DataSet of feature vectors with golden labels (H): SampledData.csv
Development Set (I): first 300 feature vectors of set H
Debugging stage:
    Training set (U): first 150 feature vectors of set I
    Testing Set (V): last 150 feature vectors of set I
Evaluation Set (J): last 100 feature vectors of set H

1. Report the precision, recall, and F-1 for each of the five learning methods obtained from performing cross validation on I

   We performed leave-one-out-cross-validation on our development set and obtained the following results for the metrics of precision, recall and F-1

	Precision	Recall	F-1 Score
Decision Tree	0.895	0.879	0.887
Random Forest	0.916	0.879	0.898
Support Vector Machine	0.911	0.879	0.895
Gaussian Naive Bayes	0.911	0.879	0.895
Logistic Regression	0.911	0.879	0.895

2. Report which learning based matcher you selected after that cross validation.

   The Random Forest classifier recorded the highest F1 score on the development set at 0.911 and was selected as the best matcher to improve upon.

3. Report all debugging iterations and cross validation iterations that you performed.

   We made three iterations of debugging, all over the Random Forest classifier. We chose to debug this classifier because it consistently showed a higher overall accuracy than the

others. During these debugging activities we tried to improve the recall score which was relatively low to begin with.

Iteration 1: Data cleaning and correcting mislabeled golden data

a. What is the matcher that you are trying to debug, and its precision/recall/F-1

We are trying to debug the Random Forest:

- Precision: 0.916
- Recall: 0.879
- F-1 Score: 0.898

b. What kind of problems you found, and what you did to fix them

We were struggling with low recall scores so we closely observed all the false negatives and false positives examples; We discovered that many false samples were resulting from mis-labelling in the golden data. So we cleaned our golden data set by correcting all of corresponding feature vectors.

We also stripped the white-spaces in the label string to improve the similarity scores.

c. The final precision/recall/F-1 that you reached.

- Precision: 0.927
- Recall: 0.957
- F-1 Score: 0.941

Iteration 2: Inverted the date/time feature vector computation

a. What is the matcher that you are trying to debug, and its precision/recall/F-1

We are trying to debug the Random Forest:

- Precision: 0.927
- Recall: 0.957
- F-1 Score: 0.941

b. What kind of problems you found, and what you did to fix them

In this iteration, we aimed to improve the precision. We used the Jaro String similarity score to measure the "similarity" of released Date. However we found that dates such as "23-NOV-08" has high similarity score with "21-DEC-06", which does not make sense . Then we re-considered the rule of similarity measure: taking the date difference of the two and normalized by the date difference between the oldest day and today. This makes more sense since date are more "different" if there are more days in between.

The second problems we found is that we applied the similar strategy on tracktime feature: Taking the time difference between two tracks and normalize it by the maximum track time, but we reversed the similarity scores in this way since close time difference between two songs will have low similarity scores. Then we correct the trend by taking the complement of this ratio respect to 1.

c. The final precision/recall/F-1 that you reached.

- Precision: 0.944
- Recall: 0.957
- F-1 Score: 0.950

Iteration 3: Designed custom similarity measure for Artist and Tracking features

a. What is the matcher that you are trying to debug, and its precision/recall/F-1

We are trying to debug the Random Forest:

- Precision: 0.944
- Recall: 0.957
- F-1 Score: 0.950

b. What kind of problems you found, and what you did to fix them

In this iteration, we aimed to fix the false negative that continue to appear in our debugging logs and improve the recall further. We found that artist name feature values have not been reasonably measured, For example, "Eminem" and "Armin

van Buuren", "Beyonce" and "Tye Tribbett & G.A." have similarity score with 0.44, 0.49, respectively, which does not make sense since string "edit distance" should not be equivalent to artist "similarity". Instead, we adopt both the feature of embracing mis-spelling and differentiating two unrelated strings. We implement following strategy: first tokenizing the string by whitespace to split the artist name by word and compute similarity score between words pairwisely. If the highest Jaro similarity score is not higher than our predefined threshold, 0.75, then we disable this feature by setting the corresponding value to 0. Else we simply adopt the jaro similarity score between two entire names.

We also noticed that in some cases where some different songs were released in a single albums at the same day, belong to the same artist and they happened to have the same track duration. In this case, even the track name is different and the similarity score is very low due to other features are perfectly matched, the classifier still predicted false positive.

To solve this problem, we appropriately re-weighted the feature value so that track names get more weighted to decide the prediction but at the same time we "prune" the cases where the previous feature, the similarity among artist name, scored 0 by dividing the original value of feature value by 3 to deprecate its performance on decision.

The idea behind that is the track name similarity score is not that important any more given we were almost sure that two artist were different.

c. The final precision/recall/F-1 that you reached.

- Precision: 0.917
- Recall: 1.000
- F-1 Score: 0.957

4. Now report the following:

a. For each of the five learning methods, train it on I, then report its precision/recall/F-1 on J

	Precision	Recall	F-1 Score
Decision Tree	0.917	1.000	0.957
Random Forest	0.917	1.000	0.957
Support Vector Machine	0.917	1.000	0.957
Gaussian Naive Bayes	0.880	1.000	0.936
Logistic Regression	0.917	1.000	0.957

b. For the final best matcher Y_*, train it on I then report its precision/recall/F-1 on J

The best (selected) matcher is the Random Forest, although after the debugging tweaks all the classifiers except for Naive Bayes have the same scores. The precision, recall and F-1 score for Random Forest classifier is

- Precision: 0.917
- Recall: 1.000
- F-1 Score: 0.957

c. List the final set of features that you are using in your feature vectors.

There are four features in the feature vector apart from the classification label

1. Custom similarity measure (based on Jaro) for Artist Attribute
2. Custom similarity measure (based on Jaro) for Track name Attribute
3. Distance measure between Release date attribute
4. Distance measure between Duration attribute

5. Report an approximate time estimate

a. How much did it take to label the data

It took 10 mins to label total of 400 sam[les of data

b. To find the best learning-based matcher

We used sklearn to run cross-validation for each classifier. We spent an hour doing that in the first pass. We spent another 3 - 4 hours debugging the random forest classifier and selecting the best matcher.

6. Discuss why you can't reach higher precision, recall, F-1

Due to the insufficiency of our positive samples, even though in the end we almost achieved accuracy 1 for prediction on V set, our selected classifier were unable to reach high precision at predicting J dataset. In our sample dataset I, they are only 58 positive samples and 75 for the total sample set. Therefore, it is highly likely to generate false positive samples due to insufficient training and significantly affect the precision rate on J set which only contain 75 - 58 = 17 true positive samples.