## Part 1.1 - Feature Engineering with Feature Subsets (10 points)

- 1.1.1 Which model had the best RMSE on the training data? (1 point)

  - **Ans: artist_reviewauthor_releaseyear_recordlabel_genre_danceability_energy_key_loudness_speechiness_acousticness_instrumentalness_liveness_valence_tempo**

- 1.1.2 Which model had the best RMSE on the test data? (1 point)

  - **Ans: artist_reviewauthor_releaseyear_recordlabel_genre_danceability_energy_key_loudness_speechiness_acousticness_instrumentalness_liveness_valence_tempo**

- 1.1.3 Which feature do you believe was the most important one? Why? (Note: There is more than one perfectly acceptable way to answer this question) (2 points)

  - **Ans: reviewauthor. Because it has got the highest weight among all the features.**

- 1.1.4 What can we say about the utility of the Spotify features based on these results? (1 point)

  - **Ans: Since the weights are negative and close to zero, they do not contribute much to the final result.**

## Part 1.2 - Feature Engineering with the LASSO (15 points)

- 1.2.1 - How many new features are introduced by Step 2 above? Provide both the number and an explanation of how you got to this number. (2 points)

  - **Ans: 680 columns. These were introduced by doing the one-hot encoding on categorical columns.**

- 1.2.2 - What was the best alpha value according to your cross-validation results? (5 points)

  - **Ans: best_alpha: 3.87224703978818e-05**

- 1.2.3 - What was the average RMSE of the model with this alpha value on the k-fold cross-validation on the training data? (3 points)

- **Ans: 0.24625592947243133**

• 1.2.4 - What was the RMSE of the model with this alpha value on the k-fold cross-validation on the test data? (5 points)

- **Ans: 0.2662673832867446**

## Part 1.3 - Interpreting Model Coefficients (15 points)

- 1.3.1 - How many non-zero coefficients are in this final model? (5 points)
  - **Ans: 468 (print('number of non-zero elements: ',np.nonzero(model.coef_)[0].size))**
- 1.3.2 - What percentage of the coefficients are non-zero in this final model? (1 point)
  - **Ans: 67.72793053545585 (print('percentage of non-zero elements: ',np.nonzero(model.coef_)[0].size/model.coef_.size*100))**
- 1.3.3 - Who were the three most critical review authors, as estimated by the model? How do you know? (3 points)
  - **Ans: Ian Cohen, Joe Tangari, Mark Richardson (print((pd.concat([training_data['reviewauthor'],pd.DataFrame(np.power(2,y_pred_test)-1,columns=['score'])], axis=1).groupby('reviewauthor').sum()).sort_values(by=['score'],ascending=False)[1:4]))**
- 1.3.4 - Who were the three artists that reviewers tended to like the most? How do you know? (3 points)
  - **Ans: Mogwai, Xiu Xiu, Mount Eerie (print((pd.concat([training_data['artist'],pd.DataFrame(np.power(2,y_pred_test)-1,columns=['score'])], axis=1).groupby('artist').sum()).sort_values(by=['score'],ascending=False)[1:4]))**
- 1.3.5 - What genre did Pitchfork reviewers tend to like the most? Which genre did they like the least? (3 points)
  - **Ans: Most: Electronic; Least: Global (print((pd.concat([training_data['genre'],pd.DataFrame(np.power(2,y_pred_test)-1,columns=['score'])], axis=1).groupby('genre').sum()).sort_values(by=['score'],ascending=False)[1:2])**

    **&**

    **print((pd.concat([training_data['genre'],pd.DataFrame(np.power(2,y_pred_test)-1,columns=['score'])], axis=1).groupby('genre').sum()).sort_values(by=['score'],ascending=False)[-2:-1]))**

## Part 1.4 - “Manual” Cross-Validation + Holdout for Model Selection and Evaluation (25 points)

1.4.1 Report, for each model, the hyperparameter setting that resulted in the best performance (3 points)

- **Ans: DTR : 5_squared_error, Ridge : 10, KNN : 10**

1.4.2 Which model performed the best overall on the cross-validation? (3 points)

- **Ans: Ridge: 10**

1.4.3 Which model performed the best overall on the final test set? (3 points)

- **Ans: Ridge: 10**

1.4.4 With respect to your answer for 1.4.3, why do you think that might be? (Note: there is more than one correct way to answer this question) (1 point)

- **Ans: Only in the RIDGE model Regularization is present, which helps in reducing the generalization error and alpha value 10 is the best fit. Since one-hot encoding has increased the number of features, Ridge works the best.**

1.4.5 Which model/hyperparameter setting had the highest standard deviation across the different folds of the cross-validation? (3 points)

- **Ans: DTR: 20_absolute_error**

1.4.6 With respect to your answer for 1.4.6, why do you think that might be? (Note: there is more than one correct way to answer this question) (2 points)

- **Ans: The features seem to be linearly correlated to the target variable. Since DTR with max depth 20 has to complex heuristic function, it overfits the data so generalization error has a very high variance during k-fold cross validation.**

## Part 2.1 - Logistic Regression with Gradient Descent (25 points)

**2.1.1** - How did you go about selecting a good step size, i.e. one that was not too big or too small? (Note: There is more than one correct answer to this) (2 points)

- **Ans: We started with a higher value but we missed the optimum solution. So, we gradually reduced it. For a very small step size, the time taken to converge was very high. We feel the current step size is best as it converges faster and also it is close to optimum.**

**2.1.2** - What is the condition under which we assume that the gradient descent algorithm has converged in the code here? (2 points)

- **Ans: If the sum of the squared difference between new and old weights is very less in magnitude, then we do not continue because it is already converged.**

**2.1.3** - What is a different convergence metric we could have used? (Note: There is more than one correct answer to this) (1 points)

- **Ans: One possible metric can be the number of iterations. If we don't converge within the specified iterations, we stop. In each iteration, we need to ensure that the difference between old and new loss is greater than a certain threshold.**