**Results Wizards**פרויקט באחזור מידע



נדב גולדין 316350768  
  
מתן נחמוכה 318185956  
  
עודד מורג 208983189  
  
יותם אהרון 208608448

**שאלה 1:**

האתר Rotten Tomatoes (<https://www.rottentomatoes.com/> ) הוא פלטפורמה מרכזית המתמקדת בביקורות על סרטים ותוכניות טלוויזיה, עם דגש על חוות דעת של משתמשים ומבקרי קולנוע מקצועיים. האתר מספק מידע נרחב על סרטים, כולל דירוגים, הכנסות בקופות, משך זמן הסרט, והבמאים. משתמשי האתר יכולים לגשת לדירוגים של סרטים והבמאים, לקרוא ביקורות מפורטות, ולהתעדכן במידע עדכני על סרטים. ניתן לבקר באתר בכתובת Rotten Tomatoes.

**שאלה 2:**

1. הז'אנר עם ממוצע זמן הסרטים הארוך ביותר.
2. הבמאים המוצלחים ביותר בז'אנר מסוים.
3. הסרטים עם ההכנסות הגדולות ביותר.

על מנת לענות על שאילתות אלו, יש לאסוף את המידע הבא: עבור הז'אנר עם ממוצע זמן הסרטים הארוך ביותר, נדרש לקבל את רשימת הסרטים לפי ז'אנר ואורך זמן הסרט (Runtime) ולחשב את הממוצע לכל ז'אנר. עבור הבמאים המוצלחים ביותר בז'אנר מסוים, יש לאסוף את הבמאים עם הציון הממוצע לסרטיהם (Tomatometer או Audience Score). עבור הסרטים עם ההכנסות הגדולות ביותר, נדרש לקבל את רשימת הסרטים עם הכנסות הקופות (Box Office).

לאחר הגדרת השאילתות בצורה מדויקת, אלו השאילתות שנתמקד בהן:

1. Which movies have the best directors?

בשאילתה זו נרצה למצוא את הבמאים הטובים ביותר באתר rotten tomatoes, להחזיר את הבמאים ואת הסרטים שלהם.

נמצא את אותם במאים על ידי מעבר על כל הסרטים לצפייה.

כאשר נחפש בדף המידע על הסרט, נתמקד בשם הבמאי שנמצא ב class = "caregory -wrap" ובדירוגים של הסרט שנמצאים ב- data-json="reviewData". נבצע חישוב המכליל את כמות הסרטים של הבמאי ואת הדירוגים של כל הסרטים השייכים לו. לבסוף, נחזיר את כל הבמאים עם הניקוד הגבוה ביותר שחושב ואת דפי הסרטים השייכים לכל במאי.

1. Which movie has the longest run time in every genre?

בשאילתה זו נרצה למצוא את הסרטים הארוכים ביותר בכל ג'אנר באתר.

נבצע זאת על ידי מעבר על כל הסרטים לצפייה באתר ונמצא את אותם סרטים בכל ז'אנר על ידי שימוש בתגית class = "caregory -wrap" שם נמצא הruntime של כל סרט. לבסוף, נחזיר את דף הסרט הארוך ביותר מכל ז'אנר.

1. Which movie has the highest box office in every genre?

בשאילתה זו נחפש את הסרט עם הכי הרבה הכנסות בכל ז'אנר.

גם בשאילתה זו נעבור על כל הסרטים ונשתמש בתגית class = "caregory -wrap" שבה נמצא סכום ההכנסות של הסרט. נחזיר את דף הסרט עם ההכנסה הגדולה ביותר מכל ז'אנר.

**כמו כן, לחישוב האינדקס נשתמש בספריית nltk לstop words ולהורדת סופיות במילים כאשר נתמקד בדף הביקורות של כל סרט (ביקורות של מומחים וביקורות של משתמשים) – רלוונטי לשאילתות שלנו.**

**שאלה 3:**

קישור לזחלן: <https://colab.research.google.com/drive/1o4iye5_s1s5w8DLF-G8-WUXQM6TKTxEf#scrollTo=XAFCjY9QuYnO> .

**שאלה 4:**

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.

**שאלה 5:**

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.

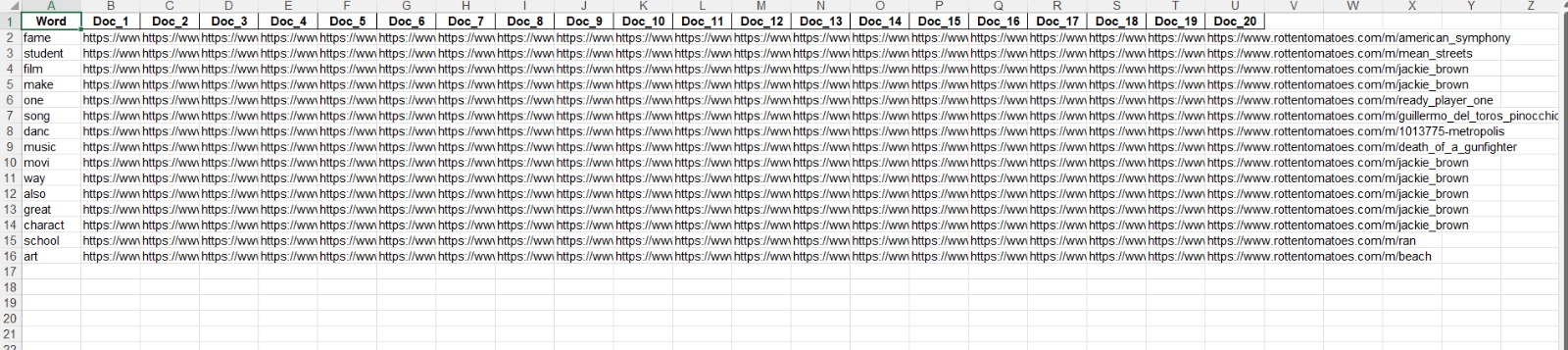
**שאלה 6:**

**קישור לדרייב:** <https://drive.google.com/drive/folders/1Nd3ITyDM8f03vOzj511P4MyTjWTMnkl1?usp=sharing>

לאחר הרצת הזחלן, קיבלנו שלושה קבצי אקסל הנמצאים בתיקייה queries\_result, כך שבכל קובץ אקסל, התוצאות של שאילתה.

לאחר בניית האינדקס על כל המסמכים שחזרו מהשאילתות, לקחנו את 15 המילים הנפוצות ביותר ועשינו להם inverted index, לכל מילה לקחנו את 20 המסמכים הראשונים שבהם היא מופיעה אם היא מופיע ביותר מ20 מסמכים (כמתבקש).

(הערה: האינדקס נמצא בדרייב בתקייה index, כך שהקובץ index\_combined מכליל את כל האינדקס).

Inverted\_index\_common\_15 בקובץ אקסל נמצא הinverted index.

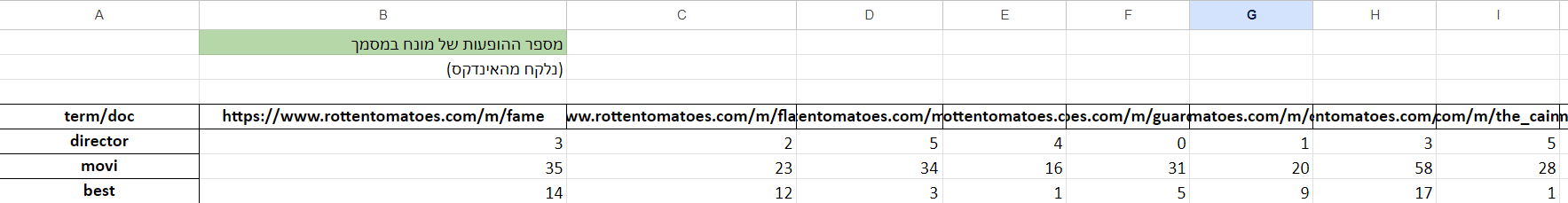
**שאלה 7:**

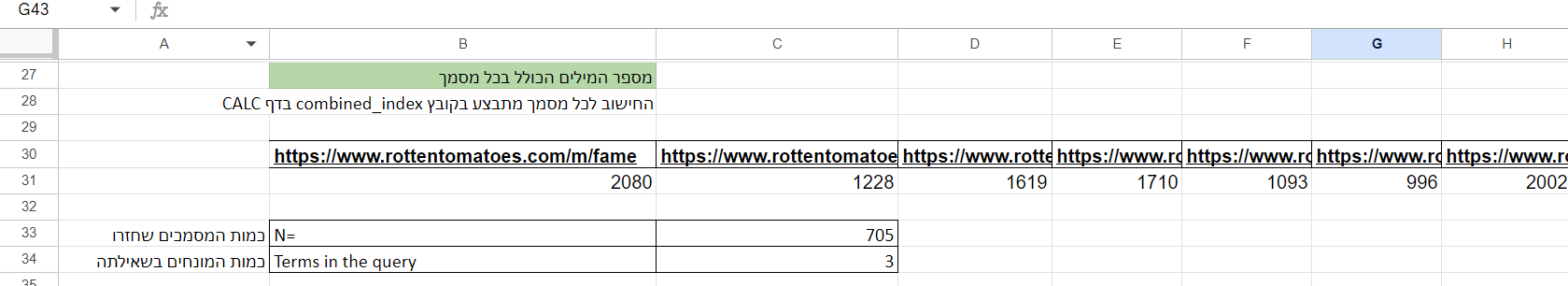
ניקח את השאילתה:

Which movies have the best directors?

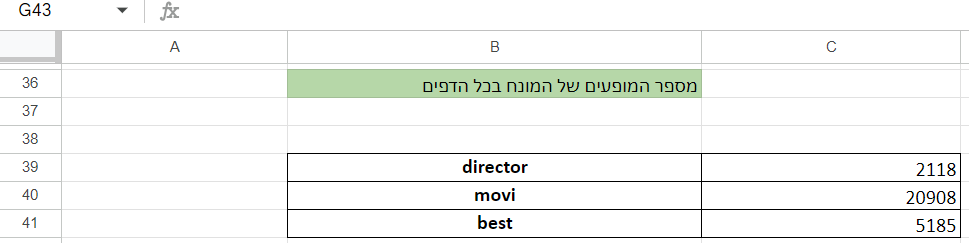
ובעזרת האינדקס, נבצע חישוב tf-df של המושגים בשאילתה על פני כל המסמכים שחזרו. את החישובים נבצע בקובץ calculates excel בדף CALCS.

תחילה, נארגן את הנתונים כך שלכל מושג בשאילתה (לאחר הורדת stop words וסופיות), יופיעו כמות ההופעות שלו בכל מסמך (נלקח מהאינדקס).

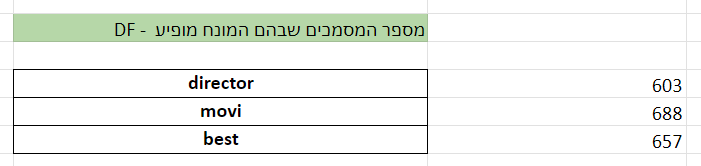


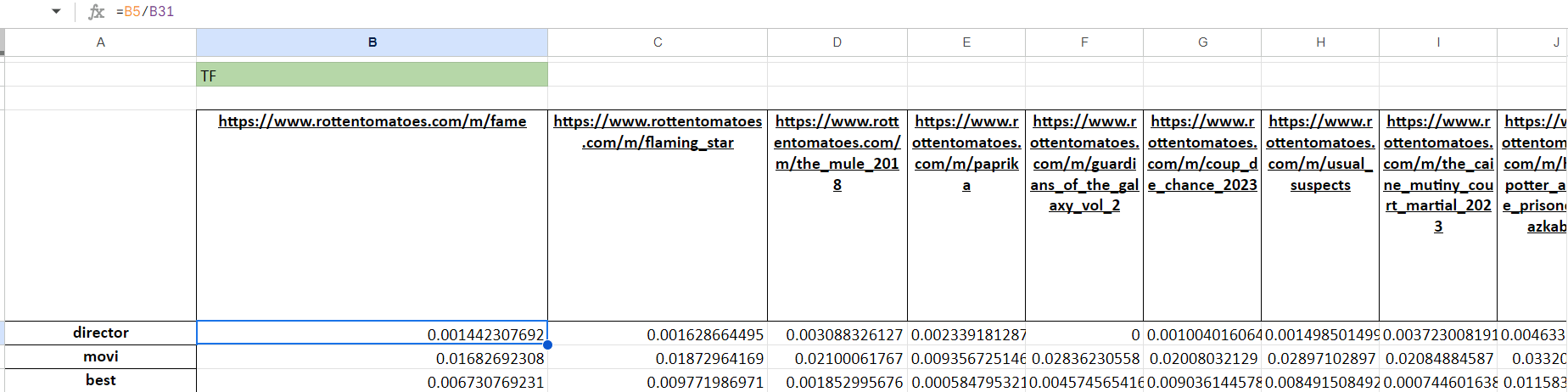
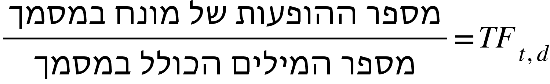
לאחר מכן, נמצא את מספר המילים הכולל בכל מסמך, את כמות המסמכים שהוחזרו, וכמות המושגים בשאילתה.

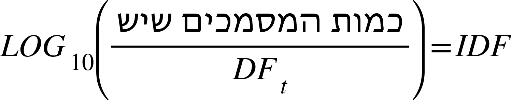
נבנה טבלה שבה יש את מספר המופעים של כל מושג בשאילתה בכל הדפים:

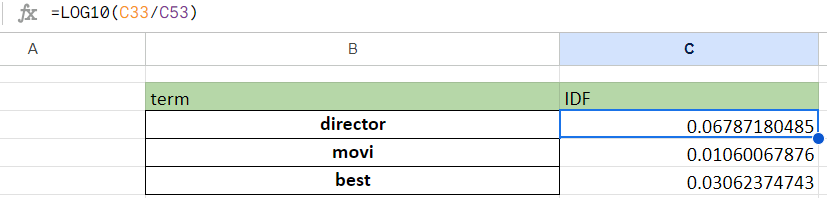


נחשב DF - מספר המסמכים שבהם מופיע כל מונח (באקסל בעזרת COUNTIF):



נחשב TF על ידי הנוסחא: 

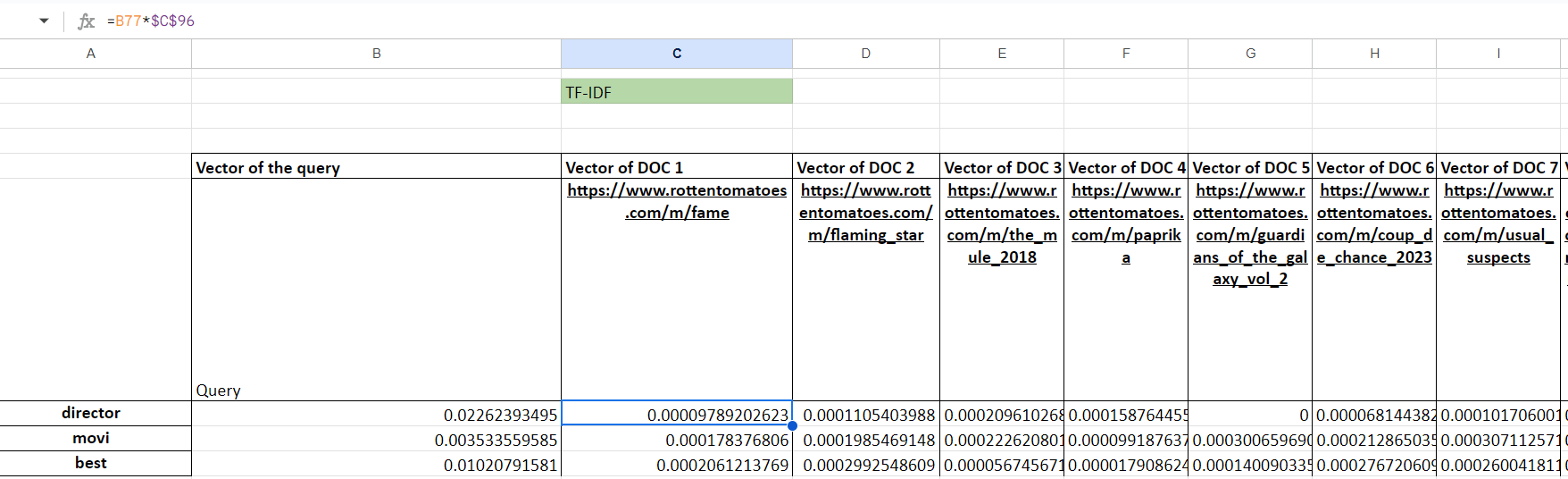
נחשב IDF על ידי הנוסחא: 



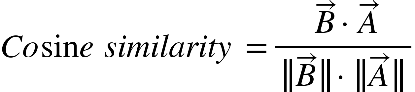
נחשב את הtf-idf- של המושגים בשאילתה על פני על כל המסמכים:

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>T</mi><mi>F</mi><mo>-</mo><mi>I</mi><mi>D</mi><msub><mi>F</mi><mrow><mi>t</mi><mo>,</mo><mo>&#xA0;</mo><mi>d</mi></mrow></msub><mo>=</mo><mi>T</mi><msub><mi>F</mi><mrow><mi>t</mi><mo>,</mo><mi>d</mi></mrow></msub><mo>&#xB7;</mo><mi>I</mi><mi>D</mi><msub><mi>F</mi><mi>t</mi></msub><mspace linebreak=\"newline\"/><mspace linebreak=\"newline\"/></mstyle></math>","origin":"MathType for Microsoft Add-in"}כך נחשב לכל מסמך ודף ובנוסף נחשב tf-idf של השאילתה עצמה.

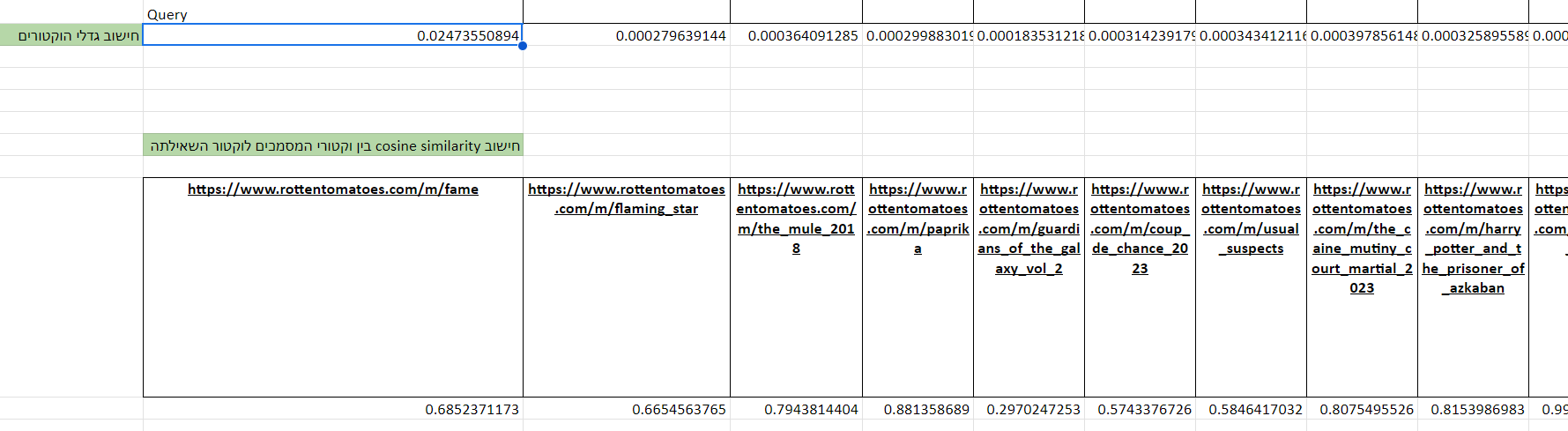
נקבל:



ניתן לראות בעזרת הטבלה שקיבלנו של tf-idf, כי **וקטורי המסמכים הם העמודות**.  
ולכן, על מנת לחשב ולהבין אילו מסמכים הם הקרובים ביותר לשאילתה. נחשב בעזרת Cosine similarity, את הדמיון בין וקטור השאילתה לבין כל אחד מוקטורי המסמכים, כך שלמעשה נקבל ערכים גבוהים עבור דמיון גבוה (כיווני וקטורים דומים יותר).



כאשר נחשב אותו עבור כל וקטור של מסמך. וקטור A – וקטור השאילתה, וקטור B – וקטור של מסמך.

נקבל:

דוגמא לחישוב גדלי הוקטורים.

דוגמא לחישוב cosine similarity.

שאלנו שני אנשים האם 10 התוצאות הראשונות רלוונטיות לשאלה:

1. סטודנט להנדסת חשמל במכללת אפקה בעל ידע נרחב בסרטים.
2. אמא של עודד אשר מכירה במאים ותיקים רבים.

להלן התוצאות:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |  |
| NR | R | NR | NR | R | R | R | R | R | R | P1 |
| R | R | R | NR | R | NR | R | NR | R | R | P2 |

נחשב את ה-Recall וה-Precision עבור כל אחד מהנשאלים.

מכיוון שלא ניתן לדעת באופן אובייקטיבי את מספר התוצאות הרלוונטיות, נשתמש בהיוריסטיקה:

ישנם כ-75 במאים זוכי פרס אוסקר, והמספר הממוצע של סרטים לכל במאי זוכה הוא כ-25. לכן, נניח כי ישנם כ-1875 סרטים העונים על הדרישה.

באתר Rotten Tomatoes ישנם כ-150,000 סרטים, מתוכם סקרנו רק 10,000. לכן, נניח שסקרנו בערך 125 סרטים העונים על הקריטריונים של סרט השייך לבמאי מוצלח.

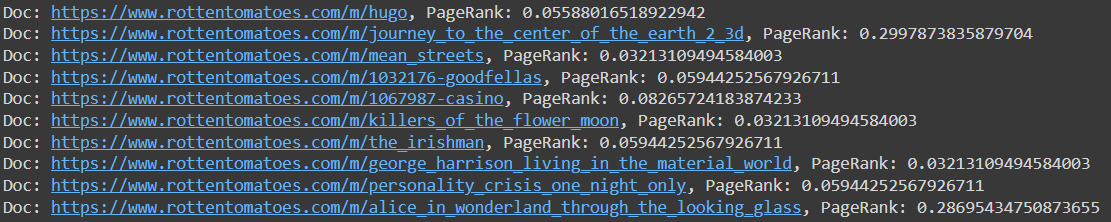
|  |  |  |
| --- | --- | --- |
| Recall | Precision |  |
| 7/125 | 7/10 | P1 |
| 7/125 | 7/10 | P2 |

**שאלה 8:**

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.

**שאלה 9:**

לקחנו עשרה דפים של סרטים שחזרו מהשאילתה: הסרטים של הבמאים הטובים ביותר.

תוצאות ה:pagerank-  


ניתן לראות כי הpagerank הגבוה ביותר שקיבלנו הוא של הסרט:



# **Journey 2: The Mysterious Island**

(journey to the center of the earth 2)

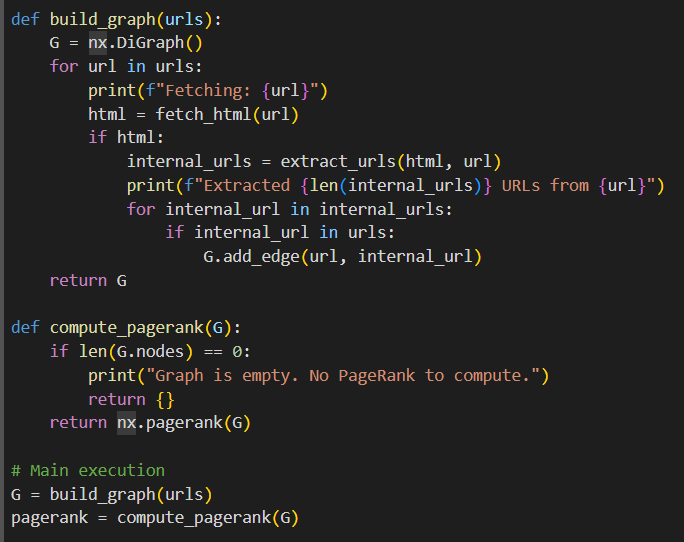
החישובים נעשו בקולאב, כך שהקישורים לסרטים נלקחים מהתגית של הסרטים הדומים ביותר לאותו הסרט בו אנחנו נמצאים. לכן, ניתר לומר כי הסרט עם הpagerank הגבוה ביותר הוא המקושר ביותר לשאר הסרטים.

(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().



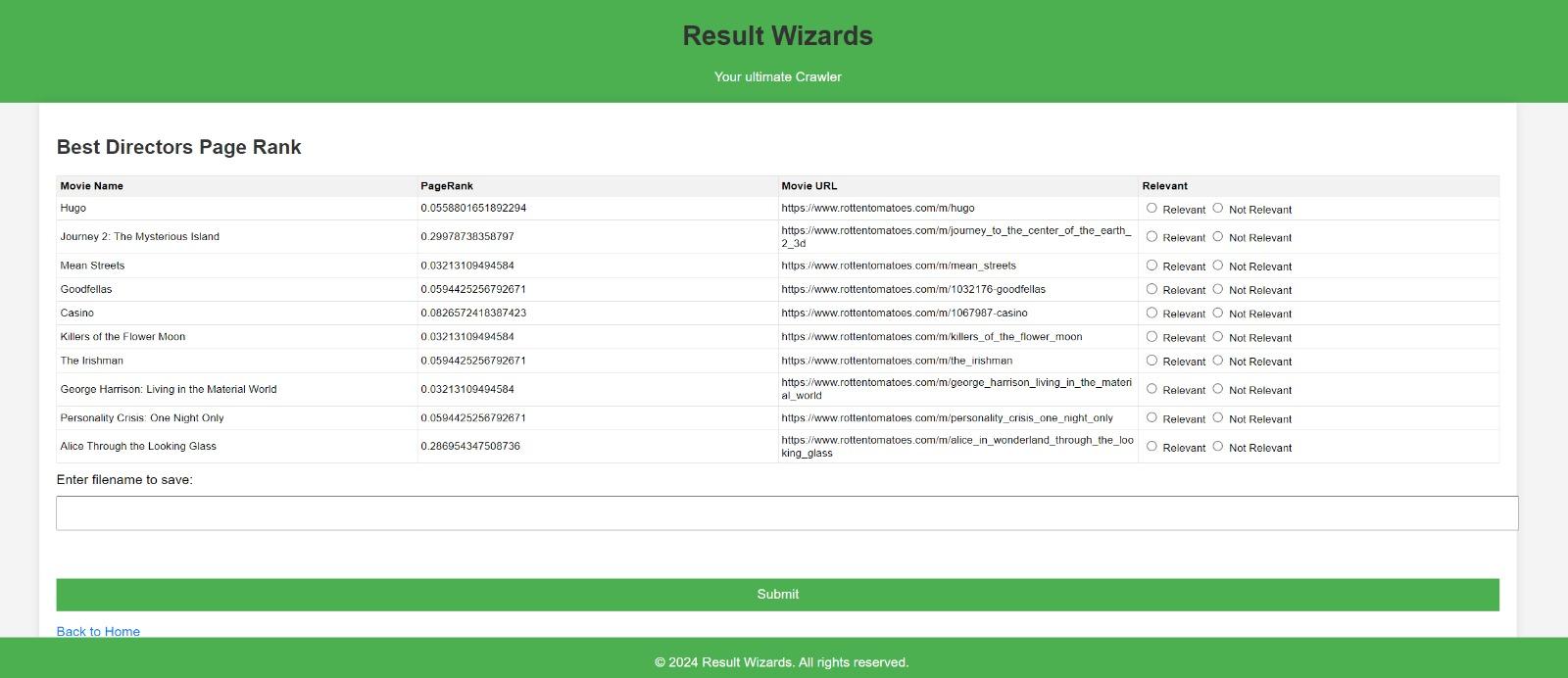


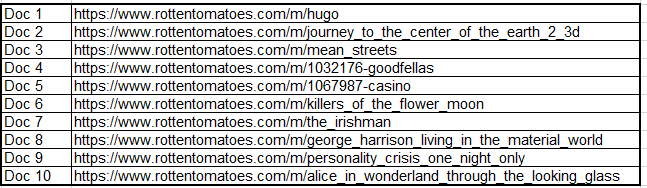


**שאלה 10:**

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".

**שאלה 11:**

<https://ndvp39.github.io/crawler/>

**שאלה 12:**

|  |  |
| --- | --- |
| איזה שינוי הוצע? | האם לדעתכם יש מקום לשיפור זה? אם כן - הסבירו כיצד ניתן לשפר. אם לא, נמקו |
| לכתוב את הקוד ללא קולאב, אלא במחשב האישי | ניתן לבצע שינוי זה וזמן הריצה כנראה יהיה מהיר יותר אך יותר נוחה העבודה בצוות בעזרת גוגל קולאב. |
| שימוש בהגדלת הקצאת הזיכרון בקולאב | ניתן לעשות זאת ולשפר את שמירת קבצי האקסל. |
| להראות את רשימות הסרטים שהתקבלו מהשאילתות בצורה יותר נוחה לעין. | ניתן לעשות זאת על ידי הצגת הסרטים עם תמונותיהם ואפילו טריילר קצר לכל סרט.  נראות יותר טובה בהצגה למשתמש. |

הערה: זמן הריצה הגדול הוא כתוצאה מזחילה על האתר ולקיחה של כ10 אלף סרטים, כולל לקיחת נתונים וביקורות של אותם הסרטים. היה צפוי שזמן הריצה יהיה גדול בגלל השימוש בscript של האתר (לדוגמא puppeteer, selenium) לצורך העברת דפים (קליק על כפתור = דילאי) וגם כתוצאה מכך שכמות הסרטים שלקחנו היא גדולה יחסית.

**שאלה 13:**

האתגרים שהיו לנו:

בפרויקט, היו לנו כמה אתגרים.

ההבנה ובחירת השאילתות המתאימות על מנת להשתמש בחומר הנלמד בהרצאות להשגת המטרה של הפרויקט. בהתחלה בחרנו שאילתות לא טובות מכיוון שלא ידענו למה להתאים אותם.

לאחר חשיבה רבה, הצלחנו להתאים ולהגדיר מחדש את השאילתות בצורה המיטבית כך שהפרויקט יענה על הדרישות בעזרת החומר הנלמד בהרצאות.

אחת הבעיות שהיו לנו היא הצלחת הזחלן באתר. מכיוון שלאתר הייתה בעיה בהעברת הדפים (כפי שמוסבר בפרויקט) נאלצנו לחפש ולהשתמש בטכנולוגיות שונות על מנת לגרום לפרויקט לעבוד. מה שגרם לקושי מכיוון שגוגל קולאב לא התממשק בצורה טובה עם סלניום. בסוף מצאנו את pupetter שנתנה לנו הצלחה בשימוש בscript של האתר לצורך העברת דפי הסרטים.

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