דו"ח פרויקט גמר – חיזוי פופולריות של שירים

דטא סיינס בתעשייה

תקציר:

מוזיקה הייתה חלק בלתי נפרד מהתרבות שלנו לאורך כל ההיסטוריה האנושית. לחיזוי פופולריות של שירים יש השלכות גדולות על עסקים כמו למשל תחנות רדיו, חברות תקליטים, זמרים ועוד. עניין אותנו לנסות ולבדוק מה בדיוק הופך שיר להיות פופולרי, במיוחד כיוון שכולנו שומעים שירים פופולריים דרך המלצות יוטיוב וספוטיפי וזה בחיי היום יום שלנו כל הזמן. בפרויקט זה במסגרת הקורס "דטא סיינס בתעשייה" נרצה לחזות פופולריות של שירים לפני פרסומם. בסקירת הספרות לרוב נחשפנו למאמרים אשר השתמשו בסיווג לפתרון הבעיה שהצגנו על חיזוי פופולריות של שירים באמצעות סיווג לשתי קטגוריות של פופולרי או לא פופולרי בלבד. לעומת זאת, אנו בחרנו לנסות לבחון מודלי חיזוי על ערכים רציפים ולא רק על ערכים קטגוריאליים. בנוסף, בחרנו לממש מודלי סיווג קטגוריאלי על יותר משני קטגוריות. במהלך העבודה התייעצנו עם מומחה תוכן (Domain expert) שנתן לנו קצת רקע תיאורטי על מוזיקה וסייע לנו בניסיונות להוצאת פיצ'רים. קבענו לבסוף את סוגי התכונות שיש להם את כוח הניבוי הגדול ביותר עבור הפיכת שיר לפופולרי, אך למרות זאת המסקנה העיקרית שלנו הייתה שאין השפעה גדולה למאפיינים טכניים של שיר על חיזוי פופולריות של שיר כמו שחשבנו שתהיה.

מבחינת חלוקת עבודה בינינו הסטודנטים, אנחנו דוגלים בשיתוף פעולה והבנה מלאה של כל החומר, לכן לא ראינו לנכון לחלק בנינו את העבודה לפי סעיפים והעדפנו לעבוד עליה בשיתוף פעולה ובמקביל כדי ששנינו נכיר את העבודה כולה מא'-ת' ושנתנסה בכל המשימות לאורך כל העבודה. עבדנו בתצורה שאחד רשם את הקוד והשני במקביל רשם את הדוח ולפעמים התחלפנו בינינו. מבחינת סקירת הספרות, חילקנו בנינו את המאמרים כך שכל אחד קרא מאמר וסיכם אותו.

1. בחירת הנתונים:

עבור משימת חיזוי זו, סט הנתונים שבחרנו הינו:

https://www.kaggle.com/vicsuperman/prediction-of-music-genre

סט הנתונים מכיל כ-50,000 רשומות של שירים שונים (5,000 שירים מכל אחד מעשרת סוגי הז'אנר) של יותר מ-6000 זמרים שונים. סט הנתונים מכיל את העמודות הבאות[1]:

	עמודה	הסבר
1	instance_id	מספר ייחודי של שיר
2	artist_name	שם האומן
3	track_name	שם השיר
4	popularity	מתאר את הפופולריות של השיר (0-99), מתפלג בצורה נורמלית

[1] התפלגויות העמודות מצורפות בנספח 2

5	acousticness	מתאר אקוסטיקה של שיר, בין 0 ל-1 כך ש-1 מציין אקוסטיות גבוהה
6	danceability	מתאר עד כמה השיר מרקיד (משלב אלמנטים מוזיקאליים כולל tempo, rhythm stability, beat strength, and overall regularity). 0 מסמן הכי פחות מרקיד, לעומת 1 שמסמן שהשיר מאוד מרקיד
7	duration_ms	משך MS של השיר
8	Energy	מתאר את העוצמה והפעילות בשיר. 0 מסמן עוצמה ופעילות נמוכה, לעומת 1 שמסמן עוצמה ופעילות גבוהה
9	instrumentalness	מתאר האם רצועת המוזיקה אינה מכילה שירה. ככל שערך האינסטרומנטליות קרוב יותר ל-1, כך גדל הסיכוי שהרצועה לא מכילה תוכן ווקאלי
10	key	סוג הסולם בשיר
11	liveness	מתאר זיהוי נוכחות של קהל בשיר, ערכי חיים גבוהים יותר מייצגים סבירות גבוהה שהשיר בוצע בשידור חי
12	loudness	מתאר את הקולניות (העוצמה) בדציבלים (dB) בשיר. ערכים טיפוסיים נעים בין מינוס 60 ל-0
13	mode	מתאר את המודאליות (מז'ור או מינור) של השיר, זהו סוג הסולם שממנו נגזר התוכן המלודי שלו. מז'ור מיוצג על ידי 1 והמינורי הוא 0
14	speechiness	מזהה נוכחות של מילים מדוברות בשיר, ככל שיהיה דומה יותר לדיבור הערך יהיה יותר קרוב ל-1
15	tempo	הקצב הכולל המשוער של רצועה בפעימות לדקה (BPM)
16	obtained_date	המקצב בשיר, רשום בצורה של תאריך כנראה הייתה טעות (המרנו במהלך הכנת הנתונים למספרי)
17	valence	מתאר את החיוביות המוזיקלית בשיר, רצועות שירים עם ערכיות גבוהה נשמעות חיוביות יותר (שמחה, עליזות, אופוריה) לעומת עם ערכיות נמוכה אשר נשמעות שליליות (עצבות, דיכאון, כעס)
18	music_genre	סוג הסגנון של השיר

2. **הגדרת המשימה:**

חיזוי הפופולריות של שיר בשני אופנים:

- כאשר נבצע מודל על ערכים רציפים, המשימה שלנו תהיה לחזות עד כמה השיר יהיה פופולרי מ 0-99 (כאשר 0 הכי נמוך, 99 הכי גבוה).
- כאשר נבצע מודל על ערכים קטגוריאליים, המשימה שלנו תהיה לסווג האם הפופולריות של השיר היא Low/Moderate/High.

3. **סקירת ספרות:** (קישורים למאמרים נמצאים על פירוט מקור המאמר)

א. **<u>מאמר ראשון:</u>**

Pham, James, Edric Kyauk, and Edwin Park. "Predicting song popularity". Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep 26 (2016).

מאמר זה הדגיש את החשיבות והשילוב בתכונות נוספות של "bag of words" לקבוצת מאפיינים קטגוריאליים מעבר לתכונות הבסיסיות בכדי לשפר חיזוי פופולריות של שיר. תרומתו העיקרית של מאמר זה הינו הפחתת המגבלה הטמונה בהשוואה בעקבות המגוון הרב של הנתונים. כלומר, שימוש במודלים בSPD (SpotGenTrack Popular Dataset) במקום MSD (Million Song

Dataset) כך שלא יכיל רק כמה תכונות אודיו מחושבות מראש אלא גם את כתובות ה-URL של תצוגות אודיו וסט מלא של המילים כדי למנוע הגבלות בחילוץ תכונות חדשות. השיטות בהם השתמשו הינם קבוצה של מסווגים: Logistic Regression, Linear & Quadratic Discriminant Analysis ו- SVM כדי לקבוע את הפופולריות של שיר כבעיית סיווג בינארי. בנוסף, הוצגו מודלים שונים של רגרסיה כדי לאחזר מידע בעל ערך לגבי רמת הפופולריות. עבור מודלי הסיווג, האלגוריתם עם ציון F1 הגבוה ביותר היה SVM(RBF), תוצאה המצביעה על קשרים לא ליניאריים בין תכונות האודיו/מטא נתונים והפופולריות. זאת לעומת SVM(Linear), אם ציון F1 נמוך יותר, אשר מצביע שכנראה הנתונים לא היו ליניאריים. עם זאת, לא היו הבדלים משמעותיים בביצועים בין כל הדגמים שנבדקו במאמר. ציוני F1 נעו כולם בין 0.5 ל-0.6 והדיוקים כולם נעו בין 0.75 ל-0.8. התוצאות הגבוהות יותר עבור SVM, כיוון ש-SVMs מושפעים בעיקר מנקודות הנתונים הקרובות ביותר לשוליים, בעוד רגרסיה לוגיסטית ומודלים אחרים מושפעים מכל נקודות הנתונים. בנוסף, SVMs גם מתפקדים טוב יותר בבעיות עם מספר רב של ממדים.

Model	AUC
SVM (Linear Kernel)	0.79
SVM (RBF Kernel)	0.81
Logistic Regression (LR)	0.69
LDA	0.71
QDA	0.64
Multilayer Perceptron (MLP)	0.79

עבור מודלי הרגרסיה, שגיאת הבדיקה הקטנה ביותר מבין המודלים הייתה באמצעות רגרסיית לאסו (Lasso):

Model	MSE	Avg Error
Baseline	0.02529	0.1590
Full Model ($n = 976$)	0.03010	0.1735
Selected Model ($n = 45$)	0.01842	0.1357
Lasso ($\lambda = 0.00238$)	0.01802	0.1342

במאמר זה מצאו שהתכונות האקוסטיות אינן משפיעות על החיזוי, סיבה סבירה לכך היא שיש הרבה וריאציות בתכונות האקוסטיות בתוך שיר בודד שמקשות על חילוץ מדדים המייצגים שיר שלם. זאת לעומת סוג ז'אנר ושנת יציאה של שיר אשר טובים בהרבה ומשקפים בצורה יותר מדויקת תכונות של שיר.

ב. **<u>מאמר שני:</u>**

Raza, A. H., & Nanath, K. (2020, July). "Predicting a Hit Song with Machine Learning: Is there an apriori secret formula?". In 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA) (pp. 111-116). IEEE.

מטרת מחקר זה הייתה לנסות לחזות האם השיר יהיה להיט או לא לפי הנתונים המוזיקליים הטכניים של השירים ובנוסף ע"פ סנטימנט של הטקסט בשיר. תרומתו העיקרית של מאמר זה הייתה בכך שזו הייתה הפעם הראשונה שבה ניסו לחזות האם שיר יהיה להיט באמצעות חיבור סנטימנט לטקסט השיר ועל פי מידע טכני של שיר ולפני פרסומו, ולא על פי מידע שנאסף לאחר פרסום השיר. השיטות בהם השתמשו במאמר הינם: Logistic Regression, Decision Tree, Random Forests ו-Naïve Bayes. תוצאות המאמר היו כי המודל המדויק ביותר הינו רגרסיה לוגיסטית, עם דיוק של 52%. לעומת זאת, המודל הפחות מדויק, היה עץ ההחלטות אם כי ההבדל לא היה משמעותי. אך הם לא הגיעו למסקנות משמעותיות בנוגע למודלים כיוון שהיו תוצאות מאוד קרובות וכמעט רנדומליות.

Model	Accuracy	Precision
Logistic Regression	52.0%	0.5
Decision Tree	50.5%	0.428
Random Forests	51.0%	0.488
Naïve Bayes	51.1%	0.49

אחת המסקנות המעניינות היו ש-Danceability היה המאפיין הקריטי ביותר בקביעת הצלחתו של שיר. בנוסף, למרות שהסנטימנט היה צפוי להיות בין התכונות המשמעותיות שיקבעו את הצלחתו של שיר, ציפייה זו לא באה לידי ביטוי בתוצאות.

4. פתרון המשימה באמצעות CrispDM :

א. Business Understanding: בנוסף לזה שחלק זה נועד להבין את כוונת הלקוח הוא נועד גם כדי שנוודא שכוונתו של הלקוח תואמת את כוונתנו ושהיא אפשרית לביצוע. שלב זה מחולק לשלושת השלבים הבאים:

(1) Statement Of Business Objective: שלב זה מגדיר את המטרה של הלקוח, היעדים שלו, מה הצורך שלו ומדוע הוא מתעניין בשירות שלנו. כלומר בעזרת שלב זה נבין את האינטרס של הלקוח. כיוון שבעבודה זו אין לנו לקוח רשמי, נסקור אפשרויות שונות של בעלי אינטרסים שמשימות החיזוי תוכל לעניין אותם:

(א) זמרים: היעד של הזמרים הוא גידול בהכנסות ובהון העצמאי שלהם, באמצעות ביצוע שירים שיש להם פוטנציאל להפוך לפופולריים. בקבלת החלטה של זמר אם לבצע שיר מסוים, יש יתרון לשירים אם פוטנציאל גבוה להיות פופולריים בעתיד. חיזוי של שירים פופולריים יסייעו לזמרים בקבלת החלטה מושכלת האם לבצע את השיר שמציעים להם. חיזוי פופולריות של שירים יכול לחסוך להם עלויות בהקלטה ובפרסום שירים שלא יהיו הצלחה גדולה ואף יוכלו לגרום להפסדים.

(ב) אמצעי תקשורת (רדיו וטלוויזיה): היעד של אמצעי תקשורת הוא גידול ברייטינג כתוצאה מהשמעת שירים בפרסומות ובתוכניות אשר ימשכו את הקהל, לכן

בקבלת ההחלטות שלהם הם יעדיפו להשמיע שירים בעלי פוטנציאל גבוה להיות פופולריים כדי להעלות את הרייטינג. לכן חיזוי פופולריות של שירים יכול להעלות את ההכנסות שלהם באמצעות עליית רייטינג.

(ג) <u>אפליקציות מוזיקה מבוססות המלצה (כמו ספוטיפי ויוטיוב):</u>

חברות כמו ספוטיפיי (שמשם הדאטה מגיע) יתעניינו מאוד בתוצר כזה. אחת הסיבות היא שהם המקום המושלם לתחזק ולשפר מוצר שכזה כי אצלם נמצא הדאטה. בנוסף, יועיל להם בקבלת החלטות לגבי תיעדוף מוזיקה ומערכות ההמלצה שלהם ללקוחותיהם. ליכולת לבצע תחזיות מדויקות לגבי הפופולריות של שירים יש גם השלכות על הצעות מוזיקה מותאמות אישית ולכן חיזוי פופולריות של שירים יעלה את ההכנסות שלהם בכך שאנשים יעדיפו מערכת המלצה טובה יותר וירצו להשתמש בשירות שלהם.

(2) <u>Statement Of Data Mining Objective</u>: בשלב זה נרצה להבין מהי בעיית ה-Machine Learning אותה נרצה לפתור, כלומר שלב זה ממיר את הגדרת בקשת הלקוח למשימה של Data Science. הגדרנו בשלב הקודם את הדרישות השונות של הלקוחות השונים, אך בסופו של דבר כל הדרישות של הלקוחות הפוטנציאלים מסתכמות לאותה דרישה אחת מרכזית מנקודת מבט של Data Science אשר הינה בניית מודל כך שבהינתן מאפיינים טכניים של שיר יחזה את רמת הפופולאריות של השיר.

(3) <u>Statement Of Success Criteria</u>: שלב זה מטרתו להגדיר מדדי הצלחה לעבודה שלנו, בשלב זה אנו בוחרים את שיטת האבלואציה ואת הגדרת המטרה אליה נרצה להגיע ע"י שיטת אבלואציה זו. כדי לבחור את המדדים הרלוונטיים נעזרנו בסקירת הספרות ובדברים שלמדנו במהלך הקורס דטא סיינס בתעשייה. כיוון שאנחנו בוחרים לנסות גם אלגוריתמי חיזוי לערכים רציפים וגם אלגוריתמי סיווג לערכים קטגוריאליים נבחר במדדים הבאים לפי הסוגים השונים:

מדד	RMSE	MAE	R^2	Accuracy	Precision	Recall	f-measure
מודלי חיזוי לערכים רציפים (MLP ,Lasso , Linear Regression)	V	V	V				
מודלי סיווג לערכים קטגוריאליים RandomForest , Logistic Regression, XGBoost) (Decision Tree, SVM,				V	V	V	V

<u>פירוט על המדדים:</u>

(א) **מדד RMSE**: שורש ממוצע סכום ריבועי הסטיות, כלומר ההבדל בין האומד לבין מה שנאמד. מייצג את הפער בין התצפיות עצמן לבין ערכי התצפיות שנחזו ע"י המודל, משמש אותנו על מנת לקבוע את המידה שבה המודל מתאים לנתונים, וכן לקבוע האם ניתן להסיר משתנים מסבירים ובכך לפשט את המודל מבלי לפגוע באופן משמעותי ביכולת החיזוי של המודל. החיסרון של

מדד זה הוא שהוא נותן משקל רב לתצפיות חריגות ולכן נשתמש בעוד מדדים ולא רק בו. במדד זה ננסה להגיע לציון כמה שיותר נמוך וקרוב לאפס.

(ב) **מדד MAE** : ממוצע הערך המוחלט של הסטיות. מדד זה בודק את הפיזור הממוצע סביב החציון. הרעיון הוא למצוא בכמה בממוצע התצפיות סוטות בערכן המוחלט מהחציון. במדד זה ננסה להגיע לציון כמה שיותר נמוך וקרוב לאפס.

(ג) **מדד R^2:** מדד לטיב התאמה של המודל, נע בין 0 ל-1 .ככל שמדד זה קרוב יותר ל-1 ההתאמה טובה יותר. לכן ננסה להגיע לתוצאה אשר קרובה ל-1.

(ד) **מדד Accuracy**: זהו מדד הביצועים האינטואטיבי ביותר והוא מתאר יחס של תצפית חזויה נכון לסך התצפיות. דיוק גבוה לא בהכרח מראה שהמודל שלנו טוב, מדד זה יסווג בצורה מדויקת רק כאשר מערך הנתונים שלנו הוא סימטרי- כאשר היחס בין ה-false positive לבין false negative יהיה כמעט זהה. לכן נצטרך להשתמש במדדים נוספים. במדד זה ננסה להגיע לדיוק של יותר מ-0.7, בניסיון לשאוף לתוצאות שקיבלנו בסקירת ספרות (אצלנו יש קושי נוסף כיוון שאנו מסווגים ל- 3 קטגוריות של פופולריות ולא 2 כמו במאמרים).

(ה) **מדד Precision**: מדד זה מראה את היחס בין הפופולריות של השירים שהמודל סיווג בצורה נכונה ביחס לסך התצפיות שהמודל סיווג כנכונות. ציון גבוה מתייחס לשיעור false positive נמוך. במקרה שלנו לא נרצה לסווג שיר כפופולרי כאשר הוא בעצם לא פופולרי, תוצאה שכזו תפגע באמינות המודל ללקוחות הפוטנציאלים. נרצה לקבל Precision גבוה, ולהגיע לציון של יותר מ-60%.

(ו) **מדד Recall**: מדד זה מראה את היחס בין שירים שהמודל סיווג בצורה נכונה ביחס לסך התצפיות של השירים במסווגים עם פופולריות גבוהה. לא נרצה במקרה שלנו לסווג שיר כלא פופולרי כאשר הוא בעצם פופולרי, תוצאה שכזו תפגע בבחירת השירים ללקוחות שלנו. נרצה לקבל Recall גבוה, ולהגיע לציון של כ- 75%.

(ז) **מדד f-measure**: מדד אשר מראה ממוצע משוקלל של recall ו-precision. נרצה להגיע לציון של כ-60%.

ב. **<u>Data understanding:</u>** בשלב זה אנחנו רוצים להבין את המידע שאיתו אנחנו עובדים. המטרה להבין את משמעות המידע אל מול הבעיה העסקית שאיתה אנחנו מתמודדים כמו למשל אילו פיצ'רים יהיו רלוונטיים להמשך וכו'. בנוסף, בשלב זה נרצה לנסות להעריך האם המידע שלם ומספק.

שלב זה הינו שלב מאוד חשוב בפרויקט מכיוון שבלי data אי אפשר לעבוד, המידע הוא הדלק של המודל- אם הdata משובש זה יכול לגרום לטעויות במודל, יכול לגרום

להתמקד בטפל ואף לייצג בצורה לא טובה את המציאות עבור המודל.

<u>להלן פירוט הדברים שאותם בדקנו בשלב זה:</u>

1) בדקנו את ממדי המידע שיש לנו על מנת להבין את הכמויות ולוודא שזוהי כמות סבירה שיכולה להתאים לטובת המודל (500005,18). אנחנו עושים את הבדיקה הזאת גם לפני הכנת המידע וגם אחריו, כרגע המספרים המוצגים הינם לפני שלב הכנת המידע ומחיקת השורות הריקות.
2) בדקנו לכל הפיצ'רים את התכונות הבאות: count, mean, std, min, 25%, 50%, 75%, max. כדי להבין את הפיצ'רים בצורה טובה יותר, להבין אילו שינויים נצטרך לעשות כדי לעבוד איתם בצורה טובה וכדי לחשוב על פיצ'רים עתידים אשר נוכל להוציא מהמידע (מצורף כנספח מס' 1).
3) הדפסנו את חמשת הרשומות הראשונות על מנת להבין כיצד נראה המידע ואילו שינויים נצטרך לעשות בו כדי המודל יוכל לעבוד עם המידע בצורה טובה.
4) בדקנו ערכים ייחודיים עבור פיצ'רים כדי שנוכל לוודא כמה המידע מגוון ואיזה שינויים צריך לעשות בהם כדי להתאים אותם למודל.
5) בדקנו את סוגי הערכים של הפיצ'רים כדי שנוכל לזהות לאילו פיצ'רים נצטרך לעשות התאמה למודל.
6) כדי להבין את המידע בצורה מעמיקה בדקנו את התפלגות הערכים וכדי שנוכל לזהות אם יש התפלגויות מעניינות. בדיקת ההתפלגויות עוזרות לנו להבין גם איזה ערכים כדי לנו לנרמל בהמשך. ראינו שאנו לא מצליחים לבדוק התפלגות לפיצר "tempo" ולאחר בדיקה ראינו שהעמודה מכילה תווים של סימני שאלה כערכים חסרים, מצב שלא מאפשר לעשות התפלגות לכל הערכים, לכן בהמשך בשלב הכנת הנתונים אחרי שנחליף ערכים אלו נבדוק שוב את התפלגות זו.

ג. **<u>Data preparation:</u>** לאחר שהבנו את המידע שיש ברשותנו לעומק, נכין את הנתונים. שלב זה כולל ניקוי מידע (למשל הורדת פיצ'רים שאינם רלוונטיים) ובחירת נתונים סופיים בהם נבחר להשתמש במודל. שלב זה הינו שלב קריטי מכיוון שבלעדיו המודל יכול להתבלבל או לחזות דברים שקריים שנבעו מחוסר איזון בנתונים שהוא קיבל. המודל לא יכול להבין לבד את איכות הנתונים או את המשמעות שלהם, אנחנו צריכים לעשות זאת עבורו כדי שיניב עבורנו את התוצאות האידיאליות ("זבל נכנס זבל יוצא").

כדי שנוכל לפתור את המשימה בצורה טובה, הורדנו תחילה שורות ריקות (היו כ-5 שורות כאלה מספר זניח כך שהמידע שנעבוד איתו עדיין מאוד משמעותי).

1) הפיכת האינדקסים להיות לפי instance_id של השירים ולא לפי קאונטר דיפולטיבי, כדי להקל על אחזור רשומות ופעולת כמו join.

2) הורדנו שורות של שמות האומנים שיש להם רק שיר אחד בלבד. כחלק מתהליך הבנת הדאטה ולאחר ניסיונות לייצר פיצ'רים חדשים ולממש מודל הבנו שהמודל שלנו לא יודע להתמודד עם שמות אמנים שמופיעים פעם אחת בלבד עבור חלוקת המודל ל-train ו-test. זאת מכיוון שבהמשך ייצרנו פיצ'ר בוליאני של פופולריות הזמר, שנקבע על פי ממוצע הפופולריות של השירים שלו ב-train. נוצר מצב שבו היו זמרים חדשים ב-test שלא היה להם שיר ב-train. לכן בחרנו להוריד אותם לאחר שראינו שהם אינם מהווים כמות גדולה, בערך 5% (2,346) מתוך כל הדאטה.
3) הורדנו עמודות: artist_name, track_name. את track_name הורדנו כיוון שהוא מתפקד כמספר סידורי ולכן לא יעזור לנו, אך בפרק הוצאת פיצ'רים כן השתמשנו בו בצורה אחרת. את artist_name הורדנו בגלל שהמידע מכיל יותר מ-6000 זמרים שונים ולכן החלטנו לתת ביטוי שונה להשפעה של הזמר על פופולריות שירו (יורחב בסעיף הוצאת פיצ'רים).
4) המידע הכיל עמודה בשם obtained_date שהכילה חמישה ימים באפריל. ממחקר על הדאטה של Spotify הבנו שיש טעות ומדובר בעצם במקצב (time signature). לכן המרנו עמודה זו להיות מספרית והחלטנו להשתמש בה כמקצב.
5) בשלב הבנת המידע הבנו שיש ערכים חסרים שלא מיוצגים ב-nan: ב-duration_ms היה 1- כערך חסר וב-tempo היה '?'. הפכנו אותם לnan כדי שיאותרו עם שאר הערכים החסרים בשלב השלמת הערכים החסרים.
6) קידוד עמודה קטגוריאלית mode לערך מספרי כדי שנוכל להכניס ולשלב אותם במודלים.
7) קידוד עמודה קטגוריאלית key לפי סדר עליית הסולמות (לכל סולם יש הפרש של חצי טון), כך שקידדנו את key לערך מספרי לפי סדר המפתחות בעולם המוזיקה.
8) קידוד עמודה קטגוריאלית music_genre בעזרת dummies, בשביל שלא תיווצר הטיה למודל רק ע"פ סדר מלאכותי לפיצ'ר.
9) לאחר הכנת הנתונים, וקידוד הערכים הקטגוריאליים בדקנו מחדש את התפלגות הפיצ'ר של tempo.
10) נוודא שאין ערכי null בdata אשר יכולים לשבש או להטעות את המודל שלנו. לאחר בדיקה ראינו שיש ערכים חסרים רק ל-duration_ms ול-tempo אשר אותם ננרמל בהמשך לאחר פיצול הדאטה לtrain ו-test.
11) פיצלנו את הdata לtrain ו-test לפי חלוקה של 20% test ו80% train.

בדקנו את מדדי הטבלאות עבור ה-train וה-test כדי להבין האם מדובר בכמות דגימות סבירה: Train size is 38123 , Test size is 9531.

במהלך הניסויים ביצענו הרצות עם seed 3 שונים (66 ,667,42) כדי לקבל חלוקות שונות (דימוי של cross validation). בשלושת המקרים ה-metrics יצאו דומים עד כדי אלפית בחיזוי ועד כדי מאית בסיווג. לכן השארנו את התוצאות מהריצה על פי seed=42.

12) מילוי ערכים חסרים עבור ערכי tempo ו-duration_ms לפי ממוצע הפיצ'ר עבור סוג הז'אנר של אותו השיר. בחרנו לעשות ממוצע לפי ז'אנר כיוון שראינו שלרוב השונות יותר נמוכה עבור ז'אנר ספציפי מאשר לכל הdata ביחד, כלומר במרבית הז'אנרים סטיית התקן של tempo ו-duration_ms הייתה נמוכה מסטיית התקן הכללית. עשינו פעולה זאת עבור ה-train ועבור ה-test בנפרד כדי לא להסתמך על מידע מהtest באימון.

13) נרמלנו את כל העמודות לתחום של (0,1) בשיטת min-max כולל משתנה המטרה פופולריות כך שכל העמודות יהיו באותו תחום (את duration_ms הימרנו קודם לשעות ורק אז לתחום שציינו).

14) <u>הוספת פיצ'רים</u> שלדעתנו יעזרו למודל לחזות בצורה טובה יותר פופולריות של שיר:

(א) key_signature אשר מתאר פיצ'ר של תווים. על פי התאוריה של המוזיקה כל תו מתאים לזוג ממכפלה קרטזית של מצב וסולם (modeXkey). לדוגמא הסולם C במצב minor מתאים לתו [2].

(ב) פרסרנו את שמות השירים ומצאנו את עשרת המילים הכי שכיחות. מהם ייצרנו פיצ'רים dummies. אנו נוטים לחשוב שמילים שכיחות בשמות של שירים יכולות להוות אינדיקטור משמעותי לפופולריות של שיר (מצורף WordCloud והתפלגות של ניתוח המילים החזקות בשירים בנספח מס' 3).

(ג) is_popular_artist אשר מתאר את האומנים לפי הפופולריות שלהם. (אם ממוצע הפופולריות של הזמר גדול מממוצע הפופולריות הכללי נסמן שהוא פופולרי, אחרת נסמן שהוא לא), לדעתנו, הסיכוי של שיר להיות פופולרי עולה כאשר הוא מבוצע ע"י אמן פופולרי.

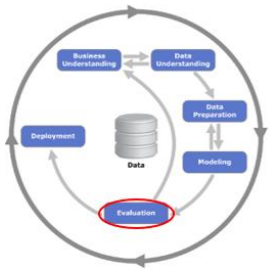
15) בדיקת קורלציה בין הפיצ'רים כדי לזהות פיצ'רים נוספים שיכולים להיות משמעותיים עבור המודל שלנו וכדי לבדוק את איכות הפיצ'רים שייצרנו (את מטריצת הקורלציה ניתן לראות בנספח מס' 4). ראינו כי יש לפופולריות השיר קורלציה גבוהה מאוד וישרה עם is_popular_artist והפוכה עם Anime. נתונים

[2] https://dsmusic.com.au/wp-content/uploads/2012/01/key_sig_worksheet.pdf

אלו גורמים לנו לצפות שפופולריות הזמר וז'אנר אנימה ישפיעו רבות על המודלים. בנוסף, ניתן ללמוד על אמינות המידע והנתונים על ידי הקורלציה הישרה או ההפוכה שמתקיימת בין פיצ'רים מסוימים. למשל רואים שיש ל-loudness קורלציה הפוכה עם acousticness ולעומת זאת קורלציה ישרה עם energy, נתונים אלה הגיוניים.

ד. **Modeling**: את שלב זה חילקנו לשני חלקים כך שבחלק הראשון ניסינו מודלים שמתאימים למשימת חיזוי לערך רציף. בחלק השני, ניסינו מודלים עבור משימת סיווג על משתנה מטרה קטגוריאלי, בהם ביצענו דיסקרטיזציה בשיטת Equal frequency על משתנה הפופולריות ל-3 קטגוריות (High, Moderate, Low) עפ"י השלישונים. בחרנו להשתמש במודלים הכי רלוונטיים אשר בהם נתקלנו בסקירת הספרות ובנוסף במודלים אשר למדנו בקורס ולדעתנו יכולים להתאים לטובת המשימה. עבור מודלי חיזוי לערכים רציפים נשתמש ב-Linear Regression, Lasso ו-MLP . עבור מודלי סיווג לערכים קטגוריאליים נשתמש ב-Logistic Regression , Decision Tree, XGBoost, RandomForest ו-SVM . מדובר במודלים הכי קלאסיים בתחום כיום והכי מתאימים למשימות שהגדרנו. בנוסף ביצענו ניסוי בעזרת GridSearch לטובת אופטימליות של מודל אחד בכל משימה: במשימת חיזוי ביצענו על MLP ובמשימת סיווג ביצענו על RandomForest (מצורף בנספח מס' 5 פרמטרים אופטימליים עבורם).

ה. **Evaluation**: בשלב זה נעריך את המודלים שאימנו. מפאת חוסר המקום נראה רק את התוצאות הסופיות. מהלך העבודה היה איטרטיבי כך שביצענו מודל וחזרנו אחורה לשפר או לבדוק אם יש לנו טעות בתהליך. כפי שציינו התמקדנו בשתי אופציות שונות של מודלים כך שהראשונה היא עבור ערכים רציפים והשנייה עבור ערכים קטגוריאליים. את הערכת המודלים ביצענו לפי הטבלה שתיארנו בשלב Statement Of Success Criteria לפי המדדים שהגדרנו לכל מודל. התוצאות שקיבלנו עבור מודלים לערכים רציפים הינם:



Model	RMSE	MAE	R^2
Linear Regression	0.09784	0.0738	0.6415
MLP	0.09684	0.0727	0.6488
Grid – MLP	**0.09582**	**0.0723**	**0.6561**
Lasso	0.09784	0.0738	0.6415

התוצאות שקיבלנו עבור מודלים לערכים קטגוריאליים הינם:

Model		Accuracy	Precision	Recall	f-measure
Logistic Regression	Low	0.716	0.722	0.849	0.781
	Moderate		0.559	0.556	0.557
	High		0.864	0.725	0.788
Decision Tree	Low	0.691	0.803	0.676	0.734
	Moderate		0.503	0.664	0.573
	High		0.825	0.727	0.772
SVM	Low	0.696	0.731	0.798	0.763
	Moderate		0.525	0.543	0.534
	High		0.822	0.728	0.772
XGBoost	Low	**0.72**	0.719	0.868	0.786
	Moderate		0.571	0.54	0.555
	High		0.858	0.731	0.789
RandomForest Grid	Low	0.717	0.701	0.904	0.79
	Moderate		0.579	0.483	0.527
	High		0.85	0.737	0.79

5. **תוצאות ומסקנות**:

מטרת המשימה הייתה לחזות פופולריות של שיר לפני יציאתו לשוק על סמך הנתונים הטכניים שלו, תוך כדי עמידה במדדים שהגדרנו. כדי לנסות ולהשיג את המטרה שלנו חקרנו את המידע שקיבלנו, ניקינו את סט הנתונים, יצרנו פיצ'רים חדשים בעזרת הסקת מידע חדש, בעזרת סקירת הספרות, בעזרת כלים שנלמדו במסגרת הקורס ובעזרת התייעצות עם domain expert. מימשנו מודלי חיזוי וסיווג כפי שלמדנו וביצענו אבלואציה לתוצאות. ביצענו משימות בדרגת קושי גבוהה יותר ממה שעלה במאמרים. ראשית ביצענו חיזוי לערך רציף ושנית ביצענו סיווג ל3 דרגות של פופולריות ולא לערך בינארי כמו במאמרים שסקרנו. לכן בהתחשב בזה אנו חושבים שעמדנו ברף של המדדים שהצבנו לעצמנו.

בתוצאות של מודלי חיזוי עבור ערכים רציפים, מודל רשתות ניורונים MLP עם Grid (הפרמטרים בנספח מספר 5) נתן את התוצאות הטובות ביותר. אומנם ההבדלים מאוד קטנים, אך ניתן לראות שMLP- טוב יותר מהרגרסיה לינארית באופן עקבי בכל המדדים. זאת כיוון שב-MLP השכבה הראשונה של הצמתים מגלמת את קבוצת התכונות המקורית, והשכבה האחרונה של הצמתים מייצגת תכונות ברמה גבוהה יותר המושפעות ממספר תכונות מקוריות, המשקלים של צמתים אלה נלמדים באמצעות התפשטות לאחור. עבור מודלי חיזוי לערכים רציפים הפיצ'רים המשמעותיים היו הז'אנרים Anime, Rap, Rock ו-duration_ms . בתוצאות של מודלי סיווג עבור ערכים קטגוריאליים, מודל XGBoost נתן את הדיוק הגבוה ביותר. נשים לב כי השיטות האחרות הביאו דיוקים דומים עד כמעט זהים אחד לשני. ניתן להסיק מכך כי הנתונים והפיצ'רים מפוזרים בצורה

טובה, ללא הטיות אשר מקבלות עדיפות באלגוריתם מסוים על פני אחר. במודלי הסיווג, הפיצ'ר המשמעותי ביותר הינו "is_popular_artist", פיצ'ר שמתאר האם הזמר פופולרי או לא. בנוסף, גם לפיצ'ר של ז'אנר Aime יש השפעה גבוהה (שלילית) על הפופולריות. ניתן לראות שהיה למודלים יותר קשה לחזות פופולריות ממוצעת(Moderate), זאת לדעתנו כיוון שכנראה הדיסקרטיזציה לא הייתה טובה, אנחנו עשינו לפי עומק שווה (אותה כמות ערכים בכל קטגוריה), אך אולי לפי רוחב שווה התוצאות יהיו טובות יותר. (מצורף בנספח מס' 6 דירוגי הפיצ'רים שקיבלנו עבור חלק מהמודלים).

מבחינת המשמעות בעולם הבעיה: הצגנו בחלק של הבנת הBusiness מספר לקוחות אפשריים, כך שהמטרה של כולם הייתה לקבל מודל אשר יחזה האם השיר פופולרי לפני יציאתו לשוק. נוכחנו לדעת כי עיקר ההשפעה על פופולריות של שיר נובעת מהז'אנר ומפופולריות הזמר ששר אותה. מסקנות אלו יכולות לעזור לתחנות רדיו, לתוכנות מהתחום ומפיקים אך לא כל כך לזמרים שלא יכולים להחליף את עצמם וגם פחות קורה שזמר מחליף ז'אנר שהוא בדרך כלל שר אותו.

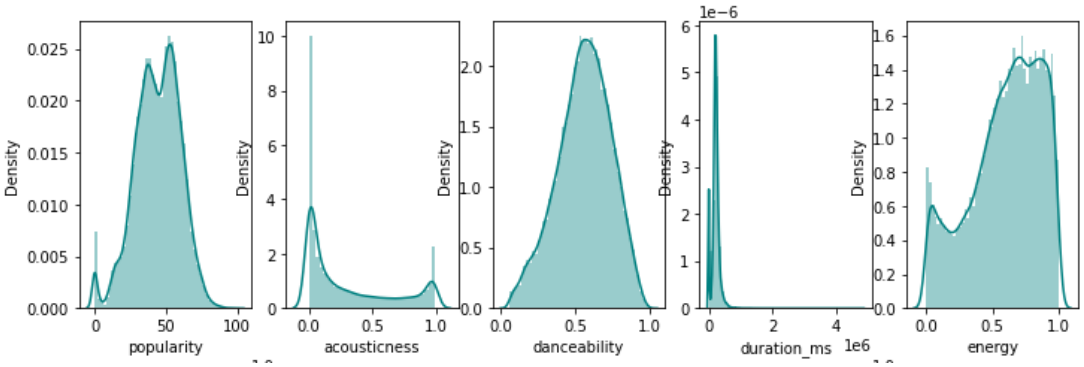
אילו היה לנו עוד מקום במסמך ועוד משאבים היינו מבצעים את הדברים הבאים:
שואבים עוד מידע מSpotify API . מהתייעצות עם ממוחה התוכן הבנו כי מורכבות של שיר אשר מאופיינת עם כמות ומגוון אקורדים הינה תופעת לוואי של שיר פופולרי. לכן זה מידע שהיינו רוצים לשאוב. בנוסף, גם המקצב מעיד על מורכבות. בשירים שלנו השונות במקצב הייתה קטנה מאוד ולכן הפיצ'ר לא השפיע, לכן היינו ממליצים לשאוב עוד שירים יותר מגוונים. אנו ממליצים בנוסף לשאוב מידע נוסף של מילות הטקסט ולנתח אותם מבחינת סנטימנט שגם יכול להשפיע על פופולריות השירים.
מבחינת קוד היינו מוסיפים עוד מודלים למשימת חיזוי הפופולריות הרציפה כדי לקבל תמונה מלאה יותר. היינו משתמשים בkbest כדי להבין אם יש פיצ'רים שמטעים את המודלים ושכדאי להסיר. בנוסף, היינו מריצים הרבה ניסויים עם חלוקות שונות של האימון והמבחן כדי לקבל מודלים אמינים יותר.

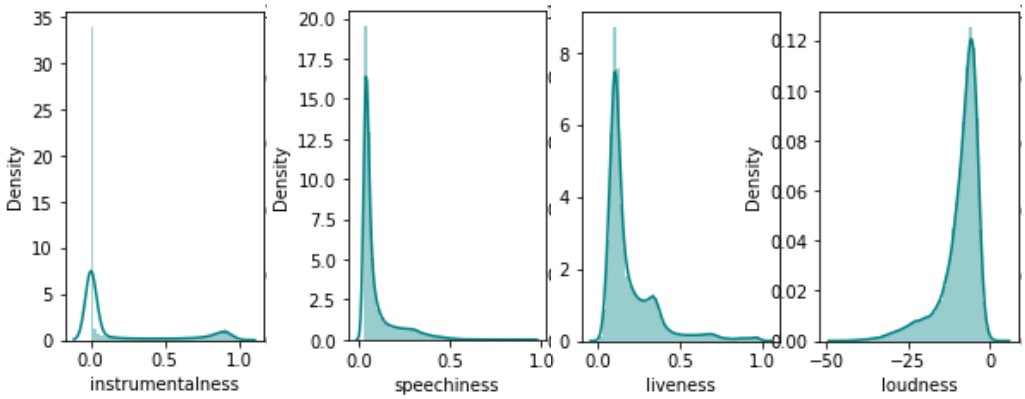
נספחים:

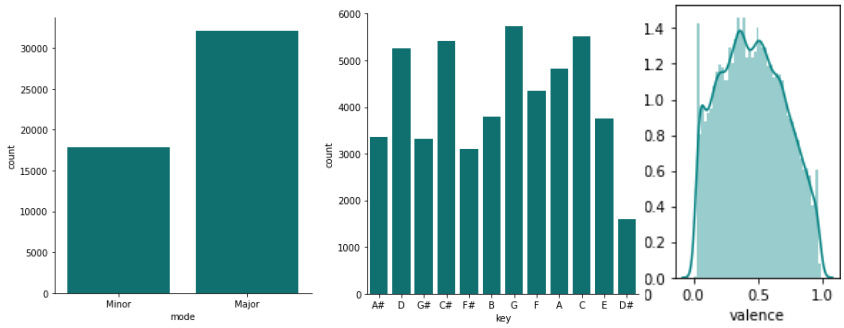
נספח מס' 1:

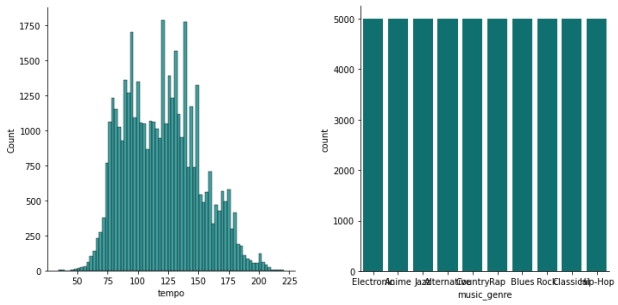
	count	mean	std	min	25%	50%	75%	max
instance_id	50000.0	55888.396360	20725.256253	20002.000000	37973.5000	55913.500000	73863.250000	91759.000
popularity	50000.0	44.220420	15.542008	0.000000	34.0000	45.000000	56.000000	99.000
acousticness	50000.0	0.306383	0.341340	0.000000	0.0200	0.144000	0.552000	0.996
danceability	50000.0	0.558241	0.178632	0.059600	0.4420	0.568000	0.687000	0.986
duration_ms	50000.0	221252.602860	128671.957157	-1.000000	174800.0000	219281.000000	268612.250000	4830606.000
energy	50000.0	0.599755	0.264559	0.000792	0.4330	0.643000	0.815000	0.999
instrumentalness	50000.0	0.181601	0.325409	0.000000	0.0000	0.000158	0.155000	0.996
liveness	50000.0	0.193896	0.161637	0.009670	0.0969	0.126000	0.244000	1.000
loudness	50000.0	-9.133761	6.162990	-47.046000	-10.8600	-7.276500	-5.173000	3.744
speechiness	50000.0	0.093586	0.101373	0.022300	0.0361	0.048900	0.098525	0.942
valence	50000.0	0.456264	0.247119	0.000000	0.2570	0.448000	0.648000	0.992

נספח מס' 2:


Density
popularity
acousticness
danceability
duration_ms
1e−6
1e6
energy


Density
instrumentalness
speechiness
liveness
loudness


count
Minor
Major
mode
A# D G# C# F# B G F A C E D#
key
valence

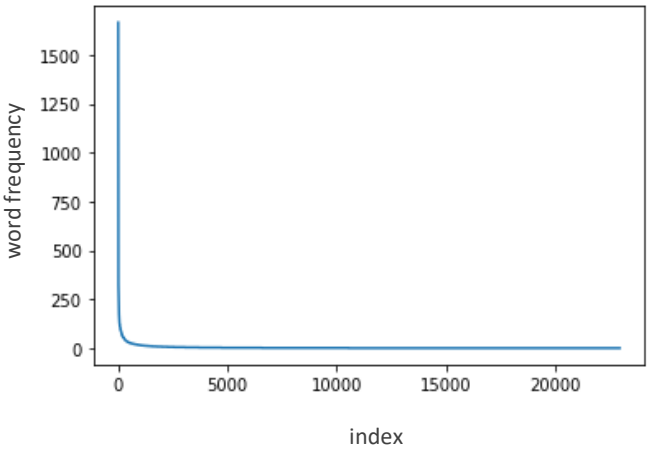

Count
tempo
count
Electronic Anime Jazz Alternative Country Rap Blues Rock Classical Hip-Hop
music_genre

<u>נספח מס' 3:</u>

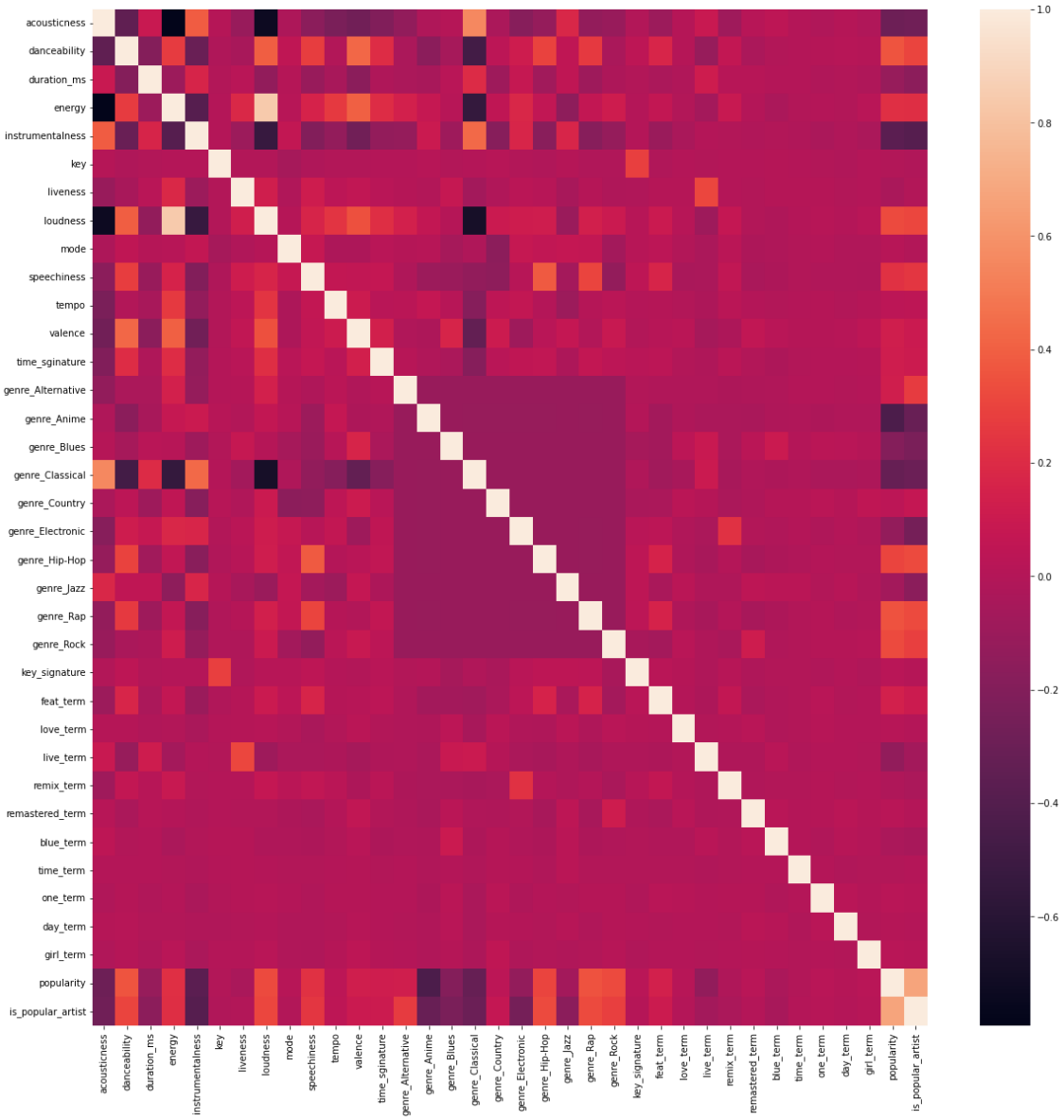
- עבור כל המילים שהיו בשמות השירים הפכנו לאותיות קטנות.

Most used words is:

התפלגות המילים:



נספח מס' 4:



נספח מס' 5:

הפרמטרים האופטימליים ברשת הניורונים MLP עם GridSearch:

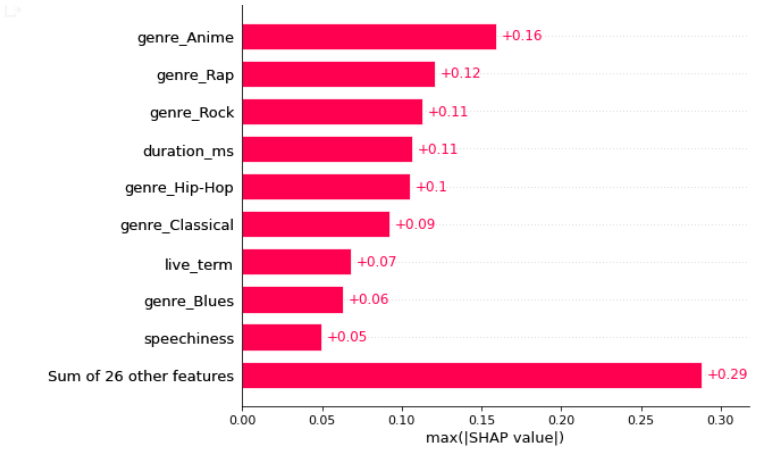
```
Best parameters found:
 {'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'adam'}
```

הפרמטרים האופטימליים RandomForest עם GridSearch:

```
Best parameters found:
 {'criterion': 'gini', 'max_depth': 8, 'max_features': 'auto', 'n_estimators': 500}
```

נספח מס' 6: SHAP עבור מודל חיזוי ועבור מודל רגרסיה

מודל חיזוי - עבור רגרסיה ליניארית:



מודל סיווג- עבור עץ החלטה: