פרויקט בינה עסקית

**סטודנטים:**

אביאל יהודה 312588056

יובל לוי 208552471

אור קינן 208499996

גל זולדן 207298092

דורין בוזגלו 208064733

**מטרת הפרויקט**

הפרויקט מתמקד בנתונים מתוך שירים המושמעים בSpotify. הנתונים כוללים מידע רחב מאוד הכולל פרטים טכניים על השיר, דירוגים על תכונות מסוימות, מדדים טכניים על תוכן השיר ועוד.  
מתוך נתונים אלו ניתן ללמוד הרבה על הקשרים בין הפרמטרים השונים. אחת השאלות העיקריות היא האם ניתן להבין מראש האם שיר יהיה פופולרי או פחות? כמובן שתחום המוזיקה הינו תחום רווחי מאוד. ככל שהשיר יהיה יותר מוצלח ויהיו לו יותר משמעויות, כך תעלה התפוקה הכלכלית ממנו והזמר יחשב טוב יותר. לכן בפרויקט זה ננסה לרכז את המידע בצורה נגישה יותר ומובנת יותר על מנת שיהיה אפשרי להסיק מנתונים אלו מסקנות פרקטיות ועסקיות.

**שאלת מחקר 1**

האם נוכל למצוא מודל הנותן חיזוי של פופולריות השיר על סמך תכונותיו?

**Kpi:**

1. דיוק החיזוי: טעות חיזוי ממוצעת קטנה מ5% במשך חודש.

התאמה למודל SMART:   
ספציפית: המדד מתאים רק למודל שלנו ונותן אינדיקציה אם המודל טוב או לא.  
מדידה: על ידי הצבה במודל נקבל את חיזוי הפופולריות.  
ברת השגה: ניתן למצוא מודל חיזוי ולהציב בו את נתוני השיר.  
רלוונטיות: טעות חיזוי הוא מדד הכרחי כדי לראות אם התוצאות תואמות את מה שקורה בפועל.  
זמנים: מדידה במשך חודש מקבלת המודל.

1. חשיבות התכונות: מציאת 5 תכונות המשפיעות ביותר על תוצאות המודל בשנה האחרונה.

התאמה למודל SMART:  
ספציפית: מיקוד בכמות תכונות שמקבלות משקל גבוה בהשפעה על הפופולריות.  
מדידה: ניתן למדוד את משקל התכונות על ידי פעולות סטטיסטיות.  
ברת השגה: על ידי תוכנות סטטיסטיות כמו excel ניתן למצוא את 5 התכונות.  
רלוונטיות: כדי להבין באיזה תכונות יש לתת יותר דגש בשידור/כתיבה/הוצאה של שיר.  
זמנים: תחום במשך שנה.

1. דירוג כל התכונות לפי חוזק ההשפעה שלהם על הפופולריות בכל השירים למשך השנה האחרונה.

התאמה למודל SMART:  
ספציפית: מתאימה לנתונים ולשאלת המחקר.  
מדידה: על ידי מקדם מתאם בן התכונה לבין הפופולריות.  
ברת השגה: ניתנת להשגה בעזרת excel.  
רלוונטיות: ניתן להבין על ידה איזו תכונות משפיעות יותר ואיזה פחות.  
זמנים: תחום במשך שנה.

**שאלת מחקר 2**

לאיזו תכונה יש את ההשפעה השלילית המשמעותית ביותר על הפופולריות של שיר?

**Kpi:**

1. מדד פירסון: מקדם מתאם r השלילי ביותר בין התכונה לבין הפופולריות בשנה האחרונה.

התאמה למודל SMART:   
ספציפית: המדד מתאים לנתונים אלה ומשקף בצורה טובה את הקשר בין המשתנים.  
מדידה: מתקבלת תוצאה עבור כל המשתנים וניתן להשוות בניהם.  
ברת השגה: ניתן למדוד אותה בקלות על ידי excel.  
רלוונטיות: מודדת את היחס ההפוך בין התכונה לבין הפופולריות  
זמנים: תחום במשך שנה.

1. התכונה הבולטת ביותר ב1,000 השירים המדורגים הכי נמוך במשך השנה האחרונה.

התאמה למודל SMART:   
ספציפית: מתאימה לנתונים ולשאלת המחקר.  
מדידה: ניתן למדוד את התכונות על ידי נרמול כל העמודות ומציאת ממוצע.  
ברת השגה: ניתנת להשגה בעזרת excel.  
רלוונטיות: ניתן האם יש איזו שהיא תכונה שבולטת יותר בשירים המדורגים נמוך.  
זמנים: תחום במשך שנה.

1. דירוג כל התכונות לפי חוזק ההשפעה שלהם על הפופולריות בכל השירים למשך השנה האחרונה.

התאמה למודל SMART:   
ספציפית: מתאימה לנתונים ולשאלת המחקר.  
מדידה: על ידי מקדם מתאם בן התכונה לבין הפופולריות.  
ברת השגה: ניתנת להשגה בעזרת excel.  
רלוונטיות: ניתן להבין על ידה איזו תכונות משפיעות יותר ואיזה פחות  
זמנים: תחום במשך שנה.

**בחירת סכמה: כוכב**

הנתונים שיש לנו הם נתונים טכניים של שירים כמו שם השיר, אורך, עוצמה, פופולריות, ערכיות ועוד. אין קשר ישיר בין הנתונים אבל אפשר לסווג אותם לפי 2-3 נושאים שונים, בנוסף לא ניתן לעשות היררכיה בין הנתונים לכן הסכמה הכי מתאים לנו היא סכמת כוכב.

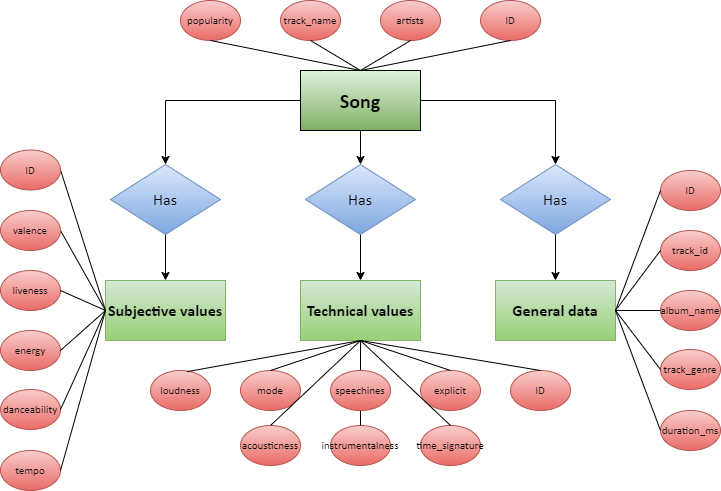
**תיאור הData :**

\*תיאור עמודות הdata הינן בקישור :

<https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset>

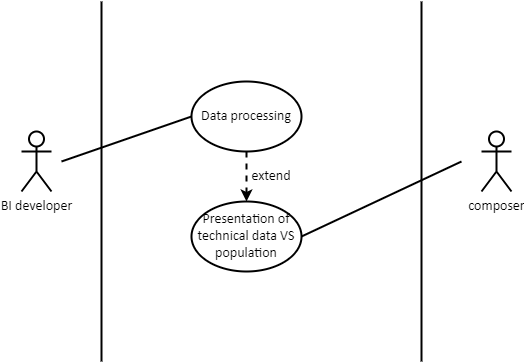
כל נתונים מפוצלים ל3 טבלאות. מה שמקשר בין כולם זה הID של השיר.  
הטבלאות מחולקות בצורה הבאה:  
1. מידע כללי – מידע בסיסי שיודעים לגבי כל שיר: אורך, שם, זמר, ז'אנר ואלבום.  
2. ערכים טכניים – נתונים לגבי כל שיר שמודדים אותם בצורה טכנית כמו עוצמה (בדציבלים), כמות מלל, אקוסטיקה ועוד  
3. ערכים סובייקטיביים – ערכים שמבטאים תחושות ורגשות לגבי השירים כמו ערכיות, אנרגטיות, התאמה לריקודים, פופולריות ועוד.  
  
הנתונים העיקריים יוצגו בטבלה המרכזית (fact).

להלן תרשים ERD המייצג את הסכמה של הData Warehouse :



**Use case לתיאור יעילות הסכמה שבחרנו:**

חלוקת הנתונים נעשתה בצורה שבה אין כל כך קשר בין הנתונים בין הטבלאות השונות וכאשר נרצה לקבל מידע לגבי השיר ולהסיק מסקנות נמזער את השילובים בין הטבלאות. אך יחד עם זאת אנחנו שומרים על תבנית מסודרת ומובנת למשתמש. המידע בשלמותו בטבלה אחת מקשה על ביצוע פעולות, שאילתות ועוד ולכן יש למצוא את האיזון בין חלוקה גדולה מדי שמקלה על הפעולות לבין פיזור גדול מדי של הנתונים.



לדוגמה אם מלחין המחובר למערכת ירצה להבין עד כמה כל פרמטר בנתונים משפיע על פופולריות השיר, אז יש טבלה אחת של נתונים טכניים שאותם נעבד ונציג אותם בטבלה המרכזית (fact) לצד הפופולריות שכבר נמצאת שם.

**שאילתות SQL**

1. חלוקה של הפופולריות לעשירונים עבור כל עשירון נבדוק את הממוצע של המדדים הסובייקטיביים (נתונים המבטאים יותר רגש כמו מתאים לריקוד, אנרגטי, מוסרי). בעזרת מידע זה תהיה אפשרי לבחון האם יש איזה שהוא מגמתיות בין אחד מין התחושות של המשתמשים לבין הפופולריות.

SELECT DISTINCT

popularity\_ntile, popularity,

AVG(danceability) OVER (PARTITION BY popularity\_ntile) AS avg\_danceability,

AVG(energy) OVER (PARTITION BY popularity\_ntile) AS avg\_energy,

AVG(tempo) OVER (PARTITION BY popularity\_ntile) AS avg\_tempo,

AVG(valence) OVER (PARTITION BY popularity\_ntile) AS avg\_valence

FROM

(SELECT

popularity, danceability, energy, tempo, valence,

NTILE(10) OVER (ORDER BY popularity) AS popularity\_ntile

FROM

musicData) AS subquery;

1. בדומה לשאילתה הקודמת רק שהפעם נחלק לפי תכונות טכניות כמו עוצמה בדציבלים, אורך השיר ועוד.

SELECT DISTINCT

popularity\_ntile, popularity,

AVG(duration\_ms) OVER (PARTITION BY popularity\_ntile) AS avg\_duration,

AVG(speechiness) OVER (PARTITION BY popularity\_ntile) AS avg\_speechiness,

AVG(acousticness) OVER (PARTITION BY popularity\_ntile) AS avg\_acousticness,

AVG(loudness) OVER (PARTITION BY popularity\_ntile) AS avg\_loudness

FROM

(SELECT

popularity, duration\_ms, speechiness, acousticness, loudness,

NTILE(10) OVER (ORDER BY popularity) AS popularity\_ntile

FROM

musicData) AS subquery;

1. טבלה המרכזת את האלבומים והז'אנרים ובנוסף ממוצע פופולריות לכל אלבום ולכל ז'אנר. בעזרת טבלה זו נבחן כל ז'אנר לגופו ונראה האם יש אלבומים ספציפיים שהשפיעו בצורה חריגה.

SELECT

album\_name, track\_genre, AVG(Popularity) OVER (partition by album\_name) as AlbumPopularity, AVG(Popularity) OVER (partition by track\_genre) as GenrePopularity

FROM musicData

ORDER by Popularity DESC;

1. טבלה שיש בה רק עשירון תחתון של הפופולריות שבה נציג את הז'אנר, פופולריות של הז'אנר ואת הכמות שהז'אנר נמצא בעשירון זה.

SELECT DISTINCT

popularity, track\_genre, COUNT(track\_genre) OVER (ORDER BY popularity) AS count\_genre

FROM

(SELECT

popularity, track\_genre, NTILE (10) OVER (ORDER BY popularity) AS popularity\_ntile

FROM

musicData) AS subquery

WHERE Popularity\_ntile == 1

ORDER BY count\_genre DESC;

1. שאלה דומה לשאלה הקודמת אך הפעם נבחן את העשירון העליון על מנת לזהות הבדלים בין העשירונים השונים .

SELECT DISTINCT

popularity, track\_genre, COUNT(track\_genre) OVER (ORDER BY popularity) AS count\_genre

FROM

(SELECT

popularity, track\_genre, NTILE (10) OVER (ORDER BY popularity) AS popularity\_ntile

FROM

musicData) AS subquery

WHERE Popularity\_ntile == 10

ORDER BY count\_genre DESC;

1. טבלה המסודרת לפי הפופולריות בסדר עולה ,ולכל דירוג אנו מציגים את המינימום והמקסימום של כמה פרמטרים שאנו חושבים שמשפיעים יותר על הפופולריות.

SELECT

popularity,

MIN(duration\_ms) OVER (PARTITION BY popularity) AS MINduration,

MAX(duration\_ms) OVER (PARTITION BY popularity) AS MAXduration,

MIN(danceability) OVER (PARTITION BY popularity) AS MINdanceability,

MAX(danceability) OVER (PARTITION BY popularity) AS MAXdanceability,

MIN(energy) OVER (PARTITION BY popularity) AS MINenergy,

MAX(energy) OVER (PARTITION BY popularity) AS MAXenergy,

MIN(loudness) OVER (PARTITION BY popularity) AS MINloudness,

MAX(loudness) OVER (PARTITION BY popularity) AS MAXloudness,

MIN(valence) OVER (PARTITION BY popularity) AS MINvalence,

MAX(valence) OVER (PARTITION BY popularity) AS MAXvalence

FROM

musicData;

1. בטבלה זו נבחן את האפשרות שהזמר הוא זה שמשפיע על הפופולריות. לשם כך אנו נציג את הזמרים, את הכמות שכל זמר הופיע בנתונים ואת ממוצע הפופולריות של השירים המשויכים אליהם בסדר יורד.

SELECT DISTINCT

COUNT(artists),artists, AVG(popularity) over (partition by artists) as artistsPopularity, track\_genre

FROM

musicData

GROUP BY artists

ORDER by artistsPopularity DESC;

1. בטבלה זו נבחן את העמודה שנותנת לנו ערכיים בינאריים , אם קיימות מילים מפורשות בשיר ונבדוק את ממוצע הפופולריות עבור כל קבוצה.

SELECT avg(popularity),explicit, COUNT(explicit) as countexplicit

FROM musicData

GROUP by explicit

ORDER by avg(popularity) DESC;

1. בטבלה זו נבחן את עמודת ה"ערכיות" שנותנת ערכים בין 0 ל1 , כאשר ערך קרוב לאפס מייצג שירים שמעוררים תחושות לא טובות (עצב, כעס) וקרוב ל1 מייצג שירים שמעוררים תחושות טובות (שמחה , ריגוש) נפצל את העמודה , נראה את כמות השירים לאחר הפיצול ואת ממוצע הפופולריות.

SELECT AVG(popularity) AS valencePopularity ,COUNT(valence)

FROM musicData

WHERE valence > 0.5

UNION

SELECT AVG(popularity) AS valencePopularity ,COUNT(valence)

FROM musicData

WHERE valence <= 0.5;

1. בשאלה זו נבדוק את הסיטואציה בה שם של שיר זהה לשם של האלבום והאם בהכרח מעידה על פופולריות גבוהה.

SELECT album\_name, track\_name, Popularity

FROM musicData

WHERE track\_name=album\_name

ORDER by Popularity DESC;

**מסקנות**

לאחר הרצת השאילתות ניתן להסיק מספר מסקנות עבור שאלת המחקר :

1. ישנם ז'אנרים שמופיעים מספר רב של פעמים בעשירון העליון ומופיעים מעט פעמים בעשירון התחתון או שכלל לא . ניתן לראות תוצאות אלו גם באופן הפוך על ז'אנרים שמופיעים מספר רב של פעמים בעשירון התחתון. מכאן ניתן להסיק שסוג הז'אנר משפיע על הפופולריות של השיר.

2. לפי השאילתה האחרונה ניתן לראות שטווח הערכים בעמודת הפופולריות הוא גדול ומכיל שירים פופולריים יותר ופופולריים פחות ומכאן ניתן להסיק כי אין השפעה חד משמעית לסיטואציה שבה שם השיר זהה לשם האלבום על הפופולריות.

3. עבור כל אומן קיבלנו את ממוצע הפופולריות של כל השירים שלו. כאשר אומן מופיע מספר רב של פעמים וממוצע הפופולריות של אותו אמן היה גבוה , נסיק כי רוב שיריו קיבלו פופולריות גבוהה . מכאן קיימת השפעה של האומן על הפופולריות של השיר.

4. זיהינו פער בין שירים חיוביים(שמחים) לשירים שלילים(עצובים). ממוצע פופולריות של השירים השלילים היה גבוה יותר מהשירים החיוביים. לכן ניתן להסיק כי שירים עצובים הם יותר פופולריים.