UNIVERSITÀ
DEGLI STUDI
DI MILANO

Applying Ridge Regression with k-fold Cross Validation to Spotify dataset to Predict Track Popularity

Master's Degree: Data Science for Economics, Machine Learning Project

Ahmet Emre Iskender

Index

1. Understanding data:

- Exploratory Data Analysis
- Data Preprocessing

2. Experiment Results and Analysis for Numerical dataset:

- Implementation with using different parameters
- Results Presentation using performance metrics

3.Feature Encoding:

- Data Visualization for Categorical Variables
- Data Preprocessing for Categorical Variables

4.Experiment Results and Analysis for Combined dataset:

- Implementation with using different parameters
- Result Presentation using performance metrics

5.Discussion:

- Comparison & Summary

6. Appendix:

- Dataset variables
- Data source
- Python code public repository link

1. Understanding data

Exploratory Data Analysis

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 114000 entries, 0 to 113999
Data columns (total 21 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Unnamed: 0        114000 non-null  int64
 1   track_id          114000 non-null  object
 2   artists           113999 non-null  object
 3   album_name        113999 non-null  object
 4   track_name        113999 non-null  object
 5   popularity        114000 non-null  int64
 6   duration_ms       114000 non-null  int64
 7   explicit          114000 non-null  bool
 8   danceability      114000 non-null  float64
 9   energy            114000 non-null  float64
 10  key               114000 non-null  int64
 11  loudness          114000 non-null  float64
 12  mode              114000 non-null  int64
 13  speechiness       114000 non-null  float64
 14  acousticness      114000 non-null  float64
 15  instrumentalness  114000 non-null  float64
 16  liveness          114000 non-null  float64
 17  valence           114000 non-null  float64
 18  tempo             114000 non-null  float64
 19  time_signature    114000 non-null  int64
 20  track_genre       114000 non-null  object
dtypes: bool(1), float64(9), int64(6), object(5)
memory usage: 17.5+ MB
```

Firstly, in order to understand the dataset size, the ".info()" function is used in order to visualize all of the columns and to see the data type for each of the features in the dataset,. In the above provided table which is the output of the ".info()" function, the total number of features, the data types of these features and the number of rows(114000) can be seen.

	Unnamed: 0	popularity	duration_ms	danceability	energy	
count	114000.000000	114000.000000	1.140000e+05	114000.000000	114000.000000	1
mean	56999.500000	33.238535	2.280292e+05	0.566800	0.641383	
std	32909.109681	22.305078	1.072977e+05	0.173542	0.251529	
min	0.000000	0.000000	0.000000e+00	0.000000	0.000000	
25%	28499.750000	17.000000	1.740660e+05	0.456000	0.472000	
50%	56999.500000	35.000000	2.129060e+05	0.580000	0.685000	
75%	85499.250000	50.000000	2.615060e+05	0.695000	0.854000	
max	113999.000000	100.000000	5.237295e+06	0.985000	1.000000	

A snapshot of the .describe() function.

In the above provided screenshot, which shows the dataset description, it is clearly seen that popularity values for each track is in the interval of 0-100. On the other hand, there the variables which have all values between 0-1 (danceability, energy etc.) Therefore, in order to avoid overflow, it is decided to normalize all of the values and use them between 0-1 value intervals. For example if a popularity score of a track is 73, in order implementation it will be converted into 0.73.

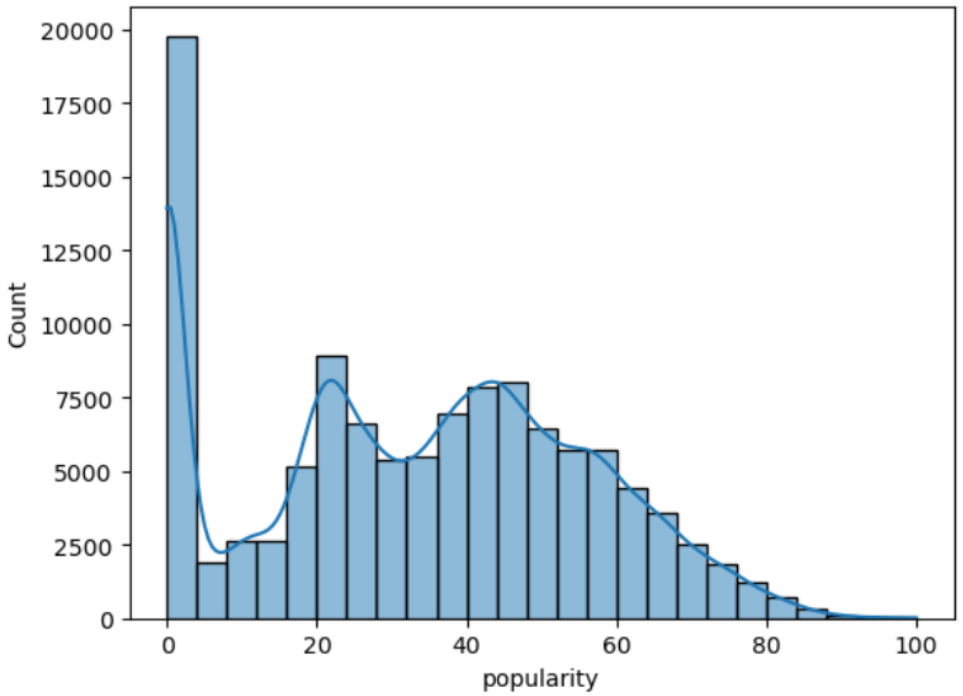


*Figure 1, Distribution of popularity values before 0-1 normalizatio*n.

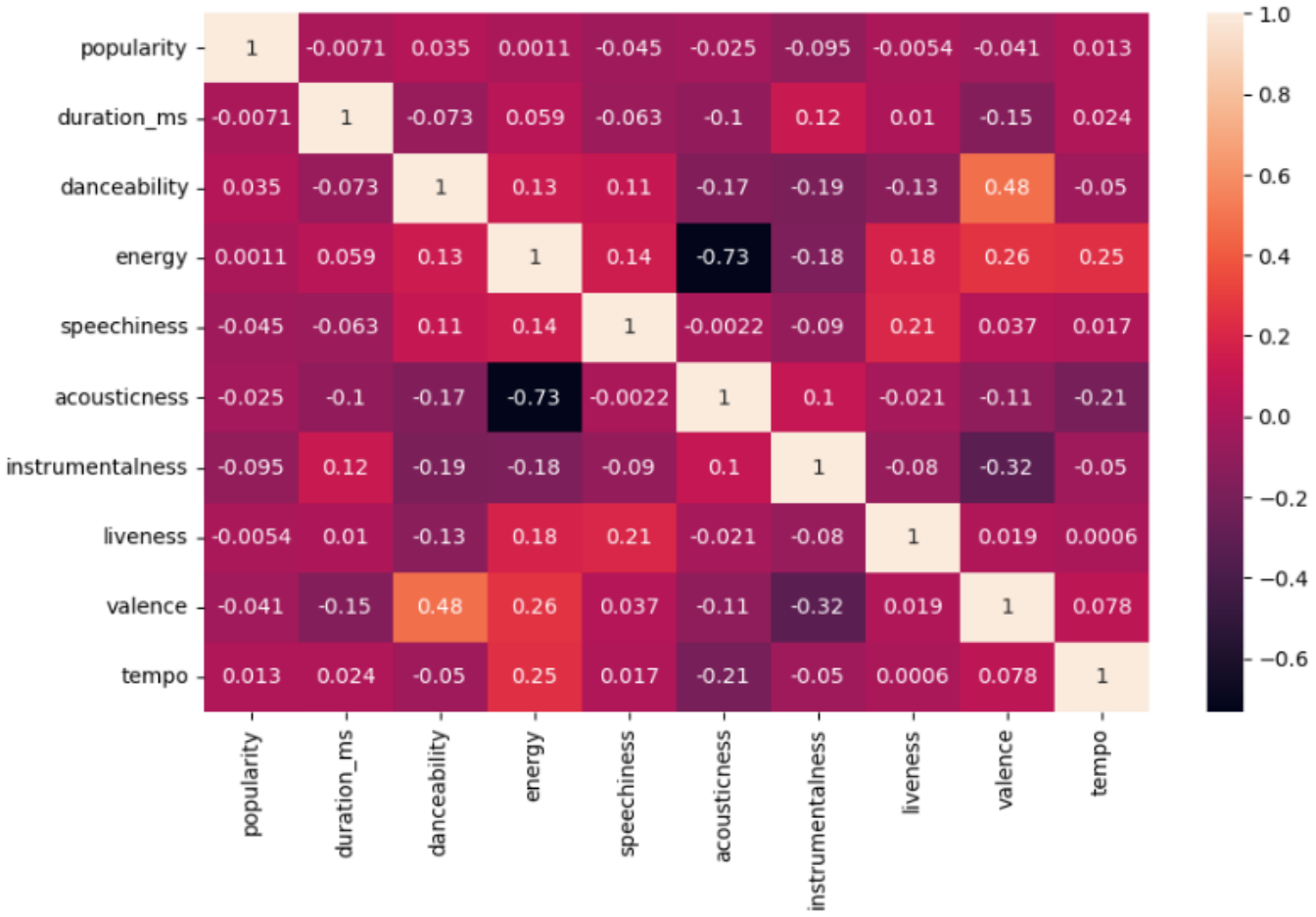


Figure 2

Before proceeding to the Ridge Regression with cross validation algorithm application, it was needed to extract the numerical features in the dataset. The numerical variables chosen that will be inputted into the algorithm are: 'duration_ms', 'danceability', 'energy', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo'. Then in order to obtain the correlation matrix to see the relation between all of these variables also with the target variables the dataset which contains only the numerical features which is extracted from spotify dataset is combined with the target variable these ‘popularity’. In Figure 2, the correlation matrix can be seen. In the provided figure, it can be seen that the most correlated variable with ‘popularity’ in ‘danceability’ but with a small magnitude.

Data Preprocessing

In order to make the dataset and the algorithm more relaxed and can be implementable easily, all of the numerical values as well as the target variable are normalized and obtained values between 0-1. Then, in order to eliminate the potential problems with the decimals, data is converted into 64-bit. Lastly, while applying the algorithm to the dataset in order to prevent the gradient of a loss function from becoming too large and overfit, the gradient clipping method is applied to the dataset. Please remember that, these Data Preprocessing methods are applied to the dataset because when running the algorithm without applying these data preprocessing methods, several errors are obtained then as mentioned above, in order to relax the algorithm and dataset these data preprocessing methods are applied.

2. Experiment Results and Analysis for Numerical dataset

Implementation with using different parameters

After writing the function for Ridge regression and k-fold cross validation and after defining which parameter values and evaluation methods will be used, the implementation started. For evaluation methods, Mean Squared Error(MSE) and Mean Absolute Error(MAE) are chosen and for parameter values alpha:(0.1, 10, 100) , for learning rate:(0.001, 0.01, 0.1), for number iterations:(1000, 5000, 10000) are chosen. The reason for choosing several parameters values and evaluating their results using the MSE and MAE was to obtain as many results as possible and to find the best parameter values without using any algorithm to determine the best parameters values like grid search. Moreover, due to having many parameter values and due to their combination various results are obtained. Furthermore, these results are dual results(i.e for number of iterations=1000, alpha=0.1 and learning rate=0.01 there is also MSE and MAE). In order to find the best solution both of the values are summed and evaluated accordingly. In the next section, the figure is provided showing each of the results for each of the parameter combinations.

Results Presentation using performance metrics

Numerical

Iterations=1000

	Learning rate = 0,001			Learning rate = 0,01			Learning rate = 0,1		
	MSE	MAE	**Sum**	MSE	MAE	**Sum**	MSE	MAE	**Sum**
α = 0,1	0,054630	0,196196	0,250826	0,051083	0,189966	0,241049	0,050164	0,187131	0,237295
α = 10	0,054632	0,196200	0,250832	0,051083	0,189968	0,241051	0,050166	0,18715	0,237316
α = 100	0,054654	0,196233	0,250887	0,051087	0,189985	0,241072	0,050183	0,187319	0,237502
		MIN	0,250826		MIN	0,241049		MIN	0,237295

Iterations=5000

	Learning rate = 0,001			Learning rate = 0,01			Learning rate = 0,1		
	MSE	MAE	**Sum**	MSE	MAE	**Sum**	MSE	MAE	**Sum**
α = 0,1	0,051296	0,190390	0,241686	0,05032	0,187933	0,238253	0,050146	0,186715	0,236861
α = 10	0,051296	0,190390	0,241686	0,050323	0,187944	0,238267	0,050144	0,186742	0,236886
α = 100	0,051297	0,190400	0,241697	0,050345	0,188044	0,238389	0,050144	0,186983	0,237127
		MIN	0,241686		MIN	0,238253		MIN	0,236861

Iterations=10000

	Learning rate = 0,001			Learning rate = 0,01			Learning rate = 0,1		
	MSE	MAE	**Sum**	MSE	MAE	**Sum**	MSE	MAE	**Sum**
α = 0,1	0,051083	0,189966	0,241049	0,050164	0,187132	0,237296	0,050162	0,186749	0,236911
α = 10	0,051083	0,189968	0,241051	0,050166	0,187151	0,237317	0,050158	0,18677	0,236928
α = 100	0,051087	0,189985	0,241072	0,050185	0,18732	0,237505	0,050146	0,186989	0,237135
		MIN	0,241049		MIN	0,237296		MIN	0,236911

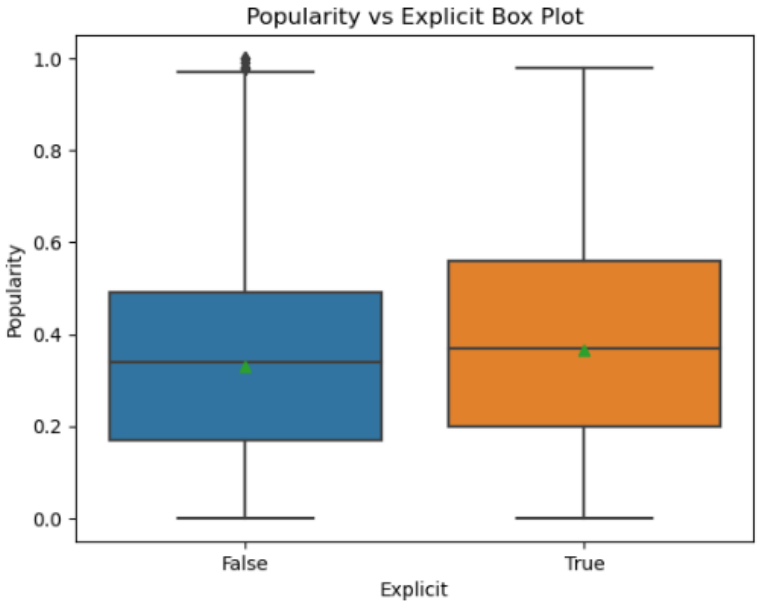
Figure 3

As obtained results shown in Figure 3, it can be seen that for all of the Number of Iterations combinations (1000, 5000, 10000) the best result is obtained through applying alpha=0.1 and learning rate=0.1. Among these 3 best results, the best of the best result is obtained using num_iterations= 5000, learning_rate=0.1 and alpha = 0.1. Furthermore, among these 3 best results, it can be seen that there are tiny differences between them. The best parameter combination score seems to be 0,236861 however the 2nd best combination score is 0.236911 and the difference between them is only 0.00005 which is a very low amount. To be able to drill down and interpret about the effect of each parameter it can be said that for each of the number of iterations combinations, if the learning parameter is increased the error rates decreases thus, by increasing the the learning rate, a positive trend is seen(we would like to see the sum value as low as possible). On the other hand, when the alpha is increased it is seen that there are cases in which the sum of the errors decreases, but generally speaking it can be easily visualized that the sum of error rates increases if we increase alpha. It can be stated that when alpha is increased our chances of predicting correctly decreases. Combining the interpretations about each of the parameters individually with the general results, it makes sense that for all number of iterations the best results are obtained using alpha=0.1 and learning rate=0.1.

3. Feature Encoding

Data Visualization for Categorical Variables

Before applying the feature encoding algorithms to the categorical features, it is necessary to firstly proceed with data visualization to understand better. Firstly, the categorical variable “explicit” is handled. This variable is a kind of a binary variable. Explicit variable shows if the lyrics of the song is explicit or not. In order to see this variables relationship with the target variable “popularity”, data is shown with the box plot also with the means in terms of popularity.



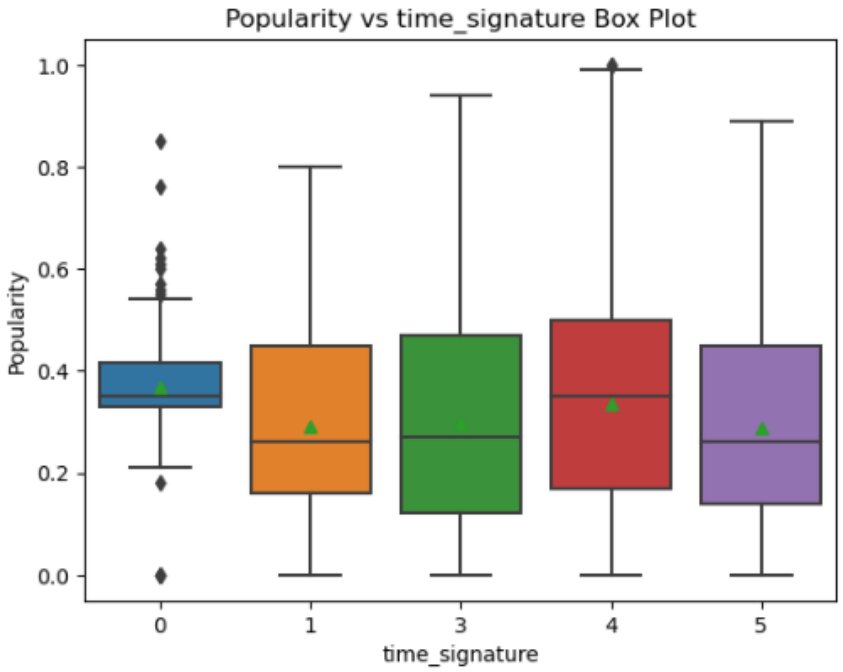

```
explicit
False    0.329379
True     0.364542
Name: popularity, dtype: float64
```

Figure 4

In Figure 4, the popularity of the tracks with the value “explicit” = False, the average popularity is 0.329. On the other hand, if the “explicit” value = True, the average popularity is 0.3645. As can be seen from the obtained results, there is only 4% of difference if the track contains explicit lyrics or not. However, it is decided to keep this variable and use their means as the numerical values. For instance, if a value for an “explicit” variable is False, then this is replaced with 0.329.

Secondly, the “time_signature” variable is analyzed again using boxplot and the average of popularity values of each categorical value. The variable “time_signature” shows is a

notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7 indicating time signatures of 3/4, to 7/4.

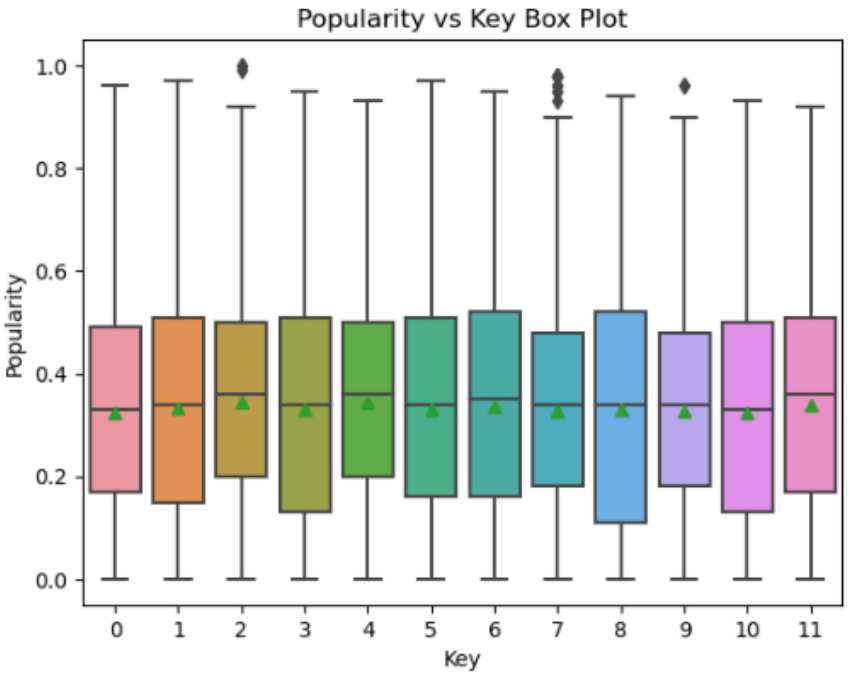



```
time_signature
0    0.369755
1    0.290740
3    0.296924
4    0.336725
5    0.287782
Name: popularity, dtype: float64
```

Figure 5

Please consider that in the x-axis the value “0” corresponds to 3/4, and the value “5” corresponds to 7/4. It is seen that the chance of a track being more popular is the highest when time_signature value is 3/7 with mean: 0.369755. The values are in a compact range however the difference between them cannot be avoidable and so that it is decided to include this categorical variable by changing their values accordingly with the values of the average of the popularity variable, For instance, if a track has 3/4 time signature value, then it is replaced with 0.369755 and the same procedure is applied for the remaining values.

Thirdly, the “Key” variable is analyzed again using boxplot and the average of popularity values of each categorical value. The values of this categorical variable are integers that mapS to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1

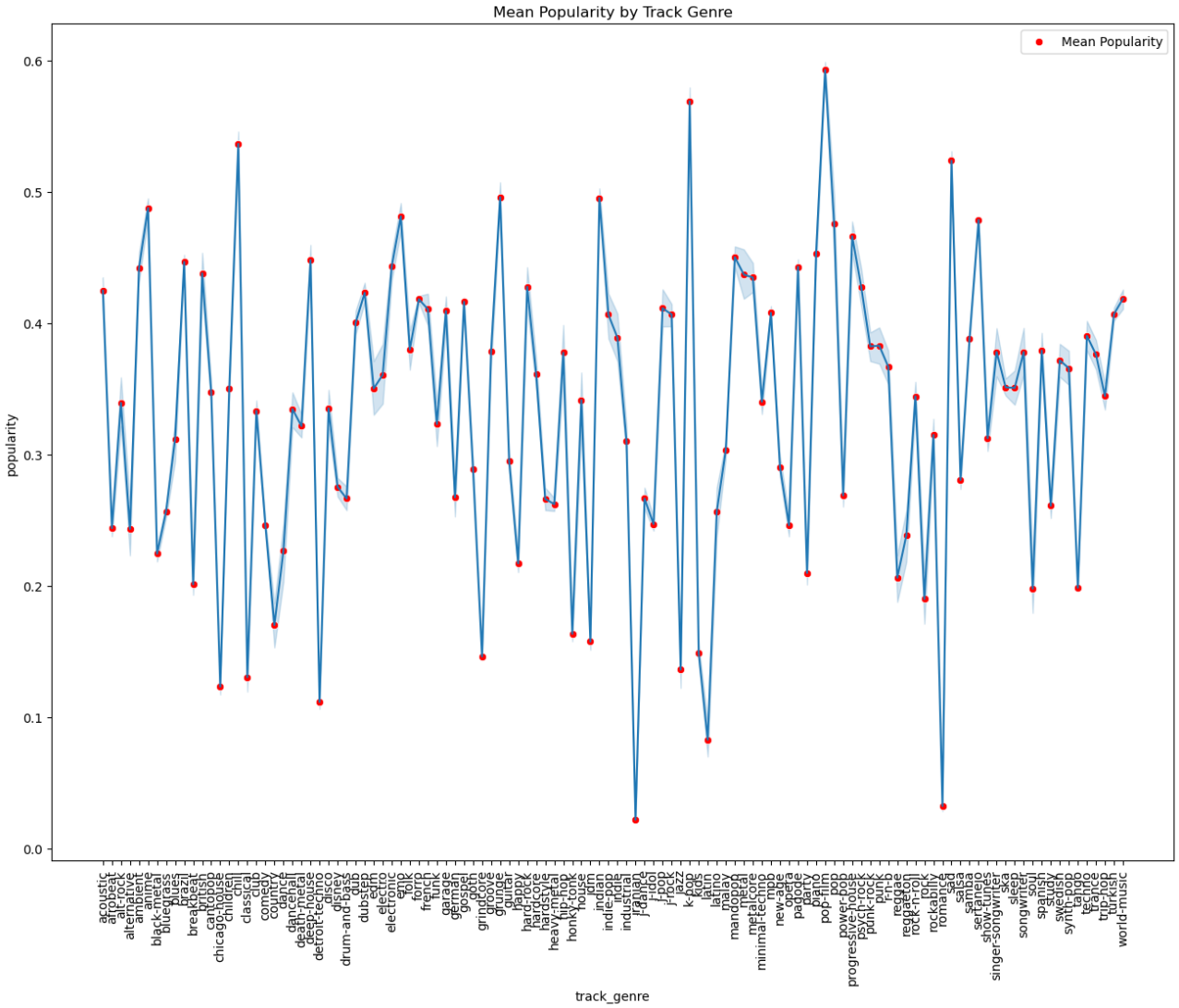



```
key
0     0.325199
1     0.332002
2     0.343776
3     0.330317
4     0.346437
5     0.329309
6     0.335855
7     0.325796
8     0.329587
9     0.328236
10    0.324614
11    0.338876
Name: popularity, dtype: float64
```

Figure 6

In the x-axis of the graph, the value “0” corresponds to -1 and the rest is so on. GAs it can be clearly seen from the graph provided and from the values provided, the averages in terms of popularity for each of the “key” values are nearly the same. Therefore, since the impact of this variable will be very slight in order to predict the popularity scores, this variable is avoided to be included in the combined dataset.(Numerical & Categorical)

Lastly, the most important and impactful variable on the popularity of a track which is track_genre is analyzed. In the dataset provided there were 114 different genres which also shows that the dataset is also a very detailed dataset.




```
count    114.000000
mean       0.332385
std        0.112948
min        0.022100
25%        0.257840
50%        0.348855
75%        0.411240
max        0.592830
Name: popularity, dtype: float64
```

Figure 7

In Figure 7, it can be seen that the popularity values are not concentrated and there are various average popularity values for the genres. Therefore, as stated before it is proven that genre is one of the most important variables in order to determine the popularity of a track. The most popular genre is “party” (0.59) and the least popular is “iranian” (0.02). The same feature encoding method is also applied to the track_genre variable and the genres are replaced by their average popularity values.

Data Preprocessing for Categorical Variables

As also applied for the numerical features, several data preprocessing steps are also applied to the new numerical versions of categorical variables. Since it is decided to proceed with the following 3 categorical variables: “track_genre”, “explicit” and “time_signature”, firstly the values of these variables are normalized even though they were between 0-1. The reason for applying normalization for these variables is to avoid inconsistency because the normalization technique was also applied to numerical variables. Then, the values of these mentioned variables are converted into 64 bit and gradient clipping is applied to these variables.

4. Experiment Results and Analysis for Combined dataset

Implementation with using different parameters

After combining the datasets with originally numerical features and the numerical variables(previously categorical), a correlation matrix is printed out to monitor the effects of the variables to population variable and also the relations between the variables.

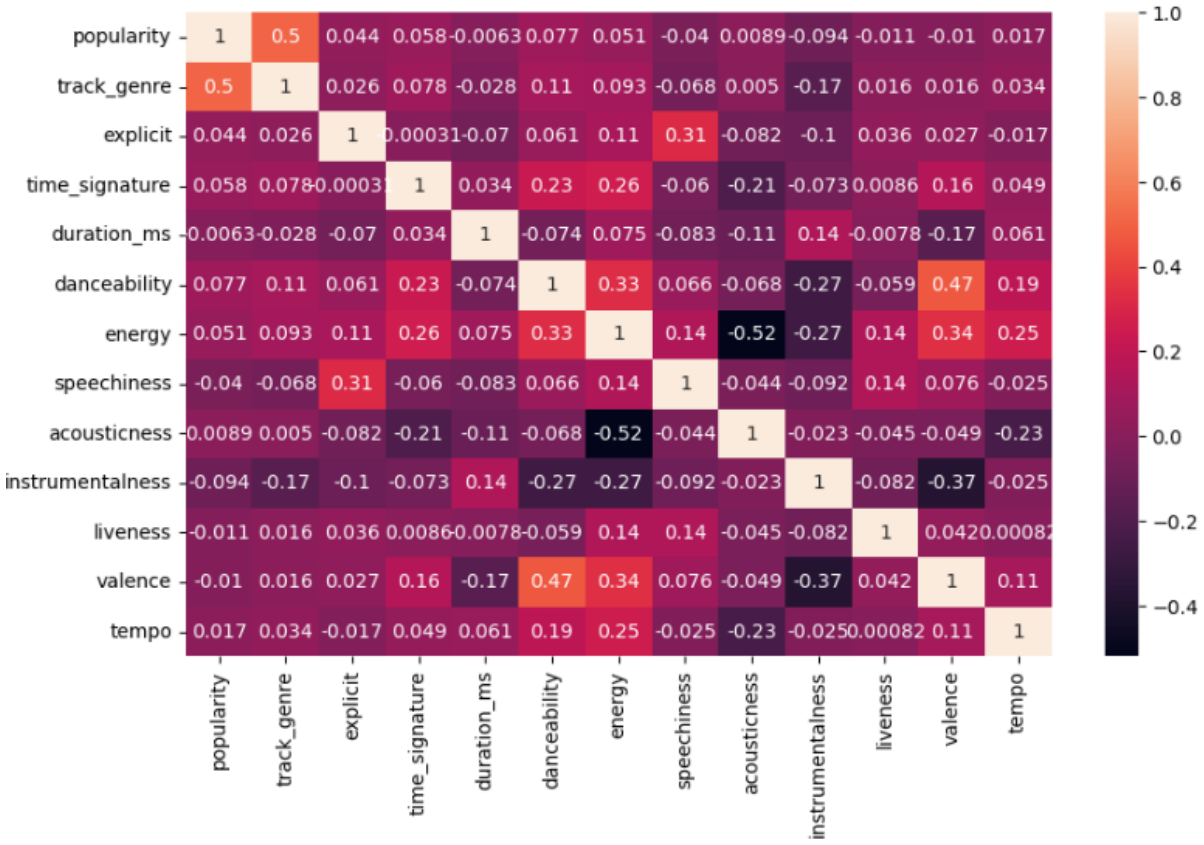


Figure 8

As it can be seen from Figure 8, the most correlated variable with popularity is track_genre. As mentioned in the section3, it is now proven that the most important variable in order to predict the popularity scores is the genre of track. It also can be said that instrumentalism is the variable which is the most negatively correlated variable with the popularity variable however the magnitude is not big.Furthermore, we see that there is also a strong correlation between danceability and valence which also makes sense since valence is a variable that shows the music positivity and if the music is positive, people are more likely to dance. Figure 8 also provides us with other details about the dataset. For example, one of the other 2 most correlated variables are energy and valence and also energy vs danceability is also correlated. On the other hand, the correlation between energy and acousticness is negative, which also makes sense since when a person listens to acoustic music he/she feels more emotional and doesn't want to dance.

Results Presentation using performance metrics

Combined

Iterations=1000

	Learning rate = 0,001			Learning rate = 0,01			Learning rate = 0,1		
	MSE	MAE	**Sum**	MSE	MAE	**Sum**	MSE	MAE	**Sum**
α = 0,1	0,048913	0,185178	0,234091	0,044218	0,173211	0,217429	0,037387	0,144982	0,182369
α = 10	0,048914	0,18518	0,234094	0,044222	0,173222	0,217444	0,037403	0,145146	0,182549
α = 100	0,04892	0,185201	0,234121	0,044257	0,173328	0,217585	0,037569	0,146655	0,184224
		MIN	0,234091		MIN	0,217429		MIN	0,182369

Iterations=5000

	Learning rate = 0,001			Learning rate = 0,01			Learning rate = 0,1		
	MSE	MAE	**Sum**	MSE	MAE	**Sum**	MSE	MAE	**Sum**
α = 0,1	0,046119	0,178395	0,224514	0,038405	0,151888	0,190293	0,037188	0,141919	0,179107
α = 10	0,046120	0,178399	0,224519	0,038425	0,151997	0,190422	0,037190	0,142130	0,179320
α = 100	0,046131	0,178434	0,224565	0,03861	0,152973	0,191583	0,037290	0,144104	0,181394
		MIN	0,224514		MIN	0,190293		MIN	0,179107

Iterations=10000

	Learning rate = 0,001			Learning rate = 0,01			Learning rate = 0,1		
	MSE	MAE	**Sum**	MSE	MAE	**Sum**	MSE	MAE	**Sum**
α = 0,1	0,044218	0,173212	0,217430	0,037388	0,144988	0,182376	0,037192	0,141891	0,179083
α = 10	0,044222	0,173224	0,217446	0,037403	0,145152	0,182555	0,037193	0,142104	0,179297
α = 100	0,044257	0,17333	0,217587	0,03757	0,146661	0,184231	0,037290	0,144094	0,181384
		MIN	0,217430		MIN	0,182376		MIN	0,179083

Figure 9

As a general overview, it can be said that the best alpha and learning rate combinations for each number iteration is alpha= 0.1 and learning rate = 0.1. The best values are : 0.179083, 0.179107, 0.182369. It can be said that there are tiny differences between these 3 best

values but when the number of iterations are increased, the error rate decreases. Furthermore, among these three best results for numbers of iterations the best of the best result is obtained through combining alpha = 0.1, learning parameter = 0.1 and number of iterations = 10000 which is 0.179083. The obtained best results's combination for the combined dataset (numerical + categorical) is exactly the same as the experiment conducted using only numerical features. In the experiment that was conducted using the combined dataset we saw that when the number of iterations increases we start to get better results, there is a decreasing trend for the overall error when we increase the number of iterations. Similarly to the first experiment conducted with only using the numerical features also where we see that when the learning rate increases the sum of errors decreases meaning that when the learning rate increases the algorithm learns better and obtains better results. On the other hand, generally we see that when the alpha value increases the algorithm performs slightly worse, increasing the sum of errors.

5.Discussion:

Comparison & Summary

It was quite surprising that for both the datasets the best results are achieved through using alpha = 0.1 and learning parameter = 0.1. However the only difference was for numerical dataset the best result is obtained by setting the number of iterations equal to 5000, but in the combined dataset it is obtained by setting the number of iterations equal to 10000. Comparing the best results obtained with using each dataset it can be stated that for near all of the parameter combinations, combined dataset overperformed the numerical dataset, which was quite expectable since by adding the categorical variables in the training process it is normally expected that the model's performance will increase. By including the categorical variables to our training process, the performance of the model is increased by 27%.

For both the dataset, we saw that when the learning rate increases with other parameters remaining the same, the model's performance increases and similarly we saw that when the alpha parameter value increases, the model's performance slightly decreases. On the other hand, for the number of iterations, it can be said that setting it to 1000 seems not enough because for both cases among the best scores, the scores with number of iterations 1000 performance was the worst.

6.Appendix

Dataset variables

- **track_id**: The Spotify ID for the track
- **artists**: The artists' names who performed the track. If there is more than one artist, they are separated by a `;`
- **album_name**: The album name in which the track appears
- **track_name**: Name of the track
- **popularity**: **The popularity of a track is a value between 0 and 100, with 100 being the most popular**. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity is derived mathematically from track popularity.
- **duration_ms**: The track length in milliseconds
- **explicit**: Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown)
- **danceability**: Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable
- **energy**: Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale
- **key**: The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. `0 = C`, `1 = C♯/D♭`, `2 = D`, and so on. If no key was detected, the value is -1
- **loudness**: The overall loudness of a track in decibels (dB)
- **mode**: Mode indicates 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**: Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such

cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks

- **acousticness**: A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic
- **instrumentalness**: Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content
- **liveness**: Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live
- **valence**: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry)
- **tempo**: The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration
- **time_signature**: An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7 indicating time signatures of `3/4`, to `7/4`.
- **track_genre**: The genre in which the track belongs

Data source

https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset

Python Code link

https://github.com/emreiskender1/ML-project-Ridgre-Regression-with-K-fold-CV