דוח מסכם – ביג דאטה

תיאור הפרויקט

מטרת הפרויקט היא לבצע ניתוח נתונים אקספלורטיבי (EDA) בנתוני Youtube וSpotify ולפתח מודל חיזוי להערכת מספר הצפיות בסרטוני Youtube. פרויקט זה נועד להבין את הגורמים המשפיעים על הפופולריות של סרטוני YouTube וליצור מודל שיכול לחזות במדויק את הצפיות שסרטון צפוי לקבל.
האתגר איתו נתמודד הוא חיזוי של מספר הצפיות שסרטון יקבל בYoutube.
אנו רואים במשימה חשובה ומשמעותית עבור יוצרי תוכן, משווקים ובעלי פלטפורמות מכיוון שהיא יכולה לעזור להם להבין את פוטנציאל החשיפה וההשפעה של סרטון. על ידי ניתוח הנתונים הזמינים וזיהוי דפוסים ויחסים, נוכל לחשוף תובנות התורמות לבניית מודל חיזוי יעיל.
הפרויקט יכלול ביצוע EDA במערך הנתונים של YouTube ו Spotify -תהליך זה יכלול בחינת התפלגות הצפיות, לייקים וההשמעות ומשתנים רלוונטיים נוספים. בחינת מגמות ומתאמים בתוך הנתונים יכולה לעזור לזהות משתנים מנבאים פוטנציאליים של צפיות בסרטון.
לאחר השגת תובנות מה ,EDA-השלב הבא הוא פיתוח מודל חיזוי. בשלה זה נשתמש באלגוריתמי למידת מכונה כדי לאמן מודל על הנתונים שברשותנו, המטרה היא ליצור מודל שיכול להעריך במדויק את מספר הצפיות שסרטון YouTube עשוי לקבל על סמך התכונות שלו, כגון כותרת, תיאור, קטגוריה, משך ותכונות רלוונטיות אחרות.
על ידי פיתוח מוצלח של מודל חיזוי צפיות, יוצרי תוכן ומשווקים עשויים להבין טוב יותר את הגורמים התורמים לפופולריות של וידאו ולבצע אופטימיזציה של אסטרטגיות הפעולה שלהם בהתאם.
בעלי פלטפורמות יכולים גם להפיק תועלת ממודלים כאלה על ידי זיהוי סרטונים בעלי פוטנציאליות להיות ויראליים וכך לשפר את חווית המשתמשים ופופולריות הפלטפורמה.

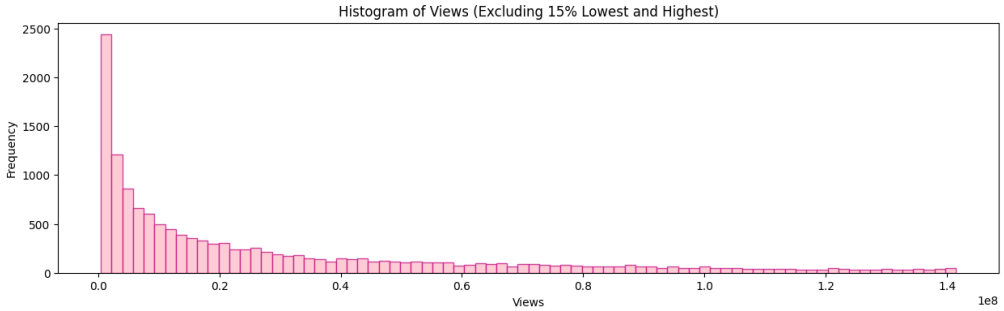
תיאור הנתונים

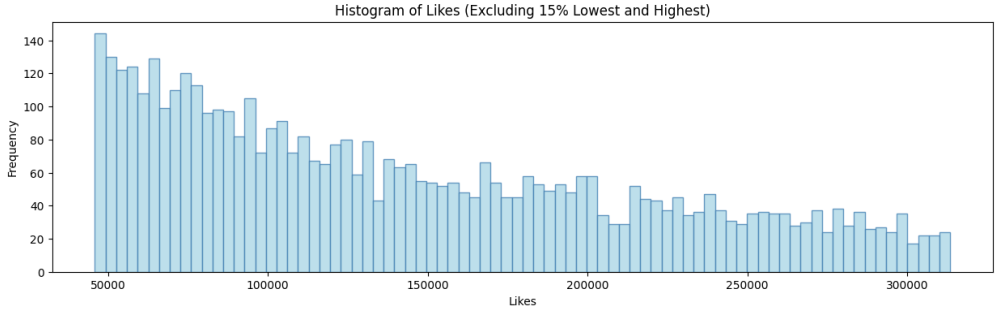
הדטסט מכיל מידע רב על שירים הקיימים בפלטפורמות ההאזנה YouTube וSpotify, המידע כולל פריטים כלליים וגם פריטים שנאספו לאחר פרסום השיר. בסה"כ במערך הנתונים יש 26 עמודות וכ-20720 רשומות(שירים).
קיימים במערך הנתונים, 14 עמודות נומריות, 8 מסוג string, 2 עמודות של כתובות URL ו4 עמודות מסוגים אחרים.
<u>מערך הנתונים כולל נתונים מהסוגים הבאים:</u>

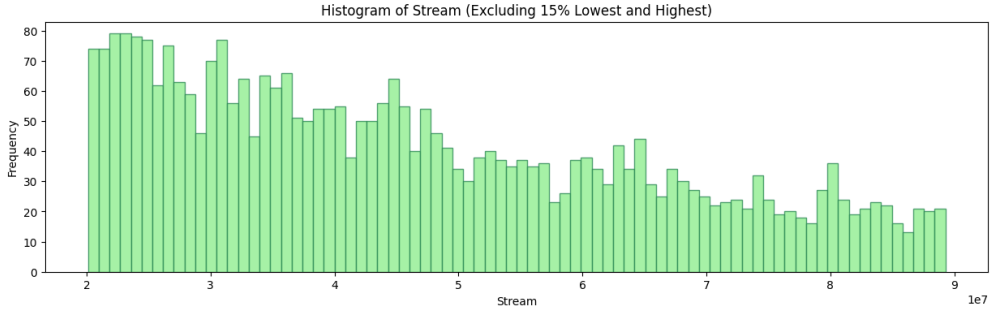
1. מטא נתונים של השיר: שם האמן, שם האלבום, שם השיר, תאריך פרסום השיר, ז'אנר, אורך השיר וכו'.
2. תכונות אודיו: תכונות המתארות את מאפייני האודיו של השיר, כגון יכולת ריקוד, אנרגיה, עוצמה, קצב, מפתח וכו'.
3. מדדי פופולריות: מידע על הפופולריות של השיר, כגון מספר ההשמעות, העוקבים ומדדי מעורבות אחרים.
4. מדדי מעורבות: מידע על אינטראקציות של משתמשים עם השירים, כמו ספירת צפיות, לייקים, האזנות, תגובות וכו'.

מערך הנתונים נאסף באמצעות שיטות שונות, כגון קריאות API לפלטפורמות Spotify ו-YouTube או טכניקות גירוד אינטרנט.

נביט בהתפלגות ההאזנות, הצפיות והלייקים לכלל השירים במאגר:







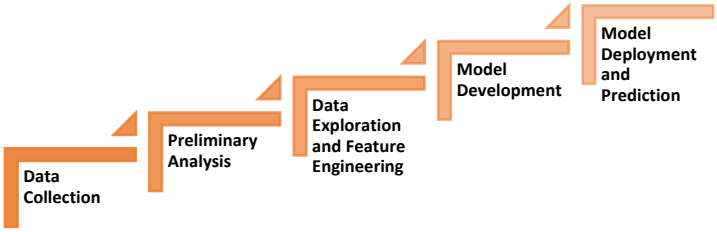
נשים לב כי יש פער משמעותי בין כמות השירים בצד הימני של שלושת הגרפים שבהם יש מעט שירים שלהם כמות גדולה מאוד של האזנות, צפיות ולייקים לעומת הצד השמאלי שבו יש כמות גדולה מאוד של שירים שלהם מעט האזנות.

נביט ברשומה לדוגמה ממערך הנתונים, ונתמקד בנתוני השיר שרלוונטיים למשימה שלנו:

Artist	Red Hot Chili Peppers
Track	Californication
Album Type	album
Danceability	0.592
Energy	0.767
Key	9
Loudness	-2.788
Speechiness	0.027
Acousticness	0.0021
Instrumentalness	0.00165
Liveness	0.127
Valence	0.328
Tempo	96.483
Duration_ms	329733
Views	1018811259
Likes	4394471

Comments	121452
Licensed	TRUE
Stream	1055738398

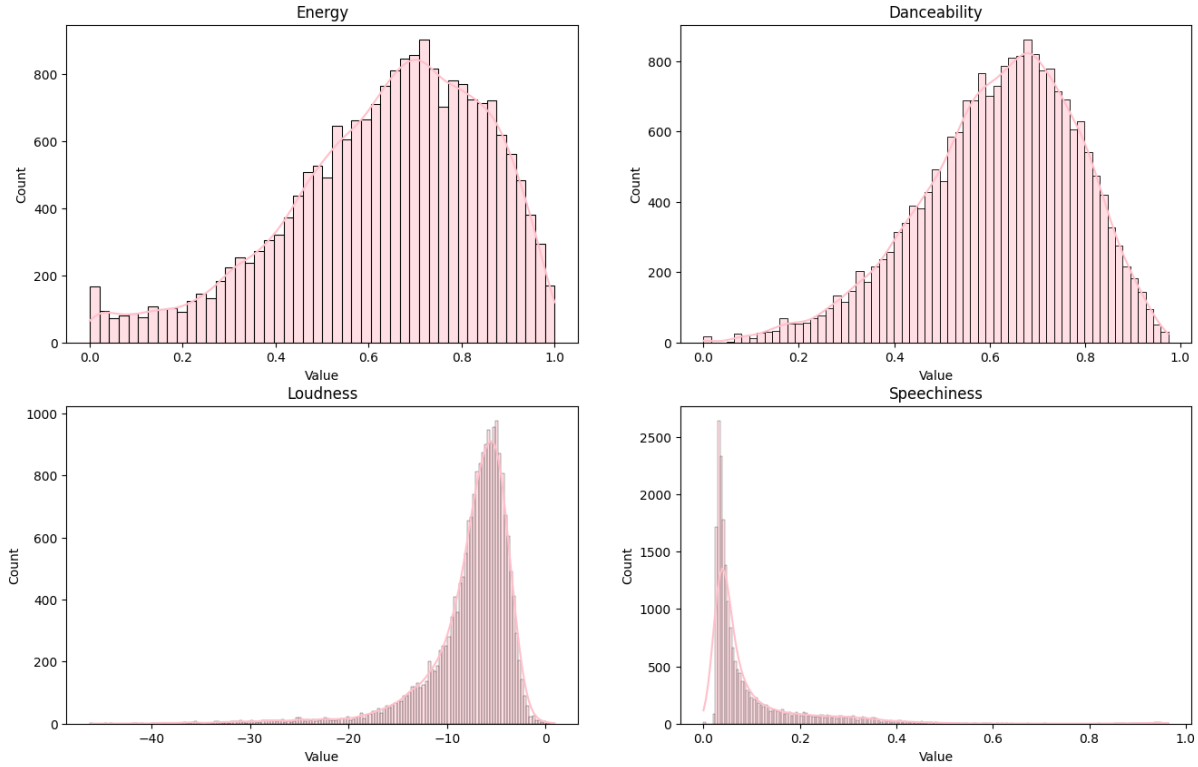
Pipeline



1. Data Collection
   א. בחירת סט הנתונים ואיסוף נתונים נוספים במידת הצורך.
2. Preliminary Analysis
   ביצוע data cleaning ו- pre-processing הכולל טיפול בערכים חסרים, אנומליות ונתונים חריגים ובחוסר עקביות.
   א. descriptive statistics כדי להבין את המאפיינים הבסיסיים של הנתונים, כגון ממוצע, חציון, סטיית תקן וכו.'
   ב. ייצוג גרפי של הנתונים באמצעות - histograms, box plots, scatter plots וכו' כדי לחקור ולהבין יותר לעומק את ההתפלגות, הקשרים והדפוסים בתוך מערך הנתונים.
3. Data Exploration and Feature Engineering
   א. מחקר והבנת הקשר בין משתנים על מנת לזהות משתנים מנבאים פוטנציאליים.
   ב. ביצוע correlation analysis כדי לקבוע את עוצמת וכיוון הקשר בין משתנים.
   ג. ביצוע feature engineering ליצירת משתנים חדשים או שינוי של משתנים קיימים כך שיסייעו לשפר את תוצאות הניבוי של המודל.
4. Model Development
   א. חלוקה של הדאטא ל-training and testing sets.
   ב. בחירת אלגוריתמי למידת מכונה מתאים (למשל regression, random forests, gradient boosting) עבור משימת החיזוי שהצגנו מעלה.
   ג. אימון המודל על ה-training sets .
   ד. ביצוע אופטימיזציה fine-tune לhyperparameters של המודל באמצעות cross-validation אוgrid search .
   ה. הערכת ביצועי המודל באמצעות mean squared error, R-squared למשל.
5. Model Deployment and Prediction
   א. לאחר השלבים הקודמים כעת ניתן להשתמש במודל לטובת משימות חיזוי על נתונים חדשים שלא נראו קודם.
   ב. הערכת ביצועי המודל.

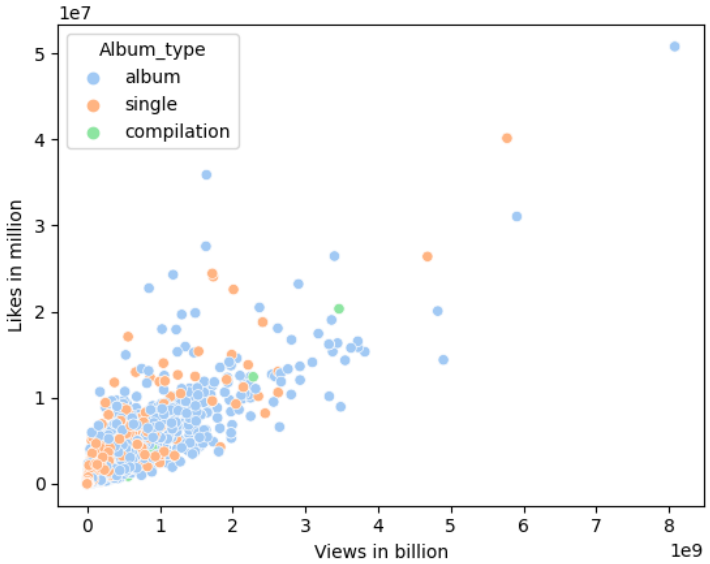
תוצאות הניתוח המקדים

1. <u>ניתוח ההתפלגויות של משתנים מסבירים נומריים</u>



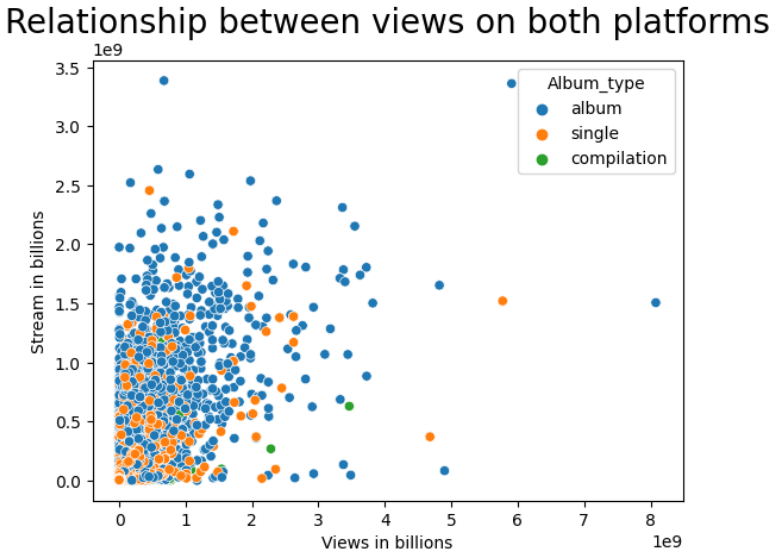
בתרשים המצורף מעלה ניתן לראות פלט היסטוגרמה של הפיצ'רים – energy, danceability, loudness, speechiness.
ציר ה-x מייצג את ערכי הפיצ'ר וציר ה-y מייצג את מספר ההתרחשויות, ההיסטוגרמה כוללת kernel density estimation המספקת smooth estimation של probability density function.
היסטוגרמות אלה מאפשרות לנו לבחון shape, central tendency, and spread של הערכים וכך להסיק תובנות לגבי המאפיינים והדפוסים של הנתונים.
כאן למשל ניתן לראות כי danceability וenergy מתפלגים בצורה יחסית גאוסיאנית, כלומר ככל שנתקרב לממוצע נצפה לראות מספר רשומות גדול יותר ולהיפך.
כמו כן, ניתן לראות כי loudness ו-speechiness מתפלגים יחסית כ- Heavy-tailed distributions, כלומר נצפה לראות את רוב הרשומות סביב ערך מסוים ועם זאת רשומות רבות עם ערכים שונים.

2. <u>ניתוח הקשר בין לייקים לצפיות</u>



הגרף מתאר את הקשר בין צפיות ולייקים, ה-scatter plot מאפשר לנו לזהות את טיב הקשר בין מספר הצפיות למספר הלייקים. כל נקודה מייצגת רשומה ספציפית, והמיקום על הגרף מראה כמה צפיות הוא קיבל לעומת כמה לייקים הוא צבר.
השימוש בצבעים שונים עבור פיצ'ר סוג_אלבום מסייע לנו לזהות בצורה חזותית דפוס או הבדלים בלייקים ובצפיות בהתבסס על סוג האלבום.
ניתן להסיק מהגרף כי קיים קשר חיובי בין מספר הצפיות לבין מספר הלייקים בלי הבדל משמעותי לגבי סוג האלבום.
כלומר, עבור כל סוג אלבום נצפה לראות עבור מספר רב של צפיות גם מספר רב של לייקים.

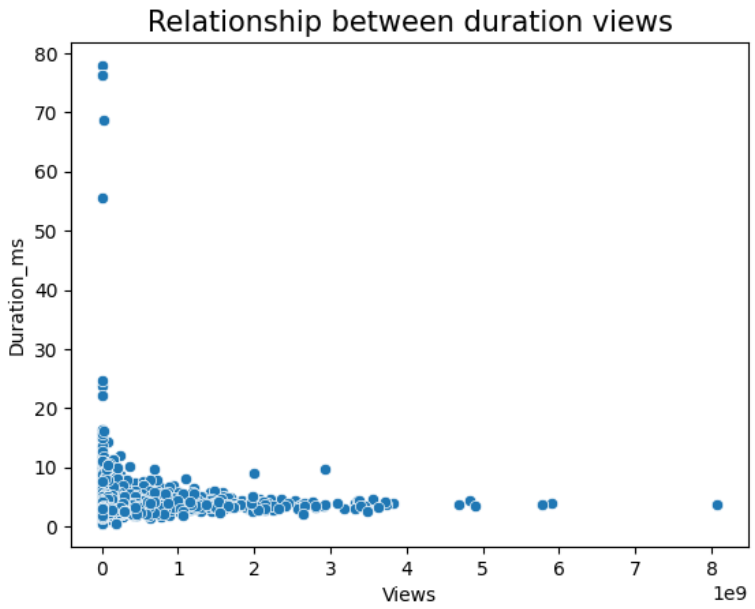
3. <u>ניתוח הקשר בין צפיות להשמעות</u>



כפי שהסברנו קודם, scatter plot מאפשר לנו לזהות את טיב הקשר בין מספר הצפיות למספר ההשמעות של השיר. כל נקודה מייצגת רשומה ספציפית, והמיקום על הגרף מראה כמה השמעות יש לשיר לעומת כמה צפיות הוא צבר.

השימוש בצבעים שונים עבור פיצ'ר סוג_אלבום מסייע לנו לזהות בצורה חזותית דפוס או הבדלים בהשמעות ובצפיות בהתבסס על סוג האלבום.
ניתן להסיק מהגרף כי לא קיים קשר משמעותי חיובי או שלילי בין מספר הצפיות לבין מספר ההשמעות וכן כי גם לא מסקנה זו לא מושפעת מסוג האלבום.

4. הקשר בין משך השיר לבין מספר הצפיות

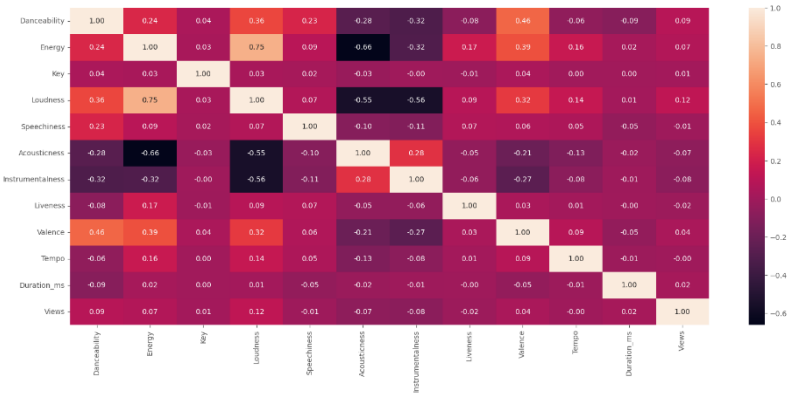


בגרף זה אנו מציגים scatter plot כדי לחקור את הקשר בין משך השיר (בדקות) לבין מספר הצפיות.
ראשית ביצענו טרנספורמציה של נתונים – נחלק הפיצ'ר 'Duration_ms' ב-1000 כדי להמיר את הערכים ממילי-שניות לשניות ולאחר מכן נחלק ב-60 כדי להמיר משניות לדקות.
המיקום של כל נקודה על הגרף מייצג את משך הזמן בדקות על ציר ה-y ואת מספר הצפיות על ציר ה-x.
ניתן להסיק מהגרף כי אין מגמה או קשר משמעותי בין הפיצ'רים. ניתן לראות ששירים שאורכן עד 10 דקות מקבלים מספר צפיות לאורך כל טווח הערכים האפשרי.
כמו כן, ניתן לראות כי שירים שאורכים חריג קיבלו מספר צפיות אפסי.

העיבוד המקדים

בשלב העיבוד המקדים בחרנו במספר שיטות שיתאימו את הנתונים בצורה הטובה ביותר למשימה שלנו, פעלנו כך:

1. הורדת עמודות עם ערכים שאינם רלוונטיים לבעיה – כגון שם האלבום.
2. הורדת השורות בהן יש ערכים חסרים.
3. שימוש בlabel encoder לצורך ביצוע המרה של עמודות קטגוריאליות לעמודות נומריות.
4. שימוש בStandardScaler לצורך נרמול עמודות שלהן שונות גבוהה.
5. הפרדת עמודת המטרה שלנו מהשאר – עמודת הviews.



בחירת המאפיינים

בחירת הפיצ'רים הרלוונטיים ביותר עבור המשימה שלנו:
לצורך בדיקת מה הפיצ'רים שלהם ההשפעה הגדולה ביותר על כמות ההאזנות לשיר, התייחסנו לערך views כמשתנה מטרה, וביצענו בדיקת f_regression של חבילת Sklearn.

אלו התוצאות שקיבלנו עבור k=10 פיצ'רים:

<u>Selected Features:</u>

1. 'Loudness'
2. 'Danceability'
3. 'Instrumentalness'
4. 'Acousticness'
5. 'Energy'
6. 'Valence'
7. 'Album_type'
8. 'Duration_ms'
9. 'Liveness'
10. 'Speechiness'

<u>תיאור אתגרים במהלך העבודה</u>

אחד האתגרים בשלב זה היה קבלת ההחלטה האם להתייחס למשתנה "Artist" כמשתנה מסביר במודל החיזוי שלנו.
הכללת תכונת "Artist" במודל החיזוי עלולה להוביל ל-overfitting , כלומר המודל הופך ספציפי מדי לנתוני האימון. אם המודל ילמד לשייך אמנים מסוימים עם צפיות גבוהות או נמוכות בהתבסס על נתוני האימון, הוא עלול לייתר הטיה גדולה עבור אמנים חדשים שלא היו נוכחים במערך האימון.

שונות בין אמנים ובין שירים של אותו אמן:
קיימת שונות בין שירים שונים של אותו אמן ולעיתים שונות זו אפילו משמעותית, יתכן שאמנים פרסמו מגוון רחב של שירים עם פופולריות משתנה.
כאשר נאמן את המודל גם על בסיס הפיצ'ר "Artist" המודל עשוי להתעלם מהפיצ'רים הספציפיים של השירים התורמים לפופולריות שלהם ולספר הצפיות.

היתרונות שלא להתייחס לפיצ'ר "Artist":

6. Unbiased Predictions: המודל ילמד ויבצע חיזוי על סמך המאפיינים שבחרנו בשלב ה- features selection, נקיטה בגישה זו עשויה לספק חיזויים שאינם מושפעים מהפופולריות של אמן ספציפי.
7. Better Generalization: ללא שימוש בפיצ'ר זה המודל יכול ללמוד דפוסים ויחסים כלליים ולזהות את המגמות והמאפיינים שתורמים לפופולריות של תוכן על פני אמנים וז'אנרים שונים ובכך לסייע למגוון רחב של קהל יעד.

החסרונות שלא להתייחס לפיצ'ר "Artist":

1. Loss of Artist-Specific Insights: כאשר לא נתחשב בפיצ'ר זה, המודל עלול להחמיץ דפוסים ומאפיינים ספציפיים לאמן שיכולים לתרום לניבוי הצפיות. לחלק מהאמנים יש רקורד עקבי של

הפקת תוכן פופולרי, ונוכחותם לבדה עשויה למשוך יותר צפיות. במקרים כאלה, המודל לא יוכל ללכוד את המגמות הספציפיות לאותו אמן.

לסיכום, בחירה שלא להשתמש במשתנה המסביר " Artist" מאפשרת תחזיות תכנים שונים ועבור מגוון רחב של אמנים. בחירה זו עשויה למנוע overfitting של המודל ולשפר את תוצאות החיזוי. לאור יתרונות אלה בחרנו שלא להשתמש במשתנה זה למרות הפשרה של החמצה פוטנציאלית של תובנות ספציפיות לאמן.

שיטות רגרסיה והייפר-פרמטרים בהם השתמשנו:

ניסינו מגוון שיטות רגרסיה הכוללות רגרסיה לינארית, עצי החלטה, SVM, שיטות אנסמבל, KNN, רגרסיה עם רגולריזציה ורשתות נוירונים.

ניסינו גם מודלים פשוטים כמו רגרסיה ועץ החלטה בודד וגם שיטות מורכבות יותר מתוך הנחה שאמנם זה בעיה מסובכת וסביר להניח שהשיטות הפשוטות יעבדו בצורה פחות טובה אבל זמן האימון שלהן קצת משמעותית אז יכלנו בקלות לעשות איתם ניסויים רבים, בנוסף למדנו שלפעמים גם שיטות "פשוטות" עשויות להפתיע ולתת תוצאות טובות.

פירוט השיטות והיפר פרמטרים – חלק מהקוד, יצרנו מילון של מודלים והיפר פרמטרים שנבחן לכל אחד מהם:

```
# Create a dictionary to store the models and hyperparameter options
models = {
    'Linear Regression': {
        'model': LinearRegression(),
        'params': {'fit_intercept': [True, False]}
    },
    'Decision Tree': {
        'model': DecisionTreeRegressor(),
        'params': {'max_depth': [None, 10, 20], 'min_samples_split':
[2, 5, 10]}
    },
    'Random Forest': {
        'model': RandomForestRegressor(),
        'params': {'n_estimators': [100, 200, 500], 'max_depth': [2, 5,
10], 'min_samples_split': [2, 5]}
    },
    'Gradient Boosting': {
        'model': GradientBoostingRegressor(),
        'params': {'n_estimators': [100, 200, 1000], 'learning_rate':
[0.1, 0.5]}
    },
    'Support Vector Regression': {
        'model': SVR(),
        'params': {'C': [1, 10, 100], 'epsilon': [0.1, 0.01]}
    },
    'K-Nearest Neighbors': {
        'model': KNeighborsRegressor(),
        'params': {'n_neighbors': [3, 5, 7], 'weights': ['uniform',
'distance']}
```

```
    },
    'Lasso Regression': {
        'model': Lasso(),
        'params': {'alpha': [0.01, 0.1, 1.0]}
    },
    'Ridge Regression': {
        'model': Ridge(),
        'params': {'alpha': [0.01, 0.1, 1.0]}
    },
    'MLP': {
        'model': MLPRegressor(max_iter=1000),
        'params': {'hidden_layer_sizes': [(50,), (100,), (50, 50)],
'activation': ['relu', 'tanh', 'logistic']}
    }
}
```

תוצאות האבלואציה

חלוקה לtrain ו-test:

חילקנו את הדאטה שלנו 80/20, את סט האימון העברנו לgrid search cross validation עם 5 פולדים לצורך אימון המודל ובדקנו כל מודל קנדידט עם סט הבדיקה.

רצינו לקבל התפלגות דומה של הפיצ'רים בין סט האימון לסט הבדיקה לכן השתמשנו ב stratify עבור פיצ'רים קטגוריים ועשינו סטנדרטיזציה לפיצ'רים הנומריים לפני החלוקה לtrain ו-test.

סוגי אבלואציה:

בחנו כמה אפשרויות-

1. אופציה 1 - שימוש בכל הפיצ'רים
2. אופציה 2 - שימוש רק בפיצ'רים אחרי feature selection
3. אופציה 3 - שימוש בפיצ'רים אחרי טרנספורמציה (וגם feature selection – אין חפיפה בין הפיצ'רים של הטרנספורמציה לפיצ'רים של פיצ'ר סלקשיין)

טבלאות/תרשמים מסודרים עם מדדי הדיוק, וזמני הריצה:

1. אופציה 1 - שימוש בכל הפיצ'רים

Option	Model	Features	Best Params	MSE	Train Time
Option 1	Random Forest	All features	{'max_depth': 10, 'min_samples_split': 5, 'n_estimators': 500}	526349087549631.75	0.14349842071533203
Option 1	Gradient Boosting	All features	{'learning_rate': 0.1, 'n_estimators': 100}	534518591303050.9	4.471486568450928
Option 1	Linear Regression	All features	{'fit_intercept': True}	567367101120821.2	472.62592792510986
Option 1	Lasso Regression	All features	{'alpha': 1.0}	567367104576062.5	199.60394740104675
Option 1	Ridge Regression	All features	{'alpha': 1.0}	567368508225488.5	45.816322803497314
Option 1	K-Nearest Neighbors	All features	{'n_neighbors': 7, 'weights': 'distance'}	577705087205960.8	4.0613486766815186
Option 1	MLP	All features	{'activation': 'relu', 'hidden_layer_sizes': (50, 50)}	620404641526831.4	0.6323153972625732
Option 1	Support Vector Regression	All features	{'C': 100, 'epsilon': 0.1}	658136240800512.4	0.34772825241088867
Option 1	Decision Tree	All features	{'max_depth': 10, 'min_samples_split': 10}	679525699236527.5	1657.572214603424

2. אופציה 2 - שימוש רק בפיצ'רים אחרי feature selection

Option	Model	Features	Best Params	MSE	Train Time
Option 2	Random Forest	Only selected features	{'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 500}	530378922380556.5	0.25960850715637207
Option 2	Gradient Boosting	Only selected features	{'learning_rate': 0.1, 'n_estimators': 100}	536155910602002.7	2.466433048248291
Option 2	Linear Regression	Only selected features	{'fit_intercept': True}	567231570382262.0	408.61419343948364
Option 2	Lasso Regression	Only selected features	{'alpha': 1.0}	567231574373551.2	171.2352955341339
Option 2	Ridge Regression	Only selected features	{'alpha': 1.0}	567233019848738.9	44.87624764442444
Option 2	K-Nearest Neighbors	Only selected features	{'n_neighbors': 7, 'weights': 'distance'}	583594700217009.8	2.809654951095581
Option 2	MLP	Only selected features	{'activation': 'relu', 'hidden_layer_sizes': (50, 50)}	606279262184929.4	0.3052842617034912
Option 2	Support Vector Regression	Only selected features	{'C': 100, 'epsilon': 0.1}	658120005221370.0	0.20212149620056152
Option 2	Decision Tree	Only selected features	{'max_depth': 10, 'min_samples_split': 10}	668536427587346.8	1605.2795128822327

3. אופציה 3 - שימוש בפיצ'רים אחרי טרנספורמציה

Option	Model	Features	Best Params	MSE	Train Time
Option 1	Random Forest	All features	{'max_depth': 10, 'min_samples_split': 5, 'n_estimators': 500}	526349087549631.75	0.14349842071533203
Option 1	Gradient Boosting	All features	{'learning_rate': 0.1, 'n_estimators': 100}	534518591303050.9	4.471486568450928
Option 1	Linear Regression	All features	{'fit_intercept': True}	567367101120821.2	472.62592792510986
Option 1	Lasso Regression	All features	{'alpha': 1.0}	567367104576062.5	199.60394740104675
Option 1	Ridge Regression	All features	{'alpha': 1.0}	567368508225488.5	45.816322803497314
Option 1	K-Nearest Neighbors	All features	{'n_neighbors': 7, 'weights': 'distance'}	577705087205960.8	4.0613486766815186
Option 1	MLP	All features	{'activation': 'relu', 'hidden_layer_sizes': (50, 50)}	620404641526831.4	0.6323153972625732
Option 1	Support Vector Regression	All features	{'C': 100, 'epsilon': 0.1}	658136240800512.4	0.34772825241088867
Option 1	Decision Tree	All features	{'max_depth': 10, 'min_samples_split': 10}	679525699236527.5	1657.572214603424

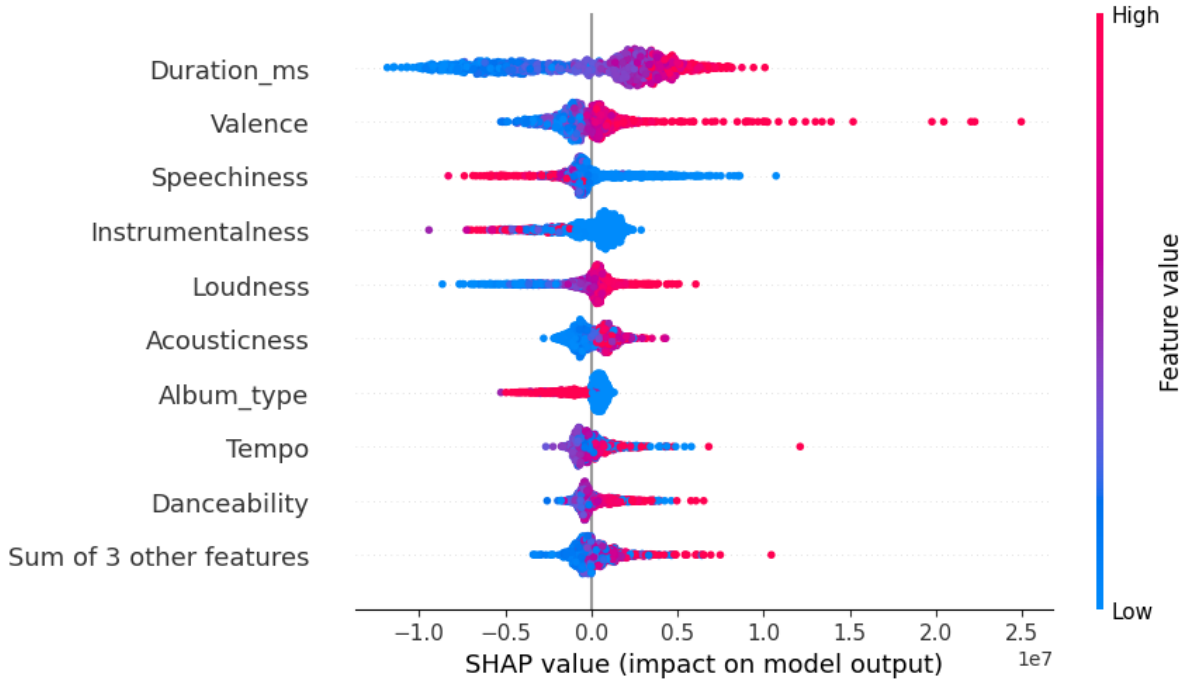
תוצאות מפורטות עבור מספר מודלים מצומצם:

אנחנו בבעיית רגרסיה לכן ROC, confusion matrix לא רלוונטים. כדי להראות תוצאות יותר מפורטות נראה מטריקות נוספות עבור 3 המודלים הכי טובים של כל אופציה (שימוש בכל הפיצ'רים, שימוש רק בפיצ'רים אחרי feature selection, שימוש בפיצ'רים אחרי טרנספורמציה)

1. אופציה 1 - שימוש בכל הפיצ'רים

Model	Best Params	MSE	MAE	Explained Variance	MAPE	Quantile Loss	Train Time
Random Forest	{'max_depth': 10, 'min_samples_split': 5, 'n_estimators': 500}	526349087549631.75	16963888.665571317	0.11609320065625495	215.7778329691747	8481944.332785659	0.14349842071533203
Gradient Boosting	{'learning_rate': 0.1, 'n_estimators': 100}	534518591303050.9	17153947.747143432	0.1022719041679867	215.4717567216456	8576973.873571716	4.471486568450928
Linear Regression	{'fit_intercept': True}	567367101120821.2	18019374.992044996	0.04711598907439141	232.96675352511488	9009687.496022498	472.62592792510986

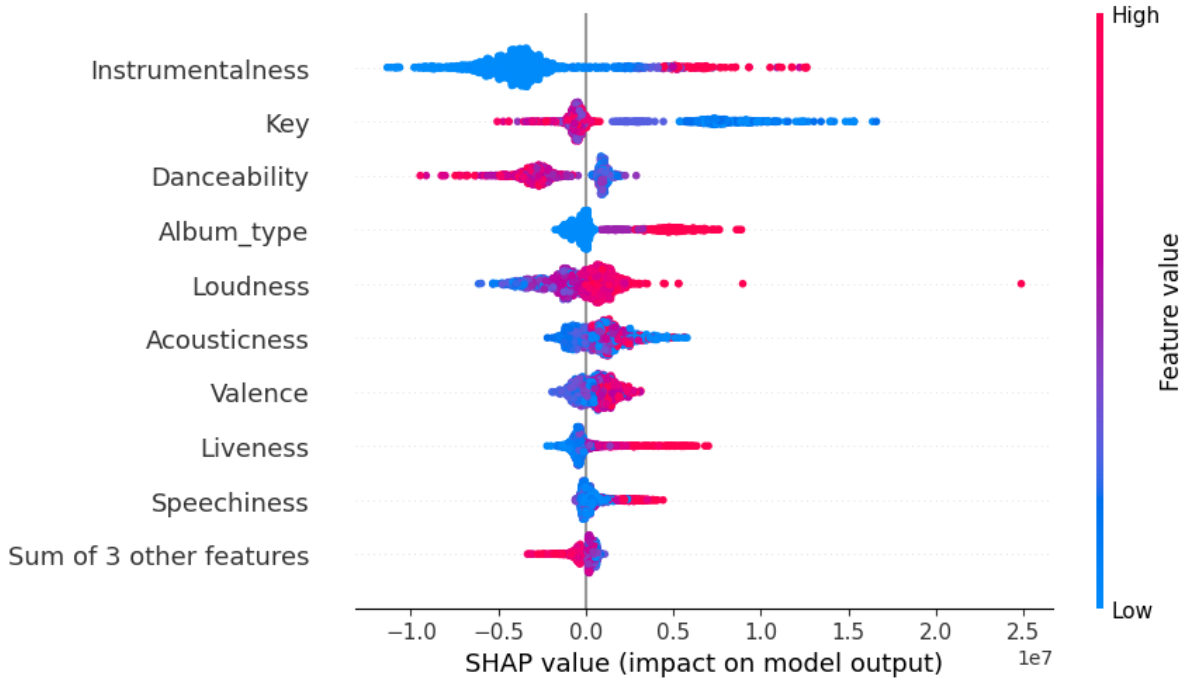
הסבר למודל הטוב ביותר בעזרת SHAP – Global explanation:



2. אופציה 2 - שימוש רק בפיצ'רים אחרי feature selection

Model	Best Params	MSE	MAE	Explained Variance	MAPE	Quantile Loss	Train Time
Random Forest	{'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 500}	530378922380556.5	17011753.81950942	0.10931084034256189	216.72765278808873	8505876.90975471	0.25960850715637207
Gradient Boosting	{'learning_rate': 0.1, 'n_estimators': 100}	536155910602002.7	17125704.675403226	0.09950544827787733	214.31550403448298	8562852.337701613	2.466433048248291
Linear Regression	{'fit_intercept': True}	567231570382262.0	18026104.74677788	0.04734525355785857	232.94880864151278	9013052.37338894	408.61419343948364

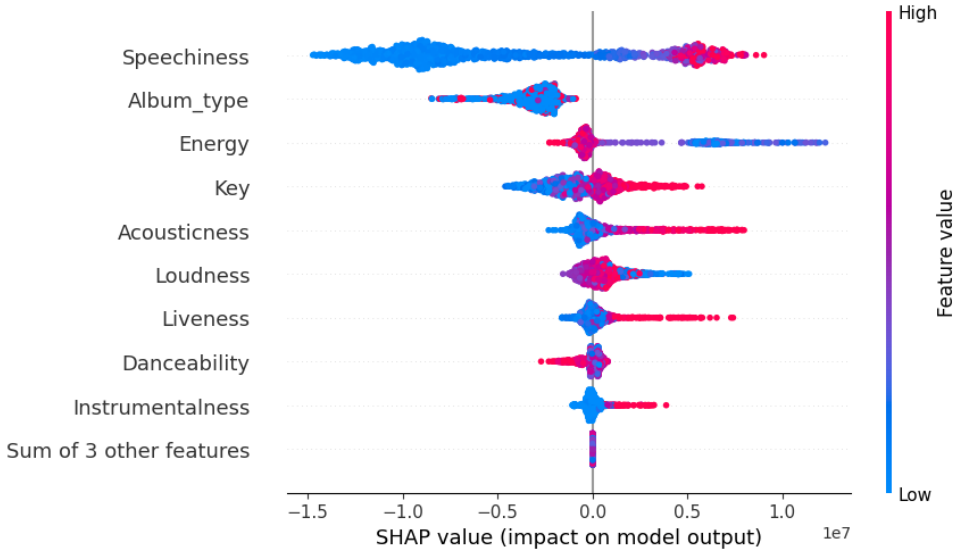
הסבר למודל הטוב ביותר עבור אופציה זו:



3. אופציה 3 - שימוש בפיצ'רים אחרי טרנספורמציה

Model	Best Params	MSE	MAE	Explained Variance	MAPE	Quantile Loss	Train Time
Random Forest	{'max_depth': 10, 'min_samples_split': 5, 'n_estimators': 500}	526349087549631.75	16963888.665571317	0.11609320065625495	215.7778329691747	8481944.332785659	0.14349842071533203
Gradient Boosting	{'learning_rate': 0.1, 'n_estimators': 100}	534518591303050.9	17153947.747143432	0.1022719041679867	215.4717567216456	8576973.873571716	4.471486568450928
Linear Regression	{'fit_intercept': True}	567367101120821.2	18019374.992044996	0.04711598907439141	232.96675352511488	9009687.496022498	472.62592792510986

הסבר למודל הטוב ביותר עבור אופציה זו:



הצעות כיצד ניתן להמשיך את העבודה - מידולים נוספים, אלו נתונים חיצוניים היה כדאי לאסוף בכדי לשפר את המודלים.

נתונים חיצוניים – שימוש בנתונים נוספים כדי להוסיף עוד מידע למודל ובכך לשפר את הדיוק, למשל מידע נוסף על השירים ממקורות נוספים (מתי השיר יצא, כמה שירים היו לאמן לפני כן, כמה שירים הצליחו עבור האמן הזה..). למשל, נוכל לייבא נתונים פרטניים יותר עבור מדינות ולבצע ניתוח פנים רוחבי ומעמיק יותר בכדי להבין את פוטנציאל ההצלחה לכל מדינה בנפרד. באופן הזה נוכל להבין אם שיר עשוי להצליח במדינה מסויימת כשבמדינות אחרות לא – מה שיאפשר לנו לקבל החלטה נקודתית עבור כל מדינה והגדיל את הרווחים שלנו. יתכן שיהיו לנו הרבה שירים שנרצה להשקיע בהם ברמה לאומית ולא גלובלית אך עדיין להרוויח מהם, הרי נושא התרבות הוא נדבך חשוב בכל הנוגע למוזיקה ולא כל שיר יצליח בהכרח בכל העולם. לפי שיטה זו, נוכל להגדיל את הרווחים וגם להבין איזה שירים מצליחים בכל מדינה. בנוסף, סביר להניח שעבור שיר שיצליח גלובלית נחזה שיצליח גם ברמת המדינה, מה שיעזור לנו לסווג שירים כשירים עם הצלחה עולמית. כמו כן, מקורות דאטה נוספים על אמנים והשירים שלהם יכולים להיות מתחנות רדיו, מצעדים שנתיים (כמו המצעדים של גלגלצ למשל), פרסים והוכרות (כמו הגרמי למשל). נוכל לשקול להשתמש במודלים מתקדמים יותר בכדי לבצע ניתוחים יותר עמוקים ולמצוא קשרים יותר מורכבים. הצעה נוספת היא שימוש במרחב אמדינג בכדי ליצור שיטה שתעזור לנו להבין כמה שיר חדש קרוב לשיר שנחשב מוצלח (עם הרבה השמעות). הרעיון הוא לבחור threshold מסוים שעבורו נקבע אם שיר נחשב למוצלח או לא (כמות השמעות / רווחים וכו'). ניקח את כל השירים שעומדים בקריטריון ונבנה וקטור עם כל הפרמטרים המתאימים עבור כל שיר, בכדי שנוכל ללמוד מרחב אמבדינג. כעת כשנרצה לבחון שיר חדש, נוכל לבחון כמה השיר קרוב או דומה לשירים אחרים במרחב, מה שיכול לעזור לנו להבין עם כדאי להשקיע בשיר מסוים לפי הקרבה שלו לשירים שידועים כשירים מוצלחים. כמובן שידרשו מספר ניסויים בכדי להבין אם השיטה עובדת או לא. אנו שמים לב כי ביצועי המודלים השונים שניסינו לא היו הכי מוצלחים מה שמרמז שיתכן והדאטה הקיימת לא מספקת. כחלק מהמשך העבודה העתידית שציינו נרצה למצוא קשרים שלא חשבנו עליהם העשויים לשפר את יכולות הפרדיקציה. בנוסף למקורות דאטה נוספים נדרש לנתח את הדאטה באופן יותר מורכב ולבצע Feature Engineering יותר מתוחכם הן לדאטה הקיימת והן לדאטה החדשה.

סיכום ותבונות עיקריות מהעבודה.

המטרה העיקרית של הפרויקט הייתה לבצע ניתוח נתונים אקספלורטיבי ולפתח מודל חיזוי להערכת מספר הצפיות בסרטוני YouTube. הפרויקט עסק בכלליות בעקרונות המשפיעים על הפופולריות של סרטוני YouTube וניסה לבנות מודל חיזוי מדויק יותר למספר הצפיות בהתבסס על מאגר הנתונים הזמין.
במהלך העבודה, בוצעו שלבים מרכזיים כגון איסוף הנתונים, ניתוח נתונים ראשוני, ופיתוח והערכת מודל חיזוי. בנוסף, נחקרו פיצ'רים רלוונטיים ובוצעה טיפוח הפיצ'רים ושימוש בשיטות רגרסיה והייפר-פרמטרים כדי לקבוע את המודל המתאים לחיזוי.
אחת התובנות העיקריות שלנו מפרויקט זה הוא חשיבות שלב עיבוד וניתוח הנתונים שכן הוא משפיע באופן מהותי על המשך שאר השלבים בתהליך, על המסקנות אליהם ניתן להגיע, ובעיקר על התוצאות שמקבלים מהמודלים השונים. לכן יש חשיבות רבה לשלב זה כיוון שהינו רגיש ובעל השפעה ניכרת.
ניסינו בפרויקט זה לכלול במודלים שלנו רק פיצ'רים בעלי קורלציה גבוה והשפעה על כמות הצפיות אותם איתרנו בעזרת החבילה f_regression של sklearn.
במהלך חקירת הפיצ'רים גילינו שאין מגמה או קשר משמעותי בין אורך השיר לבין מספר הצפיות שלו. שירים שאורכן עד 10 דקות מקבלים מספר צפיות לאורך כל טווח הערכים האפשרי.
אחד הדברים שחקרנו במהלך הפרויקט ניסינו להבין את ההשפעה של סוג האלבום על היחס של מספר הצפיות מול מספר הלייקים. כלומר עבור סוג אלבום עם מספר רב של צפיות ראינו מספר רב של ליקים .
אחד האתגרים בעבודה היה בחירת האם לכלול את המשתנה "Artist" כפיצ'ר במודל החיזוי. נקטה החלטה לא לכלול, כיוון שהוא עשוי להביא ל-overfitting ולא נתונים מייצגים לכל השירים. זאת כאשר יתרונות שימוש בפיצ'ר זה כוללים התחשבות בדפוסים והמאפיינים הספציפיים של השירים, אך יתרונות נוספים בעדיפות לא לכלול.
התוצאות של ה SHAP הראו כי ששלושת המודלים הטובים ביותר, הגיעו לתוצאות דומות אך ההסבר שלהם מאוד שונה, ניתן לייחס את זה לכך שאף אחד מהמודלים לא ממש מוצלח והצליח להתכנס לתוצאה טובה (הMSE של כולם גבוהה).

לסיכום, העבודה על הפרויקט התמקדה בניתוח נתונים אקספלורטיבי של YouTube ו-Spotify ופיתוח מודל חיזוי. המודלים שנבדקו לא הביאו לתוצאות מספקות ברמת החיזוי של מספר הצפיות בסרטוני YouTube.
המשימה התגלתה כמורכבת, והעבודה התמקדה בשיפור פיתוח המודל ושיפור המשקל בין פיצ'רים עם מתודולוגיה וניסיון שונה.