של 2 משתמשים שונים:



(רלוונטי - 1, לא רלוונטי - 0)



לאחר ניתוח הנתונים של סימון המשתמשים, מצאנו כי הם בחרו בסרטים רלוונטיים כסרטים שבהם השחקן רוברט דה נירו משחק שם, ולכן נרצה שהשאילתה הבאה תציג נתונים יותר רלוונטיים עבור אותם משתמשים. שאילתה שיכולה לשפר את התוצאות:

"מצא סרטים שבהם מופיע השחקן רוברט דה נירו".

קישור לגיט: <https://github.com/ndvp39/crawler> .