Final Project

Predicting the popularity of songs on Spotify

Ethan DeMott, Rao Fu, Alexis Wheeler, Krithi Sachithanand

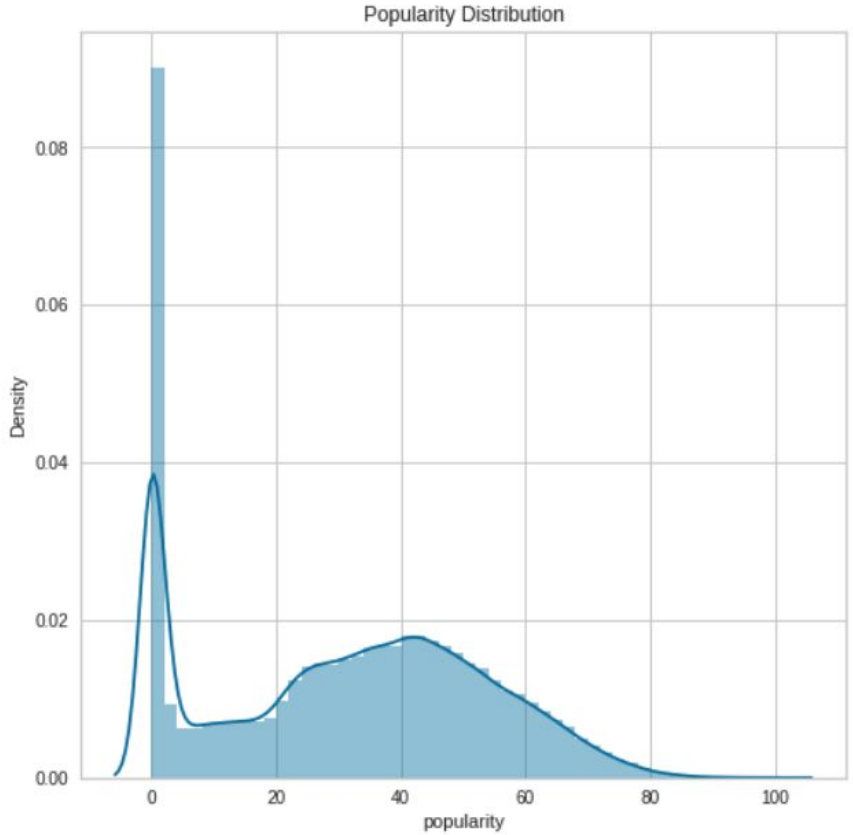
Team: Treble Clef

Background

- **What we are trying to achieve:**
  - Predict the popularity of a song (based on Spotify data)
  - The Spotify Popularity Index is a 0-to-100 score that ranks how popular an artist is relative to other artists on Spotify. As numbers grow, you'll get placed in more editorial playlists and increase your reach on algorithmic playlists and recommendations. Some say the magic number is 50! (Loudlab).
- Dataset being used is from Kaggle,

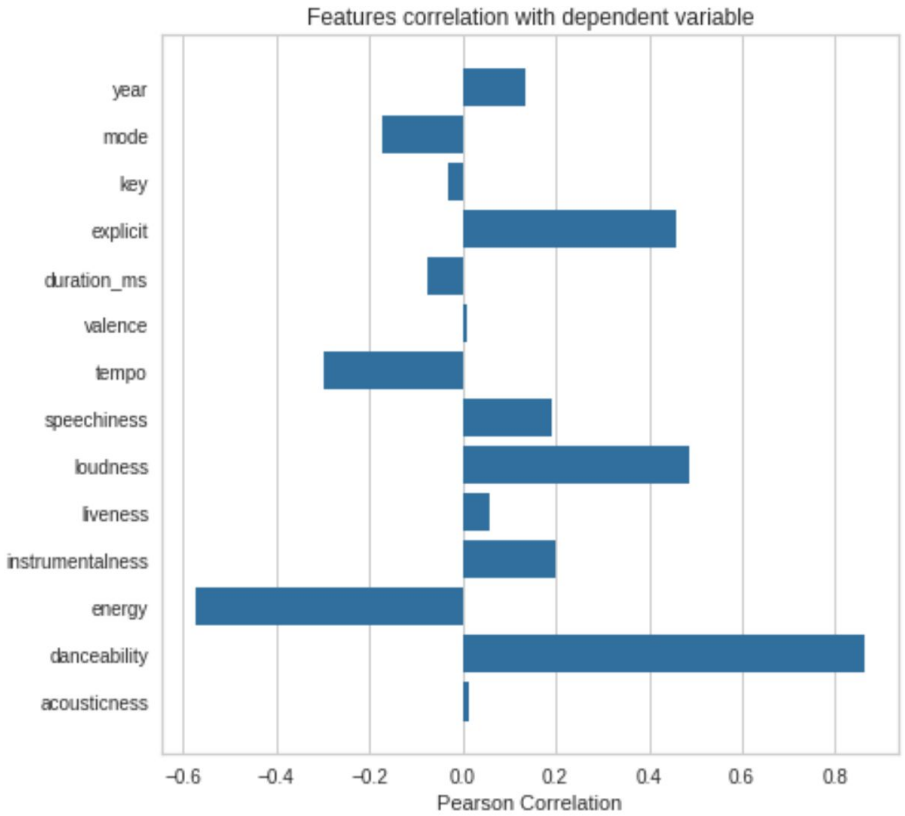
	valence	year	acousticness	danceability	duration_ms	energy	explicit	instrumentalness	key	liveness	loudness	mode	popularity	speechiness	tempo
0	0.0594	1921	0.982	0.279	831667	0.211	0	0.878000	10	0.665	-20.096	1	4	0.0366	80.954
1	0.9630	1921	0.732	0.819	180533	0.341	0	0.000000	7	0.160	-12.441	1	5	0.4150	60.936
2	0.0394	1921	0.961	0.328	500062	0.166	0	0.913000	3	0.101	-14.850	1	5	0.0339	110.339
3	0.1650	1921	0.967	0.275	210000	0.309	0	0.000028	5	0.381	-9.316	1	3	0.0354	100.109
4	0.2530	1921	0.957	0.418	166693	0.193	0	0.000002	3	0.229	-10.096	1	2	0.0380	101.665

Exploring the data: Distribution



- Displays a right skew and that having a popular song is hard/uncommon

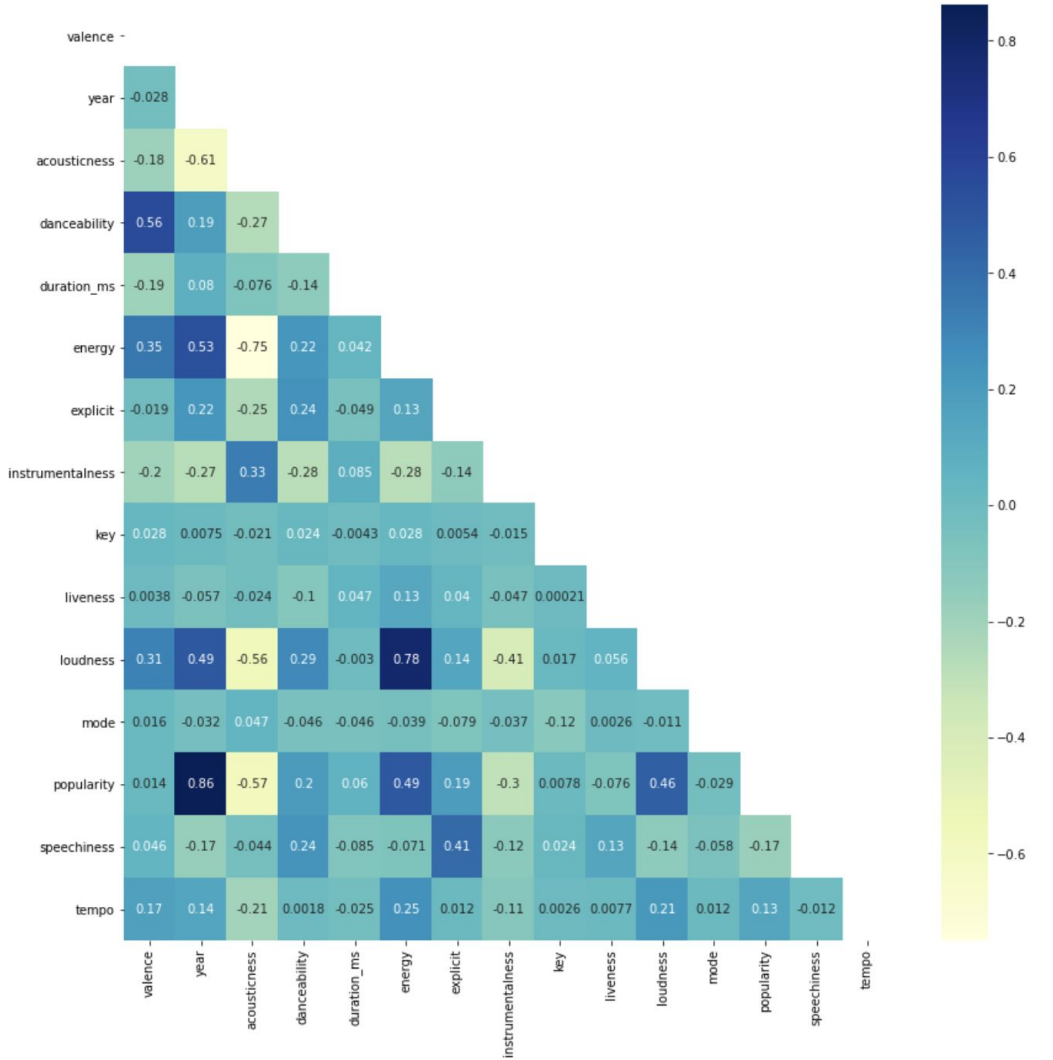
Exploring the data: Correlation



- We look at the variables independently in comparison to popularity
- Our Pearson Correlation shows our coefficient that indicates the strength of the relationship between popularity and each variable.

Exploring the Data: Heat Map Correlation

- Shows the correlation between the features of the dataset
  - Loudness and energy are strongly correlated as well
  - Popularity and year are strongly correlated



Methodology

1. After creating a couple of visualizations, we wanted to look at the multiple regression of our data.
2. We were looking at a modified dataset with the columns 'id','name','artists','release_date' dropped as we wanted to only work with numeric values
3. We used the train_test_split from scikit_learn to create our training and testing data
4. We used the OLS() method from statsmodels.api and displayed the summary of our process
5. We were looking for the summary relative to popularity

Methodology

OLS Regression Results

Dep. Variable:	y	R-squared:	0.754
Model:	OLS	Adj. R-squared:	0.754
Method:	Least Squares	F-statistic:	2.800e+04
Date:	Sun, 27 Nov 2022	Prob (F-statistic):	0.00
Time:	16:19:19	Log-Likelihood:	-4.8661e+05
No. Observations:	127989	AIC:	9.733e+05
Df Residuals:	127974	BIC:	9.734e+05
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1287.6494	3.356	-383.648	0.000	-1294.228	-1281.071
x1	0.7350	0.165	4.460	0.000	0.412	1.058
x2	0.6690	0.002	397.775	0.000	0.666	0.672
x3	-4.1949	0.140	-29.860	0.000	-4.470	-3.920
x4	2.6165	0.241	10.847	0.000	2.144	3.089
x5	-3.533e-07	2.54e-07	-1.391	0.164	-8.51e-07	1.44e-07
x6	-1.7454	0.252	-6.922	0.000	-2.240	-1.251
x7	0.8815	0.129	6.826	0.000	0.628	1.135
x8	-4.1475	0.113	-36.759	0.000	-4.369	-3.926
x9	-0.0058	0.009	-0.674	0.501	-0.023	0.011
x10	-2.9500	0.182	-16.245	0.000	-3.306	-2.594
x11	0.0147	0.010	1.545	0.122	-0.004	0.033
x12	-0.2351	0.068	-3.476	0.001	-0.368	-0.103
x13	-7.6393	0.231	-33.037	0.000	-8.092	-7.186
x14	0.0011	0.001	1.062	0.288	-0.001	0.003

Omnibus:	13542.058	Durbin-Watson:	2.010
Prob(Omnibus):	0.000	Jarque-Bera (JB):	115749.682
Skew:	0.119	Prob(JB):	0.00
Kurtosis:	7.653	Cond. No.	2.90e+07

- Our R-squared value is 0.75
- We have variables with high p-values meaning that those variables are not statistically significant
  - Duration
  - Key
  - Loudness
  - Tempo

RESULTS

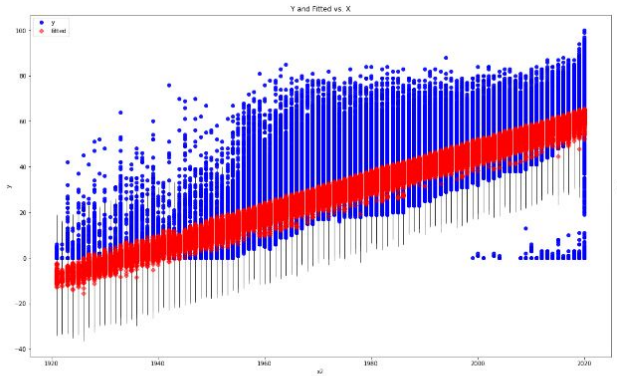
- By extrapolating our y_hat from our .predict(), we tried to compare our expected popularity value by our actual popularity value.

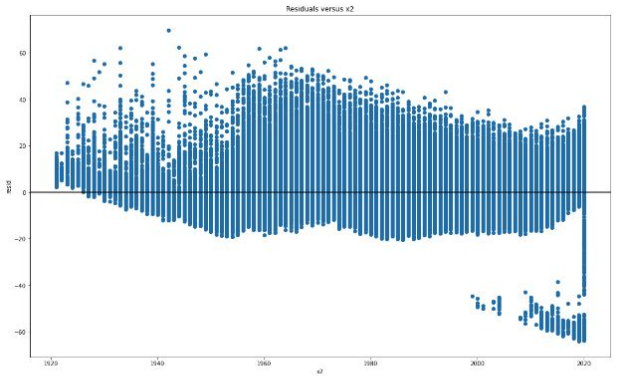
In [5]:
```
y_hat=result.predict(sm.add_constant(X_test))
print(y_hat)
print(y_test)
```

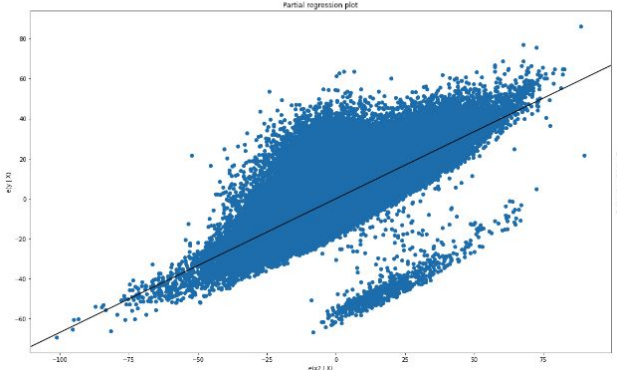
```
[13.31896453 10.54236383 10.10545305 ... 34.08111276  5.67003441
 -1.04891963]
[ 8  1  4 ... 52  0  1]
```

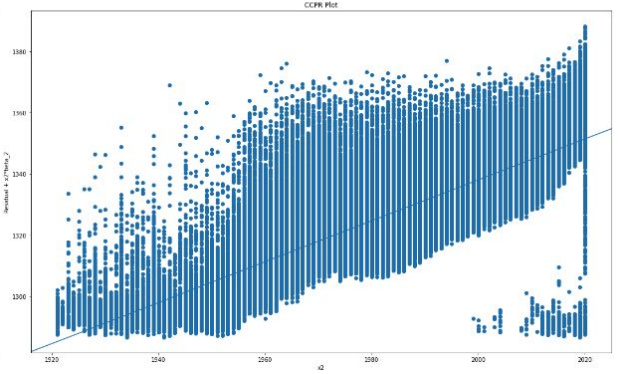
- Our first value is our popularity, which leaves our prediction at 13.3 and our actual at 8.

Results









- Examining the fit of the data using ‘year’ as the independent variable
- Songs have gotten more popular over time
- We see issues with our linearity especially prominent in our partial regression plot
    - Around the center point of zero, the expected values are severely over or undershot

Results: Expanding the Model

- Dropping insignificant values from the original model
  - Same R-squared, model did not change
- Dropping all rows with “popularity” equal to 0
  - R-squared went from .754 to .673
- Logistic regression
  - With an accuracy of .16, this was not further investigated

Ultimately our original model had the best fit for predicting Spotify song popularity.

References

- https://www.kaggle.com/code/vatsalmavani/music-recommendation-system-using-spotify-dataset/data
- https://www.loudlab.org/blog/spotify-popularity-leverage-algorithm/