

# Classic analysis on historically hittest songs in the USA via XGBoost

Haosheng Shi

Institute of Statistics and Big Data  
*Renmin University of China*

April 24, 2022

# Presentation Overview

## ① Introduction of the Data

- Problem restatement
- Dataset description

## ② Introduction of the Method

- XGBoost
- Problems encountered in implements

## ③ Results on data

# Problem restatement

- The basic question I am concerned about is how classic (or lively) the historically hottest songs are at present.
- I crystallize "hit songs" by considering those entering the Billboard. And I use the popularity of those songs on Spotify to measure their liveliness.
- I expect to design a method to easily predict the popularity with its first entry timestamp and other available features. I also want to learn which features contribute the most in the process of prediction.

# Used datasets

- The sorted US Billboard Hot100 ranking data from 1970 to 2009[2]:

```
head(PopUSA)
```

```
## # A tibble: 6 x 29
##   recording_id artist_name artist_name_cle~ track_name first_entry quarter year
##   <dbl> <chr> <chr> <chr> <date> <chr> <dbl>
## 1 1090 Candi Stat~ CANDISTATON I'm Just ~ 1970-01-03 1970 Q1 1970
## 2 2260 Dionne War~ DIONNEWARWICK I'll Neve~ 1970-01-03 1970 Q1 1970
## 3 2514 The Rascals RASCALS Hold On 1970-01-03 1970 Q1 1970
## 4 3013 Sly & The ~ SLYFAMILYSTONE Thank You~ 1970-01-03 1970 Q1 1970
## 5 4217 The 5th Di~ 5THDIMENSION Blowing A~ 1970-01-03 1970 Q1 1970
## 6 6082 Rotary Con~ ROTARYCONNECTION Want You ~ 1970-01-03 1970 Q1 1970
## # ... with 22 more variables: fiveyear <dbl>, decade <dbl>, era <dbl>,
## # cluster <dbl>, hTopic_01 <dbl>, hTopic_02 <dbl>, hTopic_03 <dbl>,
## # hTopic_04 <dbl>, hTopic_05 <dbl>, hTopic_06 <dbl>, hTopic_07 <dbl>,
## # hTopic_08 <dbl>, tTopic_01 <dbl>, tTopic_02 <dbl>, tTopic_03 <dbl>,
## # tTopic_04 <dbl>, tTopic_05 <dbl>, tTopic_06 <dbl>, tTopic_07 <dbl>,
## # tTopic_08 <dbl>, n <int>, artist_popularity <int>
```

Figure: the PopUSA dataset

- Sample tracks entering the board in 1970-1972, 1980-1982, 1990-1992 and 1974-1976.

The first three datasets were used for training and the last one for testing.

# Used datasets

- I used the R package "Spotifyr" to extract some new features.

```
##           name popularity artist_popularity count count_self
## 1           Bad Times          20           56 14          1
## 2 Remember (Walking In The Sand) 42           79  4          1
## 3           I Thank You          36           72  9          3
## 4           Got To Love Somebody 41           65  5          1
## 5           Too Hot             58           70 13          1
## 6           Back On My Feet Again 51           47 26          1
## cluster name_complexity duration_ms danceability loudness liveness
## 1           8           9      436507      0.680 -13.518  0.1130
## 2          11          30     243933      0.238  -9.613  0.3500
## 3           7          11     203973      0.792  -7.850  0.0963
## 4           1          20     413707      0.739  -5.549  0.0614
## 5          12           7     226827      0.719 -11.168  0.1100
## 6          12          21     198467      0.544  -6.619  0.1530
## speechiness energy key valence year_board time_duration hTopic_01
## 1      0.0457 0.559 7 0.802      0      42.28219 0.0417861905
## 2      0.0684 0.555 0 0.476      0      42.26301 0.0736626152
## 3      0.0463 0.663 7 0.680      0      42.24384 0.1078997358
## 4      0.1110 0.896 7 0.854      0      42.24384 0.0282657805
## 5      0.1130 0.515 7 0.697      1      42.24384 0.0726571334
## 6      0.0345 0.730 9 0.522      0      42.24384 0.0002164915
```

**Figure:** first columns for the data\_80s dataset

- loudness, speechiness, danceability of these songs
- the artist popularity and the track popularity

*Track popularity was extracted as the highest one of songs from the same singer.*

- the name complexity of the track
- the list of songs with duplicated names

I used the list of songs with duplicated names to produce some features like:

- count
- count\_self
- covering\_density

*I regarded the appearance of all duplicated ones after the original hit song as an inhomogeneous Poisson process and estimated the intensity by a Haar wavelet-based Multiresolution analysis[3].*

*I took the scale parameter  $J = 2$  and acquired 4 coefficient parameters for each original song.*

# Used datasets

Why do I use these variables?

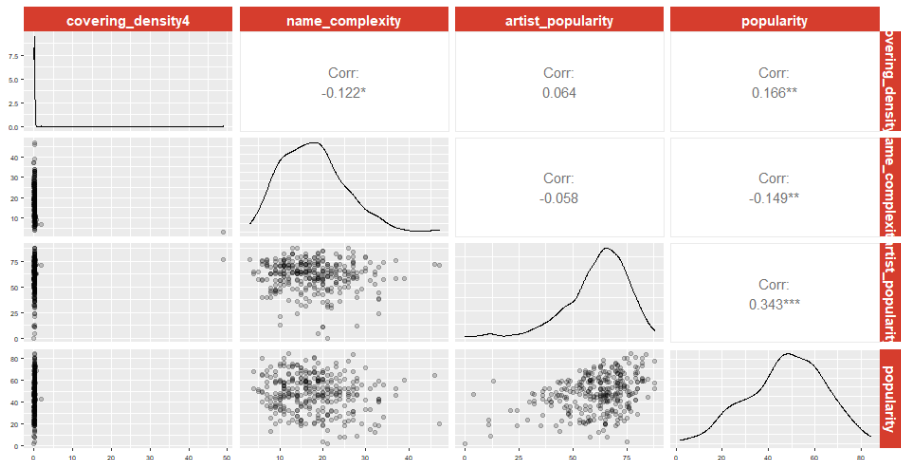


Figure: variable correlation in the data\_80s dataset

- **XGboost**(e**X**treme **G**radient **B**oosting) is one of the most popular machine learning algorithms. As its name, this method seeks to bring the extreme learning efficiency of the boosting technique into play.
- Besides the benefits it inherits from boosting algorithms, the method introduces the weighted quantile sketch algorithm to handle instance weights in approximate tree learning. [1]
- Parallel, distributed and out-of-core computation makes the learning process very fast.[1]



# XGBoost

## Classical Gradient Boosting

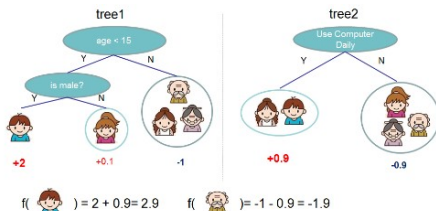
- Target classifier:

$$F(\mathbf{x}) = \sum_{m=1}^M \beta_m f(\mathbf{x}; \gamma_m)$$

where  $b(\mathbf{x}, \gamma)$  is the simple classifier parameterized by  $\gamma$ .

- Objective function:

$$\min_{\{\beta_m, \gamma_m\}_1^M} \sum_{i=1}^N L \left( y_i, \sum_{m=1}^M \beta_m f(\mathbf{x}_i; \gamma_m) \right)$$



**Figure 1: Tree Ensemble Model.** The final prediction for a given example is the sum of predictions from each tree.

### 1 Overfitting-prevention:

- 1 Except for the normal error term  $\sum_{i=1}^n L(y_i, F(\mathbf{x}_i))$ , another penalization term  $\sum_{k=1}^m \Omega(f) = \sum_{k=1}^m (\gamma T + \frac{1}{2} \lambda \|\mathbf{w}\|^2)$  is added to the goal function;
  - $T$  is the number of leaf nodes of the classifier  $f(\mathbf{x})$ ;
  - $\mathbf{w}$  is the score attributed to  $f(\mathbf{x})$ .
- 2 As opposed to the first-order approximation in classical gradient boosting, XGBoost uses the second-order approximation of  $L$  for gradient descent to improve efficiency.

$$\begin{aligned}\mathcal{L}^{(t)} &= \sum_{i=1}^n L(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \sum_{k=1}^m \Omega(f_k) \\ &\simeq \sum_{i=1}^n \left[ L(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i) \right] + \Omega(f_t)\end{aligned}$$

### 2 Splitting the tree:

#### 1 Exact Greedy Algorithm

Traverse the features and the splitting points to find which match can improve the splitting gain the most.

$$\text{Gain} = \frac{1}{2} \left[ \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma$$

where  $I_j = \{i \mid q(x_i) = j\}$ ,  $G_j = \sum_{i \in I_j} g_i$ ,  $H_j = \sum_{i \in I_j} h_i$  are defined for each leaf node.

#### 2 Approximate Algorithm

### 3 Weighted Quantile Sketch for weighted data

### 4 Sparsity-aware Split Finding for sparse data

# classification goals

- Combine the 1970-1972, 1980-1982, 1990-1992 datasets and divide them into training data (80%) and testing data (20%).  
*Use stratified sampling to maintain the distribution consistency of the testing data and the training data.*
- I want to classify the songs into three categories: forgotten, ordinary and classical.

- $\leq 30$  forgotten
- 30 – 60 ordinary
- $> 60$  classical

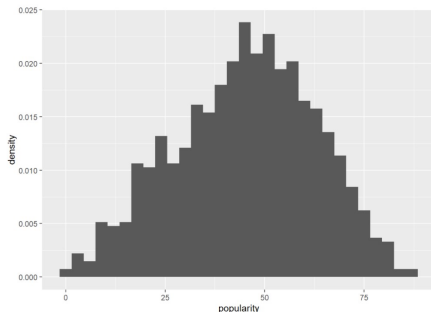


Figure: A histogram for the popularity

# A simple try on the data

- Directly run the XGboost algorithm on the training data and test the output model for 10 times:

I set the learning rate  $\eta = 0.3$ , subsample ratio used in splitting the tree 0.7, features sampled in splitting the tree 0.4, number of trees  $n = 2000$  (and a further early stopping criterion was set.)

```
kable(indicator$table/10)
```

	[0,30]	(30,60]	(60,100]
[0,30]	12.3	7.9	0.8
(30,60]	24.8	90.5	25.2
(60,100]	0.9	7.6	11.0

```
kable(t(indicator$byClass[,c(1,2,5,11)]))
```

	Class: [0,30]	Class: (30,60]	Class: (60,100]
Sensitivity	0.3236842	0.8537736	0.2972973
Specificity	0.9391608	0.3333333	0.9409722
Precision	0.5857143	0.6441281	0.5641026
Balanced Accuracy	0.6314225	0.5935535	0.6191348

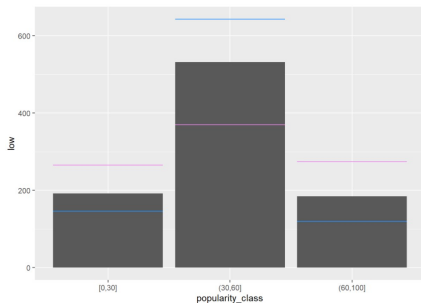
Figure: Preliminary result

- Obviously the result was unsatisfactory.

# Analysis of the Challenges

## Features observed from the data:

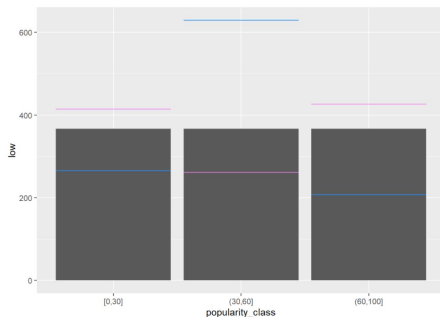
- 1 This is a highly unbalanced classification problem.  
Specifically We care more about which songs are in  $[0,30]$  and  $[60,100]$ , while those in  $(30,60]$  take account almost half of the population.  
Most of the data points will be classified into this category.
- 2 Borders between classes are not clear, hence common criteria for judging the accuracy may not be suitable.



**Figure:** Numbers in the 3 target classes; I also plot the combination of  $[0, 25]$ ,  $(25, 65]$ ,  $(65, 100]$  in blue and  $[0, 35]$ ,  $(35, 55]$ ,  $(55, 100]$  in red.

# Methods to overcome the challenges

- Correlations between input and output variables are not high, so synthesization methods such as SMOTE are not trustworthy for their characteristic of introducing too much noise and blurring the borderline.
- A simple oversampling technique may lead to unreasonable results on the border of the classification.



**Figure:** Numbers after a simple oversampling of minority classes.

## Assumption

Assume that we can get access to the full popularity information in the training data instead of the popularity class only.

# Methods to overcome the challenges

- 1 For the first problem, I decided to use the weighted oversampling technique on the training data.  
Instead of data on the border, extreme ones are highlighted.

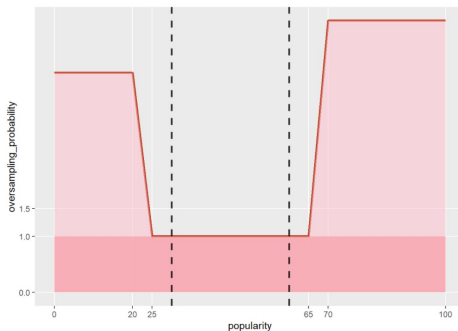


Figure: Weight1

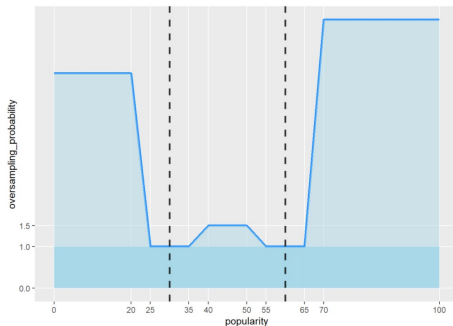


Figure: Weight2

- 2 Another technique to increase the accuracy is firstly classifying the songs into 4 classes and then merging two of them into one.



# Methods to overcome the Challenges

- ③ For the second problem, I designed some new indicators to evaluate the performance:

$$\text{Extreme Sensitivity for } [0, 30] = \frac{\#\{\text{songs in } [0, 20] \text{ predicted as } [0, 30]\}}{\#\{\text{songs in } [0, 20]\}}$$

$$\text{Extreme Sensitivity for } (60, 100] = \frac{\#\{\text{songs in } (70, 100] \text{ predicted as } (60, 100]\}}{\#\{\text{songs in } (70, 100]\}}$$

$$\text{Extreme Error for } [0, 30] = \frac{\#\{\text{songs in } (60, 100] \text{ predicted as } [0, 30]\}}{\text{songs predicted as } [0, 30]}$$

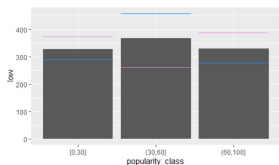
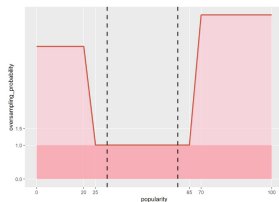
$$\text{Extreme Error for } (60, 100] = \frac{\#\{\text{songs in } [0, 30] \text{ predicted as } (60, 100]\}}{\#\{\text{songs predicted as } (60, 100]\}}$$

$$\text{Adjusted Precision for } [0, 30] = \frac{\#\{\text{songs in } [0, 45] \text{ predicted as } [0, 30]\}}{\#\{\text{songs predicted as } [0, 30]\}}$$

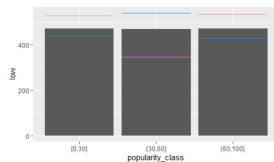
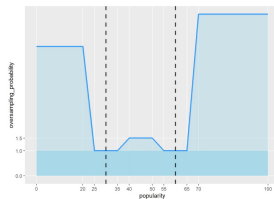
$$\text{Adjusted Precision for } (30, 60] = \frac{\#\{\text{songs in } (20, 70] \text{ predicted as } (30, 60]\}}{\#\{\text{songs predicted as } (30, 60]\}}$$

$$\text{Adjusted Precision for } (60, 100] = \frac{\#\{\text{songs in } (45, 100] \text{ predicted as } (60, 100]\}}{\#\{\text{songs predicted as } (60, 100]\}}$$

# Results of direct classification

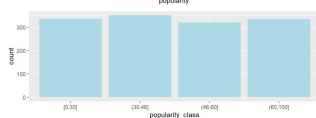
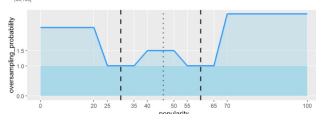
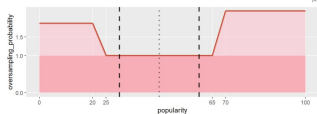
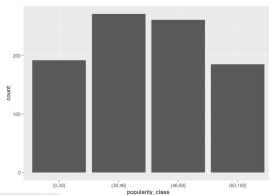


evaluation measures	Class: [0,30]	Class: (30,60]	Class: (60,100]
Balanced Accuracy	0.685	0.605	0.668
Sensitivity	0.518	0.612	0.516
Extreme Sensitivity	0.681		0.667
Precision	0.482	0.682	0.424
Adjusted Precision	0.76	0.909	0.718
Extreme Error	0.059		0.08
Specificty	0.852	0.597	0.82



evaluation measures	Class: [0,30]	Class: (30,60]	Class: (60,100]
Balanced Accuracy	0.696	0.612	0.693
Sensitivity	0.534	0.62	0.557
Extreme Sensitivity	0.681		0.72
Precision	0.501	0.689	0.457
Adjusted Precision	0.79	0.905	0.765
Extreme Error	0.035		0.067
Specificty	0.859	0.604	0.83

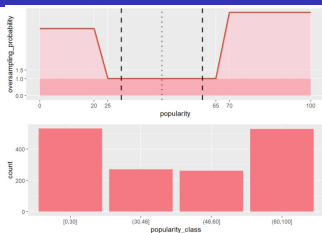
# Results of indirect classification



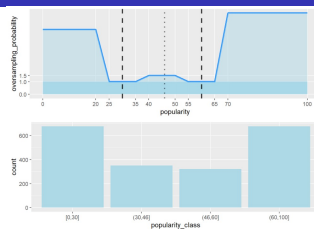
evaluation measures	Class: [0,30]	Class: (30,60]	Class: (60,100]
Balanced Accuracy	0.721	0.594	0.665
Sensitivity	0.613	0.555	0.522
Extreme Sensitivity	0.755		0.718
Precision	0.486	0.681	0.412
Adjusted Precision	0.76	0.924	0.726
Extreme Error	0.044		0.06
Specificty	0.828	0.633	0.809

evaluation measures	Class: [0,30]	Class: (30,60]	Class: (60,100]
Balanced Accuracy	0.681	0.609	0.713
Sensitivity	0.529	0.577	0.608
Extreme Sensitivity	0.686		0.762
Precision	0.456	0.694	0.462
Adjusted Precision	0.728	0.915	0.766
Extreme Error	0.045		0.07
Specificty	0.832	0.64	0.818

# Results of indirect classification



evaluation measures	Class: [0,30]	Class: (30,60]	Class: (60,100]
Balanced Accuracy	0.705	0.585	0.699
Sensitivity	0.666	0.393	0.651
Extreme Sensitivity	0.837		0.87
Precision	0.408	0.714	0.398
Adjusted Precision	0.687	0.943	0.731
Extreme Error	0.06		0.086
Specificty	0.743	0.777	0.747

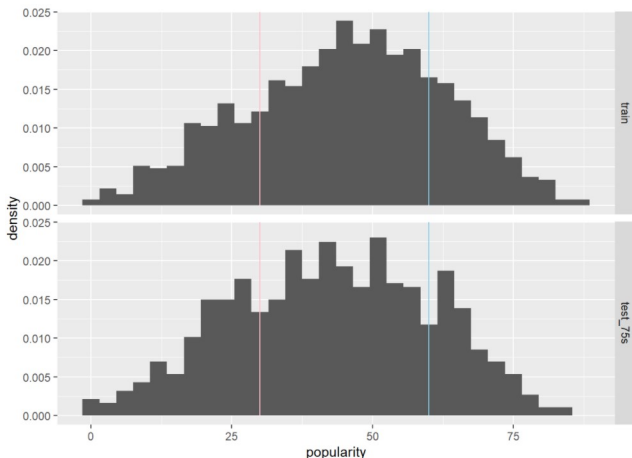


evaluation measures	Class: [0,30]	Class: (30,60]	Class: (60,100]
Balanced Accuracy	0.689	0.564	0.674
Sensitivity	0.629	0.387	0.614
Extreme Sensitivity	0.729		0.771
Precision	0.4	0.679	0.373
Adjusted Precision	0.717	0.914	0.714
Extreme Error	0.074		0.076
Specificty	0.75	0.741	0.735

- The indirect classification always performs better than the direct one.
- The first weight always performs better than the second one.
- Apparently a tradeoff exists between sensitivity and precision.

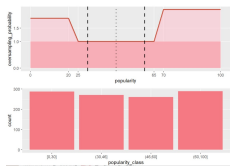
# Experiments on the 75s data

In case the improved prediction result on the original data were a fortuity, I introduced another dataset including the 1974-1976 hit songs. I used the whole combined dataset to train the model and predict the result on the new one.



# Experiments on the 75s data

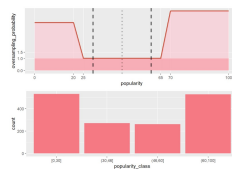
## Prediction results



```
## Reference
## Prediction [0,30] (30,60] (60,100]
## [0,30] 81.16 67.08 5.22
## (30,60] 74.24 169.70 36.30
## (60,100] 14.60 100.22 75.48
```

```
## reality
## predict [0,20] (20,30] (30,45] (45,60] (60,70] (70,100]
## [0,30] 39.58 41.58 41.76 25.32 4.56 0.66
## (30,60] 26.34 47.90 88.58 81.12 30.18 6.12
## (60,100] 5.08 9.52 40.66 59.56 50.26 25.22
```

evaluation measures	Class: [0,30]	Class: (30,60]	Class: (60,100]
Balanced Accuracy	0.659	0.559	0.709
Sensitivity	0.477	0.504	0.645
Extreme Sensitivity	0.557		0.788
Precision	0.529	0.606	0.397
Adjusted Precision	0.801	0.884	0.71
Extreme Error	0.034		0.077
Specificty	0.841	0.615	0.774



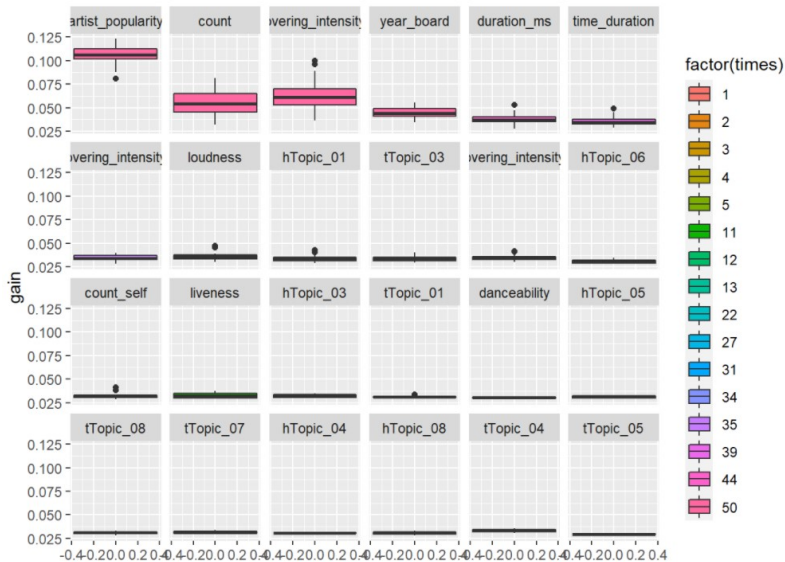
```
## Reference
## Prediction [0,30] (30,60] (60,100]
## [0,30] 102.76 94.22 8.26
## (30,60] 46.82 119.40 23.84
## (60,100] 20.42 123.38 84.90
```

```
## reality
## predict [0,20] (20,30] (30,45] (45,60] (60,70] (70,100]
## [0,30] 49.52 53.24 58.96 35.26 7.04 1.22
## (30,60] 14.24 32.58 62.96 56.44 19.94 3.90
## (60,100] 7.24 13.18 49.08 74.30 58.02 26.88
```

evaluation measures	Class: [0,30]	Class: (30,60]	Class: (60,100]
Balanced Accuracy	0.689	0.554	0.721
Sensitivity	0.604	0.354	0.726
Extreme Sensitivity	0.697		0.84
Precision	0.501	0.628	0.371
Adjusted Precision	0.788	0.905	0.696
Extreme Error	0.04		0.089
Specificty	0.774	0.754	0.716

# Experiments on the 75s data

## Feature importance



# Experiments on the 75s data

## Result Analysis

```
al=testing_data[which(pred2=="[0,30]"&reference2=="(70,100)")]%%624,]
n=(al%>%group_by(name)%>%count())
kable(full_join(n,unique(al))%>%filter(n>=10)%>%select(n,popularity,artist_popularity,covering_intensity4,year_board,count)%>%arrange(desc(n)))
```

name	n	popularity	artist_popularity	covering_intensity4	year_board	count
Hooked On A Feeling	34	77	62	0.1658548	0	11
Some Kind Of Wonderful	22	71	62	0.0456251	0	13


*Hooked On A Feeling* was covered by a Blue Swede, a Swedish band, and was made a massive hit, reaching #1 in the US, Holland, Australia, and Canada. As the first Swedish singer to score a No. 1 hit in the U.S., Blue Swede paved the way for well-known ABBA. However, they themselves have never again cracked the American market. As you can see, the song was quickly forgotten and did not hit the year board.

(<https://www.songfacts.com/facts/bj-thomas/hooked-on-a-feeling>)

In 2014, the song was used in the hot film *Guardians of the Galaxy*, which helped it return to #1 in America and revived this song. (The picture comes from <https://www.wnycstudios.org/podcasts/soundcheck/segments/that-was-a-hit-hooked-feeling-blue-swede>)

 **Soundcheck**

Podcast   Gig Alerts   Weekly Roundup   Archive   About

 **LISTEN FOR FREE**

## That Was a Hit?!?: Blue Swede, 'Hooked On A Feeling'

 **LISTEN**    Download    Embed



# Experiments on the 75s data

## Result Analysis

```
al=testing_data[which(pred2=="[0,30]"&reference2=="(70,100)")]%%624,]
n=(al%>%group_by(name)%>%count())
kable(full_join(n,unique(al))%>%filter(n>=10)%>%select(n,popularity,artist_popularity,covering_intensity4,year_board,count)%>%arrange(desc(n)))
```

name	n	popularity	artist_popularity	covering_intensity4	year_board	count
Hooked On A Feeling	34	77	62	0.1658548	0	11
Some Kind Of Wonderful	22	71	62	0.0456251	0	13

*Some Kind Of Wonderful* is also interesting because the popularity for 1999 remastered version is a lot higher than the 1974 version, possibly leading to a misclassification. I review the history of this song and found that there is another cover version of Huey Lewis and the News at 1994, and this one also hit into U.S. Billboard Hot 100. So maybe the listeners follow that version to this one and choose the latest versions (or versions recommended by Spotify) to listen to.

##	track_name	artist_name	release_date	popularity
## 1	Some Kind of Wonderful	The Drifters	1962-04-23	50
## 2	Some Kind of Wonderful	Carole King	1971-12-01	37
## 3	Some Kind Of Wonderful - Remastered	Grand Funk Railroad	1974-12-01	32
## 4	Some Kind Of Wonderful - Remastered 1999	Grand Funk Railroad	1999-01-01	71
## 5	Some Kind Of Wonderful - Remastered 1999	Grand Funk Railroad	1999-01-01	69
## 6	Some Kind Of Wonderful - Remastered 1999	Grand Funk Railroad	1999-01-01	69
## 7	Some Kind Of Wonderful	Reflection Eternal	2000-10-17	26
## 8	Some Kind Of Wonderful	Joss Stone	2003-01-01	48
## 9	Some Kind Of Wonderful	Joss Stone	2003-01-01	48
## 10	Some Kind Of Wonderful	Peter Cincotti	2004-01-01	38
## 11	Some Kind Of Wonderful	Grand Funk Railroad	2006-01-01	51
## 12	Some Kind Of Wonderful - Remastered 2002	Grand Funk Railroad	2008-01-01	30
## 13	Some Kind Of Wonderful	Grand Funk Railroad	2008-01-01	34
## 14	Some Kind of Wonderful	Michael Bublé	2009-10-09	33
## 15	Some Kind of Wonderful	.Douglas Lyons,Alan Wiggins	2014-04-01	30
## 16	Some Kind Of Wonderful	Rod Stewart	2021-11-12	41
## 17	Some Kind Of Wonderful	Mean Marc Ash	2021-12-22	23

# Experiments on the 75s data

## Result Analysis

```
a2=testing_data[which(pred2=="(60,100)"&(reference2=="[0,20]"))%%624,]  
n=(a2%>group_by(name)%>%count())  
kable(full_join(n,unique(a2))%>%filter(n>=20)%>%select(n,popularity,artist_popularity,covering_intensity4,year_board,count)%>%arrange(desc(n)))
```

name	n	popularity	artist_popularity	covering_intensity4	year_board	count
Shotgun Shuffle	50	19	66	0.1379401	0	12
TVC 15	47	16	81	0.1853351	0	12
On And On	40	11	63	0.1814385	0	21
Wake Up Susan	39	13	64	0.1856453	0	14
Good Vibrations	32	0	61	0.3862761	0	33
Put A Little Love Away	32	16	65	0.0301855	0	13
Once You Hit The Road	30	7	66	0.1076859	0	7
Kung Fu	23	0	64	0.3782997	0	32

Missing values might occur when scratching *Shotgun Shuffle*, *Put A Little Love Away* and *On and On* because none of them have records around 1974-1976.

*TVC 15* and all of its cover versions have low popularity and I guess this can be owed to its bizarre name.

The songs *Kung Fu* and *Real Man* may be misclassified because there are too many duplicated songs. As you can see, many songs I searched contain the phrase "Kung Fu" but not exactly the original track.

For *Good Vibrations*, this was a song sung by the American rock band the Beach Boys in 1966 and immediately became a commercial hit. "Characterized by its complex soundscapes, episodic structure and subversions of pop music formula, it was the most expensive single ever recorded." Many subsequent versions came out, including our target, the one covered by Todd Rundgren which peaked at 34. However, the cover version is apparently forgotten while the original track remains a high popularity.

For *Wake Up Susan* (peaked at 56), I found that the group The Spinners had other hits *I'm Coming Home*, *The Rubberband Man* and *Love or Leave*. However, except for *The Rubberband Man* (This track reached top 2 in 1976), popularity values of others are all below 30. Hence despite for the effectiveness of the "year\_board" variable, a more detailed feature can make the classification more precise. For example, we can consider whether the track ever topped in the board or not. It is a pity that we could not get that information.



# Experiments on the 75s data

## Result Analysis

```
a3=testing_data[which(pred2=="(60,100)"&(reference2=="(70,100)"))%%624,]  
a4=testing_data[which(pred2=="(60,100)"&(reference2=="(60,70)"))%%624,]  
n=(a3)%>group_by(name)%>count()  
kable(full_join(n,unique(a3))%>filter(n>=30)%>select(n,popularity,artist_popularity,covering_intensity4,year_board,count)%>arrange(desc(n)))
```

name	n	popularity	artist_popularity	covering_intensity4	year_board	count
Annie's Song	50	71	72	0.1512290	1	14
Carry On Wayward Son	50	76	69	0.1554550	0	17
Dancing Queen	50	84	82	0.2486229	0	26
Jolene	50	74	75	0.1657559	0	19
Killer Queen	50	72	88	0.2589617	1	20
Mamma Mia	50	79	82	0.1544459	0	43
Night Moves	50	74	71	0.1243115	0	10
Ob-La-Di, Ob-La-Da	50	75	88	0.1552909	0	10
Piano Man	50	80	80	0.1357264	0	13
Rebel Rebel	50	74	81	0.1209832	0	10
Rock And Roll All Nite	50	79	76	0.1833183	0	23
Somebody To Love	50	73	88	0.1708561	0	22
Sweet Home Alabama	50	85	75	0.1060302	0	14
Waterloo	50	72	82	0.1361061	1	15
Bohemian Rhapsody	49	78	88	0.1999468	1	24
I'm Not In Love	49	74	65	0.1222122	0	17
Sara Smile	49	71	74	0.0615731	1	14
50 Ways To Leave Your Lover	48	73	72	0.1537416	0	16
You Sexy Thing	48	73	62	0.1535195	1	14
Free Bird	45	75	75	0.0911948	0	12
One Of These Nights	44	71	79	0.1222621	0	10
Sweet Emotion	44	77	79	0.0917341	0	15
Come And Get Your Love	37	80	65	0.3313802	0	34
Feel Like Makin' Love	34	71	64	0.0305968	1	18
The 100th Day	34	75	66	0.2145305	0	18

# Conclusion

- Although predicting the popularity of old songs is not an easy task because of the complex unknown affecting factors, the artist popularity and the appearance frequency of covering versions are still the gold standards in measuring the classic of the song.
- Our XGboost with designed weighted oversampling method is effective in lifting the performance of predicion.



Tianqi Chen and Carlos Guestrin.

## Xgboost: A scalable tree boosting system.

ACM, 2016.



M. Mauch, R. M. Maccallum, M. Levy, and A. M. Leroi.

## The evolution of popular music: Usa 1960-2010.

*R Soc Open Sci*, 2(5):150081, 2015.



Y. Taleb and EaK Cohen.

Multiresolution analysis of point processes and statistical thresholding for haar wavelet-based intensity estimation.

*Annals of the Institute of Statistical Mathematics*, 73, 2021.