Prediction on Song Popularity with Audio and Social Media Features

---

Group 10 - Members

Jessie Chen (tyc360@nyu.edu)

Yu-Jui Chen (yjc464@nyu.edu)

Anusha Sanka (hns@nyu.edu)

Anisha Salunkhe (ass722@nyu.edu)

BUSINESS PROBLEM STATEMENT :

- Considering many factors which have been involved in making blockbuster music possible, there are a plethora of factors that could decide its fate. We try to solve this dilemma by considering various factors that vary for each “hot” song, and in turn, affect how the record company should invest in a specific song.

- Potential question from Business - Imagine the record company receives a demo, how do they know whether this song or the composer of the song is worth the investment?

- Song popularity can be defined in various ways like its place on the billboard list, volume, and sentiment related to song and artist on social-platforms or awards the song has obtained. Details of music semantics (acoustics and lyrics), social context, artist reputation, and on-line digital platforms play a huge role in the popularity of the song.

- We are attempting to build a model that can take major attributes that influence the song’s popularity and predict the popularity of the songs to help record companies make investment.

- For helping in the decision, we want to analyse the impact of social media on song popularity and see a correlation in song title among the proposed and HIT songs over the years.

I. DATA UNDERSTANDING

1. Data Collection :

- We considered six data sources considering the problem statement. To get a variety of songs from all genres, we used that as our base to collect songs. In a total we had ~2GB data (song details). We have used Spotify API's to get acoustic features of songs, details of artists and their popularity. We have further used Twitter API to extract tweets of songs after one week of release date.

- We have built two models to explore acoustic and social platforms' impact on song popularity, the acoustic model is built using all acoustic and artist related attributes of song and the latter model uses tweets along with preliminary data.

  The dataset and idea of using them are described below.

No.	Dataset source	Volume of Data	Attributes targeted
1	Spotify API – search	2000 songs per genre per year from 1960 - 2019 2.2 million songs in total	Basic information including song name, artist, album, and popularity for 35 genres
2	Spotify API – audio features	As above	13 audio features
3	Spotify API – artists	4.6k artists in total	Artist followers
4	Spotify API – albums	29.3k albums in total	Release date
5	Billboard API	100 songs per week from 1960 - 2019 27.8k songs in total	As "HIT" (positive) instances
6	Twitter API	songs released in years 2013 - 2019	Tweets tagged by song name or song and artist name.

- We have used unique identification numbers to join the data, UID used were *track_id*(for songs), for *artist_id*(for artist details) and *album_id(* for album) from Spotify data. Tweets were joined using song names similarly billboard hit list was joined using song name and artist name.

2. Data Preparation :

- For training data, we had to tweek certain data, short description of those tweeks are following:

2.1 Converted release date to number of months (till present date):

- To account release date impact on target variable we have converted release date to numeric form by calculating number of months from release date to present date.

2.2 Converted popularity from numeric variable to be categorical (binary):

- We have defined our target variable - "popularity" of a song using two metrics. One is defined from numeric popularity provided from spotify and the other is defined by the billboard list. If a song is displayed in the billboard list - it is tagged as a "hot" song.

- In order to compare the prediction performance, we converted the numeric popularity from Spotify to the binary variable. We divided the number of songs on the Billboard chart by the number of songs in the main dataset we have to get a popular song rate. Then we use this rate to determine the popularity threshold, which is 56 in our case, from Spotify. Any song with more than 56 of popularity is considered a popular song.

2.3 Convert categorical variable genre to a dummy variable:

- We have 35 genres in total. However, some songs are labeled with more than one genre. Therefore, we create dummy variables for genres, and if the song is labeled with "acoustic" and "blues", then the genre_acoustic column and the genre_blues are set to '1', and other genre dummies are set '0'.

2.4 Filtered Tweetiment Analysis:

- Tweets were cleaned by removing links and special characters to perform sentiment analysis of song popularity.

** ***DATA SIZE INFO*** - After all the data processing, there were 1.3 million songs in total. About 1.75% of songs are popular/hot.

- An instance in our dataset for the models contains the following attributes:

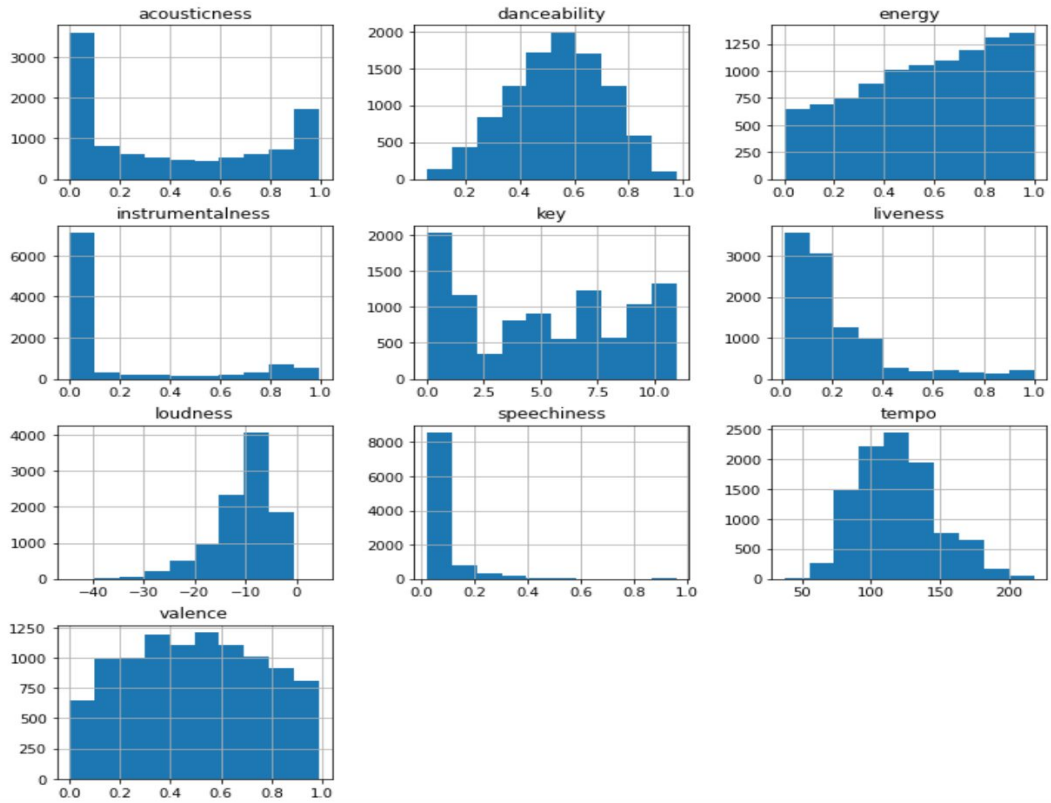
ATTRIBUTE	**DEFINITION**	**DATA TYPE** (and range)
genre	The genre the track is associated with.	categorical
month	The number of months from the release date to the present.	numeric
artist_followers	The total number of followers of the artist.	numeric
popularity*	The popularity of the track. The closer to 100 represents the song is more popular.	numeric (0 – 100) → categorical (0, 1)
hot*	Indicates whether the song was once on the billboard chart. 1 is yes while 0 is no.	categorical (0, 1)
acousticness	Whether the track is acoustic. The closer to 1 represents the high confidence the track is acoustic.	numeric (0 – 1)
danceability	How suitable a track is for dancing. The closer to 1 represents the track is most danceable.	numeric (0 – 1)
duration_ms	The duration of the track in milliseconds.	numeric
energy	Represents a measure of intensity and activity. The closer to 1 represents the track is full of energy.	numeric (0 – 1)
instrumentalness	Predicts whether a track contains no vocals.	numeric (0 – 1)
key	Estimates the overall key of the track. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on.	Categorical (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)
liveness	Detects the presence of an audience in the recording. The closer to 1 represents an increased probability that the track was performed live.	numeric (0 – 1)
loudness	The overall loudness of a track in decibels (dB). Values typically range between -60 and 0 db.	numeric
mode	Indicates the modality (major or minor) of a track.	categorical (0, 1)
speechiness	Detects the presence of spoken words in a track. The closer to 1 represents the track is speech-like.	numeric (0 – 1)
tempo	The overall estimated tempo of a track in beats per minute (BPM).	numeric
time_signature	An estimated overall time signature of a track.	Categorical
valence	Describes the musical positiveness conveyed by a track.	numeric (0 – 1)
Tweets(** for second model)	Describes the sentiment of the song analyzed.	Positive, negative or neutral .

- **Data Exploration :**
    - We started our analysis by examining the dataset in aggregate and computing basic statistics, the results we as following:
    - We could infer the spectrum of value for each attribute.

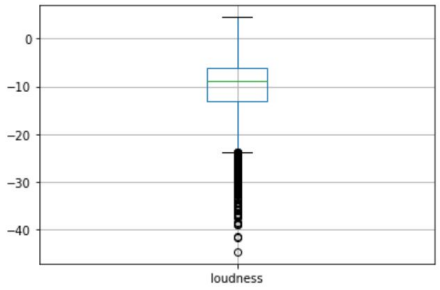
	month	artist_followers	popularity	hot	acousticness	danceability	duration_ms	energy	instrumentalness	key
count	1.317502e+06	1.317502e+06	1.317502e+06	1.317502e+06	1.317502e+06	1.317502e+06	1.317502e+06	1.317502e+06	1.317502e+06	1.317502e+06
mean	2.167816e+02	6.304383e+05	1.597477e+01	1.751117e-02	4.116768e-01	5.357386e-01	2.486143e+05	5.613691e-01	1.877731e-01	5.223002e+00
std	1.741842e+02	2.268822e+06	1.546507e+01	1.311661e-01	3.742588e-01	1.740816e-01	1.235882e+05	2.801122e-01	3.252116e-01	3.526177e+00
min	3.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	4.090000e-02	1.503300e+04	2.000000e-05	0.000000e+00	0.000000e+00
25%	7.800000e+01	1.075900e+04	2.000000e+00	0.000000e+00	3.050000e-02	4.140000e-01	1.839330e+05	3.340000e-01	1.200000e-06	2.000000e+00
50%	1.700000e+02	7.408900e+04	1.200000e+01	0.000000e+00	3.170000e-01	5.430000e-01	2.291070e+05	5.880000e-01	5.130000e-04	5.000000e+00
75%	3.140000e+02	3.684460e+05	2.600000e+01	0.000000e+00	7.980000e-01	6.650000e-01	2.849600e+05	8.070000e-01	2.080000e-01	8.000000e+00
max	7.220000e+02	6.191375e+07	1.000000e+02	1.000000e+00	9.960000e-01	9.950000e-01	6.025542e+06	1.000000e+00	1.000000e+00	1.100000e+01

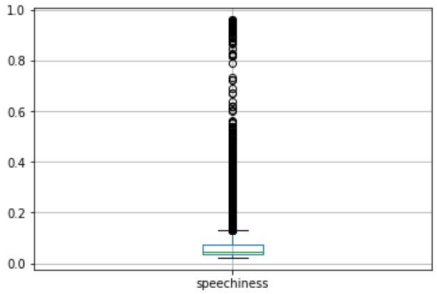
- Further, we checked how is the data skewed and find outliers if they exist using graphical analysis, and the results are as following:

    1. Left skewed data: Liveness, acousticness, spechiness.
    2. Right skewed data: Energy, loudness.



- Few plots which displayed some outliers were:





- **Feature selection:**

1. Using a random forest model we selected attributes that depicted a strong correlation to target variables.

```
Feature importances:
[0.10371623 0.10315742 0.06924289 0.10624307 0.1027767  0.10752282
 0.09670443 0.10206514 0.10182748 0.10674381]
```

	Danceability	Energy	Key	Loudness	Speechiness	Acousticness	Instrumentalness	Liveness	Valence	Temp
Importance	0.103716	0.103157	0.069243	0.106243	0.102777	0.107523	0.096704	0.102065	0.101827	0.106744

2. We analyzed word clouds to understand the most common words in popular songs.

II. Modeling:

- We tried to fit the data into **Five** classifiers to get the best model that depicts that data. Models and their parameters are described in the table.

- Parameter tuning is done over "Pop" and "Hot" attributes to get better performance in the models (as shown in Table X). We make the parameter we chose to tune be bold.

Table: Models For "POP" and "HOT" Parameters

MODEL	**PARAMETERS** (FOR POP)	**PARAMETERS** (FOR HOT)
Logistic Regression	**C=0.01**, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='auto', n_jobs=None, penalty='l2', random_state=42, solver='lbfgs', tol=0.0001, verbose=0, warm_start=False	**C=0.0001**, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100, multi_class='auto', n_jobs=None, penalty='l2', random_state=42, solver='lbfgs', tol=0.0001, verbose=0, warm_start=False
Decision Tree	ccp_alpha=0.0, class_weight=None, criterion='gini', **max_depth=7**, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort='deprecated', random_state=42, splitter='best'	ccp_alpha=0.0, class_weight=None, criterion='gini', **max_depth=7,** max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort='deprecated', random_state=42, splitter='best'
XGBoost	base_score=0.5, booster=None, colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1, impo rtance_type='gain', interaction_constraints=None, learning_rate=0.300000012, max_delta_step=0, **max_depth=6,** min_child_weight=1, missing=nan, monotone_constraints=None, n_estimators=100, n_jobs=0, num_parallel_tree=1, objective='binary:logistic', random_state=42, reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method=None, validate_parameters=False, verbosity=None	base_scbase_score=0.5, booster=None, colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1, importance_type='gain', interaction_constraints=None, learning_rate=0.300000012, max_delta_step=0, **max_depth=7,** min_child_weight=1, missing=nan, monotone_constraints=None, n_estimators=100, n_jobs=0, num_parallel_tree=1, objective='binary:logistic', random_state=42, reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1, tree_method=None, validate_parameters=False, verbosity=None
Random Forest	bootstrap=True, ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, max_samples=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, **n_estimators=100,** n_jobs=None, oob_score=False, random_state=42, verbose=0, warm_start=False	bootstrap=True, ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=None, max_features='auto', max_leaf_nodes=None, max_samples=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, **n_estimators=100,** n_jobs=None, oob_score=False, random_state=42, verbose=0, warm_start=False
Neural Network	activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08, **hidden_layer_sizes=15,** learning_rate='constant', learning_rate_init=0.001, max_fun=15000, max_iter=200, momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5, random_state=42, shuffle=True, solver='adam', tol=0.0001,	activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08, **hidden_layer_sizes=1,** learning_rate='constant', learning_rate_init=0.001, max_fun=15000, max_iter=200, momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5, random_state=42, shuffle=True, solver='adam', tol=0.0001, validation_fraction=0.1, verbose=False, warm_start=False

- **Inference and conclusions of parameter tuning :**

1. We used GridSearch to do parameter tuning. Due to restrictions on the running speed, if we can speed up, we can try more values of the parameters or more parameters to get better performance.

2. We could try other scoring methods (such as f1 score) in our approach. In our case, the data is imbalanced. Imbalanced data might give us the illusion of high accuracy. Thus, F1-score might be a better metric to be considered to evaluate the performance in some cases.

III. EVALUATION :

- Evaluation of models was done using data and visualizations of the following:
  - Accuracy
  - ROC-AUC
  - PR-AUC
  - Confusion Matrix
  - Cost and Benefits Analysis

1. **For POP models (doing Spotify popularity as our target variable),** we conclude that:

   - **ROC-AUC:**
     1. Tree-based models have better performance overall (See. Figure 1)
     2. If we can have more data to support the Neural Network model, we might have better performance.

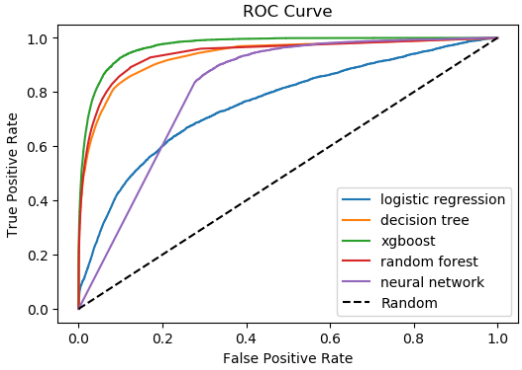


Figure 1.

- **PR-AUC:**

1. Tree-based models have better performance in general (See Figure 2.).
2. We might have better performance in the Neural Network model if we have more data.

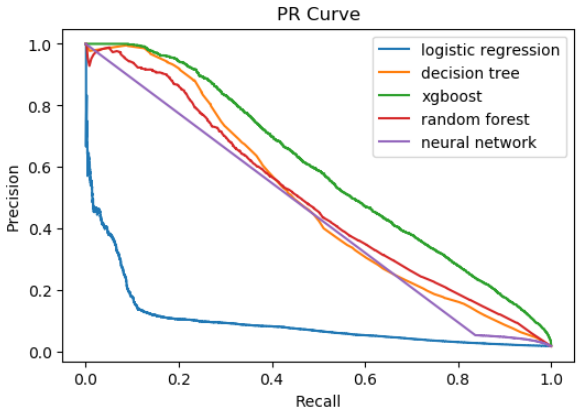


Figure 2.

- **Observation from ROC-AUC & PR-AUC :**

1. Xgboost is the best from ROC-AUC and PR-AUC.
2. Decision Tree and Random Forest are in the middle performance. Neural Network model and Logistic Regression are performing the worst among.
3. Due to imbalanced data, our classifiers have high ROC-AUC scores while low PR-AUC scores.

Model	Accuracy	ROC-AUC	Precision	PR-AUC
Logistic Regression	0.981	0.758	0.098	0.097
Decision Tree	0.985	0.935	0.485	0.494
XGBoost	0.986	0.972	0.591	0.591
Random Forest	0.984	0.942	0.488	0.494
Neural Network	0.981	0.816	0.053	0.449

- Next for Confusion Matrix:

  According to Figure 3., Xgboost shows the best performance. The Decision Tree and Random Forest presents the middle. Neural Network model and Logistic Regression are of the worst performance.

Figure 3.

- **COST and PROFIT ANALYSIS :**

1. Confusion Matrix * Cost Matrix

   - We multiplied the confusion matrix with the cost matrix to see an overall cost and benefits for all our classifiers. The number in the cost matrix is from IFPI (See Reference 1), which is an organization representing the recording industries.

   - As we can see in figure 4., Xgboost is the best in profit at a lower cost: 2,061,840 thousand USD, and a larger cost for 8,247,360 thousand USD. The Decision Tree and Random Forest are in the middle. Neural Network and Logistic Regression are presenting the worst performance. (See figure 4.)

```
Revenue with lower cost using logistic regression :  136062
Revenue with larger cost using logistic regression :  544248


Revenue with lower cost using decision tree :  1731090
Revenue with larger cost using decision tree :  6924360


Revenue with lower cost using xgboost :  2061840
Revenue with larger cost using xgboost :  8247360


Revenue with lower cost using random forest :  1706317
Revenue with larger cost using random forest :  6825268


Revenue with lower cost using neural network :  0
Revenue with larger cost using neural network :  0
```

Figure 4.

2. Profit-Curve:

a. We have two scenarios which will be "**With a smaller budget for individuals**" and "**With a larger budget for individuals**". (See figure 5.)

1) **For both scenarios,** we can target the first 4,290 songs ranked by the expected profit. We prove that Xgboost is the best among all the models, in which the maximum profits are the largest.

2) Random Forest and Decision Tree are both in the middle, and Logistic Regression and Neural Network are of the worst performance.

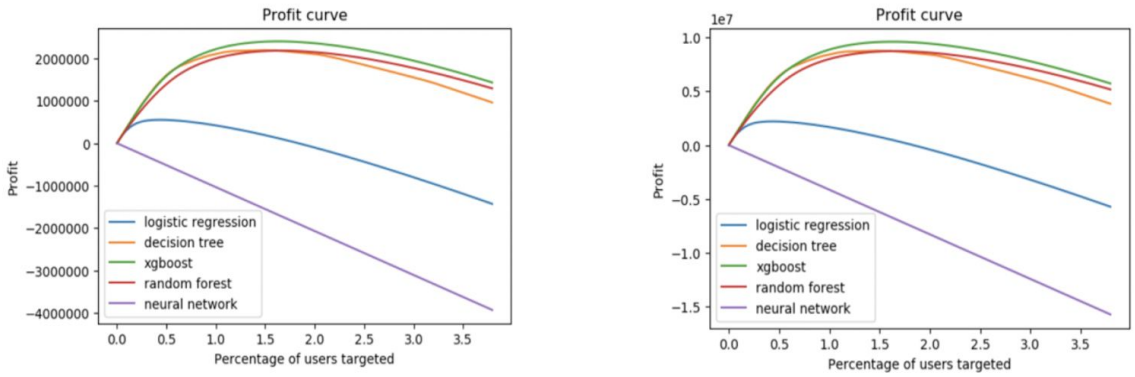


Figure 5.

FOR "HOT" MODEL (doing billboard hot as our target variable):

- **ROC-AUC:**

1. Tree-based models have better performance overall (See. Figure 6.)
2. If we can get more data to support the Neural Network model, we might have better performance. From figure 7, we can see the performance of the Neural Network is no different from random guesses.

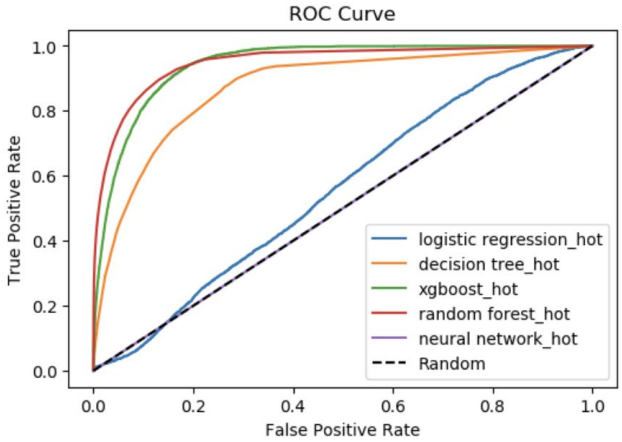


Figure 6.

- **PR-AUC :**

1. Tree-based models have better performance in general (See Figure 7).
2. If we have more data to support the Neural Network model, we might have better performance.

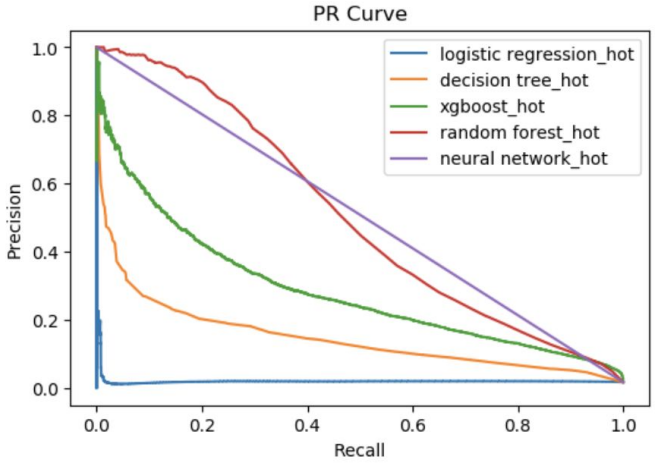


Figure 7.

- **Observation from ROC-AUC & PR-AUC :**

1. Random Forest is the best from ROC-AUC and PR-AUC.
2. Xgboost is in the middle. Decision tree, Neural Network, and Logistic Regression are performing the worst.
3. Due to imbalanced data, our classifiers have high ROC-AUC scores while low PR-AUC scores.

Model	Accuracy	ROC-AUC	Precision	PR-AUC
Logistic Regression	0.982	0.557	0.021	0.021
Decision Tree	0.982	0.873	0.141	0.149
XGBoost	0.983	0.943	0.285	0.284
Random Forest	0.985	0.949	0.496	0.505
Neural Network	0.982	0.500	0.018	0.509

- **Next for Confusion Matrix:**

1. According to Figure 8., Random Forest shows the best performance. The Decision Tree and Xgboost are in the middle. Neural Network model and Logistic Regression are the worst performance.

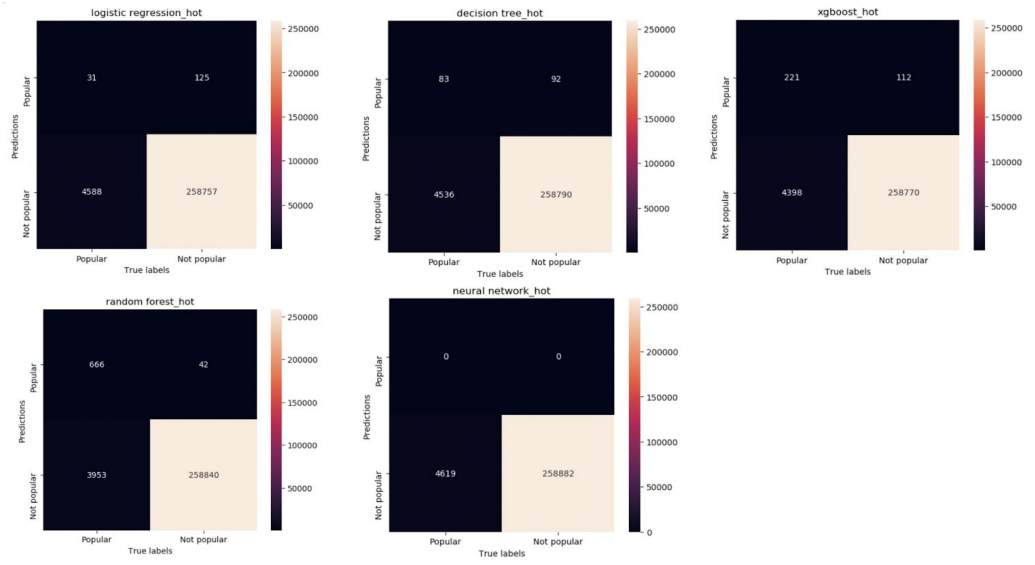


Figure 8.

- **Cost and Profit Analysis** :

1. Confusion Matrix * Cost Matrix

- We multiplied the confusion matrix with the cost matrix to see an overall cost and benefits for all our classifiers. The number in the cost matrix is from IFPI (reference 1), which is an organization representing the recording industries.

- As we can see in Figure 9. Random Forest is the best in Profit at lower cost: 926718$, and a larger cost for 3706872$. The Decision Tree and Xgboost are in the middle. The Neural Network model and Logistic Regression are presenting the worst performance. (See figure 10.)

```
Revenue with lower cost using logistic regression_hot :  -18387
Revenue with larger cost using logistic regression_hot :  -73548


Revenue with lower cost using decision tree_hot :  72109
Revenue with larger cost using decision tree_hot :  288436


Revenue with lower cost using xgboost_hot :  258483
Revenue with larger cost using xgboost_hot :  1033932


Revenue with lower cost using random forest_hot :  926718
Revenue with larger cost using random forest_hot :  3706872


Revenue with lower cost using neural network_hot :  0
Revenue with larger cost using neural network_hot :  0
```

Figure 9.

2. Profit-Curve:

We have two scenarios which will be "**With a smaller budget for individuals**" and "**With a larger budget for individuals**". (See figure 10.)

- **For both scenarios**, we can target the first 2190 songs ranked by the expected profit. We prove that Random Forest is the best among all the models, in which the maximum profits are the largest.

- Further, we prove that Xgboost and Decision Tree are both in the middle performance, and the Neural Network and Decision Tree are in the worst performance. Also, one important observation is that Logistic Regression delivers a good performance in the expected profits.

- However, according to the performance on ROC-AUC, PR-AUC, Confusion Matrix, we can infer that the probabilities of class Logistic Regression give us might likely be wrong. That is, the classifier is less reliable. Again, Neural Networks only gives negative profits.

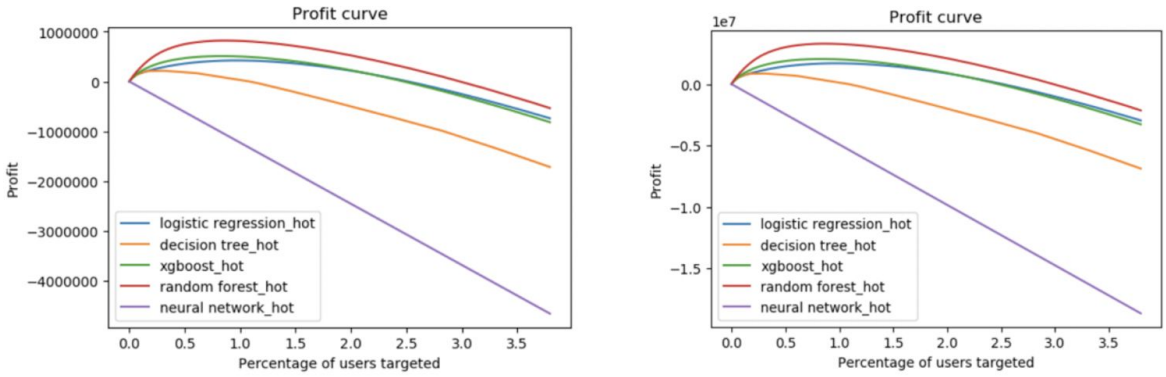


Figure 10.

Inference and conclusions of Evaluation:

1. Establishing Spotify's popularity as our target variable, it shows that Xgboost is the best classifier among 5 models. Though Xgboost is not an excellent classifier. Yet, it has the best accuracy, ROC-AUC, precision score, and cost-benefits compared to the other 4 models.

2. For billboard hot being our target variable, it shows that Random Forest is the best classifier. Nevertheless, billboard hot might not be a good choice for the target variable compared to Spotify's popularity. With billboard hot as our target variable, all classifiers seem to have lower performance in PR-AUC and Confusion Matrix.

3. Generally, it looks like Spotify's popularity is a better choice over billboard hot as our target variable.

IV. EXPERIMENT ANALYSIS :

1. **On Tweets:**

   From sentiment analysis on tweets we inferred following correlation between tweets and popularity (Spotify parameter):

   1. Volume of tweets is directly proportional to artist popularity i.e. social platform reflect artist audience reach parameter.

   2. Positive sentiment of a song shows a positive correlation with popularity after the release date.

   3. Negative sentiment of a song doesn't show any correlation with song popularity after release date. If the sentiment of the song is "negative" doesn't mean that it is not a popular song. (tweets can be due external factors like social or political issues).

2. **On word cloud:**

   a. Few visualisations of words used in popular songs billboard data of years - 2015,2010,2000,1990,1980,1970) are:

   - We can see from the data that the most common words used in all hit songs are - "LOVE", "BABY","YOU'RE"," GIRL" etc.

V. Deployment :

We imagine the scene of the business process as follows:

Demo → Audio Features → ML Models → Prob. of Popularity of the Song

Accordingly, as the company receives the Demo from the potential/existing singers, the investment team of the record company will use the existing techniques to get audio features of songs. Further, data scientists will use the machine learning model to predict whether it will be a hit song with other available data. The probability of the popularity of songs helps the record company to determine if the demo is worth investment.

However, the firm should be aware of the following issues when deployment:

- **Accessibility:** How fast can we collect the data, such as getting the audio features from the song? For instance, every record company wants to sign competitive singers in the first place. Hence, the whole process, including audio features and the use phase of the machine learning model should be fast enough to make a response.

- **Popular Trends:** The investment team should keep up with the latest trends in various music styles to ensure the performance of the model by renewing the model and adding the latest songs into training data routinely.

- **The importance of Subjective Judgment:** In the initial phase of the Machine Learning Models, human reviewers should be involved to avoid investment loss because music preference is a subjective consciousness to everyone. The ML model might not be the total representative of market preference.

In terms of related risks, the record companies that adopt this machine learning model should follow the internal principles of the procurement process to prevent ethical problems. Further, for this proposal, we could consider adding more attributes that might have an impact on our result. In addition, we could try more models and more parameter tuning to mitigate any obstacles to the proposal.

Team Contributions:

Jessie Chen: Business Problem Formulation, Data Understanding, Data Preparation, Modeling, Evaluation, Cost And Profit Analysis, Documentation

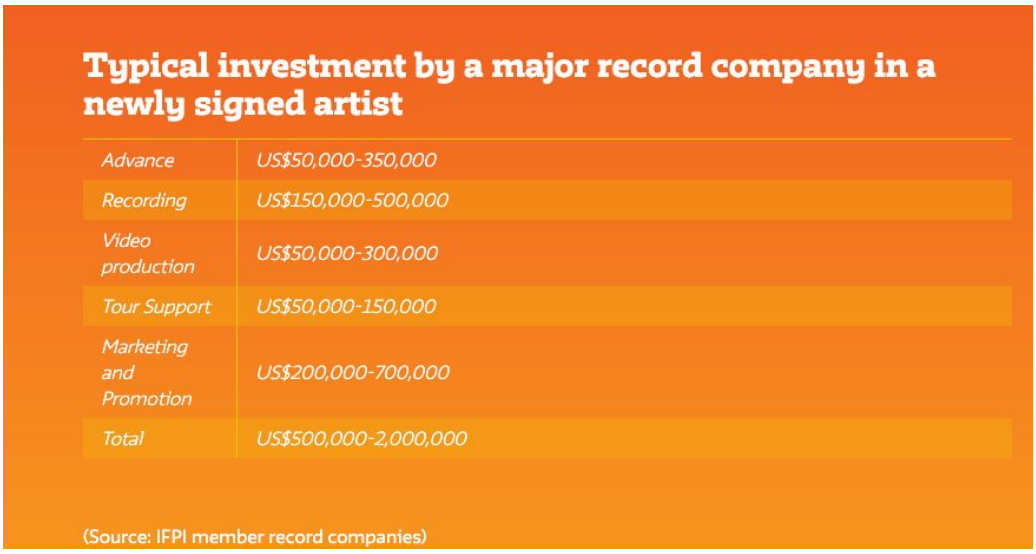
Yu-Jui Chen: Business Problem Formulation, Modeling, Evaluation, Cost And Profit Analysis, Documentation

Anusha Sanka: Business Problem Formulation, Data Understanding, Data Preparation, Experimental Analysis, Documentation

Anisha Salunkhe: Business Problem Formulation, Modeling, Cost And Profit Analysis, Documentation

EXHIBIT:

1. Exhibit : How we get the number to form the cost matrix (the reference)



Typical investment by a major record company in a newly signed artist

Advance	US$50,000-350,000
Recording	US$150,000-500,000
Video production	US$50,000-300,000
Tour Support	US$50,000-150,000
Marketing and Promotion	US$200,000-700,000
Total	US$500,000-2,000,000

(Source: IFPI member record companies)

REFERENCES:

1. Record companies discover, nurture, and promote artistic talent. They are by far the largest investors in artists' careers. (n.d.). Retrieved from https://www.ifpi.org/how-record-labels-invest.php
2. http://the.echonest.com