Evaluation of a song's performance based on its features

I. Summary

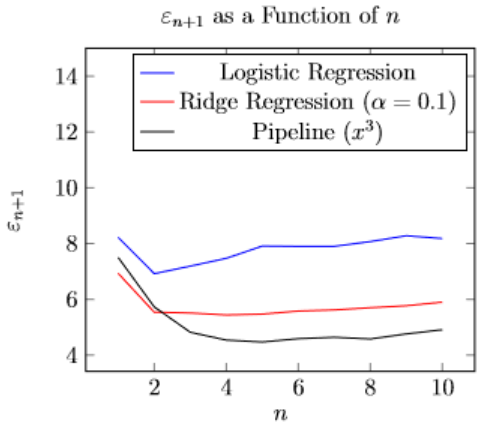
Billions of USD are invested in new artists and songs by the music industry every year. This project provides a strategy for assessing the hit potential of songs, which can help record companies support their investment decisions. The million song subset which contain the song features and the billboard top 100 dataset is used in our project. Data is pre-processed using correlation plot,heat map,correlation matrix and the most significant attributes are chosen for predicting if the song is a hit or not hit.Various models such as Decision trees, Support Vector machines were modeled with an accuracy of 90 percent ,multiple linear regression was modeled with an accuracy of 47 percent , ridge regression is used when there's multicollinearity between attributes and an accuracy of 49 percent is obtained .It was observed that SVM's provide the best accuracy and is more efficient as the non-linear inputs are mapped to a high dimensional feature spaces.

II. Introduction to the context of the problem

The theory of course is that popular songs share a certain set of features that make them appealing to the majority of people, and that you can test any new song against those success markers to predict its commercial potential.Billions of USD are invested in new artists and songs by the music industry every year. This project provides a strategy for assessing the hit potential of songs, which can help record companies support their investment decisions.

III. Previous work done

[1]In this paper , Exploring the space of predicting the (n + 1)th position in the Billboard Hot 100 charts given the rst n positions and musical data. We can then use this prediction as the next entry to predict the (n + 2)th entry, and keep repeating this regression until the song drops out of the charts .An array of dierent algorithms such as Ridge regression , logistic regression, perceptron , pipeline were used to see which produced the best results. it is also noted that for a year in which there are more songs, the songs tend to take on shorter paths. It is hypothesized that these shorter paths are more direct in their representation of the general trends. When features were added from Gracenote, namely Mood and WeeksInChart, it only cluttered the same classiers that would produce better results without it. The best performing algorithm in this case was pipeline with n = 4. The graph including these features are.



[2]We explore the automatic analysis of music to identify likely hit songs. We extract both acoustic and lyric information from each song and separate hits from non-hits using standard classifiers, specifically Support Vector Machines and boosting classifiers.Their results suggest that there is indeed some distinguishable thread connecting hit songs. Results indicate that for the features used, lyric-based features are slightly more effective than audio- based features at distinguishing hits. Combining features does not significantly improve performance. Figures 1 and 2 show the ROC area averaged over the 10 cross validation cuts of the experimental database for SVM and boosting classifiers trained on the acoustic-based and lyrics-based features. They also try combining acoustic and lyric-based features. They achieved this by concatenating the vectors for the two representations. Figure 3 shows results for this experiment.

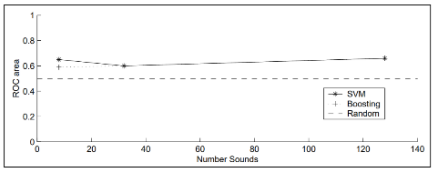


Figure 1: Average ROC area for acoustic-based features with various numbers of sounds for SVM and boosting classifiers

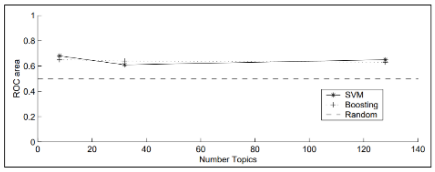


Figure 2: Average ROC area for lyric-based features with various numbers of topics for SVM and boosting classifiers

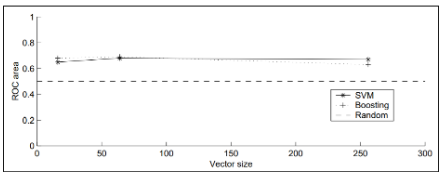
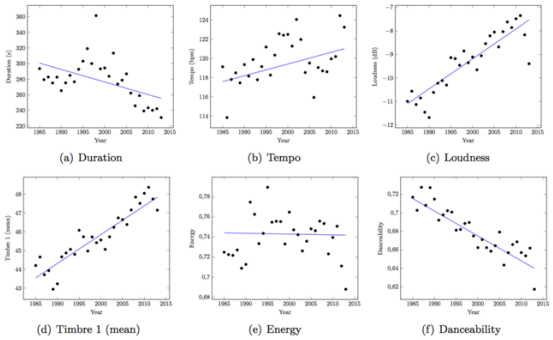


Figure 3: Average ROC area for combined acoustic and lyrics features with varying vector sizes for SVM and boosting classifiers

They performed analysis for the 8-topic lyric vectors, using weights of the weak learners so as to identify which dimensions of the feature vectors are most helpful for classification. From the boosting models analysis, the most important features for distinguishing hits were Topic 1 and Topic 6. These appear to describe "heavy metal" and "peaceful/new age" music. All of the weak learners learnt negative boundaries. That is, the absence of Topic 1 or Topic 6 meant the song was more likely to be a hit.

[3] To quantify the building blocks of a hit, you first have to break down a song into its various audio characteristics. They pulled this data from The Echo Nest, the music intelligence software that analyses and categorizes audio data to power the search and recommendation features on services like Spotify.Echonest has been acquired by Spotify, this means that the API has been thus updated. They looked at 139 different musical aspects to analyse songs. This includes basic features like length, tempo (measured in beats per minute or bpm), time signature, key, and loudness; more subjective features like beat, energy, and danceability; and even more intangible qualities like a song's timbre or tone colour, or what we might think of as its general feel.Popular songs share a certain set of features that make them appealing to the majority of people, and that you can test any new song against those success markers to predict its commercial potential.They developed an online app that allows you to upload a dance song and calculate the probability that this song will be a dance hit.The algorithm predicted a 65 per cent or higher probability of a hit for all of the top 10, and over 70 per cent probability for 6 out of 10 songs. A graphical representation of the features is shown below:



(a) Duration (b) Tempo (c) Loudness

(d) Timbre 1 (mean) (e) Energy (f) Danceability

IV. Data Resources

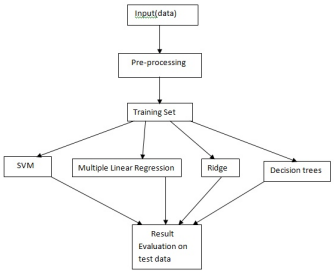
Our project evaluates the songs features and predicts if the song is a hit or not. Data was collected from the one million data set and the billboard top 100 songs. Hierarchical Data Format (HDF) is a set of file formats (HDF4, HDF5) designed to store and organize large amounts of data. The data are stored in the form of an hierarchy. This is useful for large amounts of numeric data. The Million Song Dataset is a freely-available collection of audio features and metadata for a million contemporary popular music tracks.The core of the dataset is the feature analysis and metadata for one million songs, provided by The Echo Nest. The dataset does not include any audio, only the derived features.

The original code makes use of example code provided at the Million Song Dataset website. It has been modified to extract appropriate features using the hdf5getters python file that is included in The Million Song Database website.

The class song must be defined with constructor and must have objects to point to each of the attribute that need to be extracted. An attribute string is required to be made, in the order of obtaining attributes, to use as header for the corresponding csv file to be generated. Each attribute must be obtained from the return value of the corresponding function from the hdf5getters file provided. Each attribute value for every row of the hierarchal file must be concatenated into a single string and this string is then appended to the csv file. This process happens iteratively till the end of the process.

The dataset containing billboard top 100 songs with their corresponding ranks is used to match songs in the million song dataset .A new CSV file is created which contains the song features from the million song dataset and their corresponding rank and year of release.

V. Block diagram of the project

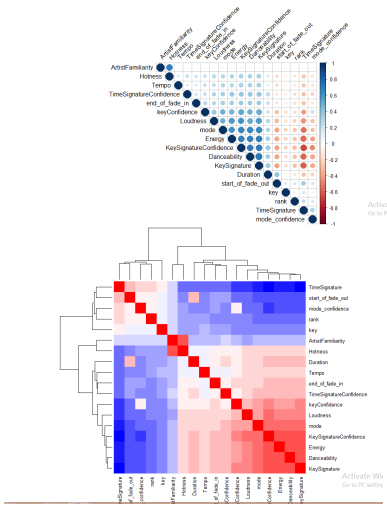


VI. Experiments Done

1-Data preprocessing

Correlated attributes are identified and removed because they are similar in behavior and will have similar impact in prediction calculations, so keeping attributes with similar impacts is redundant. Using the correlation matrix, heat map and correlation plot , the attributes with high correlation such as Key signature and key, key confidence and mode confidence , start of fade out and duration .

A heat map and correlation plot is used for better visualization and understanding the correlation between attributes. The heatmap and corrplot function is used to achieve this in R. The attributes such as energy and danceability were removed because it contained only 0 values. Numeric attributes such as duration , key confidence , key signature and start of fade out are dropped. The correlation plot and heat maps are shown below



2-Support Vector machines

A supervised learning is being conducted on our dataset using a Support Vector Machine. The support vector machine is a binary classifier, which classifies our data as either a hit song or not a hit song represented as 1 and 0 respectively in the dataset. The Support Vector Machine maps all the points, or songs, into an n dimensional space. Each dimension of our n dimensional space represents our attributes. SVM training algorithm builds a model that classifies test data to one category or the other, making it a non-probabilistic binary linear classifier. However here we have many attributes and thus we cannot use a linear classifier. SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. The model attempts to find natural clustering and then map new data to these formed groups. The clustering algorithm which provides an improvement to the support vector machines is called support vector clustering.

We have not manually written the code for the support vector clustering algorithm. The e1071 package in R is used to create Support Vector Machines. It has several helper functions as well as code for the Naive Bayes Classifier. To improve the performance of the model we utilise the various parameters provided. The explicitly expressed parameters are kernel, type and cost. Here, we have various options available with kernel like, linear, radial,poly and others (default value is rbf). We have used radial which is useful for non-linear hyper-plane. The cost is penalty parameter C of the error term. It controls the trade-off between smooth decision boundary and classifying the training points correctly.It avoids overfitting. The type classification is given as C or Classification SVM Type 1. This is in accordance to the error function. For this type of SVM, training involves the minimization of the error function.

Before the code is passed to the svm() function we must first clean the data appropriately. The NA values must be replaced appropriately. The dataset is divided in a test and train set to conduct supervised learning accordingly. The hit column has been modified to contain either 0 or 1 according to the rank, as in if the rank is less than 50 then we categorise it as a hit or 1 else 0.

After the train dataset is passed in to svm() and a model is obtained we use the test dataset to predict values. The corresponding confusion matrix and results are observed. The tuning function can be used to appropriately decide the parameter values for greatest accuracy. It will check for all possible cases of parameter values.

We can also check the accuracy of our SVM with in a time interval. This has been done because intuitively we can tell due to cultural changes a song that might have been a hit in the 40s will not necessarily be a hit today. However it can be noted that this has been taken into account by the SVM which considers it as another dimension and can thus distinguish songs of different eras. The accuracy of the new model of an era is same as that of the SVM of the whole datset. Below attached confusion matrix and corresponding results of the SVM of the whole dataset and that of a particular era.

Results obtained are represented in the form of a confusion matrix for the dataset

```
Confusion Matrix and Statistics


pred   0   1
   0 754   1
   1   8 862

               Accuracy : 0.9945
                 95% CI : (0.9895, 0.9975)
    No Information Rate : 0.5311
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9889
 Mcnemar's Test P-Value : 0.0455

            Sensitivity : 0.9895
            Specificity : 0.9988
         Pos Pred Value : 0.9987
         Neg Pred Value : 0.9908
             Prevalence : 0.4689
         Detection Rate : 0.4640
   Detection Prevalence : 0.4646
      Balanced Accuracy : 0.9942

       'Positive' Class : 0
```

Results obtained are represented in the form of a confusion matrix for that entire era

```
Confusion Matrix and Statistics


pred5   0   1
    0 733   0
    1  29 863

               Accuracy : 0.9822
                 95% CI : (0.9745, 0.988)
    No Information Rate : 0.5311
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9641
 Mcnemar's Test P-Value : 1.999e-07

            Sensitivity : 0.9619
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9675
             Prevalence : 0.4689
         Detection Rate : 0.4511
   Detection Prevalence : 0.4511
      Balanced Accuracy : 0.9810

       'Positive' Class : 0
```

3-Multiple Linear Regression

Multiple linear regression is the most common form of linear regression analysis. As a predictive analysis, the multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables. The independent variables can be continuous . A linear relationship is assumed between the dependent variable and the independent variables. The residuals are homoscedastic and approximately rectangular-shaped. Absence of multicollinearity is assumed in the model, meaning that the independent variables are not too highly correlated.

Songs which lie in the top 50 of the billboard chart are considered hit (have the values 1) and the rest have the value 0.

In multiple linear regression, the hit column is the dependant variable and the song features such as KeySignatureConfidence , TimeSignature ,TimeSignatureConfidence ,ArtistFamiliarity , Hotness , end-of-fade-in , key, Loudness , mode , mode-confidence are considered as independant variable

Train and test sets are created to train the model and to check the accuracy of the model. The function lm in e1071 is used to build the multiple linear regression model in R and the predict function is used to predict the values on the test data. The confusion matrix is used to find the accuracy, precision, specificity of the mode , using the confusion matrix function in the caret package. The function accuracy is also used to find the accuracy of the model. Below is the confusion matrix and the corresponding results of the multiple linear regression model.

```
Confusion Matrix and Statistics

          0   1
  0 503 559
  1   0   0

                Accuracy : 0.4736
                  95% CI : (0.4432, 0.5042)
     No Information Rate : 0.5264
     P-Value [Acc > NIR] : 0.9997

                   Kappa : 0
  Mcnemar's Test P-Value : <2e-16

             Sensitivity : 1.0000
             Specificity : 0.0000
          Pos Pred Value : 0.4736
          Neg Pred Value :    NaN
              Prevalence : 0.4736
          Detection Rate : 0.4736
    Detection Prevalence : 1.0000
       Balanced Accuracy : 0.5000

        'Positive' Class : 0


> accuracy(test$hit, pred1, threshold = 0.5)
  threshold AUC omission.rate sensitivity specificity prop.correct Kappa
1       0.5 0.5             1           0           1    0.4736347     0
```

4-Ridge Regression

Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. This penalty term is known as lamba. Ridge regression is used as there is high correlation between many attributes.

Library glmnet is used to model ridge regression in R , cv.glmnet function is used to calculate the most suitable value of lambda which is 1, in this case . The values for the test dataset are predicted using the predict function and confusionMatrix is used to find the accuracy of the model in R.

Ridge regression is modelled using attributes such as Key Signature confidence, time signature, time signature confidence, artist familiarity , hotness , end of fade in , key, loudness, mode and mode confidence. And the dependant variable is the hit attribute. Below is the confusion matrix and the corresponding results of the ridge regression model.

```
Confusion Matrix and Statistics


     0   1
  0 511 551
  1   0   0

               Accuracy : 0.4812
                 95% CI : (0.4507, 0.5117)
    No Information Rate : 0.5188
    P-Value [Acc > NIR] : 0.9935

                  Kappa : 0
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.4812
         Neg Pred Value :    NaN
             Prevalence : 0.4812
         Detection Rate : 0.4812
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : 0
```

5-Decision Trees

We want to predict the attributes that influences the rank or hit-flop attribute which is the out outcome variable of our dataset at the most using a decision tree model.

Decision tree is one of the supervised learning algorithms. It works for input and output variables for both categorical and continuous that data points or variables. So basically, we split the population or sample into homogeneous sets (could be two or more) based on the variable that has the most significant splitter or differentiator and the purity of the node increases with respect to the target variable.

It is used because it requires little data transformation. Other techniques often require the data to be normalized, dummy variables need to be created and na or blank values to be removed.

Decision trees use a white box model. If in a model, a given situation is observable, there is an explanation for the condition explained by boolean logic. By contrast, in a black box model, for example in an AAN, it is often more difficult to interpret the results.

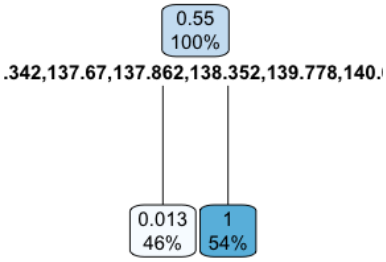
Here in this model that we are building , we take all the attributes that sound to have an influence of the rank of the song, however in the end we come up with the attributes that really do. So we will be looking at the feature hit-flop in our model.

The output of the plot however only shows the root node. We could control this by the attributes that package rpart provides namely minsplit and minbucket. Decision trees generally use a criteria, but in fact there are multiple criteria available to decide to split a node in two or more sub-nodes. By default, rpart uses gini impurity to select the splits when it performs the classification. However we have used information gain to perform the split.

Entropy gives measure of impurity in a node. We need to make two important decisionsŁwhile building the decision tree model Łwhat is/are the best split/splits and which is the best variable to split a node.

Information Gain is one of those criterias that helps in making these decisions. So we use an independent variable value/values, in order to create the child nodes. And in order to calculate the information gain due to the splits, we need to calculate entropy of parent and child nodes . A variable with highest information gain is selected for the split.

So as we run the model, the decision tree correctly identifies tempo to be highly influencing a song to be in any of the two categories, that is 1 - 50 a hit and 51 - 100 a flop (Its intact a hit, but relative to top 50, taken as a flop).



VII. Result Analysis

What went wrong with the linear regression which lea to an accuracy of around 50

All the linear regression methods, have a major drawback that most systems are not linear in reality. Linear models attempt to fit a line through one dimensional, a plane through two dimensional, and a generalization of a plane through higher dimensional data sets.

So it is quite probable that even with an infinite number of training points , linear methods often fail to do a good job at making predictions.

It intuitively seems as though it is easier to make predictions when we have more information about a system . However as with many commonly used algorithms that is not the case. While it never seems to be a problem to have a large amount of training data, having too many features can cause serious difficulties. Least squares regression is one of the models prone to this problem.

Well what exactly is the problem with too many attributes?

So when too many variables are used with the least squares method, the model begins to find ways to fit itself to not only the underlying training set, but also to the noise in the training set, which is why using too many features leads to bad prediction results.

However its important that even if the underlying system that we are attempting to model is in fact linear, and even if the best way of measuring error truly is the sum of squared errors, and even if our training data does not have significant outliers or dependence between independent variables, it still is not necessary that least squares is the optimal model to use. The difficulty is that the level of noise in our data may be dependent on what region of our feature space we are in.

Having said that, which other model do we go for then?

One possible solution to the problem of non-linearities is to apply transformations to the independent variables of the data that maps these variables into a higher dimension space.

A related solution to the non-linearity problem is to directly apply a kernel method like support vector regression or or kernelized ridge regression which is what we are going to do.

Why use these models?

These methods automatically apply linear regression in a non-linearly transformed version of your feature space.

Why not logistic regression when svm and logistic sound quite comparable for the problem discussed above?

The SVM algorithm is much more geometrically motivated. We try to find a particular optimal separating hyperplane, instead of assuming a probabilistic model. So here we define optimality in the context of the support vectors. We don't really have anything that resembles the statistical model we use in logistic regression, however the linear case will give us similar results.

VIII. Contributions

Simran Choudhary- 01FB15ECS293-Decision trees and Result analysis

Michelle Maria Roy- 01FB15ECS171-Data Extraction and SVM's

Asmita Rao-01FB15ECS056-Preprocessing,Multiple linear, Ridge Regression

References

[1] Cibils, Cristian, Zachary Meza, and Greg Ramel. "Predicting a Songs Path through the Billboard Hot 100. , Stanford University , California

[2] Dhanaraj, Ruth, and Beth Logan. "Automatic Prediction of Hit Songs." In ISMIR, pp. 488-491. 2005.

[3] DorienHerremans, David Martens and Kenneth Srensen of the ANT/OR research group and the Applied Data Mining Group from the University of Antwerp