Predicting Song Downloads

Ankita

December 6, 2017

I Introduction

Predicting song download is particularly important in keeping businesses competitive within a growing music industry.But what exactly increases a songs downloads? Starting with the Million Song Dataset, a collection of audio features and metadata for approximately one million songs, we evaluated different regression algorithms on their ability to predict downloads and determined the types of features that hold the most predictive power.

II Dataset and Features

A. Data

We used the music data form Million song dataset,generated from The Echo Nest,a music intelligence platform and Downloads information provided to us. For preparing the data we performed the following :

i) *Filter*

- Removed all non alpha-numeric values except ' and space
- Removed all leading and trailing spaces in the columns
- Removed NULL values
- Removed empty strings
- Removed N/A values
- Converted all to lower case

ii) *Join*

The dataframes were joined where the "artist name" was ***equal*** and "song title" ***starts with***. ***Starts with*** was used for song title as the song title in "dowloads" dataframe contained dirty data. The join resulted in 940,912 rows.But because of using ***starts with*** to join, there were around 20,0000 duplicates which were dropped. So,we finally extracted 920,838 rows after cleaning and joining both the datasets.

iii) *Impute values*

In the resulting dataset,the fields artist familiarity,artist hotness and song hotness contained significant invalid values. Inorder to resolve this we imputed the values for artist familiarity and artist hotness with their median values using mlLib imputer. However, we decided not to do the same for song hotness as there were only 10,000 valid values out of 920,000, if we were to impute the values almost all 89% values will be imputed values.

B. Features

The Million Song Dataset contains a plethora of feature types, ranging from number-type features such as those that measure the general loudness of a song,string-type features such as names of artists and albums, and array-type features such as those that contain pitches across the length of the song etc.

The downloads Dataset contained features like string-type features like artist and title, float-type mean.price, integer-type downloads, string-type categorical confidence, number-type year. The following features were chosen and considered for training the model based on their co-relation to downloads:

	Type	Description
mean price	float	price of each download
confidence	string	confidence in download numbers
artist familiarity	double	indicates familiarity of artist
artist hotness	double	hotness of the artist
artist name	string	name of the artist
title	string	song title

Fig.1 - Model features and their types

III Methods

A. Feature Selection

Our final dataset consisted of a number of features, many of which were not relevant to predicting downloads.As a result using all the features didn't seem feasible.We limited our model to 6 relevant features.

The features were selected by calculating their co-relation with downloads and the RMSE values in the predictions when using these features.In the feature selection phase, we considered other available datasets like Taste profile and This is my Jam, as well. However, after careful analysis, features from these datasets were not included to train the model.

B. Random Forest Regression

We decided on Random forests as it trains a set of decision trees separately, so the training can be done in parallel. The algorithm injects randomness into the training process so that each decision tree is a bit different. Combining the predictions from each tree reduces the variance of the predictions, improving the performance on test data.

For prediction the model, RF uses *Averaging* Each tree predicts a real value. The label is predicted to be the average of the tree predictions.

While creating the model, the following parameters to tune the models behavior:

- ***numTrees*** : Number of trees in the forest.

* Increasing the number of trees will decrease the variance in predictions, improving the model's test-time accuracy.
* Training time increases roughly linearly in the number of trees.

- ***maxDepth*** : Maximum depth of each tree in the forest.

* Increasing the depth makes the model more expressive and powerful. However, deep trees take longer to train and are also more prone to overfitting.
* In general, it is acceptable to train deeper trees when using random forests than when using a single decision tree. One tree is more likely to overfit than a random forest (because of the variance reduction from averaging multiple trees in the forest).

- ***featureSubsetStrategy*** : Number of features to use as candidates for splitting at each tree node. The number is specified as a fraction or function of the total number of features. Decreasing this number will speed up training, but can sometimes impact performance if too low.

Given that the input test data contains only two values artist and song title we used two different approaches to train the model and predict values.

1) *Fuzzy Match Prediction*
In this approach we fetch all the features needed for predicting downloads, given input values, *artist name* and *song title*. Hence we do a fuzzy match of artist name and song title with all values in the cleaned dataset using *starts with*. This results in a subset data, that has all the features matching for the given artist name and song title. We then drop duplicate values that occur due to fuzzy match from this subset data. Now this subset data becomes our test data and we use it to predict the download value for the given input value.

2) *Progressive Prediction*
For this approach, since we only have the song title and the artist to predict downloads we took a step-wise progressive prediction approach where,in ecah step we predict one new feature and then use this predicted value along with the given feature to predict the next feature. We follow these till we have predicted all the features of our model and then predict the downloads. The following steps were executed:

	Features	Predicted.Value
Step-1	Artist, Song title	Artist Familiarity
Step-2	Artist, Song title,Artist Familiarity	Artist hotness
Step-3	Artist, Song title,Artist Familiarity,Artist hotness	Price
Step-4	Artist, Song title,Artist Familiarity,Artist hotness,Price	Downloads

Fig.2 - Steps of the progressive prediction model

IV Experimental Results

A. Datasets

Downloads

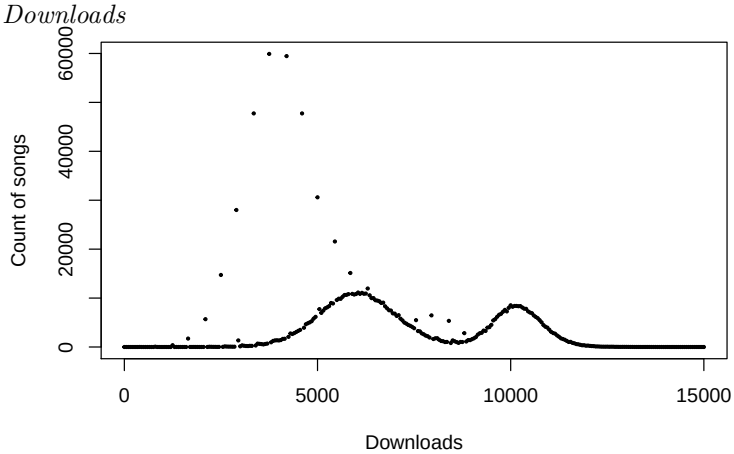

Fig.3 - Number of songs vs downloads
From the above graph we observe a pattern for song downloads with number of songs. However, there is an outlier at ~4000 downloads which cannot be ignored as around 60,000 songs have 4000 downloads.

Million Song

Facts
1,000,000 songs / files
273 GB of data
44,745 unique artists
7,643 unique terms (The Echo Nest tags)
2,321 unique musicbrainz tags
43,943 artists with at least one term
2,201,916 asymmetric similarity relationships
515,576 dated tracks starting from 1922
18,196 cover songs identified

Fig.4 - Million song dataset summarized
This provides enough data with multitude of features to train a model and predict with accuracy. B. Feature Selection
For feature slection we calculated the co-relation of different features with downloads:

Feature	Corelation
Price	0.9496
Duration	-0.0051
Tempo	0.0177
Loudness	0.0487
Year	0.09417
Artist term frequency	-0.0013
Count Jam Id	-0.0347
Count user Id	0.0637
Playcount	0.0531
Confidence	0.5229
Artist Hotness	0.254
Artist Familiarity	0.184

Fig.4 - Corelation of features with downloads

The second parameter was the RMSE values:

	Features	RMSE
CO-1	Price	697.763
CO-2	price, confidenceId	489.70
CO-3	price, confidenceId, artistFamiliarity	339.27
CO-4	price, ConfidenceId, artistFamiliarity, artistHotness	105.23
CO-5	price,ConfidenceId, artistFamiliarity, artistHotness, releaseVector	108.96
CO-6	price,ConfidenceId, artistFamiliarity, artistHotness, artistName	105.23
CO-7	price,ConfidenceId,artistFamiliarity, artistHotness, artistName,title	98.65

Fig.5 - RMSE with different combination of features

Based on the RMSE values for different features and combination of features, along with the co-relation values we decided on our model features.

C. Regression
1) Metrics: In the regression setting, we use root mean squared error (RMSE) to evaluate how well our model predicts downloads. The RMSE,approximates the standard error of our prediction. Therefore, the smaller the RMSE, the higher confidence we have about our predictions.

Depth	Number.of.trees	RMSE
5	70	632.3507
10	20	389.9157
10	25	377.3613
10	32	334.5522
10	100	381.9370
12	28	344.8572
12	30	345.2609
15	30	312.4947
15	120	310.4178

Fig-6: Variation in RMSE with change in depth and number of trees in the Random Forest

2) Results: The Random Forest regression model, with depth - 17 and 30 trees with selected features - *Price, Confidence,Artist Familiarity, Artist Hotness, Artist Name, Title* gave a RMSE value of 98.65 for *Fuzzy Match Prediction*. However, predictions for *Progressive Prediction* remain in the 7,000-8,000 range irrespective of the given input data. We believe this is the cumulative effect of errors introduced in each step of the model.

Input	Actual.Download	Prediction
William Shatner; I Can't Get Behind That	10,571	10,457
Madonna;Like a virgin	8,474	7,701
Iron Maiden ;Aces High	7,573	7,364
Tenacious D; Tribute	8,248	8,015

Fig-7: Prediction of the model- Fuzzy Match Prediction

3) Observation: Based on the results obtained as part of the experiments, we see that the features like *Price,Confidence,Artist Familiarity,Artist Hotness* have a high co-relation with downloads and when used as a combination of features we get the lowest RMSE for the model.

V System Specification

- Macintosh 2.5Ghz i7 Quad Core
- 16 GB RAM
- macOS Sierra Version 10.12.6
- Java version : 1.8
- Scala version : 2.11.11
- Spark version : 2.2.0
- MLlib

VI Conclusion

Through various experiments we realised that the price of a song is the most important predictor for the downloads of that particular song and that the data cleaning and feature extraction steps play the most crucial role in creating a model. The model shows a large variations in RMSE when the features are changed.

VII References

[1]https://spark.apache.org/docs/2.2.0/ml-statistics.html
[2]https://spark.apache.org/docs/2.2.0/mllib-ensembles.html
[3]https://docs.cloud.databricks.com/docs/spark/1.6/examples/ML%20Pipeline%20Persistence.html
[4]https://labrosa.ee.columbia.edu/millionsong/pages/getting-dataset