SONG POPULARITY PREDICTION & MUSIC RECOMMENDATION SYSTEM

Group 74

Karthik Panghat
Kaushiki Ambi
Shreyans Hiteshkumar Trivedi

panghat.k@northeastern.edu
ambi.k@northeastern.edu
trivedi.shre@northeastern.edu

Percentage of Effort Contributed by Student 1: 34%
Percentage of Effort Contributed by Student 2: 33%
Percentage of Effort Contributed by Student 3: 33%

Signature of Student 1: *Karthik Panghat*
Signature of Student 2: *Kaushiki Ambi*
Signature of Student 2: *Shreyans Trivedi*

Submission Date: 04/21/2023

TABLE OF CONTENTS

1. PROBLEM SETTING 02
2. PROBLEM DEFINITION 02
3. DATA SOURCES 03
4. DATA DESCRIPTION 03
5. DATA MINING TASKS 03
    a. Data Understanding 03
    b. Data Preprocessing 03
6. DATA EXPLORATION 04
7. DIMENSION REDUCTION 07
8. DATA MINING MODELS 07
    a. Linear Regression 07
    b. Polynomial Regression 08
    c. Lasso Regression 08
    d. Ridge Regression 09
    e. Logistic Regression 09
    f. KNN Classifier 10
    g. Decision Tree Classifier 11
    h. Cosine Similarity 11
9. MODEL PERFORMANCE EVALUATION 12
    a. Regression Models 12
    b. Classification Models 13
10. SONG RECOMMENDATION SYSTEM 15
11. PROJECT RESULTS 16
12. PROJECT OUTCOMES 17
13. REFERENCES 18

1. Problem Setting

In today's digital environment, streaming services for movies and music have gained immense popularity. The success of over-the-top (OTT) and audio/video streaming platforms depends largely on the user experience they provide. A recommendation system plays a significant role in ensuring a positive user experience by suggesting songs, playlists, etc. based on the user's historical data. This report presents a project aimed at building a recommendation system for songs by analyzing the correlation between audio features and song popularity, predicting the popularity of a song using audio feature attributes, classifying tracks into different genres based on their audio features, and finally building a recommendation model based on the user's input song.

2. Problem Definition:

The primary objective of this project is to build a recommendation system for songs per user's input. The first step towards achieving this objective is to assess the correlation between audio features and song popularity. Based on this analysis, the subsequent task is to predict the popularity of a song using its audio feature attributes. The dataset used for this project contains 20 columns (attributes) and 114,000 rows (songs), including track id, artist, track name, popularity, danceability, energy loudness, valence, track genre, and more. 16 attributes are of numeric data type while 4 attributes are of character data type. Audio features such as acoustics (a confidence measure of whether the track is acoustic), valence (a measure describing the musical positiveness conveyed by a track), and tempo have been identified as important factors in predicting a song's popularity score.

In conclusion, this project aims to build a recommendation system for songs based on the analysis of audio features and song popularity. The project involves assessing the correlation between audio features and song popularity, predicting a song's popularity score using its audio feature attributes, classifying tracks into different genres based on their audio features, and building a recommendation model based on the user's favorite song. By leveraging the power of machine learning and data analytics, we can create an efficient recommendation system that provides maximum user engagement, increased customer service, easy music discoverability, and more subscriptions.

3. Data Sources

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

4. Data Description

This is a dataset of Spotify tracks over a range of different genres. It has been taken from Kaggle, an online community of data scientists and machine learning practitioners. The dataset has 20 columns (attributes) and 114000 rows (attributes). Each track has some associated audio features, such as track id, artists, track name, popularity, danceability, energy loudness, valence, track genre, etc. 16 attributes are of numeric data type while 4 attributes are of character data type. The audio features acoustic ness (confidence measure of whether the track is acoustic), valence (a measure describing the musical positiveness conveyed by a track.), and loudness (the overall loudness of a track in decibels (dB)) which seem to be important in predicting the popularity score for the song.

5. Data Mining Tasks

a. **Data Understanding**: The dataset comprises 114000 instances and 20 variables and target variable popularity. By analyzing the dataset, the data types of attributes have been observed to be numeric and categorical. The variable information and associated datatype are provided in Table No. 1.

b. **Data Preprocessing**: The unnamed column was removed as it's the index number and provides no value to the analysis. In total, there was one missing value each in artists, album name, and track name. As these values won't have enough significance, those rows were dropped. Additionally, there were duplicate records that were dropped during further analysis. We did PCA analysis to reduce the dimension by removing the numerical variables. In order to keep a 99% variance, all 14 numerical variables are required. So, no variables were dropped then. However, a good correlation was identified between energy and loudness and therefore, we dropped the loudness attribute. As an outcome of the above cleaning steps, 46589 instances and 19 attributes remained for further analysis. Some "outliers" were identified using box plot analysis. However, from a domain knowledge, we decided to keep those outliers in our analysis. Additionally, categorical variables key and mode were one hot encoded which increased the dimensions to 32 columns. Chi square test was conducted on the one hot encoded data and this helped drop and reduce column dimension to 25.

Table 1: Description of Variables

No	Variable	Description	Type
1	track_id	The Spotify ID for the track	Numeric
2	artists	The artists' names who performed the track. If there is more than one artist, they are separated by a ;	Category
3	album_name	The album name in which the track appears	Category
4	track_name	Name of the track	Category
5	popularity	The popularity of a track is a value between 0 and 100, with 100 being the most popular.	Numeric
6	duration_ms	The track length in milliseconds	Numeric
7	explicit	Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown)	Category
8	danceability	Danceability describes how suitable a track is for dancing . A value of 0.0 is least danceable and 1.0 is most danceable	Numeric
9	energy	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.	Numeric
10	key	The key the track is in. Integers map to pitches using standard Pitch Class notation.	Numeric
11	loudness	The overall loudness of a track in decibels (dB)	Numeric
12	mode	Mode indicates the modality (major or minor) of a track. Major is represented by 1 and minor is 0	Numeric
13	speechiness	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), value will be 1 or closer to 1	Numeric
14	acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic	Numeric
15	instrumentalness	The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content	Numeric
16	liveness	Detects the presence of an audience in the recording. A value above 0.8 provides strong likelihood that the track is live	Numeric
17	valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track.	Numeric
18	tempo	The overall estimated tempo of a track in beats per minute (BPM).	Numeric
19	time_signature	An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).	Numeric
20	track_genre	The genre in which the track belongs	Category

6. Data Exploration

Descriptive statistics of each numerical variable were observed. It was evident from this information that re-scaling needs to be done on variables before the model-building phase. Through the univariate exploratory data analysis using histograms as shown in Fig. 1, the observations are,

- Popularity, danceability, valence, and tempo are mostly normally distributed, and energy is left-skewed.
- Speechiness and acoustics follow the chi-square distribution.

From the multivariate exploratory data analysis of important numerical variables (fig.2), Speechiness, liveness, and danceability have significant outliers which have to be taken care of while model building.

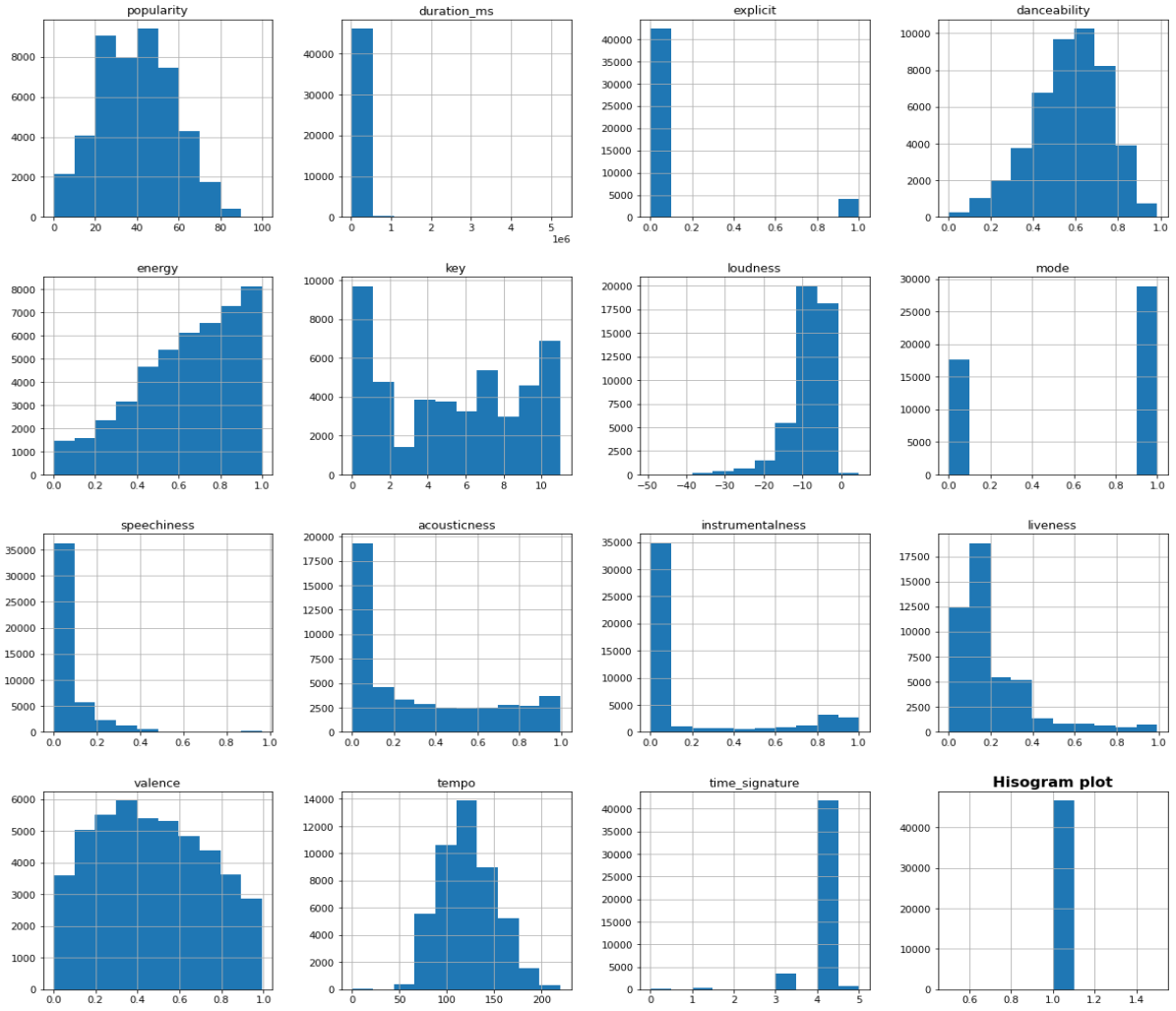


Fig.1- Histogram for Univariate Analysis on Numerical Variables

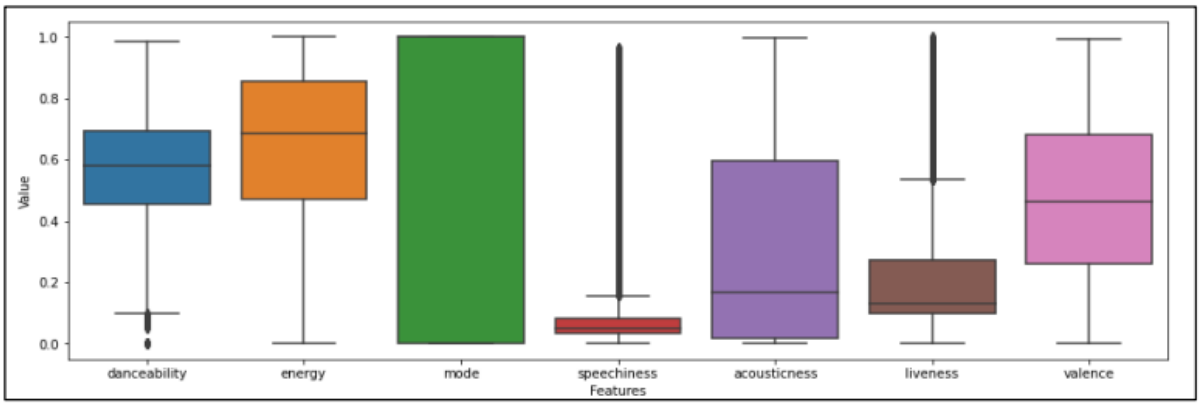


Fig.2- Boxplot for Multivariate Analysis on selected Numerical Variables

Correlation Analysis of Numerical Variables using Pearson's Correlation

Fig. 3 below shows a high correlation (P= 0.76) between energy and loudness. There is a significant correlation between valence and danceability (P= 0.46) and an inverse correlation between energy and acousticness. However, a significant correlation between popularity and other attributes was not identified. Also, loudness is correlated with energy and acousticness, exhibiting multicollinearity, so we can drop this loudness.

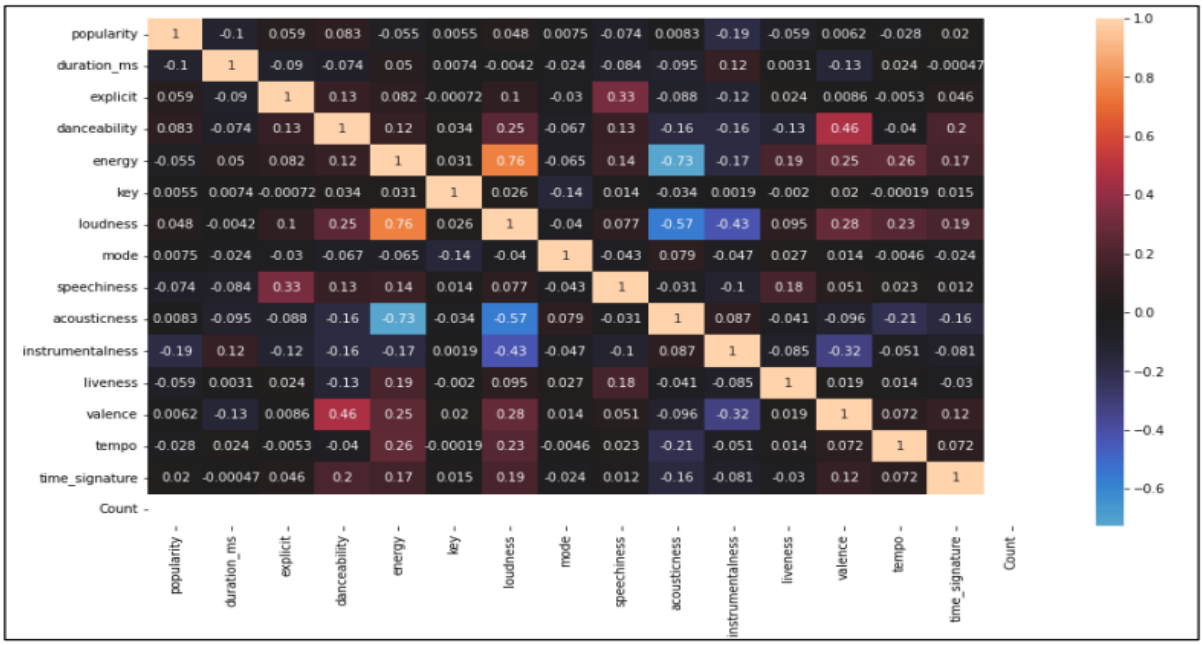


Fig.3- Correlation Heatmap- Pearson's Correlation

Other Data Analysis

Top 5 Albums from Top 20 Genre and Top 10 Popular Tracks Fig.4 and Fig.5 show the Top 5 Albums from Top 20 Genre and Top 10 Popular Tracks respectively. Dance, Latin, Hip-Hop, Pop, and Latino were popular among the music listeners. Unholy by Sam Smith was the most popular among the listeners (and the most listened to track) among approximately 38500 instances. We also identified artists with more content on Spotify.

Fig.4- Treemap- Top 5 Albums in Each Top 20 Genre

artists	album_name	track_name	popularity
Sam Smith;Kim Petras	Unholy (feat. Kim Petras)	Unholy (feat. Kim Petras)	100
Bizarrap;Quevedo	Quevedo: Bzrp Music Sessions, Vol. 52	Quevedo: Bzrp Music Sessions, Vol. 52	99
David Guetta;Bebe Rexha	I'm Good (Blue)	I'm Good (Blue)	98
Manuel Turizo	La Bachata	La Bachata	98
Bad Bunny;Chencho Corleone	Un Verano Sin Ti	Me Porto Bonito	97
OneRepublic	I Ain't Worried (Music From The Motion Picture "Top Gun: Maverick")	I Ain't Worried	96
Chris Brown	Indigo (Extended)	Under The Influence	96
The Neighbourhood	I Love You.	Sweater Weather	93
Tom Odell	Long Way Down (Deluxe)	Another Love	93
Beyoncé	RENAISSANCE	CUFF IT	93

Fig.5- Top 10 Popular Tracks

7. Dimension Reduction

From the summary statistics, it was observed that there is a scale difference between numerical variables. Therefore, to handle this difference, the dataset was standardized using the StandardScalar() library from the Scikit learning package, to ensure that all predictor variables were given equal importance in terms of variability. PCA was done on standardized data. For each of the components, the explained variance, proportion of variance, and cumulative proportion of variance were determined. Fig 6 shows the proportion of variance for each component (in %). It has been observed that to contain 99% variance, we must use a minimum of 13 variables out of 14 numerical variables. So, we conclude that PCA won't be useful in this scenario and rather we will depend on the feature selection dimension reduction process for model development which is explained in the data preprocessing section where we were able to reduce dimension from 32 to 25.

8. Data Mining Models:

The models we used are as follows:

1. *Linear Regression:*

   Linear regression is a linear approach for modeling the relationship between a scalar response and one or more explanatory variables (also known as dependent and independent variables). The case of one explanatory variable is called simple linear regression; for more than one, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

Advantages:

- Linear regression performs exceptionally well for linearly separable data
- It is easier to implement, interpret, and efficiently trained

Disadvantages

- The assumption of linearity between dependent and independent variables
- It is often quite prone to noise and overfitting.

2. *Polynomial Regression:*

Polynomial regression is a type of regression analysis that allows for non-linear relationships between variables to be evaluated. Compared to Linear Regression, which assumes a linear relationship, it models an elevated polynomial function to fit the data. The degree of the polynomial determines the accuracy of the best-fitting curve to the data. It can, however, lead to overfitting since it assumes the relationship between variables is constant.

Advantages:

- Polynomial Regression can model non-linear relationships between variables, which cannot be captured by Linear Regression.
- It allows for more flexibility in fitting the data, as higher-degree polynomials can better fit complex patterns in the data.

Disadvantages:

- Higher-degree polynomials can lead to overfitting, where the model fits the training data too closely and performs poorly on new, unseen data.
- Polynomial Regression assumes the relationship between the variables is continuous and smooth, which may not always be the case.

3. *Lasso Regression:*

Lasso Regression is a type of regression analysis that uses variable selection as well as regularization to improve the predictive accuracy and interpretability of the model. It engages a penalty term that simply eliminates some features from the model by forcing some of the regression coefficients to be zero. This helps to avoid overfitting and reduces the model's complexity.

Advantages:

- Lasso Regression performs both variable selection and regularization, which can improve the model's predictive accuracy and interpretability.
- Lasso Regression forces some of the regression coefficients to be zero, effectively removing some features from the model, which helps to avoid overfitting and reduce the complexity of the model.

Disadvantages:

- Lasso Regression may not be appropriate when all features are important for the model, as it can remove some useful features.
- It can be computationally expensive, especially with large datasets and many features.

4. *Ridge Regression:*

Ridge Regression is a type of regression analysis that uses variable selection as well as regularization to improve the predictive accuracy and interpretability of the model. It engages a penalty term that reduces the magnitude of the regression coefficients by downsizing them to zero. This helps to avoid overfitting and reduces the model's complexity.

Advantages:

- Ridge Regression performs both variable selection and regularization, which can improve the model's predictive accuracy and interpretability.
- Ridge Regression shrinks the regression coefficients towards zero, effectively reducing the magnitude of the coefficients, which helps to avoid overfitting and reduce the complexity of the model.

Disadvantages:

- Ridge Regression may not be appropriate when all features are important for the model, as it can reduce the magnitude of useful features.
- It is sensitive to the choice of the regularization parameter, which can be challenging to select in practice.

5. *Logistic Regression*

The method of modeling the likelihood of a discrete result given an input variable is known as logistic regression. Logistic regression, despite its name, is a classification model which is a very simple algorithm and achieves very good performance with

linearly separable classes. It essentially uses the sigmoid function to model a binary class.

Advantages

- Easier to implement and interpret the results. It is also a good algorithm for training the data efficiently.
- It makes no assumption about the data distribution.
- It can be easily extended to multi classification problems.

Disadvantages

- It constructs a linear boundary which can cause high misclassification rates.
- Non- Linear problems can't be solved with logistic regression since it has a linear decision surface.

6. *Decision Tree:*

A Decision Tree is a predictive modeling approach that applies to classification and regression tasks. It is a graphical representation of all possible decisions based on specific conditions. It begins with a single node representing the whole dataset and then recursively divides it into smaller subsets based on the values of the predictor variables until a stopping criterion is met. Each split in the tree generates a new decision node, which represents a new decision based on the values of a predictor variable. The terminal nodes of the tree, also known as the leaves, represent the predicted values of the response variable.

Advantages:

- Decision Trees can handle both categorical and numerical data, making them versatile for a variety of data types.
- They are non-parametric, meaning they do not make any assumptions about the distribution of the data.

Disadvantages

- Decision Trees are prone to overfitting, especially when the data is noisy.
- Decision Trees can have a bias toward features with more levels or values, as they have more opportunities to split the data.

7. *K Nearest Neighbors Classifier*

The K-Nearest Neighbors (K-NN) Classifier is a type of non-parametric machine learning algorithm used for classification problems. A lazy learning approach classifies new instances by finding the most similar instances in the training dataset based on their distance to the new instance. The K-NN Classifier is called "K" nearest neighbors because it considers the "K" closest neighbors in the training data to determine the class label of the new instance. The distance metric used to determine similarity can vary, but commonly used metrics include Euclidean distance, Manhattan distance, and cosine similarity. K-NN Classifier is simple, easy to understand and implement, and can handle complex decision boundaries. However, it can be sensitive to irrelevant features, and its performance can be affected by the choice of K and the distance metric.

Advantages:

- K-NN is a simple, non-parametric, and easy-to-understand algorithm that can be applied to both classification and regression tasks.
- K-NN does not make any assumptions about the underlying data distribution, making it useful for a variety of datasets.
- K-NN can handle multi-class classification problems.

Disadvantages:

- K-NN is computationally expensive, especially for large datasets or high-dimensional data, as it requires calculating the distance between the new instance and all instances in the training dataset.
- It is sensitive to irrelevant features or noise in the data, which can affect the accuracy of the model.

8. *Cosine Similarity- For Recommendation System*

Cosine similarity is a metric used to measure the similarity of two vectors. Specifically, it measures the similarity in the direction or orientation of the vectors ignoring differences in their magnitude or scale. Both vectors need to be part of the same inner product space, meaning they must produce a scalar through inner product multiplication. The similarity of two vectors is measured by the cosine of the angle between them.

Advantages:

- The cosine similarity is beneficial because even if the two similar data objects are far apart by the Euclidean distance because of their size, they could still have a smaller angle between them. The smaller the angle, the higher the similarity.
- Both continuous and categorical variables may be used.

Disadvantages:

- Doesn't work efficiently with nominal data.
- The magnitude of vectors is not taken into account, merely their direction.

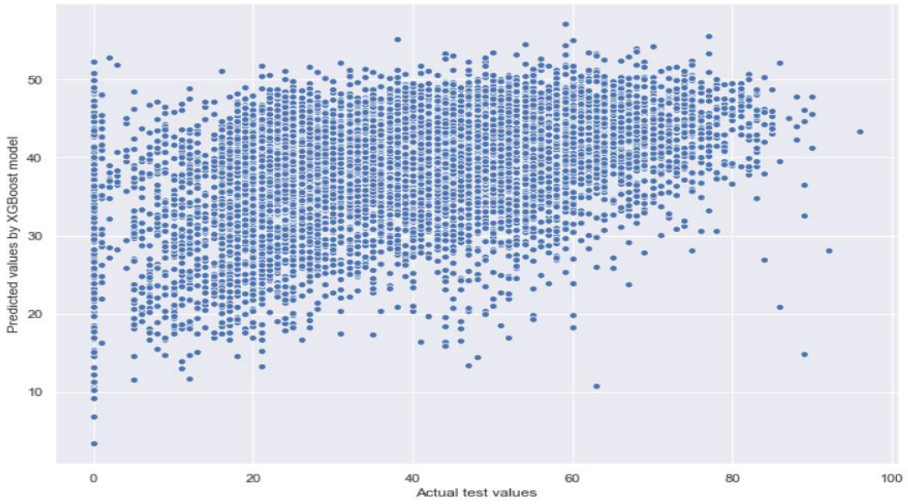
9. Model Performance Evaluation

a. Regression Models

Primarily, regression models were modeled to predict popularity. We have considered simple linear regression, lasso regression, ridge regression, polynomial regression, and XG booster for model building. All the models performed poorly. This was expected as we didn't find any good correlation between any of the predictor variables with the target variable, which is, popularity. Also, from metadata, we understood that the popularity numbers are highly correlated with the number of views (this data is not available) rather than any of the other predictors in the dataset. The highest R^2 value was given by XG Booster which is as low as 17.30% with a very high MSE of 270.39 which shows that regression models are ineffective for this dataset. So, classification models were tested to predict popularity in terms of Top, Average, and Bottom.

	model	mean_squared_error	R-Squared
0	XGBRegressor	270.39122	0.17330
0	PolynomiyalRegression_2_degrees	289.60753	0.11455
5	LinearRegression	304.12692	0.07016
1	Ridge	304.12707	0.07016
4	BayesianRidge	304.13263	0.07014
2	Lasso	306.36421	0.06332
1	PolynomiyalRegression_3_degrees	320.30301	0.02070
3	DecisionTreeRegressor	532.19773	-0.62715

Fig.6- Model Performance of Regression Models



	feature	importance
4	instrumentalness	0.089735
0	acousticness	0.076559
15	tempo	0.076087
16	valence	0.074726
2	duration_ms	0.074079
3	energy	0.072343
14	speechiness	0.070157
1	danceability	0.063290
11	liveness	0.062798
13	mode_1	0.047782
9	key_8	0.044265
12	mode_0	0.044233
7	key_6	0.043460
5	key_1	0.042986
10	key_10	0.042376
8	key_7	0.037679
6	key_2	0.037445

Fig.7- XGB Regressor model prediction and feature importance (R2= 0.173)

b. Classification Models

In order to consider this as a classification problem, the popularity column was binned as top, average, and bottom. A popularity of more than 67 was tagged as Top and less than 33 was tagged as Bottom and anything in between was tagged as Average. Upon a quick analysis, understood that the dataset is imbalanced. So, introduced the oversampling technique and oversampled the training set. Our class of interest is “Top” songs and it is necessary that we have balanced recall and precision values so that top songs are not classified as the other category songs while ensuring other category songs are not getting predicted as “Top” songs which could affect customer satisfaction.

Logistic Regression

We have selected Logistic Regression as our baseline model as it is a simple yet effective model for linear separable cases. The model gave an accuracy of 47.27% and provided poor recall and precision values of 67% and 48% respectively.

	precision	recall	f1-score	support
Average	0.39	0.25	0.31	4089
Bottom	0.50	0.47	0.49	3979
Top	0.46	0.65	0.54	4058
accuracy			0.46	12126
macro avg	0.45	0.46	0.44	12126
weighted avg	0.45	0.46	0.44	12126

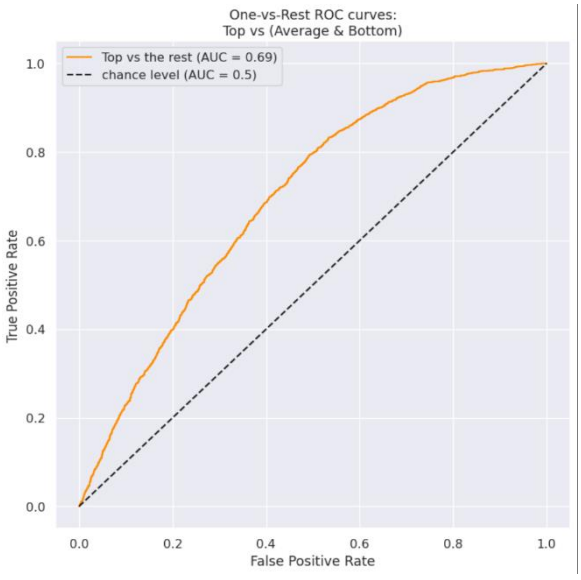


Fig.8 Logistic Regression Model Performance and ROC curve

Higher misclassification rates on this model compelled us to investigate other effective machine learning algorithms such as the KNN classifier and Decision Tree classifier.

KNN Classifier

In the KNN classifier model, a random search was performed to obtain the best metrics. Through this test, we understood that the hyperparameter, n_neighbors, k should be equal to 3 to get good accuracy from this model. The overall accuracy provided by this model was 70%. The model provided a very high recall value of 98% which established that Top songs aren't getting misclassified. However, the precision value of 77% showed that many of the other category songs are predicted as "Top" songs. So, we understood that this model is great for predicting "Top" songs as seen from the AUC graph. However, it is not so great at predicting other category songs correctly. So, we need to check another algorithm, the decision tree, to understand its effectiveness.

	precision	recall	f1-score	support
Average	0.64	0.52	0.57	4089
Bottom	0.66	0.61	0.63	3979
Top	0.79	0.99	0.88	4058
accuracy			0.71	12126
macro avg	0.70	0.71	0.70	12126
weighted avg	0.70	0.71	0.70	12126

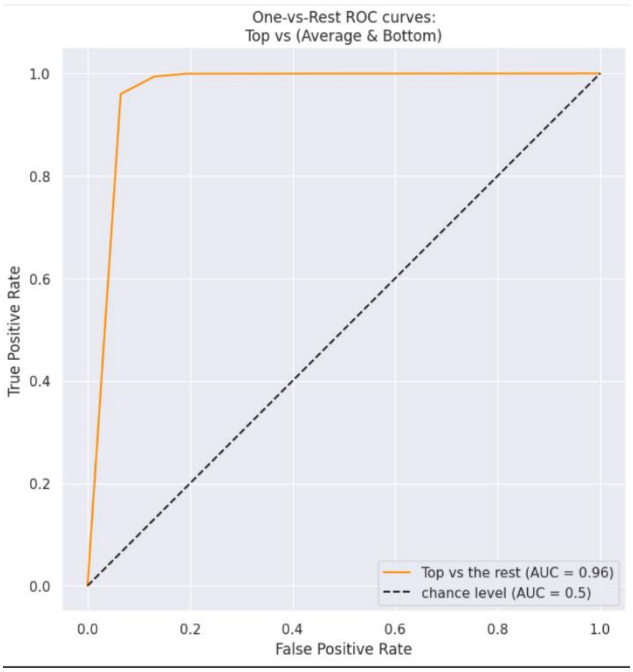


Fig.9- KNN Classifier Model Performance and ROC curve

Decision Tree Classifier

To find a better classifier, the Decision Tree Classifier was implemented. Accuracy of 71.3% obtained from this model. The recall value of 96% for "Top" songs implied that the model was able to identify True Positives for Top song instances. A higher precision score of 82% for "Top" songs indicates low False Positives, showing improved performance in predicting "Other category" songs. The ROC curve was closer to the top-left corner, with an AUC value of 0.95, indicating nearly perfect discrimination between the Top songs and other category songs. This model was also able to have a better recall and f1-score for other category songs.

	precision	recall	f1-score	support
Average	0.67	0.57	0.62	4089
Bottom	0.66	0.65	0.66	3979
Top	0.84	0.98	0.91	4058
accuracy			0.74	12126
macro avg	0.73	0.74	0.73	12126
weighted avg	0.73	0.74	0.73	12126

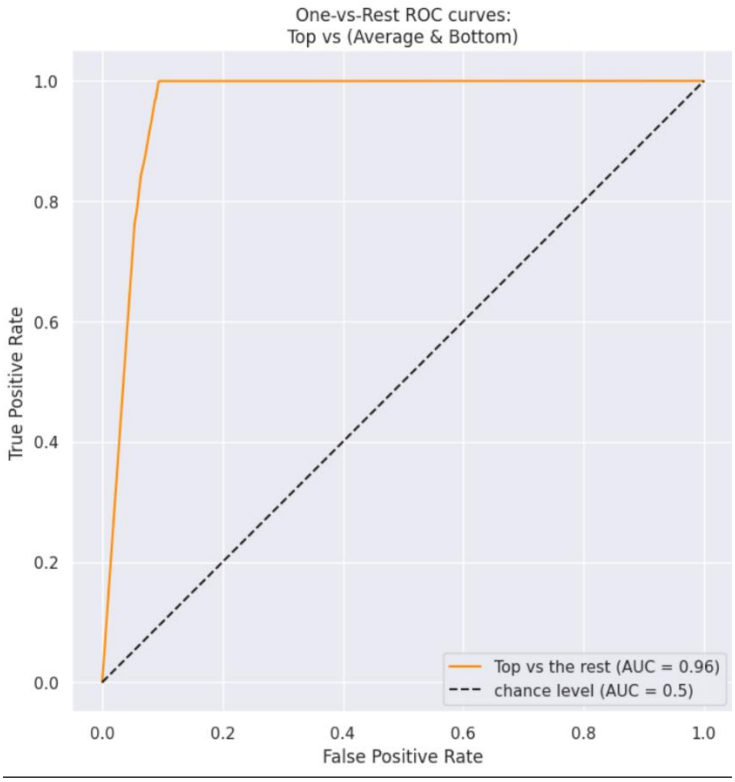


Fig.10- Decision Tree Classifier Model Performance and ROC curve

10. Song Recommendation System

We implemented a simple yet effective song recommendation system that leverages the cosine similarity metric and the TF-IDF vectorization method to calculate the similarity scores between songs in a given dataset. The system prompts the user to input a song name and retrieves the top 10 most similar songs based on the calculated cosine similarity scores. The implementation of the system is clear and concise, making it easy to understand and use. The system's performance is measured using the time and memory profiler modules, providing valuable insights into its execution time and memory usage.

One of the system's strengths is its ability to recommend songs based on user preferences and the quality of the input data. The system's accuracy depends on the quality of the data used to

compute the cosine similarity scores. Therefore, using high-quality data can lead to better recommendations.

Overall, the system presents a valuable tool for recommending similar songs to users based on their preferences and has the potential for use in various music recommendation systems. Its simplicity and effectiveness make it a promising addition to any music recommendation system.

```
# Prompt the user to input a song name and handle errors
while True:
    song_input = input("Enter a song name: ")
    if song_input in df8['track_name'].values:
        break
    else:
        print("Invalid song name. Please try again.")
```

```
# Print the top 10 similar songs
for i, score in sorted_similar_songs:
    print("{}: {}".format(i, df8.loc[i, 'track_name']))

# Measure the execution time and memory usage
execution_time = end_time - start_time
memory_usage = max(memory_usage())
print("Execution time: {:.2f} seconds".format(execution_time))
print("Memory usage: {:.2f} MiB".format(memory_usage))
```

```
Enter a song name: Hold On
38178: My Boy
37224: Eu Sou Perfeito (Soldier Boy)
21735: Solus
681: Wind in Your Sails
208: Almost Lover
1348: Can't Stop
24862: Waves - Robin Schulz Radio Edit
531: So Are We
45751: Seyre Dursun Aşk - Akustik
118: get better
Execution time: 0.03 seconds
Memory usage: 927.86 MiB
```

Fig.11- Recommendation provided by Model for User Input: “Hold On”

11. Project results

1. The decision tree classifier model has the highest overall accuracy of 74% and error rate of 26%. in classifying songs between Top vas others. Recall of 98% shows it has a great ability to classify true positives. The model also has the highest precision score of 84% and F1 score of 91% which shows improvement in predicting alternate classes. The AUC score of 0.96 represents that the model nearly perfectly classifies Top songs.

2. The KNN classifier has the next higher overall accuracy of 71%. The recall is as high as 99% showing the best performance in terms of classifying True Positives. AUC score of 0.96 implies the same. However, it is lagging the decision tree model because of its low precision value as we are looking for a balance in our results.

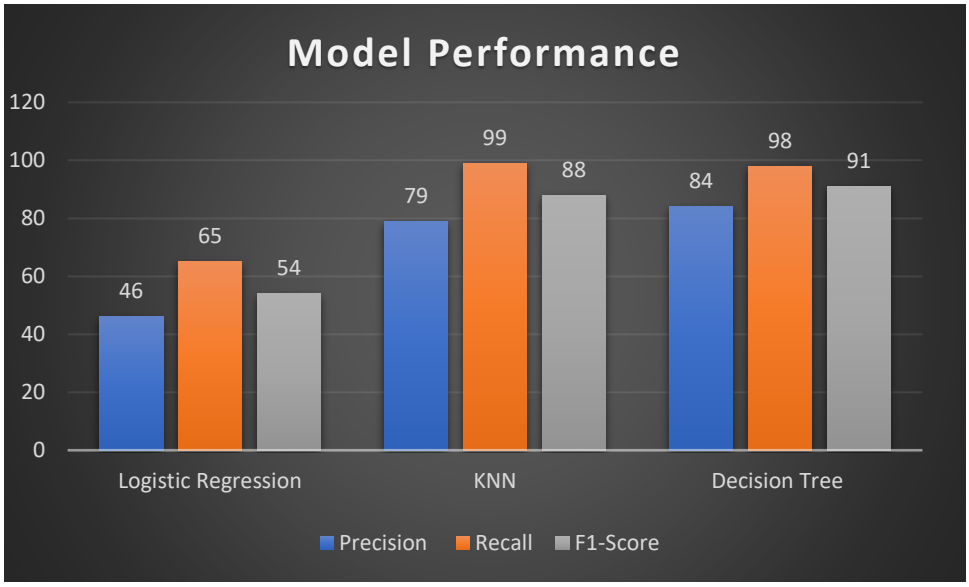


Fig.12- Model Performance by each of Classification Models

3. Logistic regression along with all the linear and polynomial regression models performed poorly in predicting popularity. This is essential because the response variable was not linearly dependent on any of the predictor variables.
4. Utilized NLP techniques like Bag of Words, TF-IDF, and Latent Semantic Analysis along with cosine similarity to make a song recommendation system based on the user's favorite song or input.

12. Project Outcome

In this project, the goal was to predict popularity and provide a recommendation system. The popularity values of the dataset came from Spotify's own model which provided popularity scores based on the listening volume of a particular song and not based on the song features or track genre. Due to this, the regression models miserably failed to predict popularity. However, the classification models were excellent in classifying the Top songs which was the real requirement.

A future work in this project might be popularity prediction using Neural Network algorithm. We are confident that more powerful and compact algorithms such as Neural Networks can

easily outperform and can obtain complex nonlinear relationships between the features and popularity which will be a great learning outcome.

It was also evident that the songs recommended per user's input were very close in terms of their song features in the real world which shows that this is a good model for a recommendation if the user's preference is the song features rather than its views.

In all, Decision Tree Classifier was the best model with a very high Recall, precision, and AUC score for classifying Top songs. A content-based recommender system was also modeled using cosine similarity.

13. References

1) Kaggle Dataset: https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset
2) IE 7275 Class Lecture Notes for Data Mining Models and Performance Evaluation.
3) Rabel, Marine Chameque. "Content Based Music Recommendation System." *degree project in computer science and engineering*, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, 18 Aug. 2020, http://www.diva-portal.org/smash/get/diva2:1515217/FULLTEXT01.pdf
4) Cosine Similarity: https://www.learndatasci.com/glossary/cosine-similarity/
5) Performance Evaluation: https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15