MITx: 15.071x The Analytics Edge

POPULARITY OF MUSIC RECORDS

The music industry has a well-developed market with a global annual revenue around $15 billion. The recording industry is highly competitive and is dominated by three big production companies which make up nearly 82% of the total annual album sales.

Artists are at the core of the music industry and record labels provide them with the necessary resources to sell their music on a large scale. A record label incurs numerous costs (studio recording, marketing, distribution, and touring) in exchange for a percentage of the profits from album sales, singles and concert tickets.

Unfortunately, the success of an artist's release is highly uncertain: a single may be extremely popular, resulting in widespread radio play and digital downloads, while another single may turn out quite unpopular, and therefore unprofitable.

Knowing the competitive nature of the recording industry, record labels face the fundamental decision problem of which musical releases to support to maximize their financial success.

How can we use analytics to predict the popularity of a song? In this assignment, we challenge ourselves to predict whether a song will reach a spot in the Top 10 of the Billboard Hot 100 Chart.

Taking an analytics approach, we aim to use information about a song's properties to predict its popularity. The dataset songs.csv consists of all songs which made it to the Top 10 of the Billboard Hot 100 Chart from 1990-2010 plus a sample of additional songs that didn't make the Top 10. This data comes from three sources: Wikipedia, Billboard.com, and EchoNest.

The variables included in the dataset either describe the artist or the song, or they are associated with the following song attributes: time signature, loudness, key, pitch, tempo, and timbre.

Here's a detailed description of the variables:

- **year** = the year the song was released
- **songtitle** = the title of the song
- **artistname** = the name of the artist of the song
- **songID** and **artistID** = identifying variables for the song and artist
- **timesignature** and **timesignature_confidence** = a variable estimating the time signature of the song, and the confidence in the estimate
- **loudness** = a continuous variable indicating the average amplitude of the audio in decibels
- **tempo** and **tempo_confidence** = a variable indicating the estimated beats per minute of the song, and the confidence in the estimate
- **key** and **key_confidence** = a variable with twelve levels indicating the estimated key of the song (C, C#, . . ., B), and the confidence in the estimate
- **energy** = a variable that represents the overall acoustic energy of the song, using a mix of features such as loudness
- **pitch** = a continuous variable that indicates the pitch of the song
- **timbre_0_min**, **timbre_0_max**, **timbre_1_min**, **timbre_1_max**, . . . , **timbre_11_min**, and **timbre_11_max** = variables that indicate the minimum/maximum values over all segments for each of the twelve values in the timbre vector (resulting in 24 continuous variables)
- **Top10** = a binary variable indicating whether or not the song made it to the Top 10 of the Billboard Hot 100 Chart (1 if it was in the top 10, and 0 if it was not)

PROBLEM 1.1 - UNDERSTANDING THE DATA (1/1 point)

Use the read.csv function to load the dataset "songs.csv" into R.

How many observations (songs) are from the year 2010?

373

Answer: 373

EXPLANATION

First, navigate to the directory on your computer containing the file "songs.csv". You can load the dataset by using the command:

```
songs = read.csv("songs.csv")
```

Then, you can count the number of songs from 2010 by using the table function:

```
table(songs$year)
```

You have used 2 of 3 submissions

PROBLEM 1.2 - UNDERSTANDING THE DATA (1/1 point)

How many songs does the dataset include for which the artist name is "Michael Jackson"?

18

Answer: 18

EXPLANATION

If you look at the structure of the dataset by typing str(songs), you can see that there are 1032 different values of the variable "artistname". So if we create a table of artistname, it will be challenging to find Michael Jackson. Instead, we can use subset:

```
MichaelJackson = subset(songs, artistname == "Michael Jackson")
```

Then, by typing str(MichaelJackson) or nrow(MichaelJackson), we can see that there are 18 observations.

You have used 1 of 3 submissions

PROBLEM 1.3 - UNDERSTANDING THE DATA (1/1 point)

Which of these songs by Michael Jackson made it to the Top 10? Select all that apply.

- ☐ Beat It
- ☑ You Rock My World ✓
- ☐ Billie Jean
- ☑ You Are Not Alone ✓

EXPLANATION

We can answer this question by using our subset MichaelJackson from the previous question. If you output the vector MichaelJackson$songtitle, you can see the row number of each of the songs. Then, you can see whether or not that song made it to the top 10 by outputing the value of Top10 for that row. For example, "Beat It" is the 13th song in our subset. So then if we type:

MichaelJackson$Top10[13]

we get 0, which means that this song did not make it to the Top 10. The song "You Rock My World" is first on the list, so if we type:

MichaelJackson$Top10[1]

we get 1, which means that this song did make it to the Top 10.

As a shortcut, you could just output:

MichaelJackson[c("songtitle", "Top10")]

You have used 1 of 2 submissions

PROBLEM 1.4 - UNDERSTANDING THE DATA (2/2 points)

The variable corresponding to the estimated time signature (timesignature) is discrete, meaning that it only takes integer values (0, 1, 2, 3, . . .). What are the values of this variable that occur in our dataset? Select all that apply.

- [✓ 0] ✓
- [✓ 1] ✓
- 2
- [✓ 3] ✓
- [✓ 4] ✓
- [✓ 5] ✓
- 6
- [✓ 7] ✓
- 8

Which timesignature value is the most frequent among songs in our dataset?

- 0
- 1
- 2
- 3
- [✓ 4] ✓
- 5
- 6
- 7
- 8

EXPLANATION

You can answer these questions by using the table command:

table(songs$timesignature)

The only values that appear in the table for timesignature are 0, 1, 3, 4, 5, and 7. We can also read from the table that 6787 songs have a value of 4 for the timesignature, which is the highest count out of all of the possible timesignature values.

You have used 1 of 2 submissions

PROBLEM 1.5 - UNDERSTANDING THE DATA (1/1 point)

Out of all of the songs in our dataset, the song with the highest tempo is one of the following songs. Which one is it?

- Until The Day I Die
- [✓ Wanna Be Startin' Somethin'] ✓
- My Happy Ending
- You Make Me Wanna...

EXPLANATION

You can answer this question by using the which.max function. The output of which.max(songs$tempo) is 6206, meaning that the song with the highest tempo is the row 6206. We can output the song title by typing:

songs$songtitle[6206]

The song title is: Wanna be Startin' Somethin'.

You have used 1 of 2 submissions

PROBLEM 2.1 - CREATING OUR PREDICTION MODEL (1/1 point)

We wish to predict whether or not a song will make it to the Top 10. To do this, first use the subset function to split the data into a training set "SongsTrain" consisting of all the observations up to and including 2009 song releases, and a testing set "SongsTest", consisting of the 2010 song releases.

How many observations (songs) are in the training set?

7201

Answer: 7201

EXPLANATION

You can split the data into the training set and the test set by using the following commands:

SongsTrain = subset(songs, year <= 2009)

SongsTest = subset(songs, year == 2010)

The training set has 7201 observations, which can be found by looking at the structure with str(SongsTrain) or by typing nrow(SongsTrain).

You have used 1 of 3 submissions

PROBLEM 2.2 - CREATING OUR PREDICTION MODEL (2/2 points)

In this problem, our outcome variable is "Top10" - we are trying to predict whether or not a song will make it to the Top 10 of the Billboard Hot 100 Chart. Since the outcome variable is binary, we will build a logistic regression model. We'll start by using all song attributes as our independent variables, which we'll call Model 1.

We will only use the variables in our dataset that describe the numerical attributes of the song in our logistic regression model. So we won't use the variables "year", "songtitle", "artistname", "songID" or "artistID".

We have seen in the lecture that, to build the logistic regression model, we would normally explicitly input the formula including all the independent variables in R. However, in this case, this is a tedious amount of work since we have a large number of independent variables.

There is a nice trick to avoid doing so. Let's suppose that, except for the outcome variable Top10, all other variables in the training set are inputs to Model 1. Then, we can use the formula

SongsLog1 = glm(Top10 ~ ., data=SongsTrain, family=binomial)

to build our model. Notice that the "." is used in place of enumerating all the independent variables. (Also, keep in mind that you can choose to put quotes around binomial, or leave out the quotes. R can understand this argument either way.)

However, in our case, we want to exclude some of the variables in our dataset from being used as independent variables ("year", "songtitle", "artistname", "songID", and "artistID"). To do this, we can use the following trick. First define a vector of variable names called nonvars - these are the variables that we won't use in our model.

nonvars = c("year", "songtitle", "artistname", "songID", "artistID")

To remove these variables from your training and testing sets, type the following commands in your R console:

SongsTrain = SongsTrain[, !(names(SongsTrain) %in% nonvars)]

SongsTest = SongsTest[, !(names(SongsTest) %in% nonvars)]

Now, use the glm function to build a logistic regression model to predict Top10 using all of the other variables as the independent variables. You should use SongsTrain to build the model.

Looking at the summary of your model, what is the value of the Akaike Information Criterion (AIC)?

4827.2

Answer: 4827.2

EXPLANATION

To answer this question, you first need to run the three given commands to remove the variables that we won't use in the model from the datasets:

nonvars = c("year", "songtitle", "artistname", "songID", "artistID")

SongsTrain = SongsTrain[, !(names(SongsTrain) %in% nonvars)]

```
SongsTest = SongsTest[ , !(names(SongsTest) %in% nonvars) ]
```

Then, you can create the logistic regression model with the following command:

```
SongsLog1 = glm(Top10 ~ ., data=SongsTrain, family=binomial)
```

Looking at the bottom of the summary(SongsLog1) output, we can see that the AIC value is 4827.2.

You have used 1 of 5 submissions

PROBLEM 2.3 - CREATING OUR PREDICTION MODEL (1/1 point)

Let's now think about the variables in our dataset related to the confidence of the time signature, key and tempo (timesignature_confidence, key_confidence, and tempo_confidence). Our model seems to indicate that these confidence variables are significant (rather than the variables timesignature, key and tempo themselves). What does the model suggest?

- The lower our confidence about time signature, key and tempo, the more likely the song is to be in the Top 10
- [✓ The higher our confidence about time signature, key and tempo, the more likely the song is to be in the Top 10] ✓

EXPLANATION

If you look at the output summary(model), where model is the name of your logistic regression model, you can see that the coefficient estimates for the confidence variables (timesignature_confidence, key_confidence, and tempo_confidence) are positive. This means that higher confidence leads to a higher predicted probability of a Top 10 hit.

You have used 1 of 1 submissions

PROBLEM 2.4 - CREATING OUR PREDICTION MODEL (1/1 point)

In general, if the confidence is low for the time signature, tempo, and key, then the song is more likely to be complex. What does Model 1 suggest in terms of complexity?

- Mainstream listeners tend to prefer more complex songs
- [✓ Mainstream listeners tend to prefer less complex songs] ✓

EXPLANATION

Since the coefficient values for timesignature_confidence, tempo_confidence, and key_confidence are all positive, lower confidence leads to a lower predicted probability of a song being a hit. So mainstream listeners tend to prefer less complex songs.

You have used 1 of 1 submissions

PROBLEM 2.5 - CREATING OUR PREDICTION MODEL (2/2 points)

Songs with heavier instrumentation tend to be louder (have higher values in the variable "loudness") and more energetic (have higher values in the variable "energy").

By inspecting the coefficient of the variable "loudness", what does Model 1 suggest?

- [✓ Mainstream listeners prefer songs with heavy instrumentation] ✓
- Mainstream listeners prefer songs with light instrumentation

By inspecting the coefficient of the variable "energy", do we draw the same conclusions as above?

[No] No

EXPLANATION

The coefficient estimate for loudness is positive, meaning that mainstream listeners prefer louder songs, which are those with heavier instrumentation. However, the coefficient estimate for energy is negative, meaning that mainstream listeners prefer songs that are less energetic, which are those with light instrumentation. These coefficients lead us to different conclusions!

You have used 1 of 1 submissions

PROBLEM 3.1 - BEWARE OF MULTICOLLINEARITY ISSUES! (1/1 point)

What is the correlation between the variables "loudness" and "energy" in the training set?

0.7399067 **Answer:** 0.73991

EXPLANATION

The correlation can be computed with the following command:

```
cor(SongsTrain$loudness, SongsTrain$energy)
```

Given that these two variables are highly correlated, Model 1 suffers from multicollinearity. To avoid this issue, we will omit one of these two variables and rerun the logistic regression. In the rest of this problem, we'll build two variations of our original model: Model 2, in which we keep "energy" and omit "loudness", and Model 3, in which we keep "loudness" and omit "energy".

You have used 1 of 3 submissions

PROBLEM 3.2 - BEWARE OF MULTICOLLINEARITY ISSUES! (1/1 point)

Create Model 2, which is Model 1 without the independent variable "loudness". This can be done with the following command:

```
SongsLog2 = glm(Top10 ~ . - loudness, data=SongsTrain, family=binomial)
```

We just subtracted the variable loudness. We couldn't do this with the variables "songtitle" and "artistname", because they are not numeric variables, and we might get different values in the test set that the training set has never seen. But this approach (subtracting the variable from the model formula) will always work when you want to remove numeric variables.

Look at the summary of SongsLog2, and inspect the coefficient of the variable "energy". What do you observe?

- [✓ Model 2 suggests that songs with high energy levels tend to be more popular. This contradicts our observation in Model 1.] ✓
- Model 2 suggests that, similarly to Model 1, songs with low energy levels tend to be more popular.

EXPLANATION

The coefficient estimate for energy is positive in Model 2, suggesting that songs with higher energy levels tend to be more popular. However, note that the variable energy is not significant in this model.

You have used 1 of 1 submissions

PROBLEM 3.3 - BEWARE OF MULTICOLLINEARITY ISSUES! (1/1 point)

Now, create Model 3, which should be exactly like Model 1, but without the variable "energy".

Look at the summary of Model 3 and inspect the coefficient of the variable "loudness". Remembering that higher loudness and energy both occur in songs with heavier instrumentation, do we make the same observation about the popularity of heavy instrumentation as we did with Model 2?

Yes Yes

EXPLANATION

Model 3 can be created with the following command:

SongsLog3 = glm(Top10 ~ . - energy, data=SongsTrain, family=binomial)

Looking at the output of summary(SongsLog3), we can see that loudness has a positive coefficient estimate, meaning that our model predicts that songs with heavier instrumentation tend to be more popular. This is the same conclusion we got from Model 2.

In the remainder of this problem, we'll just use Model 3.

You have used 1 of 1 submissions

PROBLEM 4.1 - VALIDATING OUR MODEL (2/2 points)

Make predictions on the test set using Model 3. What is the accuracy of Model 3 on the test set, using a threshold of 0.45? (Compute the accuracy as a number between 0 and 1.)

0.8793566 **Answer:** 0.87936

EXPLANATION

You can make predictions on the test set by using the command:

testPredict = predict(SongsLog3, newdata=SongsTest, type="response")

Then, you can create a confusion matrix with a threshold of 0.45 by using the command:

table(SongsTest$Top10, testPredict >= 0.45)

The accuracy of the model is (309+19)/(309+5+40+19) = 0.87936

You have used 1 of 5 submissions

PROBLEM 4.2 - VALIDATING OUR MODEL (1/1 point)

Let's check if there's any incremental benefit in using Model 3 instead of a baseline model. Given the difficulty of guessing which song is going to be a hit, an easier model would be to pick the most frequent outcome (a song is not a Top 10 hit) for all songs. What would the accuracy of the baseline model be on the test set? (Give your answer as a number between 0 and 1.)

0.8418231

0.8418231

Answer: 0.8418231

EXPLANATION

You can compute the baseline accuracy by tabling the outcome variable in the test set:

table(SongsTest$Top10)

The baseline model would get 314 observations correct, and 59 wrong, for an accuracy of 314/(314+59) = 0.8418231.

You have used 1 of 3 submissions

PROBLEM 4.3 - VALIDATING OUR MODEL (2/2 points)

It seems that Model 3 gives us a small improvement over the baseline model. Still, does it create an edge?

Let's view the two models from an investment perspective. A production company is interested in investing in songs that are highly likely to make it to the Top 10. The company's objective is to minimize its risk of financial losses attributed to investing in songs that end up unpopular.

A competitive edge can therefore be achieved if we can provide the production company a list of songs that are highly likely to end up in the Top 10. We note that the baseline model does not prove useful, as it simply does not label any song as a hit. Let us see what our model has to offer.

How many songs does Model 3 correctly predict as Top 10 hits in 2010 (remember that all songs in 2010 went into our test set), using a threshold of 0.45?

19

19

Answer: 19

How many non-hit songs does Model 3 predict will be Top 10 hits (again, looking at the test set), using a threshold of 0.45?

5

5

Answer: 5

EXPLANATION

According to our model's confusion matrix:

table(SongsTest$Top10, testPredict >= 0.45)

We have 19 true positives (Top 10 hits that we predict correctly), and 5 false positives (songs that we predict will be Top 10 hits, but end up not being Top 10 hits).

You have used 2 of 3 submissions

PROBLEM 4.4 - VALIDATING OUR MODEL (2/2 points)

What is the sensitivity of Model 3 on the test set, using a threshold of 0.45?

0.3220339 **Answer:** 0.3220339

What is the specificity of Model 3 on the test set, using a threshold of 0.45?

0.9840764 **Answer:** 0.9840764

EXPLANATION

Using the confusion matrix:

table(SongsTest$Top10, testPredict >= 0.45)

We can compute the sensitivity to be 19/(19+40) = 0.3220339, and the specificity to be 309/(309+5) = 0.9840764.

You have used 1 of 3 submissions

PROBLEM 4.5 - VALIDATING OUR MODEL (1/1 point)

What conclusions can you make about our model? (Select all that apply.)

- [✓] Model 3 favors specificity over sensitivity. ✓
- [ ] Model 3 favors sensitivity over specificity.
- [ ] Model 3 captures less than half of Top 10 songs in 2010. Model 3 therefore does not provide a useful list of candidate songs to investors, and hence offers no competitive edge.
- [✓] Model 3 provides conservative predictions, and predicts that a song will make it to the Top 10 very rarely. So while it detects less than half of the Top 10 songs, we can be very confident in the songs that it does predict to be Top 10 hits. ✓

EXPLANATION

Model 3 has a very high specificity, meaning that it favors specificity over sensitivity. While Model 3 only captures less than half of the Top 10 songs, it still can offer a competitive edge, since it is very conservative in its predictions.

You have used 1 of 2 submissions

Please remember not to ask for or post complete answers to homework questions in this discussion forum.

Show Discussion

New Post