Supervised Learning with Regression on Song Popularity Set

Kevin Luu

Northwestern University: CIS 435 Dr. Sunil Kakade

Table of Content

1. Business Problem

A record label company is trying to determine how to make its next song a hit. In the past, it was accomplished mainly by following trends and understanding what themes and genres were popular. However, due to the sheer number of genres and competing agencies in today's music industry, a song expert can no longer predict what will be a hit on a consistent basis. There are simply too many variables to rely on gut instinct. Data analytics supports decision-makers by providing statistical evidence that will let them know what to focus on and what to remove.

Several companies in the music industry are attempting to embrace data analytics by determining what characteristics of a song contribute to its popularity and what characteristics hurt its popularity. We, as the data experts, were provided an excel file that includes the history of several different songs. The file contains information about the popularity of the song as well as its qualities, such as song duration, acoustics, danceability, energy, and so on.

2. Machine Learning Application

The music industry is one of many industries that benefit from machine learning applications. As previously stated, more and more record labels are competing to create the next best hit, and the industry is now using data analytics to grow and expand by curating music production to their overall market as well as optimizing sales and advertisement. World tours are no longer enough to gain traction to stay relevant for record labels. The rise of streaming services has also reduced the importance of cd sales and concert merchandise sales.

Record labels have shifted priority to ad revenue, online merchandise, and brand deals. This has resulted in a greater need for record labels to understand who they are targeting and how to target them effectively to improve their revenue streams. Google Ads/YouTube can help determine the target demographic willing to partake in their events. With the gathered data, teams can accurately and automatically target ads and or deals to individuals to get them to become a fan.

2.1 Recommendation System: Collaborative Filtering

Spotify, the music streaming service, is a major data-driven music juggernaut. Like Netflix, it has built a recommendation engine to help curate taste profiles for its listeners using machine learning, specifically “Collaborative filtering,” a recommendation system aimed at finding similarities between data. (Ajao,2022) They accomplished this by building a data warehouse of customer and music data. The system will learn what type of music the person listens to and recommend similar songs to them. Depending on if they continue to listen to the song, like the song, or add it to a playlist, it further increases the reliability of the system.

2.2 Reinforcement Learning: Contextual Bandit

Spotify will use machine learning to further map out the listener’s full taste profile as well as get more buy-in time so that users spend more time on their platform. Reinforcement learning is the process of testing actions and recording results, and learning if someone likes or dislikes the action. The overall goal for the model is to learn what the person likes and test new music to see if they are willing to take it. Contextual Multi-Arm Bandit is used to find the best results while considering previous observations meaning that if you had preferred rock music two years ago and have not been on Spotify for a while, it will still recommend rock but also possibly recommend new songs or genres. As some have mentioned, you could call this better A/B testing (Surmenok, 2017).

2.3 Deep Learning: Audio Engineering

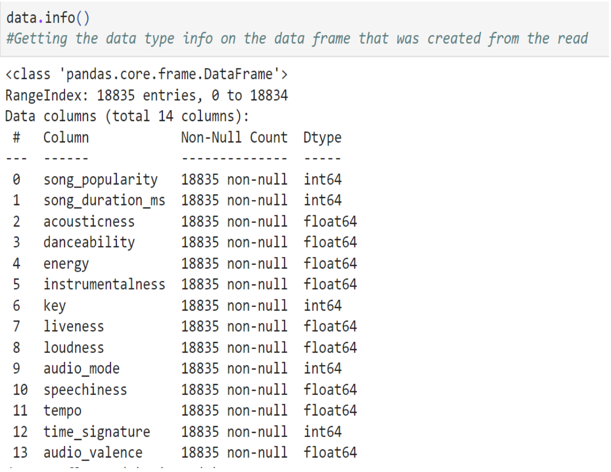
The audio quality for music has been affected by machine learning there are currently commercial applications where recording studios and post-production environments are already utilizing tools to identify and remove background noise and improve audio quality. There is currently research toward creating high-quality sounds to meet the demands of the current era at scale(Opentrack, 2022). Even here at the university, TEAMuP received $1.8 million to continue research in this space (McCormick, 2022).

2.4 Machine Learning: Music video generating

There are even tools that assist in creating videos for music, such as Revolver.AI, which will listen to songs and generate a video that will attempt to match the mood of the song. This application has already had commercial success for influential groups such as Imagine Dragons. AI has also been capable of creating music on its own while the system is not as widely successful, there are possibilities for it to be in the future.

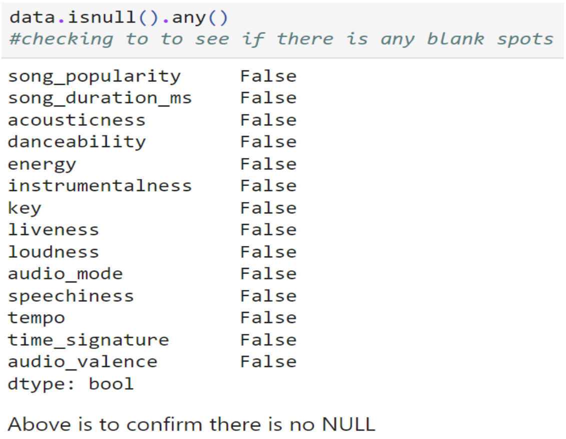
3. Data Preprocessing

Before we perform the supervised learning models, we will inspect and clean the data in this file to ensure that it is usable for transformation with Juypter notes. After we import the excel/CSV, we will start by running “data.info()” to understand what we are working with, followed by “data. describe()” to get an overview of the data’s general breakdown. Once that is all complete, I will confirm if there is any null cell by running “data.IsNull().any()”. From my initial data inspection, this data set seems mostly clean, so we can proceed.



```
data.info()
#Getting the data type info on the data frame that was created from the read

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18835 entries, 0 to 18834
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   song_popularity   18835 non-null  int64
 1   song_duration_ms  18835 non-null  int64
 2   acousticness      18835 non-null  float64
 3   danceability      18835 non-null  float64
 4   energy            18835 non-null  float64
 5   instrumentalness  18835 non-null  float64
 6   key               18835 non-null  int64
 7   liveness          18835 non-null  float64
 8   loudness          18835 non-null  float64
 9   audio_mode        18835 non-null  int64
 10  speechiness       18835 non-null  float64
 11  tempo             18835 non-null  float64
 12  time_signature    18835 non-null  int64
 13  audio_valence     18835 non-null  float64
```



```
data.isnull().any()
#checking to to see if there is any blank spots

song_popularity     False
song_duration_ms    False
acousticness        False
danceability        False
energy              False
instrumentalness    False
key                 False
liveness            False
loudness            False
audio_mode          False
speechiness         False
tempo               False
time_signature      False
audio_valence       False
dtype: bool
```

Above is to confirm there is no NULL

Sweetviz analysis was ran to have a second viewpoint of what data was doing. It still looks like the data set is overall clean but there seem to be a few questionable columns that do not matter in music development. For each of the three models, I defaulted to using 50/50 on splitting tests and training as I felt I had a large amount of data. Looking at other journal articles, I see that 70/30 is commonly used but looking at the result, I don't believe that will be necessary. If needed, we could rerun at 70/30.

4. Explaining Metrics

This will complete the preliminary data cleanup, but there are a few things that I did not include in my model. When creating my arrays for the testing, I specifically did not include a few characteristics/columns from the tests as I felt that they did not have any significant importance in impacting popularity. These variables were instrumentals, audio_mode, and time_signature. Due to the fact we are targeting listeners' preferences and some of the data were simply represented as "on or off with 0 and 1s," In the final selection process, we will be only including the following.

4.1 Explaining Variables

Sound_duration_ms	How long is the song
Acousticness	How much is the acoustic
Danceability	How danceable.
Energy	The energy level
Key	What key it is higher or lower
Liveness	How lively the music is.
loudness	Level of Loudness
speechiness	How much speech is involved
tempo	Speed/tempo
Audio_Valence	Mood of the song so higher means happier usually while lower is sad.

5. Machine Learning Algorithms

Once all the preliminary cleaning has been completed, we can run supervised learning with multiple different regression models. Linear, Ridge, and Lasso regression models were employed. Once the models have been developed, we can determine which variable has the most significant positive impact and the most negatively impacting.

For linear model regression, the advantage of using this model is that it is the most linear and the least complex model to understand it works with the multi-variable / characteristics precisely like what we have here in our song popularity file. One of the reasons I chose this method was that one of the constraints that limit this model is that it relies on that the variables being heavily independent, which in our situation is the same. The con to this model is that it is very quickly ruined by random errors, which can occur in data collection, however, because this data is not that large, and we have already cleaned it; I don't expect this to be an issue (Molnar, 2022).

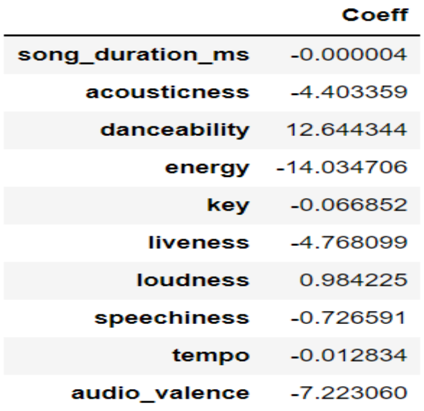
The Ridge model aims to remove multi-variable correlation. Ridge does what the linear model is flawed at where if the data is colinear or multi-colinear, it will attempt to remove that and or correct those issues by adding biases to the model to correct it. I chose this model as a secondary to go alongside the linear model for comparative reasons. if the ending results were greatly different, I could make the assumption either the data or model were flawed. Overall the Ridge model aims at correcting errors if there are no multi-collinearity issues. To begin with, the Ridge model will result similarly to the Linear model, which we will see in the final results (Corporate Finance Institute. 2023).

Lasso regression model can set coefficient variables automatically, meaning that it can set coefficients to zero if it deems it not necessary, similar to removing it from the dataset as I did in the data preparation phase. Like the other models selected, this model prefers low to no collinearity, or it would fail. While automatically setting coefficients to zero to improve the visibility of critical variables could be a benefit, but at the same can be a con. Automatically selecting coefficients could make results skewed. If a result ends up being near zero already, it might all together get set to zero, meaning you may not notice that characteristic had any impact. (Ellis, 2022)

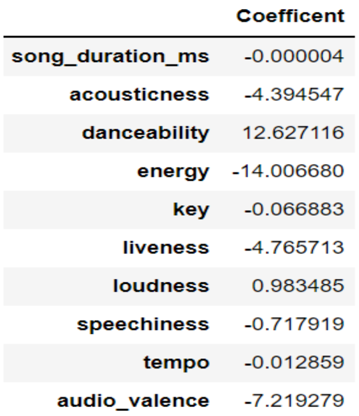
Overall each regression model has its pros and cons. Still, if we look at it collectively, we can build off each of them and understand what the results mean, and have higher confidence in recommending the company their next steps and quantifying the results.

6 Interpreting Results

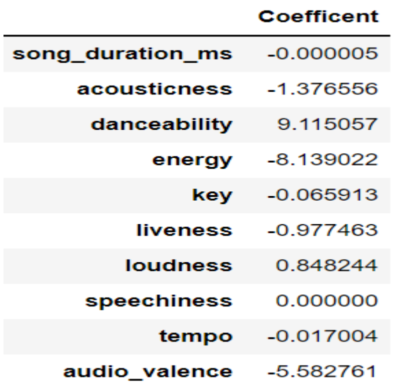
Linear Model	Ridge Model	Lasso Model



	Coeff
song_duration_ms	-0.000004
acousticness	-4.403359
danceability	12.644344
energy	-14.034706
key	-0.066852
liveness	-4.768099
loudness	0.984225
speechiness	-0.726591
tempo	-0.012834
audio_valence	-7.223060



	Coefficent
song_duration_ms	-0.000004
acousticness	-4.394547
danceability	12.627116
energy	-14.006680
key	-0.066883
liveness	-4.765713
loudness	0.983485
speechiness	-0.717919
tempo	-0.012859
audio_valence	-7.219279



	Coefficent
song_duration_ms	-0.000005
acousticness	-1.376556
danceability	9.115057
energy	-8.139022
key	-0.065913
liveness	-0.977463
loudness	0.848244
speechiness	0.000000
tempo	-0.017004
audio_valence	-5.582761

As you can see from the 3 model results, we can see that there is a pattern of positive and negative coefficients. Danceability across all three models results in a high favorable rating of song popularity, while high energy results in lower song popularity.

Because all three tests are generally telling the same story, I am confident in recommending Record labels to produce songs that are highly danceable and avoid high energy. Taking the average results, for about every score in with Danceability, we get approximately 11.46 more song popularity scores. In contrast, with each energy, we get a -12.15 in popularity for the song.

7. Performance Evaluation

After the model was created, we needed to perform several performance evaluations. Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error was used in the evaluation. Here are my results.

	Linear Model	Ridge Model	Lasso
Mean Absolute Error	0.5112270109317519	0.4972566007489901	0.4687082109415523
Mean Squared Error	0.4121268840503631	0.3990403631753102	0.3826728813450098
Root Mean Squared Error	0.6419710928463703	0.6316964169403767	0.6186055943369813

- Mean Absolute Error:
  - MAE is average of absolute errors for a group of predictions and observations as a measurement of the magnitude of errors for the entire group. MAE is not great with datasets with outliers (C3, 2022).

- Mean Squared Error
  - MSE assesses the average squared difference between the observed and predicted values. When a model has no error, the MSE equals zero.MSE is also effective with outliers working well alongside MAE (Gupta, 2022).

- Root Mean Squared Error
  - RMSE is a negatively oriented score representing absolute fit. It is calculated by square rooting MSE to determine the Standard deviation (Gunjal, 2021).

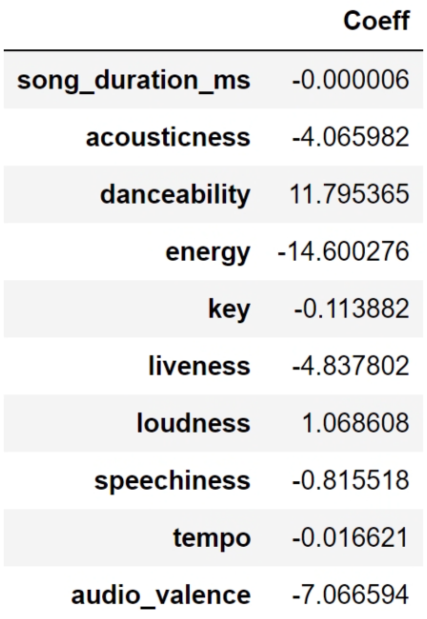
Looking at the tables, I now understand that maybe doing a 50/50 test split might have been a wrong choice just looking at the root mean squared error, which is a negatively oriented metric, I seem to have quite a high RMSE which is an issue with fitting in my data set. I could simply change that with train_test_split, setting it to 70%, and then rerun my performance metric to see if that made a positive or negative on the results.

Overall my performance metrics were low across the board, not including the lasso model, as that is negatively oriented. I could have done better with setting better parameters, I will attempt to recreate the model with 70/30. At least with a similar story from the coefficient results, we can tell which one is positively and negatively affected but, in terms of how much that value might be up for review.

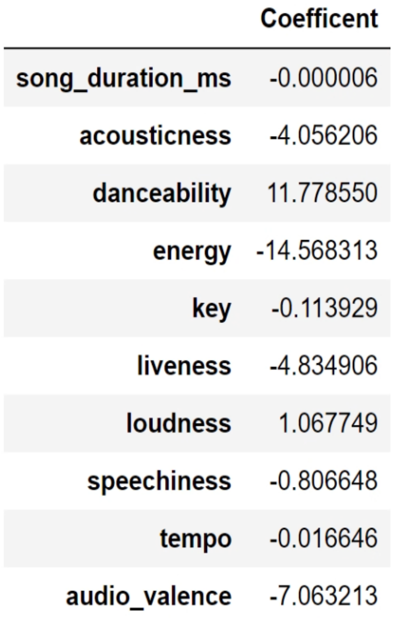
8. New train and test ratio

I went ahead and attempted to test a 70/30 split by using train_test_split(x,y, train_size=0.7, test_size=0.3) my results for coefficients were similar, but my performance metrics were vastly different.

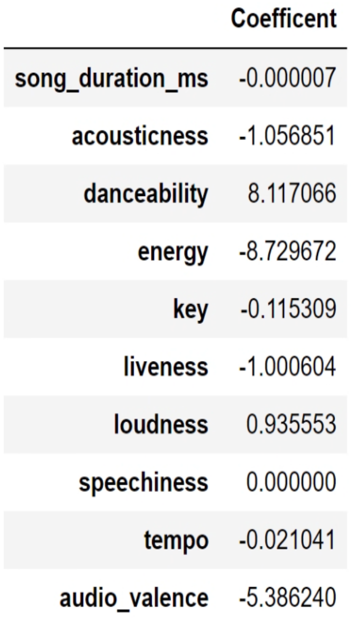
Linear Model	Ridge Model	Lasso Model



	Coeff
song_duration_ms	-0.000006
acousticness	-4.065982
danceability	11.795365
energy	-14.600276
key	-0.113882
liveness	-4.837802
loudness	1.068608
speechiness	-0.815518
tempo	-0.016621
audio_valence	-7.066594



	Coefficent
song_duration_ms	-0.000006
acousticness	-4.056206
danceability	11.778550
energy	-14.568313
key	-0.113929
liveness	-4.834906
loudness	1.067749
speechiness	-0.806648
tempo	-0.016646
audio_valence	-7.063213



	Coefficent
song_duration_ms	-0.000007
acousticness	-1.056851
danceability	8.117066
energy	-8.729672
key	-0.115309
liveness	-1.000604
loudness	0.935553
speechiness	0.000000
tempo	-0.021041
audio_valence	-5.386240

The overall story is the same Danceability is essential, and energy should be avoided let us look at the performance metrics now.

	Linear Model	Ridge Model	Lasso Model
Mean Absolute Error	17.12459751149626	17.12459030363128	17.1515520350878
Mean Squared Error	457.4982292502639	457.5013747231986	459.9288672354815 6
Root Mean Squared Error	21.38920824271585 6	21.38928177202775	21.44595223429077 5

Now this is where my issue lies after my change for the test and train size my performance evaluation have significantly changed. I am not entirely sure if, what I have is expected or if something is wrong, and when I removed my “ train_size=0.7, test_size=0.3,” my results are not reverting to my original results. I am unsure if this is due to my system caching results or if the random state is static. I am also now questioning if leaving train_test_split on default is not actually 50/50.

9. Recommended Steps

After completing our data analysis using supervised learning, specifically with Linear, Ridge, and Lasso models, we have discovered that danceability is a characteristic for songs to focus on, and energy is one to be avoided. Record labels should shift their focus and continue to collect data from future songs and make adjustments as needed. I would also like to recommend that while the metric shows danceability should be focused on now, however, one day it could change. Record labels should proactively track if a particular type of song becomes more favorable to continue to stay relevant.

REFERENCES

Davenport, T. H., & Harris, J. G. (2017). Competing on Analytics with internal processes. In Competing on analytics, the new science of winning. essay, Harvard Business School Press.

Ellis, C., & About The Author Christina Ellis I am a practicing Senior Data Scientist with a master's degree in statistics. Between academic research experience and industry experience. (2022, May 30). When to use Lasso. Crunching the Data. Retrieved January 16, 2023, from https://crunchingthedata.com/when-to-use-lasso/#:~:text=Advantages%20of%20LASSO%20regression&text=The%20main%20advantage%20of%20a,be%20included%20on%20its%20own.

How data science is revolutionizing the music industry. Opentracker. (2022, May 25). Retrieved January 16, 2023, from https://www.opentracker.net/article/data-science-music

Molnar, C. (2022, December 14). Interpretable machine learning. 5.1 Linear Regression. Retrieved January 16, 2023, from https://christophm.github.io/interpretable-ml-book/limo.html

Ridge. Corporate Finance Institute. (2023, January 9). Retrieved January 16, 2023, from https://corporatefinanceinstitute.com/resources/data-science/ridge/#:~:text=Ridge%20regression%20aims%20at%20reducing,the%20reliability%20of%20the%20estimates.

Thecleverprogrammer. (2022, April 11). Spotify recommendation system with Machine Learning. thecleverprogrammer. Retrieved January 16, 2023, from https://thecleverprogrammer.com/2021/03/03/spotify-recommendation-system-with-machine-learning/

Ajao, E. (2022, April 4). Spotify personalizes audio experiences with Machine Learning: TechTarget. Enterprise AI. Retrieved January 21, 2023, from https://www.techtarget.com/searchenterpriseai/feature/Spotify-personalizes-audio-experiences-with-machine-learning

Connecting Deep Learning developers with sound artists. Northwestern Engineering. (n.d.). Retrieved January 21, 2023, from https://www.mccormick.northwestern.edu/computer-science/news-events/news/articles/2022/connecting-deep-learning-developers-with-sound-artists.html

Gunjal, S. (n.d.). What is root mean square error (RMSE): Data Science and Machine Learning. Kaggle. Retrieved January 21, 2023, from https://www.kaggle.com/general/215997

Gupta, A. (2022, September 27). Mean squared error: Overview, examples, concepts, and more: Simplilearn. Simplilearn.com. Retrieved January 21, 2023, from https://www.simplilearn.com/tutorials/statistics-tutorial/mean-squared-error

Konieczka, J. (2022, August 26). The increasingly influential role of AI in the music industry. Arek Skuza. Retrieved January 21, 2023, from https://arekskuza.com/the-innovation-blog/the-increasingly-influential-role-of-ai-in-the-music-industry/

Mean absolute error. C3 AI. (2022, March 31). Retrieved January 21, 2023, from https://c3.ai/glossary/data-science/mean-absolute-error/#:~:text=What%20is%20Mean%20Absolute%20Error,true%20value%20of%20that%20observation.

Surmenok, P. (2017, October 18). Contextual bandits and reinforcement learning. Medium. Retrieved January 21, 2023, from https://towardsdatascience.com/contextual-bandits-and-reinforcement-learning-6bdfeaece72a