ASSIGNMENT 2

[CSE 574]

1. **Rajiv Nagesh – UBIT: rajivnag**
2. **Ishansh Sahni – UBIT: ishanshm**
3. **Shreya Joshi – UBIT: shreyajo**

Part 1.1 Feature Engineering with Feature Subsets

1.1.1 Which model had the best RMSE on the training data?

The model that is trained with the features from the feature set containing ['artist', 'reviewauthor', 'releaseyear', 'recordlabel', 'genre', 'danceability', 'energy', 'key', 'loudness', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo']] has the best **RMSE on the training data with a value of 0.1847**

1.1.2 Which model had the best RMSE on the test data?

The model that is tested with the features from the feature set containing ['artist', 'reviewauthor', 'releaseyear', 'recordlabel', 'genre', 'danceability', 'energy', 'key', 'loudness', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo']] has the best **RMSE on the test data with a value of 0.19177**

1.1.3 Which feature do you believe was the most important one? Why?

The feature set consists of features from the Spotify API as well the Pitchfork API, a combination of the features results in modeling each feature as an independent model against the target variable. This leads to an inference of the feature **['artist', 'reviewauthor', 'releaseyear', 'recordlabel', 'genre', 'danceability', 'energy', 'key', 'loudness', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo']] as the most important one** among all the features from the given feature set.

This inference is based upon observing the RMSE of both the train and test data. Furthermore, we can see that this list is a comprehensive collection of various features like – author, reviewauthor, and various other features that describe the type of music and breaks down a record to its details. Hence, this is my inference on why this feature from the given feature set is the most important one, “Keeps the RMSE in mind”, “Keeps the number of attributes in that feature in mind”.

1.1.4 What can we say about the utility of the Spotify features based on these results?

According to the Spotify API documentation given at
(https://developer.spotify.com/discover/#audio-features-analysis)
(https://developer.spotify.com/documentation/web-api/reference/#/)

We can draw an inference that the Spotify’s API captures information like Mood, Properties, Context, Segments, etc. of every song. These features play a vital role in deciding the review for an author on Pitchfork. Without the features captured from the Spotify API, the Pitchfork authors won’t have many details to rate a song/artist/album. As discussed earlier, the features like **['artist', 'reviewauthor', 'releaseyear', 'recordlabel', 'genre', 'danceability', 'energy', 'key', 'loudness', 'speechiness', 'acousticness',**

'instrumentalness', 'liveness', 'valence', 'tempo']] are the most important and they draw conclusions on the review score of a particular artist/song, etc. based on these features.

Part 1.1 Feature Engineering with LASSO

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.

There are a total of 680 new features that have been introduced. They replaced the 4 categorical features that were mentioned in the code block. The addition of new features is because of the "One-Hot-Encoding" step done. What One Hot Encoding does is, it takes your categorical features and creates a separate column of each of these features and assigns values as 1 or 0 if that feature is present in independent variable against the target variable.

1.2.2 What is the best alpha value according to your cross-validation results?

Best alpha value: 3.4212169735207094e-05. Alpha value. α (alpha) is the parameter which balances the amount of emphasis given to minimizing RSS vs minimizing sum of square of coefficients. It helps in choosing which features are the most important features given the entire feature set. In other words, alpha is said to be a hyperparameter. It is the amount of penalization chosen by Cross-Validation.

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?

On keeping the alpha value to 3.4212 using the LASSO Regression the **Average RMSE was computed to of 0.1746 of the train_data.**

1.2.4 What was the **RMSE** of the model with this alpha value on the k-fold cross validation on the *test* data?

On keeping the alpha value to 3.4212 using the LASSO Regression the **Average RMSE was computed to of 0.1762 of the test_data.**

Part 1.3 Interpreting Model Coefficients

1.3.1 How many non-zero coefficients are in this final model?

There are 397 non-zero-coefficients in this final model.

1.3.2 What percentage of the coefficients are non-zero in this final model?

There is a 57.45% of non-zero coefficients in this final model.

1.3.3 Who were the three most critical review authors, as estimated by the model? How do you know?


1.3.3:

</>

	Author	Coeff
0	Alison Fields	-0.239852
1	Brian James	-0.204856
2	Mark Martelli	-0.186225

These are the three most critical review authors estimated by the model. This inference is drawn as “critical” is often synonymous to being harsh or negative. Keeping this in mind, we decided to sort the authors by their coefficients in an ascending way. This gives us an understanding that a review by the author “Alison Fields” is going to negatively the rating of an artist/song/album, etc. For instance, if there is a review about a song and if Alison Fields has given a review about the same song, that review is going to have a -0.23% effect on the overall rating of that song.

1.3.4 Who were the three artists that reviewers tended to like the most? How do you know?

1.3.4:

	Artist	Coeff
0	R.E.M.	0.091655
1	Miles Davis	0.074612
2	Deerhoof	0.058392

The top three artists according to the reviews and reviewers are listed above. This was done by grabbing the coefficients of artists and reviewers and then sorting them in a descending way. This gave us the above table where we can see that the artist R.E.M is the most liked artist on Pitchfork by the reviewers. Furthermore, every review by an author for the artist R.E.M contributes to 0.091% of the overall rating.

1.3.5 What genre did Pitchfork reviewers tend to like the most? Which genre did they like the least?

Genre that reviewers tend to like the most are –

1.3.5:

Users tend to like the most:

	Genre	Coeff
0	Jazz	0.007220
1	Global	0.005753
2	Pop/R&B	0.004058

Genre that reviewers tend to like the least are –

Users tend to like the least:

	Genre	Coeff
0	NaN	-0.035546
1	Electronic	-0.026255
2	Metal	-0.024287

Part 1.4 "Manual" Cross-Validation + Holdout for Model Selection and Evaluation

1.4.1 Report, for each model, the hyper parameter setting that resulted in the best performance

1.4.1 The hyperparameter setting for each model that resulted in the best perfomance:

	model_name	hyperparameter_setting	mean_training_rmse	std_training_rmse	test_rmse
1	DTR	20_squared_error	0.119369	0.006454	0.195066

	model_name	hyperparameter_setting	mean_training_rmse	std_training_rmse	test_rmse
6	Ridge	1e-05	0.175918	0.005099	0.198775

	model_name	hyperparameter_setting	mean_training_rmse	std_training_rmse	test_rmse
17	KNN	10	0.187466	0.004458	0.186756

1.4.2 Which model performed the best overall on cross-validation?

1.4.2 The model that performed the best overall on cross validation:

	model_name	hyperparameter_setting	mean_training_rmse	std_training_rmse	test_rmse
1	DTR	20_squared_error	0.119369	0.006454	0.195066

1.4.3 Which model performed the best overall on the final test set?

1.4.3 The model that performed the best overall on the final test set:

	model_name	hyperparameter_setting	mean_training_rmse	std_training_rmse	test_rmse
8	Ridge	1000.0	0.185534	0.004412	0.18552

1.4.4 With respect to your answer for 1.4.3, why do you think that might be?

The bigger the alpha, the more you penalize. Ridge regression shrinks the regression coefficients, so that variables, with minor contribution to the outcome, have their coefficients close to zero. The penalization stated to us were [10^{-5}, 10^{-4}, ….], this explains us that the alpha is big which would eventually make the coefficients to zero. The Ridge Regression hence performs better on the final test set as it converges the coefficients to zero which we want ultimately and that would generally make the RMSE lesser than the other models.

1.4.5 Which model/hyperparameter setting had the highest standard deviation across the different folds of the cross validation?

1.4.5 The model that had the highest standard deviation across different fold of cross validation:

	model_name	hyperparameter_setting	mean_training_rmse	std_training_rmse	test_rmse
2	DTR	20_absolute_error	0.133311	0.014386	0.194472

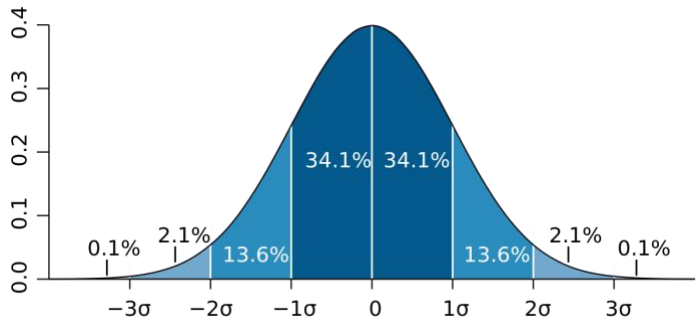
1.4.6 With respect to your answer for 1.4.5, why do you think that might be?

Regression Trees are another way of calling Decision Trees that are used for regression and it can be useful in a lot of areas where the relationship between the variables is found to be non-linear.

The algorithm is prone to overfitting. So, it is better to always specify the minimum number of children per leaf node in advance and use cross-validation to find this Value.

The hyperparameter setting was set with a tree depth of 20 and an absolute error. The depth of tree indicates the splitting of dataset into how many levels of parent child nodes. So, there is a tradeoff between smaller and larger value of the depth of the tree. Higher the value of the depth, more is the bias and vice versa. 20 for the depth of the tree here could be saying as the best fit for the tree as the standard deviation attained in this model for that same hyperparameter setting is 0.014386.

What do we know about standard deviation?



img src: https://en.wikipedia.org/wiki/Standard_deviation

As we can see how the bell curve lies, the standard deviation achieved in our case was 0.014386 and this could be stated as a robust model as it lies within 1-Standard Deviation away as seen in the figure above. We can then comment that the following model with a hyperparameter setting of 20 levels of depth and absolute error results in a medium bias low variance model and hence the computed S.D.

Part 2.1 Logistic Regression with Gradient Descent

2.1.1 How did you go about selecting a good step size, i.e., one that was not too big nor too small?

We first kept the step size as 0.1 and we converged to the gradient of the function at 17.3, we then kept reducing the step size and saw the gradient converge to a lower value. After 5 iterations, we kept the step size to 0.0001 and then saw the gradient of the function converge at 13.70. This was the lowest possible step size that we could give and then the gradient kept converging at the same value as above i.e., 13.70. This step size is optimal.

2.1.2 What is the condition under which we assume that the gradient descent algorithm has converged in the code here?

The condition where it states - `while ((new_w-old_w)**2).sum() > .0000000001:`

When this condition is met, the gradient converges. In simpler words, the value in the while function is called the precision and the function will keep converging unless and until it is above that threshold. When it goes lower than the precision value as stated above, the function converges to a local minima and we can say that the algorithm has converged.

2.1.3 What is a different convergence metric we could have used?

Different Metric for finding convergence –

1) **Linear Convergence**

(This is where we change the number of iterations ‘k’ to find the minima of a given function.)

Part 2.2 Logistic Regression with Netwon-Rhapson

This part is in the (.ipynb) code file only.