CSC 522 – Spring 2015: Project Report

MUSIC RATING PREDICTION SYSTEM

1. Project idea and description

Music rating prediction system. This system intends to predict how much rating a user would give on a scale of 1-100 for an EMI company music track they are made to hear. This helps the company know how a particular track or an artist would be received upon launch and also helps them to design a music recommendation system by predicting songs for which the user will give a high rating.

This prediction had to be made considering various attributes of users and artists. The data available in the data set about various users and artists was merged into a single data frame. The resulting data frame was input to various regression models and their outputs were analyzed. The model parameters for each of the models were varied and the best suited values were selected.

The resulting models were validated against the validation data set and the error rates were captured. The performance of each of the model was analyzed using these results. Among those models, we chose the model that was best able to identify the interesting data available in the data set (like correlation between the ratings and the words used, importance\significance of various attributes, how attributes contributed to ratings etc.). The chosen model was used to predict the values of the test data set and the error rates of the test data set were captured and the performance of each of the models were evaluated.

2. Material & Methods

2.1 Data

User specific data: User data in the *users.csv* file had information about user specific interests in music (answers to a set of 19 questions whose values were numeric in the range 0-100), demographic information, age, sex. The data set also had information about how long the users listened to the music they own and how long they listened to other music tracks.

Artist specific data: Artist specific information was available in *words.csv* file. This file had information about how different users described each artist. Each row had list of 82 words (0 indicating the word was used to describe the artist and 1 indicating the word was not used to describe the artist). "LIKE_ARTIST" attribute indicated how much the given user liked the artist on a scale of 0-100.

Training data: Training data from *train.csv* had information about how various users rated tracks of various artists. It had 4 attributes: User ID, Artist ID, Track ID and Rating (0-100).

2.2 Data Preprocessing

As the required data for analysis was spread across 3 different data files, the challenge was to merge all the required data into a single file such that all the information about the user and artist was available in a single data frame. This was done with the help of merge operation, by matching data set based on combination of artist ID and user IDs. Missing values were handled by replacing them with the median values for numerical attributes and with the most frequent value (mode) for categorical attributes.

2.3 Methods

2.3.1 Linear model

Linear model was correctly able to identify the correlation between ratings and some of the attributes. However the model failed on many occasions to identify the positive correlation between some of the positive attributes and the ratings.

The model was also tested against the validation data set and was found to have a high RMSE value of 27.6. Hence we concluded the linear model was not suitable for the complex data set.

2.3.2 Linear model split by artist

As the linear model was very simple and failed to identify the complexities involved in the data, we tried to simplify the data by splitting the data by artist ID and built separate models for each of the artist. This significantly reduced the complexity of the data for the linear model.

When this model was trained, surprisingly error rates of the model was reduced compared to linear model. The resulting model gave an RMSE of 24.76.

2.3.3 Gradient Boosting Model

The general idea in GBM is to compute a sequence of very simple trees, where each successive tree is built for predicting residuals of the preceding tree. At each step of boosting (boosting trees algorithm), a simple (best) partitioning of the data is determined, and the deviations of the observed values from the respective means (residuals for each partition) are computed. The next tree will then be fitted to those residuals, to find another partition that will further reduce the residual (error) variance for the data, given the preceding sequence of trees. This method is repeated till an acceptable residual error is reached.

GBM model was built with *shrinkage* = 0.08 and *interaction.depth* = 10. The model was run with different values of parameters and ones that could accommodate the complexity of data were chosen. Lower shrinkage level meant higher the time to train the model and higher the *interaction.depth*

meant higher the depth of each of the trees. GBM model was able to predict the significance of each of the attributes in determining the prediction ratings of songs. Shrinkage value was kept as low as possible to avoid overfitting of the data.

GBM model's prediction for the test data was captured and RMSE value was calculated. It resulted in an RMSE of 25.

2.3.4 Random Forest

Random Forest uses the concept of ensembling multiple decision trees. For each of the decision trees, subset of the attributes are taken and a subset of the data set is used to train the model. Multiple such decision trees are built and are ensembled to get the resulting random forest model. Random forest models are known to be very good suit for complex data models and random forests seldom overfit the data.

Random forest was built with sample size of 50,000 and ntrees=100. Random forest model was also able to output the importance of each of the attributes in determining the prediction ratings. The importance of attributes was used to prune the data set and random forest algorithm was rerun.

The RMSE value that was calculated from the validation results before pruning the attributes was found to be 25.46. After pruning the RMSE value was 26.3 which was comparable with the model built with unpruned data.

2.3.5 Random Forest split by artist

As we saw with the linear model, when the data set was split by artist and a separate model was built for each artist the prediction accuracy improved. We applied the same method in case of Random forest. Random forest was able to specifically learn the model for each artist and we got an improved error rate as compared to random forest.

The error rate for this model was 23.87.

3. Results

What was discovered from the analysis?
In the process of building the regression models, we were able to discover some important characteristics about the data.

3.1 Correlation coefficients

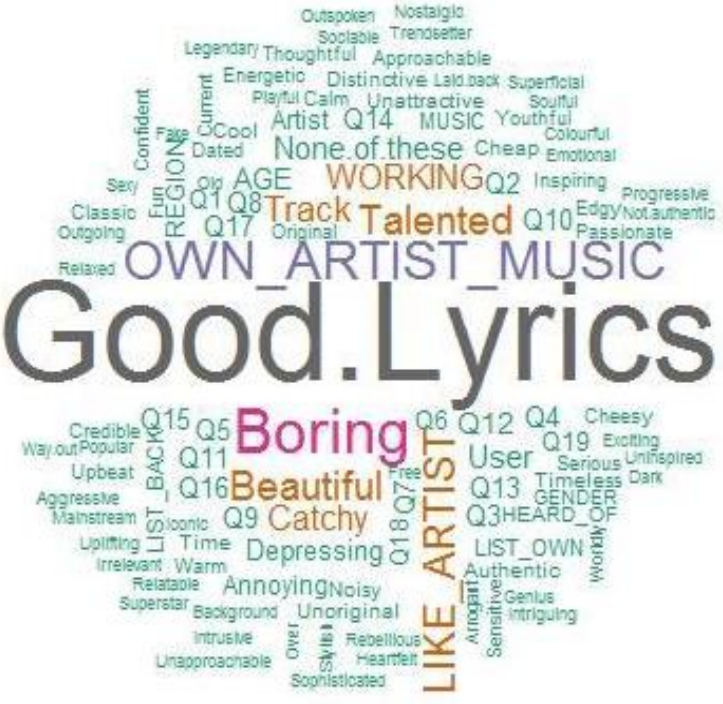
With the help of linear model, we were able to determine the correlation coefficients of various attributes. Below is a table that tells us the list of attributes and their correlation coefficients:

Attribute	Correlation Coefficient
Depressing	-23.982
Timeless	19.460
Good.Lyrics	27.685
Unattractive	-18.669
Wholesome	-4.925

3.2 Attribute importance

With the help of Random Forest and Gradient Boosting models, we were able to identify the significance of various attributes in contributing to the prediction.

The word cloud visualization below shows how important each attribute is:



Below is a table that depicts the relative influence of attributes as an output from Gradient Boosting Model:

Attribute	Relative Influence
Good.Lyrics	16.694848802
Talented	8.984563847
LIKE_ARTIST	8.016529305
Beautiful	7.552150134
Catchy	7.008568715

3.3 Error rates

RMSE values for prediction of various models against the test data:

Model	RMSE value
GBM	25.08098
Linear model	27.6154
Linear model by artist	24.76308
Random forest	25.46859
Random forest by artist	23.87086

4. Discussion

4.1 What did not work?

Demographic Information based prediction: We tried to use the demographic information about the users and come up with some predictions as shown below:

1) Track Ratings by Age of User:
We tried to divide the age group of users into 6 buckets and plot the average ratings provided by these age groups. Based on these average ratings we tried to predict ratings of new tracks but this turned out to be a pretty terrible method.

2) Track Ratings by User Work Information:
Here the data was grouped by User work information and the average ratings were calculated for each group. Based on these average ratings, new track ratings were predicted and this method also failed to predict track ratings accurately.

3) Track ratings by user similarity:
We tried to find similar users with the help of pearson coefficient between every pair of users. Values for all pairs of users above a certain threshold were considered to be similar. To predict the rating of a song by a user we tried to calculate the mean of ratings given by all the similar users to the given user. But this method failed as there were not enough ratings by many users for a given track.

4.2 What worked?

Artist Based Prediction:
We tried to group data based on artist and build separate models for each group. This improved our accuracy in both Random Forest and Linear Models. The RMSE values improved by a significant amount in both cases.

5. References

[1] Dataset: http://www.kaggle.com/c/MusicHackathon/data
[2] Gradient Boosting Model: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3885826/
[3] Random Forest: http://www.math.usu.edu/adele/randomforests/uofu2013.pdf