Top 10 Billboard Prediction Analysis

Anirudh Madhavan
Jocelyn Jasso
Michael Opiela
Justin Ward
Mia Nguyen

Challenge Statement

Our challenge: How can we reveal what tracks are most likely to be hits before we commit resources towards promoting them?

By remedying this problem, we can:

- Look at possible long-term and short-term investment for new artists
- Look at average and variance of music sales/popularity of songs and compare
- Open up new product development opportunities

Marketing Opportunity

billboard

HOT 100

	SONG	ARTIST
1	**Anti-Hero**	Taylor Swift
2	**Lavender Haze**	Taylor Swift
3	**Maroon**	Taylor Swift
4	**Snow On The Beach**	Taylor Swift ft. Lana Del Rey
5	**Midnight Rain**	Taylor Swift
6	**Bejeweled**	Taylor Swift
7	**Question...?**	Taylor Swift
8	**You're On Your Own, Kid**	Taylor Swift
9	**Karma**	Taylor Swift
10	**Vigilante Shit**	Taylor Swift

chart dated November 5, 2022

We want to know which tracks will "blow up" so we can put our marketing resources behind the right ones and maximize our return. By getting a clearer understanding of possible costs & profits for labels looking to release new records, we can help them allocate their resources in the most profit-maximizing way given the information we have.

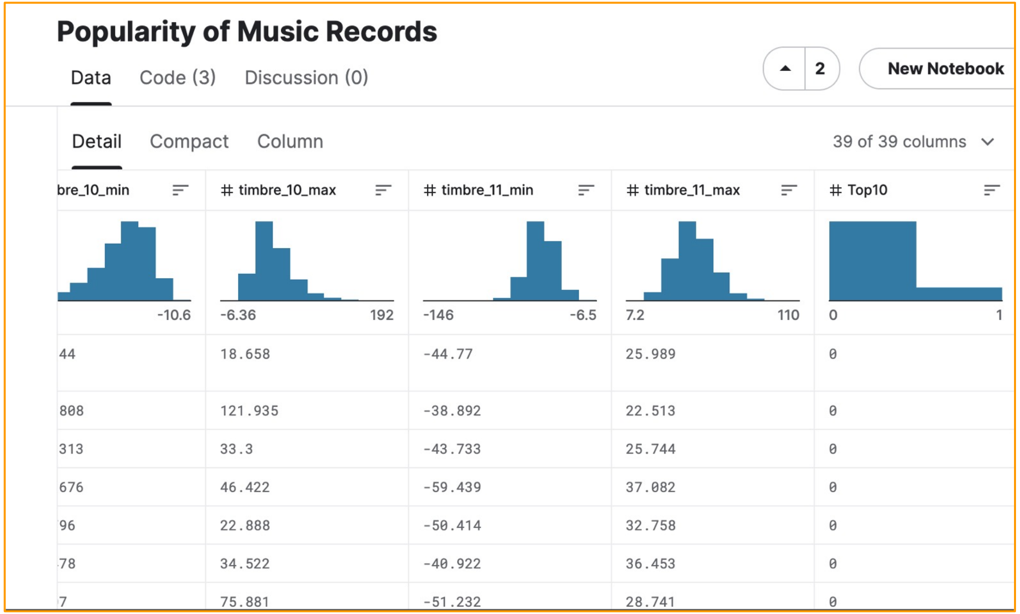
Dataset

Source: Kaggle

Contains more than **30 columns** with different song characteristics, such as tempo, estimatekey, timbre, and timesignature, including artistID and songID

We used "Top 10" binary column to showcase predicted popularity of each song

Key is to **determine probability of songs success** (more difficult), not probability of failure (easier)



Clean Data

Data was surprisingly completely clean, there were no NA values for any of the variables

```
      year        songtitle          artistname           songID
 Min.   :1990   Length:7574        Length:7574        Length:7574
 1st Qu.:1997   Class :character   Class :character   Class :character
 Median :2002   Mode  :character   Mode  :character   Mode  :character
 Mean   :2001
 3rd Qu.:2006
 Max.   :2010
   artistID          timesignature   timesignature_confidence    loudness
 Length:7574        Min.   :0.000   Min.   :0.0000           Min.   :-42.451
 Class :character   1st Qu.:4.000   1st Qu.:0.8193           1st Qu.:-10.847
 Mode  :character   Median :4.000   Median :0.9790           Median : -7.649
                    Mean   :3.894   Mean   :0.8533           Mean   : -8.817
                    3rd Qu.:4.000   3rd Qu.:1.0000           3rd Qu.: -5.640
                    Max.   :7.000   Max.   :1.0000           Max.   :  1.305
     tempo        tempo_confidence      key          key_confidence
 Min.   :  0.00   Min.   :0.0000   Min.   : 0.000   Min.   :0.0000
 1st Qu.: 88.86   1st Qu.:0.3720   1st Qu.: 2.000   1st Qu.:0.2040
 Median :103.27   Median :0.7015   Median : 6.000   Median :0.4515
 Mean   :107.35   Mean   :0.6229   Mean   : 5.385   Mean   :0.4338
 3rd Qu.:124.80   3rd Qu.:0.8920   3rd Qu.: 9.000   3rd Qu.:0.6460
 Max.   :244.31   Max.   :1.0000   Max.   :11.000   Max.   :1.0000
     energy            pitch           timbre_0_min      timbre_0_max
 Min.   :0.00002   Min.   :0.00000   Min.   : 0.000   Min.   :12.58
 1st Qu.:0.50014   1st Qu.:0.00300   1st Qu.: 0.000   1st Qu.:53.12
 Median :0.71816   Median :0.00700   Median : 0.027   Median :55.53
 Mean   :0.67547   Mean   :0.01082   Mean   : 4.123   Mean   :54.46
 3rd Qu.:0.88740   3rd Qu.:0.01400   3rd Qu.: 2.772   3rd Qu.:57.08
 Max.   :0.99849   Max.   :0.54100   Max.   :48.353   Max.   :64.01
  timbre_1_min      timbre_1_max      timbre_2_min      timbre_2_max
 Min.   :-333.72   Min.   :-74.37   Min.   :-324.86   Min.   : -0.832
 1st Qu.:-160.12   1st Qu.:171.13   1st Qu.:-167.64   1st Qu.:100.519
 Median :-107.75   Median :194.40   Median :-136.60   Median :129.908
 Mean   :-110.79   Mean   :212.34   Mean   :-136.89   Mean   :136.673
 3rd Qu.: -59.71   3rd Qu.:239.24   3rd Qu.:-106.51   3rd Qu.:166.121
 Max.   : 123.73   Max.   :549.97   Max.   :  34.57   Max.   :397.095
```

```
  timbre_3_min      timbre_3_max      timbre_4_min      timbre_4_max
 Min.   :-495.36   Min.   : 12.85   Min.   :-207.07   Min.   : -0.651
 1st Qu.:-226.87   1st Qu.:127.14   1st Qu.: -77.69   1st Qu.: 83.966
 Median :-170.61   Median :189.50   Median : -63.83   Median :107.422
 Mean   :-186.11   Mean   :211.81   Mean   : -65.28   Mean   :108.227
 3rd Qu.:-131.56   3rd Qu.:290.72   3rd Qu.: -51.34   3rd Qu.:130.286
 Max.   : -21.55   Max.   :499.62   Max.   :  51.43   Max.   :257.801
  timbre_5_min      timbre_5_max      timbre_6_min       timbre_6_max
 Min.   :-262.48   Min.   :-22.41   Min.   :-152.170   Min.   : 12.70
 1st Qu.:-113.58   1st Qu.: 84.64   1st Qu.: -94.792   1st Qu.: 59.04
 Median : -95.47   Median :119.90   Median : -80.418   Median : 70.47
 Mean   :-104.00   Mean   :127.04   Mean   : -80.944   Mean   : 72.17
 3rd Qu.: -81.02   3rd Qu.:162.34   3rd Qu.: -66.521   3rd Qu.: 83.19
 Max.   : -42.17   Max.   :350.94   Max.   :   4.503   Max.   :208.39
  timbre_7_min        timbre_7_max      timbre_8_min       timbre_8_max
 Min.   :-214.791   Min.   : 15.70   Min.   :-158.756   Min.   :-25.95
 1st Qu.:-101.171   1st Qu.: 76.50   1st Qu.: -73.051   1st Qu.: 40.58
 Median : -81.797   Median : 94.63   Median : -62.661   Median : 49.22
 Mean   : -84.313   Mean   : 95.65   Mean   : -63.704   Mean   : 50.06
 3rd Qu.: -64.301   3rd Qu.:112.71   3rd Qu.: -52.983   3rd Qu.: 58.46
 Max.   :   5.153   Max.   :214.82   Max.   :  -2.382   Max.   :144.99
  timbre_9_min      timbre_9_max      timbre_10_min      timbre_10_max
 Min.   :-149.51   Min.   :  8.415   Min.   :-208.82   Min.   : -6.359
 1st Qu.: -70.28   1st Qu.: 53.037   1st Qu.:-105.13   1st Qu.: 39.196
 Median : -58.65   Median : 65.935   Median : -83.07   Median : 50.895
 Mean   : -59.52   Mean   : 68.028   Mean   : -87.34   Mean   : 55.521
 3rd Qu.: -47.70   3rd Qu.: 81.267   3rd Qu.: -64.52   3rd Qu.: 66.593
 Max.   :   1.14   Max.   :161.518   Max.   : -10.64   Max.   :192.417
 timbre_11_min       timbre_11_max       Top10
 Min.   :-145.599   Min.   :  7.20   Min.   :0.0000
 1st Qu.: -58.058   1st Qu.: 38.98   1st Qu.:0.0000
 Median : -50.892   Median : 46.44   Median :0.0000
 Mean   : -50.868   Mean   : 47.49   Mean   :0.1477
```

First Approach - Cost-Based

First, we evaluated that the data is unbalanced

- 6455 (~83% of the dataset) being songs that doesn't make it to the top 10 billboard
- 1119 (~17% of the dataset) being songs that become a Top 10 hit

```
table(songs_data$Top10)
# This is not a 'balanced' dataset - the probability of the 1 event (song
# charting is significantly low in the dataset)
```

```
   0    1
6455 1119
```

First Approach - Cost-Based

To provide more accurate model building and minimizing discrepancies:

- Split data into training and test data (70% and 30% roughly)
- Eliminating 5 columns that have qualitative data
- Running a regression and getting rid of variables that are statistically insignificant

First Approach - Cost-Based

```
model_glm <- glm(Top10 ~ timesignature_confidence + loudness + tempo_confidence +
  energy + pitch + timbre_0_min + timbre_0_max + timbre_1_min + timbre_3_max +
    timbre_4_min + timbre_4_max + timbre_5_min + timbre_6_min + timbre_6_max +
      timbre_10_max + timbre_11_max + timbre_11_min, data = songs_train2, family = binomial(logit))
summary_glm <- summary(model_glm)
```

A matrix: 34 × 4 of type dbl

1.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.634801e+01	2.1261222004	7.68912063	1.481497e-14
timesignature	8.910904e-02	0.1006059061	0.88572371	3.757664e-01
timesignature_confidence	7.158259e-01	0.2194244529	3.26228843	1.105166e-03
loudness	3.062692e-01	0.0345066771	8.87565040	6.952591e-19
tempo	-6.239175e-04	0.0019405466	-0.32151639	7.478191e-01
tempo_confidence	5.824733e-01	0.1644896251	3.54109430	3.984711e-04
key	1.798239e-02	0.0120249120	1.49542764	1.348029e-01
key_confidence	3.034189e-01	0.1624953328	1.86724694	6.186711e-02
energy	-9.635887e-01	0.3634033427	-2.65156812	8.011895e-03
pitch	-4.408435e+01	7.7230397837	-5.70816100	1.142034e-08
timbre_0_min	2.541313e-02	0.0049090528	5.17678937	2.257368e-07

Half of the variables are eliminated

The remaining explanatory variables used on the logit regression model (in R) are:

- Time signature
- Loudness
- Tempo
- Energy
- Pitch
- Timbre min (0, 1, 4, 5, 6, 11)
- Timbre max (0, 3, 4, 6, 10, 11)

Results - Cost-Based

Not great... The accuracy of predicting flops are VERY high, but what we actually care about - **the hits** - is low

- **98%** accuracy on the flops predictions
- Only **19%** accuracy on hits predictions

```
               What the model predicts
Actual value FALSE TRUE
           0  4363   89
           1   638  153
0.980008984725966
0.193426042983565
```

Problems with First Approach

Since the dataset is imbalance to begin with, the naturally assume cutoff point of 0.5 backfired because we are weighing both outcomes equally.

Because of that, the model leans towards classifying the songs as flops since that is the majority of the data presented in the set.

However, we care about the hits infinitely MORE! Let's see if we can remedy this issue

First Approach - Cost-Based (modified)

Knowing the issue, we decided to find the optimized cutoff point:

Penalizing False Negative a lot more and False Positive a lot less

False Negative 6x as much as False Positive

Using R, we defined a cost metric, an index sequence, and utilizing coding power, sorted through every probability (by 0.01 threshold) to find a place where the cost is minimized

How the Magic Happens - Cost-Based

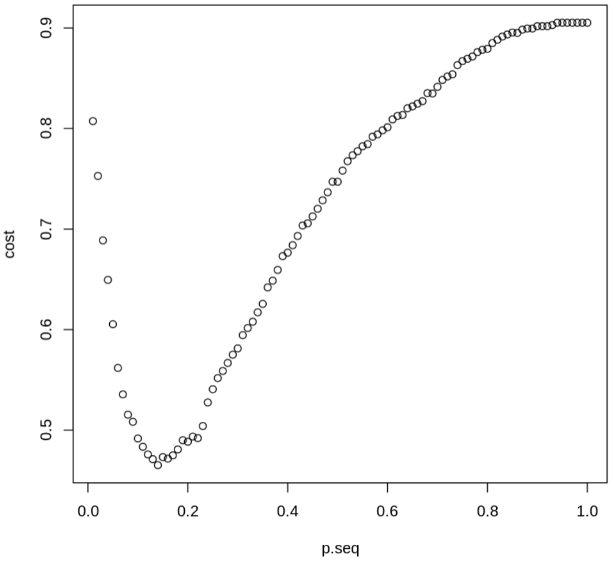
```r
cost_function = function(what_happened , model_probability , cutoff_probability)
{ weight_1 = 6 # Define the cost multiple associated with true = 1, but prediction = 0 (FN - penalize the False Negative a lot more)
  weight_2 = 1 # Define the cost multiple associated with true = 0, but prediction = 1 (FP - penalize the False Positive a lot less)
  c1 = (what_happened == 1) & (model_probability < cutoff_probability) # Counting up False Negatives
  c0 = (what_happened == 0) & (model_probability > cutoff_probability) # Counting up False Positives
  # Define a cost metric using weighted averages
  cost = mean(weight_1 * c1 + weight_2 * c0)
  return(cost)
}
```

```r
# Going through every probability to find the best cut - off

p.seq = seq(from = 0.01, to = 1 ,by = 0.01)
```

```r
# Looping through in order to find the p - cut that can lower the cost function
# to the maximum extent possible

cost = rep(0,length(p.seq))
for(i in 1 : length(p.seq)){
  cost[i] = cost_function(what_happened = songs_train2$Top10 , model_probability = songs_train2$prediction , cutoff_probability = p.seq[i])
}
```

Results - Cost-Based (modified)

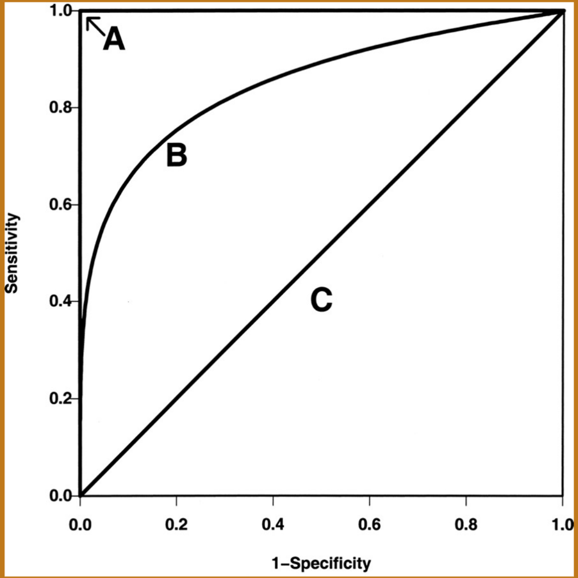
The minimized cutoff point is **0.14**



Results Improved significantly!

Cutoff probability	0.14	
	CONFUSION MATRIX	
	Predicted	
Actual	0	1
0	3147	1305
1	189	602
	Hit prediction rate	**76.11%**
	Flop prediction rate	**70.69%**
	We improved our hit prediction rate, sacrificed a little on the flop prediction rate to achieve this - we think this is likely to be a trade off record companies may be willing to accept!	

Second Approach - ROC Curve



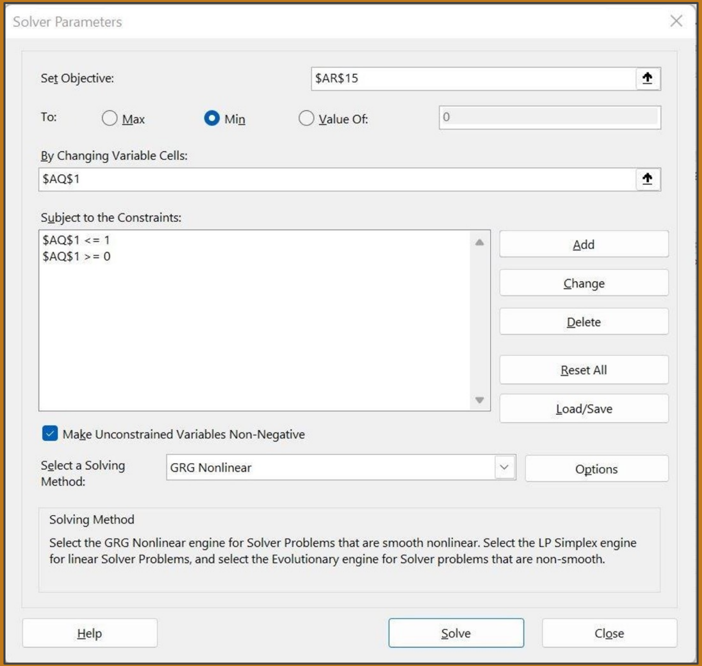
Since only 15% of songs are billboard hits in the dataset, the data was skewed and better predicted if songs were flops instead of hits. Therefore, we needed to find a better threshold probability that was lower than 0.5 to predict if a song will be a hit.

Second Approach - ROC Curve

To find the point in the ROC curve that is closest to the perfect classifier (0,1), we calculated the specificity (x-axis) and sensitivity (y-axis) using the confusion matrix values. These numbers then allowed us to calculate the distance from (0,1).

Using Solver, we found the threshold probability that **minimized the distance from the perfect classifier** (subject to the constraint that the probability is less than or equal to 1 and greater than or equal to 0).

Using Solver - ROC Curve



	CONFUSION MATRIX	
	Predicted	
Actual	0	1
0	3481	971
1	259	532
	Sensitivity	0.67256637
	Specificity	0.78189578
	x axis on the ROC curve	0.21810422
	y axis on the ROC curve	0.67256637
	Distance from (0,1)	0.39342373

Results - ROC Curve

The new **threshold probability is 0.178 ~ 0.18**, meaning that the model will classify the song as hit if it has a probability that is at or greater than this number. With a lower threshold, the model can learn more about the true positive hit songs that were previously classified as true negatives.

We tend to predict hits with about **67.26% accuracy and predict flops with about 78.19% accuracy.** While the flop prediction rate was lower than the first approach, our true aim was to build a model that would more accurately classify potential Billboard hits to inform decisions about marketing resource allocations and maximize profits off of these top songs.

Hit Prediction Rate	67.26%
Flop Prediction Rate	78.19%

Third Approach - Oversampled Data

Since we know that the dataset is not properly balanced with the proportion of hits and flops, another approach is to manually influence it.

- Build a simulated dataset where 50% are hits and 50% are flops
  - Take 250 flops
  - Take 250 hits
  - Concatenate to become one dataset

Third Approach - Oversampled Data

Using only statistically significant variables

- End up with 9 total explanatory variables after stripping insignificant variables

Run a regression on the newly created dataset

Applying a formula to mathematically correct the model

Third Approach - Oversampled Data

```r
# Sample 250 hits
songs_biased_data <- songs_data[sample(which(songs_data$Top10 == 1 ) , 250 ) , ]

# Sample 250 flops
songs_biased_data2 <- songs_data[sample(which(songs_data$Top10 == 0 ) , 250 ), ]

# Concatenate the 2 dataframes
songs_biased_data <- rbind(songs_biased_data,songs_biased_data2)
```

```r
model_glm <- glm(Top10 ~ timesignature_confidence + loudness + tempo_confidence +
  energy + pitch + timbre_0_min + timbre_0_max + timbre_1_min + timbre_3_max +
    timbre_4_min + timbre_4_max + timbre_5_min + timbre_6_min + timbre_6_max +
      timbre_10_max + timbre_11_max + timbre_11_min, data = songs_train2, family = binomial(logit))
summary_glm <- summary(model_glm)
```

$$Offset = \ln\left[\frac{\left(1 - purchase\ rate_{population}\right)/\left(purchase\ rate_{population}\right)}{\left(1 - purchase\ rate_{sample}\right)/\left(purchase\ rate_{sample}\right)}\right]$$

$$Bias\ Corrected\ Intercept = Estimated\ Intercept - Offset$$

Results - Oversampled Data

	Predicted	
Actual	0	1
0	245	5
1	191	59
Hit prediction rate	23.60%	
Flop prediction rate	98.00%	

Model does not offer significant improvements over basecase.

Summary

For each approach, we **performed the analysis again with the test data** and everything measured up to the results of the training data analysis.

The **ROC approach** and **modified Cost-Based** approach yielded the most effective results of increasing largely the probability of hits prediction even if they sacrifice some prediction of the flops, which is a tradeoff we are willing to take!

Application of Models and Marketing Effects

New Product Development

- Labels will know key metrics of top charting music
- Will know whether it make sense to acquire/develop artist in different genres

Predict Popularity/Sales

- Labels will be able to predict popularity of songs and sales
- Will know what songs to market

Artist Investments

- Labels will be able to properly evaluate artist sales potential

Appendix & Our Work

Dataset: https://www.kaggle.com/datasets/econdata/popularity-of-music-records

Code in R: https://colab.research.google.com/drive/1slRldGp1me0Apcsgl6g4l_nKdwPcjFlc?usp=sharing

Model(s) Analysis - Training Dataset of Cost-Based and ROC Approaches:
https://docs.google.com/spreadsheets/d/1M4WowOgh-LkMs8XzlKBYPmYYh-09a_yl/edit?usp=sharing&ouid=115746272252023575668&rtpof=true&sd=true

Biased (Oversampled) Data Analysis:
https://docs.google.com/spreadsheets/d/1hPn8jrrJxWqFjoG2PmheHMpjv4EFXDak/edit?usp=sharing&ouid=115746272252023575668&rtpof=true&sd=true

Testing data Results:
https://docs.google.com/spreadsheets/d/1FM6gB7bvbbvOF68NfKGY-gflPwGdYhTJ/edit?usp=sharing&ouid=115746272252023575668&rtpof=true&sd=true

Our Folder:
https://docs.google.com/spreadsheets/d/1FM6gB7bvbbvOF68NfKGY-gflPwGdYhTJ/edit?usp=sharing&ouid=115746272252023575668&rtpof=true&sd=true

Thank You!

Questions?