Hit Song Prediction

EE 660 Course Project

Project Type: (1) Design a system based on real-world data;

Number of student authors: 1

Name: Yingfeng Lou, Email: louy@usc.edu

Date: 11/28/2020

1. Abstract

In this project, we need to deal with the real-world dataset that hit song from 2000 to 2019, consider the acoustic characteristics as the features to predict if a new track would be a hit song.

The whole dataset contains 12272 data points, first I divide them to training set and test set and use 5-fold cross-validation. Then I do some preprocessing, like deal with the outliers and standardized the dataset. I also created some missing values and use KNN fill in them. In the training process, see how missing values effects the performance. For feature selection, we use confusion matrix heatmap and random forest to see which features are correlated and which features are more important.

Then I select Logistic Regression, Decision Tree, Random Forest and Adaboost and semi-supervised algorithm in the training process. Find the minimum mean cross-validation error to select the best hyperparameters and the best model (algorithm). Finally, train the training set on our best model and calculate the test error and out of sample error.

2. Introduction

2.1. Problem Type, Statement and Goals

In the music industry, accessing the potential of a song with some acoustic characteristics, like danceability, loudness, is a very important task. It will help music companies to plan how to make the publicity plan to help them to get the most profit.

Based on dataset of track features from year 2000-2019, this project conducted the research on the prediction about if a song would be a hit song

(If it will be on Billboard Hot100). Therefore, this is a binary classification problem. This project tried several machine-learning methods, including semi-supervised learning method, to see which one performs better on this dataset.

During the process of dealing with this problem, I need to solve the challenges as following: the outliers, limited number of training samples and missing data (created artificially) to see how that affects performance. And the exploration of semi-supervised learning that how missing labels affect performance compared to the original supervised case.

2.2. Literature Review (Optional)

There are some publications that do the hit song prediction by some machine-learning algorithms, including Logistic Regression, Support Vector Machines, Decision Trees and Neural Networks, etc. And they often use the accuracy to show their results. The accuracy is about 75%.

2.3. Our Prior and Related Work -None

2.4. Overview of Our Approach

In preprocessing, I deal with the outliers and standardization the dataset. Then, the heatmap is used to express the confusion matrix between each two features and use random forest to see the importance of these features.

In the training process, I use logistic regression with L1 regularization), decision tree, random forest and Adaboost algorithms. 5-cross validation is used to select the best hyperparameters and get the main validation error. Then the algorithm with the minimum validation error is selected. To explore more, I create some missing data artificially. After I filling these missing data by KNN method, I used the algorithms above to see how that affects performance.

Finally, I deleted some labels, using label propagation algorithm to see how semi-supervised learning performs compared to the original supervised case.

3. Implementation

3.1. Data Set

Our original dataset contains 12272 data points and 18 features. The output is 1 if this song has featured in the weekly Billboards Hot-100 Charts, otherwise, the output is 0. The table 3.1 listed all features in our original dataset.

Table 3.1: the list of all features and description

Feature name	Data type	Description
Track	String	The Name of the track.
Artist	String	The Name of the Artist.
Uri	String	The resource identifier for the track.
Danceability	Decimal	How suitable a track is for dancing
Energy	Decimal	A perceptual measure of intensity and activity.
Key	Integer	The estimated overall key of the track. E.g. C=0, D=2, and so on. If no key was detected, the value is -1.
Loudness	Decimal	The overall loudness of a track in decibels(dB)
Mode	Binary	The modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
Speechiness	Decimal	The presence of spoken words in a track.
Acousticness	Decimal	A confidence measure of whether the track is acoustic.
Instrumentalness	Decimal	Predicts whether a track contains no vocals.
Liveness	Decimal	Detects the presence of an audience in the recording.
Valence	Decimal	A measure describing the musical positiveness conveyed by a track.
Tempo	Decimal	The overall estimated tempo of a track in beats per minute (BPM).
Duration_ms	Decimal	The duration of the track in milliseconds.
Time_signature	Integer	An estimated overall time signature of a track.
Chorus_hit	Decimal	Author's best estimate of when the chorus would start for the track.
Sections	Integer	The number of sections the particular track has.

3.2. Dataset Methodology

The following is the procedure we followed in the use of the dataset.

Because the original dataset has limited data points, so we divide the dataset to training set (9817 data points) and test set (2453 data points) with the proportion of 8:2 and use 5-fold cross-validation. No pre-training set and validation set.

The 5-fold cross-validation was applied when we select the best hyperparameters of each model. For example, we used cross-validation to find the best regularization strength in logistic regression with L1 penalty. If there are more than one hyperparameters, like decision tree, random forest and Adaboost, we used grid search to implement cross-validation to find the best model based on minimum of the mean error of cross-validation. The minimum error is also used to choose algorithm with the best performance on our dataset.

As for test set, we use the test set only once on the selected best model to get a tighter bound of out-of-sample bound. After use the whole training set on the best model, use test set to see the performance.

Figure 3.1 shows the methodology that we use of dataset.

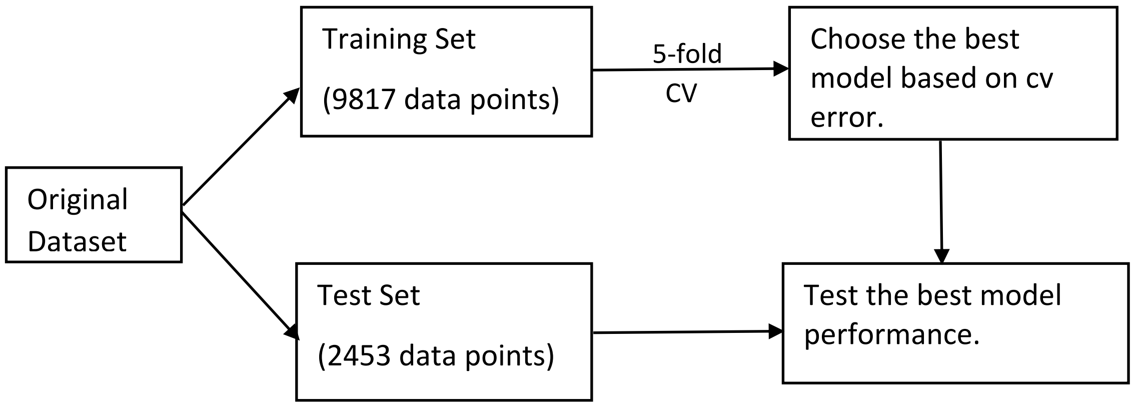
Original Dataset
Training Set (9817 data points)
Test Set (2453 data points)
5-fold CV
Choose the best model based on cv error.
Test the best model performance.

Figure 3.1 Dataset Methodology

3.3. Preprocessing, Feature Extraction, Dimensionality Adjustment

3.3.1. Preprocessing

1) The features that are string type

   There are three string-type features. They are: Track name, artist and uri. These just the name of the track and don't have significant influence to our prediction, so we just ignore them and use the rest of the numerical features to do the prediction.

2) Missing values in features

   In the original dataset, there is no missing values in it. However, to see how the missing values effect the performance of the model, we created some missing values artificially in some features. We use the completed dataset and dataset with missing values to train different models and compare the results. This will be stated in the training process.

For the dataset with missing values, there are two methods: delete if the number of missing values is very small, otherwise, we need to fill in the missing values. Figure 3.2 shows the number of missing values in each feature.

As the figure shows, there are more than 12% missing values in feature "danceability", 5.8% in feature "acousticness", 0,78% in "uri", 0.62% in "artist", 0.40% in "speachiness", 0.39% in "instrumentalness". Feature "uri" and "artist" we have ignored them.

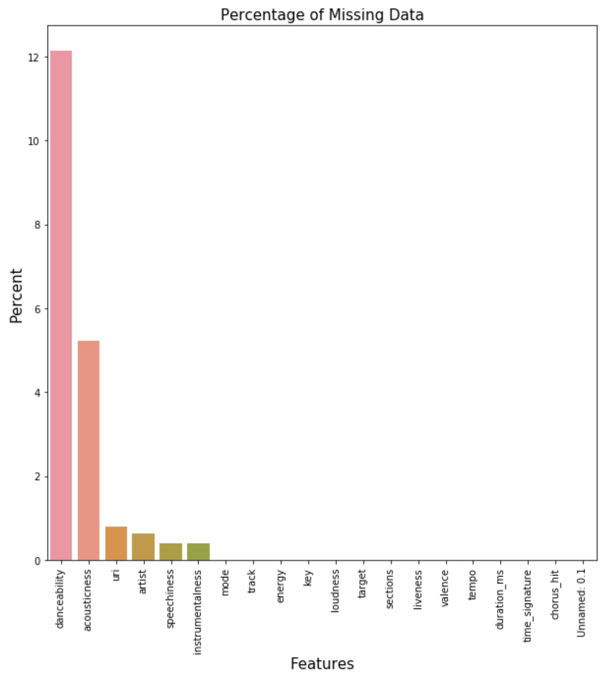

Figure 3.2 The number of missing values in each feature

For feature "speechiness" and "instrumentalness", the missing values are less than 0.5%, which is a very small data points to the whole set, so we just delete the missing values in these two features.

For feature "danceability" and "acousticness", the missing values are in larger proportion, so we need to fill these missing values. We decide that complete missing values using k-Nearest Neighbors. For each missing value, fill the mean value from n nearest neighbors in the training set, then, use these KNN model on training set applied to the test set and fill in all missing values.

3) Outliers

An outlier is an abnormal distance from other values in the dataset. So we define outlier by 25th (Q1) and 75 (Q3) percentiles. If the data point is not in the range from Q1-1.5*IQ to Q3+1.5*IQ, where IQ=Q1-Q3, the data point would be the outlier.

Check outlier of each feature using definition above. For the outlier less than the lower bound, we replace it with the lower bound, otherwise, replace it with the upper bound.

4) Standardization

First, we standardize the training set. The formulate is the following:

$$x' = \frac{x_i - \bar{x}}{s}$$

Where, $\bar{x}$ is the mean value, s is the standard deviation.

Then, applied the mean and standard deviation to test set to do the standardization of the test set.

3.3.2. Dimensionality Adjustment

After handle the outlier, we find out that in feature "time_signature" are all the same. So it will be a useless feature that we need to ignore it.

To prevent overfitting because high dimension features, we plot the heatmap of confusion matrix of the rest features to reduce the complexity as much as possible. As Figure 3.3.

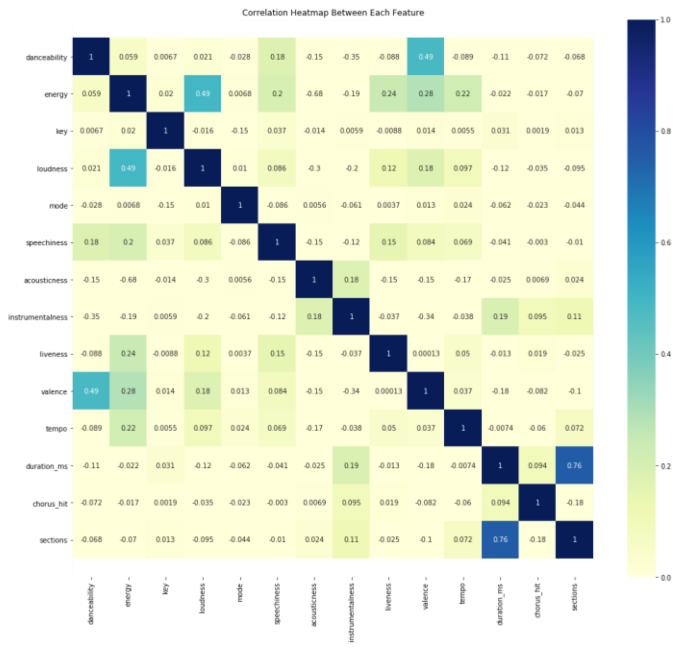

Figure 3.3 Confusion matrix heatmap

From the heat map, except the feature duration_ms_have relatively higher correlation coefficients. For other features, coefficients are relatively small, which means these features are not depend on each other very much, which is good for our learning process.

However, when we run Random forest algorithm, we get the importance of each features. As table 3.2 shows.

Table 3.2 Importance of each feature

Feature	Weight (Importance)
Danceability	0.119241
Energy	0.089545
Key	0.021190
Loudness	0.024782
Mode	0.005958
Speechiness	0.045117
Acousticness	0.112833
Instrumentalness	0.283649
Liveness	0.036160
Valence	0.067785
Tempo	0.038000
Duration_ms	0.091831
Chorus_hit	0.035635
Sections	0.028364

From the table, we can see that the most important 5 features are Instrumentalness, danceability, acousticness, duration_ms and energy. The duration_ms is relatively important feature. So we keep it for training. The scatter/box plots of these most important 5 features are shown in Figure 3.4.

From the figure, two classes are separated not quite so good, but some degree, they can be separated. That means our prediction result would not be extremely high, but would be good somewhat.

Finally, we choose 14 features after ignore feature track, artist, uri and time_signature.

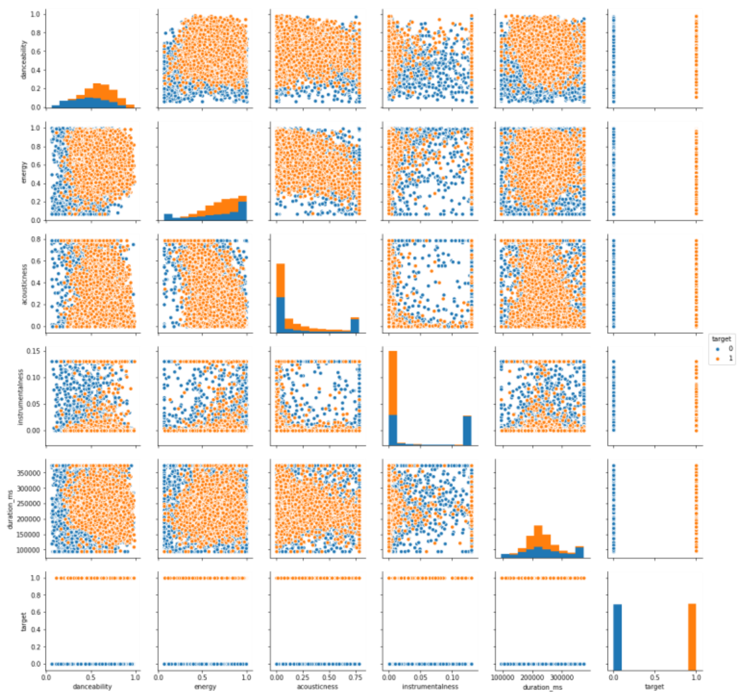

Figure 3.4 scatter/box plots of these most important 5 features

Form the objective function, the hyperparameter I need to choose is the strength of regularization λ. In the process of implementing the algorithm, the parameter we can set is C, inverse of the strength of regularization, smaller values specify stronger regularization. We set C from -3 to 1 for 10 values. Use complete training set and training set with missing values, and compare the results. Table 3.3 shows the parameter the two data set selects and the error rate. Figure 3.5 shows the different of error rate between the two sets.

From the table and figure, we can tell the missing data would increase the error rate of the model and strength regularization is stronger even if we have fill in the missing data with suitable method.

For the original training set, we get best validation accuracy 0.797 (minimum error rate 0.203) by selecting hyperparameter C=3.59. The complexity can measured by VC dimension, for logistic regression, it is VC-dimension=14+1=15

Table 3.3 Accuracy of cross-validation and parameter

Dataset	Error rate	Parameter
Complete training set	0.203	C= 3.59
Training set with missing values(after fill in)	0.217	C= 1.29

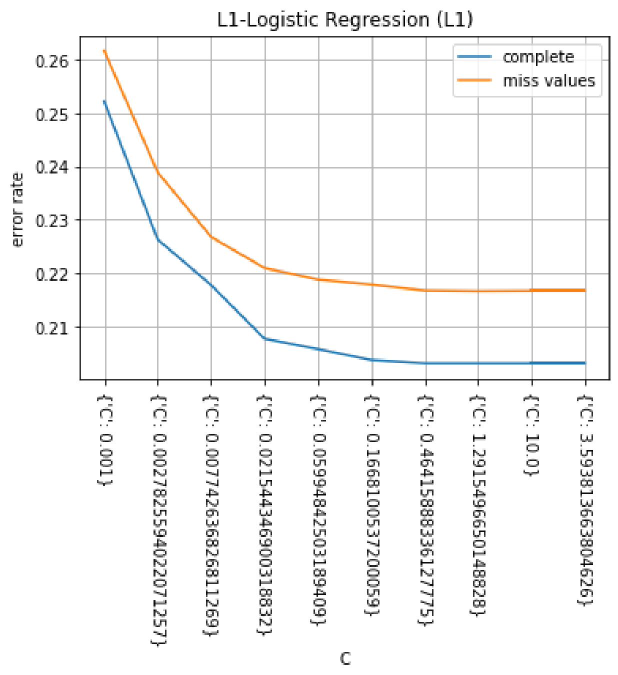

Figure 3.5 Error rate in training process

3.4.2 Decision Tree

We would like to use decision tree method, because it is a simple classifier, too.

In decision tree, we choose two hyperparameters: the maximum depth of the tree and the minimum number of samples required to be at a leaf node. We set max depth from 10 to 100 for 10 values, min sample leaf from 1 to 16 for 5 values.

Because there are more than 1 parameter for us to choose, so we use grid search to find the best parameter combo. The scores depending on Max_depth and min_samples_leaf are showing in figure 3.6.

For complete training set, the minimum mean 5-fold cross-validation error rate is 0.186 when max depth is 10 and min samples leaf is 8, which is perform better than logistic regression with L1 regularization.

Use complete training set and training set with missing values, and compare the results. Table 3.4 shows the parameter the two data set selects and the accuracy. The missing values would also influence the performance of decision tree.

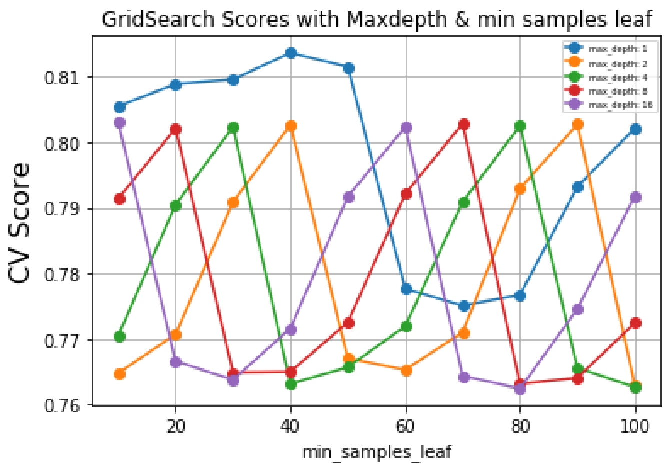

Figure 3.6 Scores depending on Max depth and min samples leaf

Table 3.4 Accuracy of cross-validation and parameter

Dataset	Error rate	Parameter
Complete training set	0.186	Max_depth=10 min_samples_leaf= 8
Training set with missing values (after fill in)	0.195	Max_depth=10 min_samples_leaf= 16

3.4.3. Random Forest

Because decision tree is easily overfitting, so we would like to try random forest to deal with the problem and maybe we will get better performance.

In this method, we have more hyperparameters needed to choose. They are: the max depth of the tree (max_depth), the minimum number of samples required to be at a leaf node (min_samples_leaf), the minimum number of samples required to split an internal node (min_samples_split) and the number of trees in the forest (n_estimators). We set max_depth from 2 to 10 for 4 values, min_samples _leaf from 1 to 4, 3 values, min_samples_split from 2 to 10 fir 3 values, and finally n_estimators from 10 to 500 for 7 values. The values in details are following:

Max_depth=[2,5,10,20,None];min_samples_leaf=[1,2,4];
min_samples_split=[2, 5, 10]; n_estimators=[10,50,100,200, 300, 400, 500]

Because there are 4 parameters needed to choose, we choose random search to reduce the search time. The method of search is select each values of each parameter randomly and combine them. In this random forest algorithm, we pick 10 combination of parameters and choose the best model with the minimum error rate.

Figure 3.7 shows that the 10 combination and their error rate. Finally we choose n_estimators= 100, min_samples_split= 5, min_samples_leaf=1, max_depth= None with the minimum error rate 0.153.

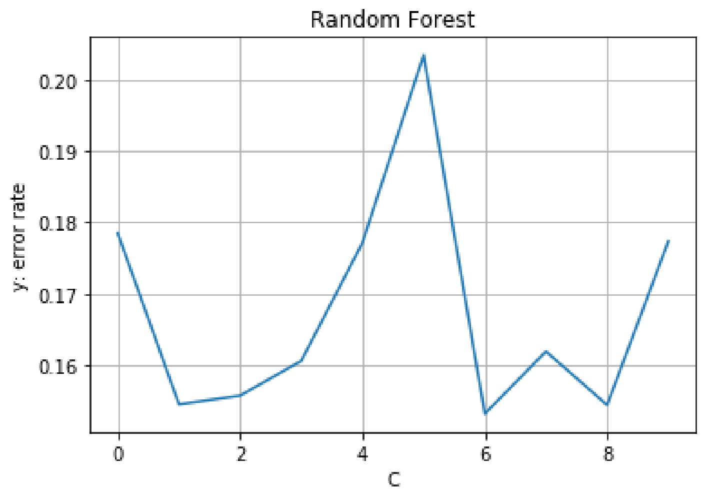

Figure 3.7 Random forest error rate with different parameters

In the same way as logistic regression and decision tree, we applied random forest to the training set with missing data. The missing data will influence on random forest even if only little effects. Table 3.5 shows the results.

Table 3.5 Accuracy of cross-validation and parameter

Dataset	Error rate	Parameter
Complete training set	0.153	n_estimators=100, min_samples_split=5, min_samples_leaf=1, max_depth=None
Training set with missing values (after fill in)	0.161	n_estimators=500, min_samples_split=5, min_samples_leaf=4, max_depth=20

3.4.4. AdaBoost

To compare two ensemble learning method we use Adaboost.

In Adaboost, the hyperparameters we need to choose are: the maximum number of estimators at which boosting is terminated (n_estimators) and learning rate shrinks the contribution of each classifier (learning_rate).

We choose n_estimators form 100 to 500 for 5 values (100,200,300,400,500) , learning rate from 0.025 to 0.30 for 8 values (0.025,0.05,0.1,0.15,0.20,0.25,0.30). There are only 2 parameters, so I use grid search to find the best parameter. Figure 3.8 shows that different combination with different error rate. We select the parameters with minimum error rate as our best hyperparameters. Our final selection is: learning rate= 0.1, n_estimators=400, and the min error in validation is 0.165.

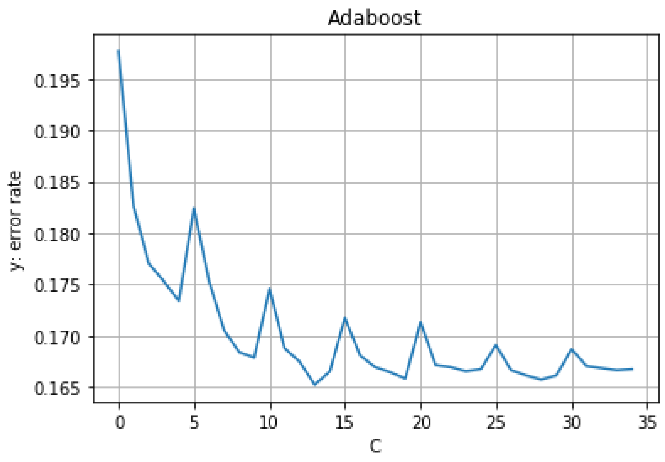

Figure 3.8 Error rate & parameters

As for missing values, results of the error rate and parameters selection compared to complete train data, shown as table 3.6.

Table 3.6 Accuracy of cross-validation and parameter

Dataset	Error rate	Parameter
Complete training set	0.165	n_estimators=400, learning rate= 0.1
Training set with missing values (after fill in)	0.174	n_estimators=400, learning rate= 0.1

3.4.5. Semi-supervised Learning

To compare the semi-supervised learning to the original supervised case, we delete about 30% labels to create the semi-supervised learning case. We use label propagation to do the semi-supervised learning. First, we delete 30% labels randomly and label them as -1. And then, use the propagation method to predict the data point labeled as -1. Finally, calculate the accuracy between the real labels and our prediction results.

The accuracy is only 0.507, which is lower a lot than supervised learning case. So This supervised learning case is not suit for semi-supervised learning.

3.4. Model Selection and Comparison of Results

We used 4 supervised learning models and 1 semi-supervised model in the beginning. We select the model based on the minimum of the error rate (highest of accuracy) of mean cross-validation error. The results are combines as below table 3.7.

Table 3.7 Model Selection and Comparison of Results

Model	Error rate	Parameter
Logistic Regression (L1)	0.203	C= 3.59

Decision Tree	0.186	Max_depth=10
		min_samples_leaf= 8

Random Forest	0.153	n_estimators=100,
		min_samples_split=5,
		min_samples_leaf=1,
		max_depth=None

Adaboost	0.165	n_estimators=400, learning rate= 0.1
Label- propagation	0.493	/

By comparison, we select the Random Forest with 100 trees in forest, 5 minimum samples leaf, 1 minimum samples leaf and max depth with None. From the research on random forest before, we select Instrumentalness and danceability to plot the decision boundary with restrict to them in 2D. The figure 3.9 shows the decision boundary.

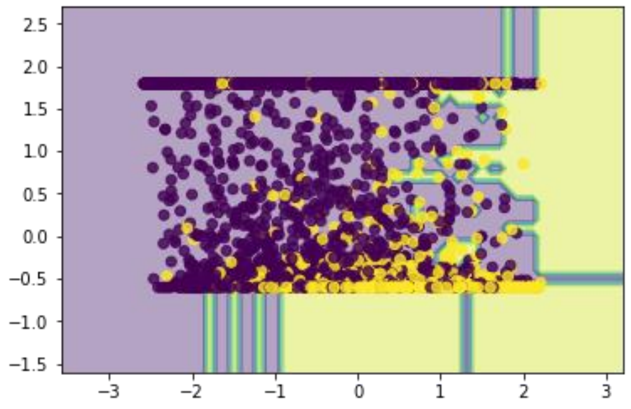

Figure 3.9 Decision boundary of 2 features

In the figure we can see that, the purple part and yellow part represent the 2 outputs, and the decision boundary can be found easily from the plot.

4. Final Results and Interpretation

Finally, we choose random forest as our best model based on the average error on multiple runs of cross-validation, and the hyperparameters are 100 trees in forest, 5 minimum samples leaf, 1 minimum samples leaf and max depth with None. In the training and test set, we only use 14 features. We use this model to train the whole training set and calculate the test error rate and accuracy. In table 4.1, we show the final model and results.

Table 4.1 Final model and results

Algorithm	Hyperparameters	Baseline Error rate	Results in literature	My Test Error rate	My Test accuracy
Random Forest	n_estimators=100, min_samples_split=5, min_samples_leaf=1, max_depth=None	0.509	0.235	0.168	0.832

Calculate the bound for out of sample error:

$$E_{out}(h_g) \leq E_{test}(h_g) + \sqrt{\frac{1}{2N_{test}} \ln\left(\frac{2M}{\delta}\right)}$$

$$E_{test}(h_g) = 0.168, N_{test} = 2453, M = 1, \delta = 0.05$$

$$E_{out}(h_g) \leq 0.168 + \sqrt{\frac{1}{2 \times 2453} \ln\left(\frac{2}{0.05}\right)} \approx 0.195$$

Results:

1. The missing values can influence the performance on different methods by increase the average error of crossvalidation, especially logistic regression. Random forest is less sensitive than the other methods. Random forest can fit in missing value during the training process. When we use Random forest, we don't need to fill in the missing values.

2. The feature "time signature" is not a very related feature in this problem based on our research, so in the future, we can reduce the effort to obtain this kind of data.

3. In Hit song prediction problem, the most 2 important features are "Instrumentalness" and "danceability". If the danceability is higher, more possible the song will be a hit song. If the instrumentalness is low, which means the vocals is clearer, and the song is more likely be a hit song. However, from the confusion

matrix heatmap, each feature is not highly correlated. So each feature is important also.

4. In this problem, for supervised learning case, Random forest performs the best, then decision tree and Adaboost, finally logistic regression. Maybe it is because logistic regression assumes these features and target are linear relationship, which is not a true assumption in this problem.

5. In this problem, for the semi-supervised learning case, the accuracy of matching their actual labels is relatively poor, so we concluded that the semi-supervised learning algorithms are not suit for this supervised learning problem.

6. For decision tree and random forest, random forest performs better in this problem. Because random forest consists of many trees and decision tree is more likely overfitting and may cause high variance.

5. Contributions of each team member

The work of the whole project is completed by Yingfeng Lou, myself

6. Summary and conclusions

To predict if a song will be a hit song, we find out that there are some important features, like “instrumentalness”, “danceability”, “acousticness”, “deration_ms” and “energy, that that has more correction with the target——if it will be a hit song.

We also explore how to deal with missing values and outliers and see their influence on different models. Then we compare different models and select the best model, random forest with suitable parameters, improving the test error from base line 0.509 to 0.168. And the out of sample error is less than 0.195

7. References

[1] "HIT SONG PREDICTION:LEVERAGING LOW- AND HIGH-LEVEL AUDIO FEATURES," 4 November 2019. [Online].Available: https://archives.ismir.net/ismir2019/paper/000037.pdf

[2] “7.1.6.What are outliers in the data?” [Online].Available:

https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm

[2] “HITPREDICT: PREDICTING HIT SONGS USING SPOTIFY DATA,” [Online].Available: http://cs229.stanford.edu/proj2018/report/16.pdf