Do You Feel The Music?

Kevin Kang, Raymond Sutanto

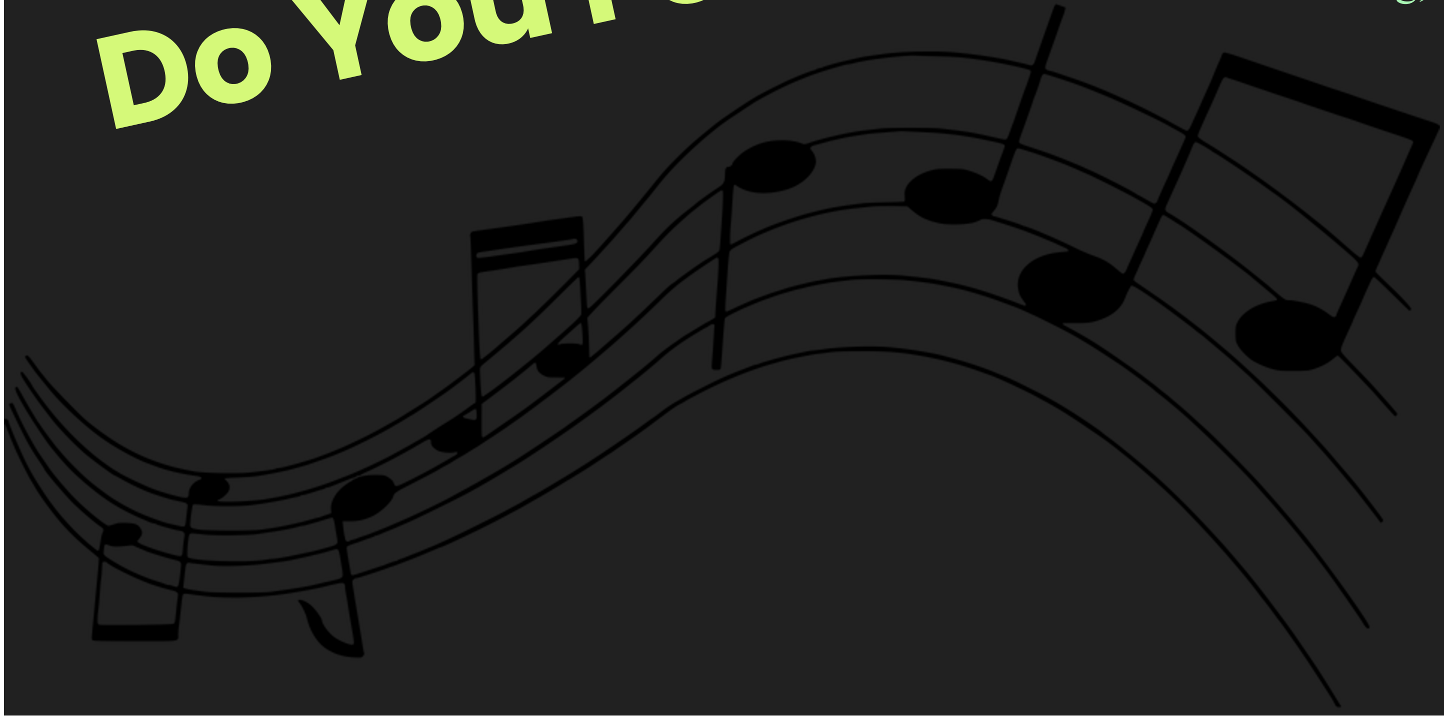


Table of contents

01 Motivation

02 Our Dataset

03 Cleaning & Exploration

04 Analysis

05 Recommendations

Motivation

- Correlation or Causation?
- Finding The Formula
    - *Analysis*
    - *How variables affect the level of valence*
- “Aiding an algorithm”
    - *Accuracy*
    - *Personalization*

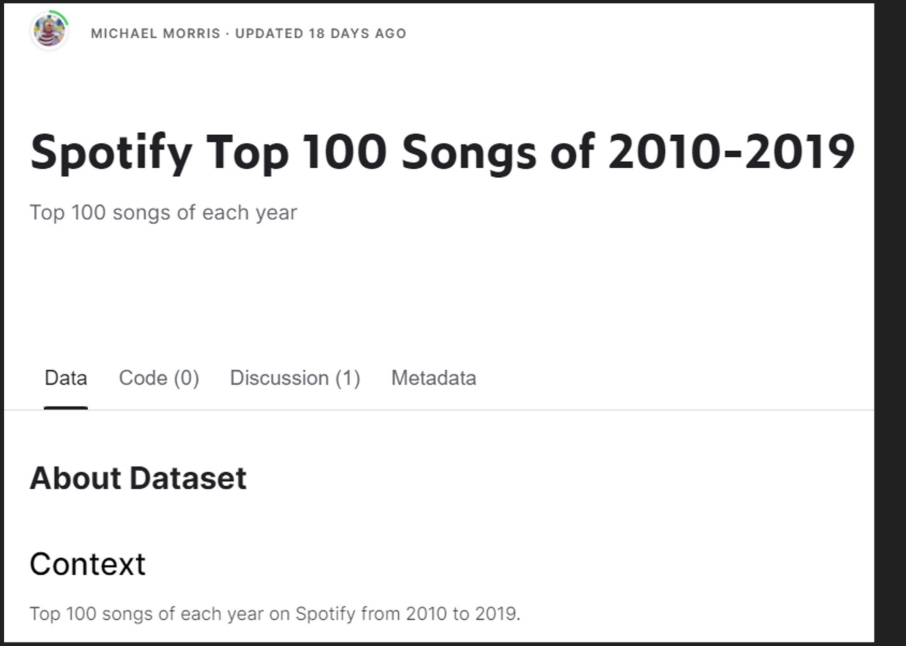
Our Dataset

kaggle

- Spotify Top 100 Songs:
  - *Timeframe: 2010-2019*

- Dataset URL (from KAGGLE):

https://www.kaggle.com/datasets/muhmores/spotify-top-100-songs-of-20152019



MICHAEL MORRIS · UPDATED 18 DAYS AGO

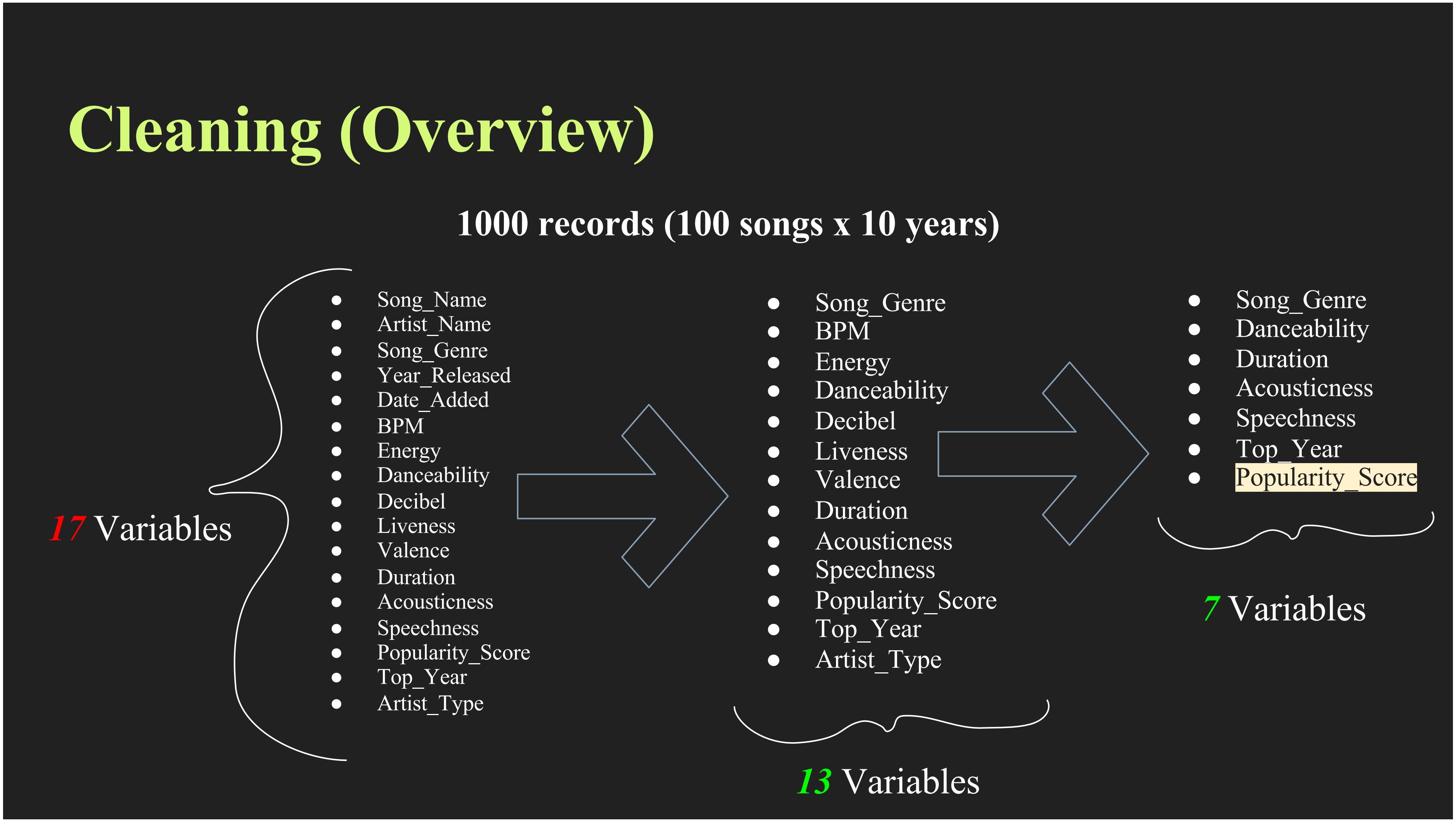
Spotify Top 100 Songs of 2010-2019

Top 100 songs of each year

About Dataset

Context

Top 100 songs of each year on Spotify from 2010 to 2019.



Cleaning (Overview)

1000 records (100 songs x 10 years)

17 Variables

- Song_Name
- Artist_Name
- Song_Genre
- Year_Released
- Date_Added
- BPM
- Energy
- Danceability
- Decibel
- Liveness
- Valence
- Duration
- Acousticness
- Speechness
- Popularity_Score
- Top_Year
- Artist_Type

- Song_Genre
- BPM
- Energy
- Danceability
- Decibel
- Liveness
- Valence
- Duration
- Acousticness
- Speechness
- Popularity_Score
- Top_Year
- Artist_Type

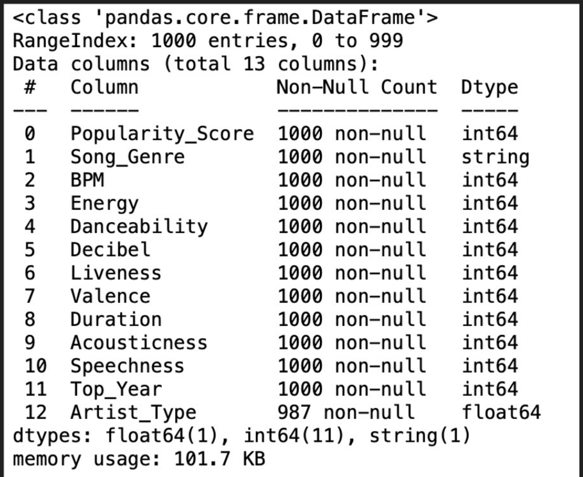
13 Variables

- Song_Genre
- Danceability
- Duration
- Acousticness
- Speechness
- Top_Year
- Popularity_Score

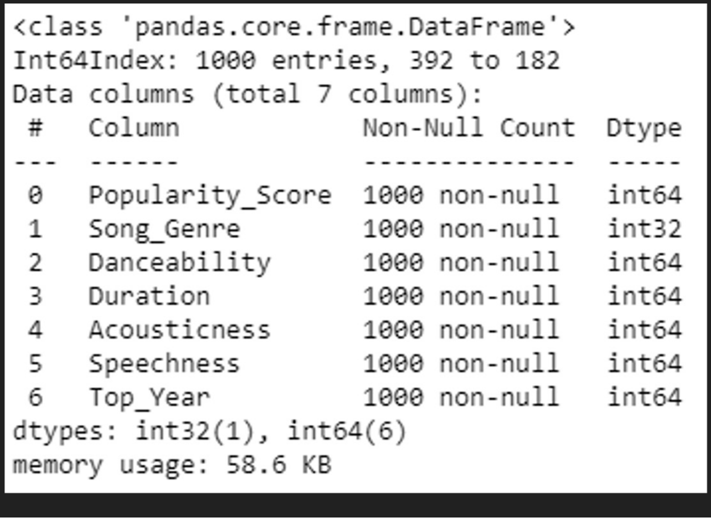
7 Variables

Cleaning

- Check & change necessary column data types to numerical
- Remove irrelevant variables
- Use python libraries in exploration & analysis

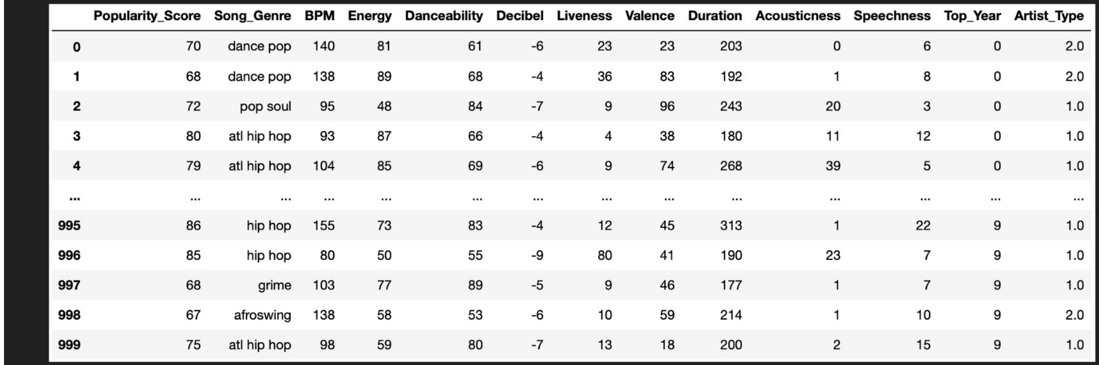


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Popularity_Score  1000 non-null   int64
 1   Song_Genre        1000 non-null   string
 2   BPM               1000 non-null   int64
 3   Energy            1000 non-null   int64
 4   Danceability      1000 non-null   int64
 5   Decibel           1000 non-null   int64
 6   Liveness          1000 non-null   int64
 7   Valence           1000 non-null   int64
 8   Duration          1000 non-null   int64
 9   Acousticness      1000 non-null   int64
 10  Speechness        1000 non-null   int64
 11  Top_Year          1000 non-null   int64
 12  Artist_Type       987 non-null    float64
dtypes: float64(1), int64(11), string(1)
memory usage: 101.7 KB
```



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000 entries, 392 to 182
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Popularity_Score  1000 non-null   int64
 1   Song_Genre        1000 non-null   int32
 2   Danceability      1000 non-null   int64
 3   Duration          1000 non-null   int64
 4   Acousticness      1000 non-null   int64
 5   Speechness        1000 non-null   int64
 6   Top_Year          1000 non-null   int64
dtypes: int32(1), int64(6)
memory usage: 58.6 KB
```

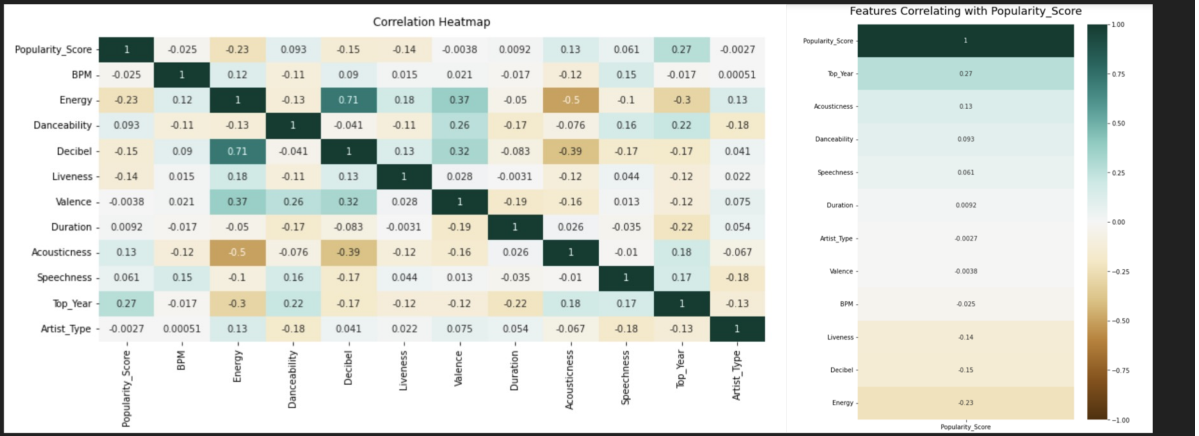
Dataset Preparation (Cleaning)



	Popularity_Score	Song_Genre	BPM	Energy	Danceability	Decibel	Liveness	Valence	Duration	Acousticness	Speechness	Top_Year	Artist_Type
0	70	dance pop	140	81	61	-6	23	23	203	0	6	0	2.0
1	68	dance pop	138	89	68	-4	36	83	192	1	8	0	2.0
2	72	pop soul	95	48	84	-7	9	96	243	20	3	0	1.0
3	80	atl hip hop	93	87	66	-4	4	38	180	11	12	0	1.0
4	79	atl hip hop	104	85	69	-6	9	74	268	39	5	0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	86	hip hop	155	73	83	-4	12	45	313	1	22	9	1.0
996	85	hip hop	80	50	55	-9	80	41	190	23	7	9	1.0
997	68	grime	103	77	89	-5	9	46	177	1	7	9	1.0
998	67	afroswing	138	58	53	-6	10	59	214	1	10	9	2.0
999	75	atl hip hop	98	59	80	-7	13	18	200	2	15	9	1.0

Chart Pre-Cleaning

Heatmap Pre-Cleaning



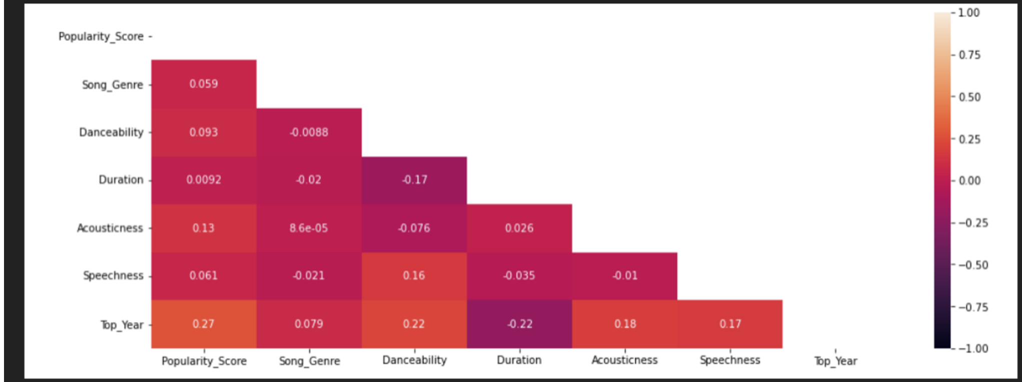
Correlation Heatmap

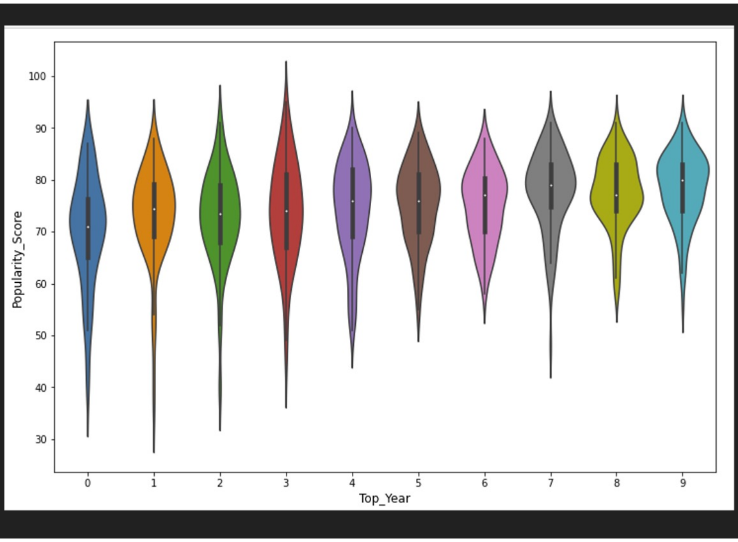
	Popularity_Score	BPM	Energy	Danceability	Decibel	Liveness	Valence	Duration	Acousticness	Speechness	Top_Year	Artist_Type
Popularity_Score	1	-0.025	-0.23	0.093	-0.15	-0.14	-0.0038	0.0092	0.13	0.061	0.27	-0.0027
BPM	-0.025	1	0.12	-0.11	0.09	0.015	0.021	-0.017	-0.12	0.15	-0.017	0.00051
Energy	-0.23	0.12	1	-0.13	0.71	0.18	0.37	-0.05	-0.5	-0.1	-0.3	0.13
Danceability	0.093	-0.11	-0.13	1	-0.041	-0.11	0.26	-0.17	-0.076	0.16	0.22	-0.18
Decibel	-0.15	0.09	0.71	-0.041	1	0.13	0.32	-0.083	-0.39	-0.17	-0.17	0.041
Liveness	-0.14	0.015	0.18	-0.11	0.13	1	0.028	-0.0031	-0.12	0.044	-0.12	0.022
Valence	-0.0038	0.021	0.37	0.26	0.32	0.028	1	-0.19	-0.16	0.013	-0.12	0.075
Duration	0.0092	-0.017	-0.05	-0.17	-0.083	-0.0031	-0.19	1	0.026	-0.035	-0.22	0.054
Acousticness	0.13	-0.12	-0.5	-0.076	-0.39	-0.12	-0.16	0.026	1	-0.01	0.18	-0.067
Speechness	0.061	0.15	-0.1	0.16	-0.17	0.044	0.013	-0.035	-0.01	1	0.17	-0.18
Top_Year	0.27	-0.017	-0.3	0.22	-0.17	-0.12	-0.12	-0.22	0.18	0.17	1	-0.13
Artist_Type	-0.0027	0.00051	0.13	-0.18	0.041	0.022	0.075	0.054	-0.067	-0.18	-0.13	1

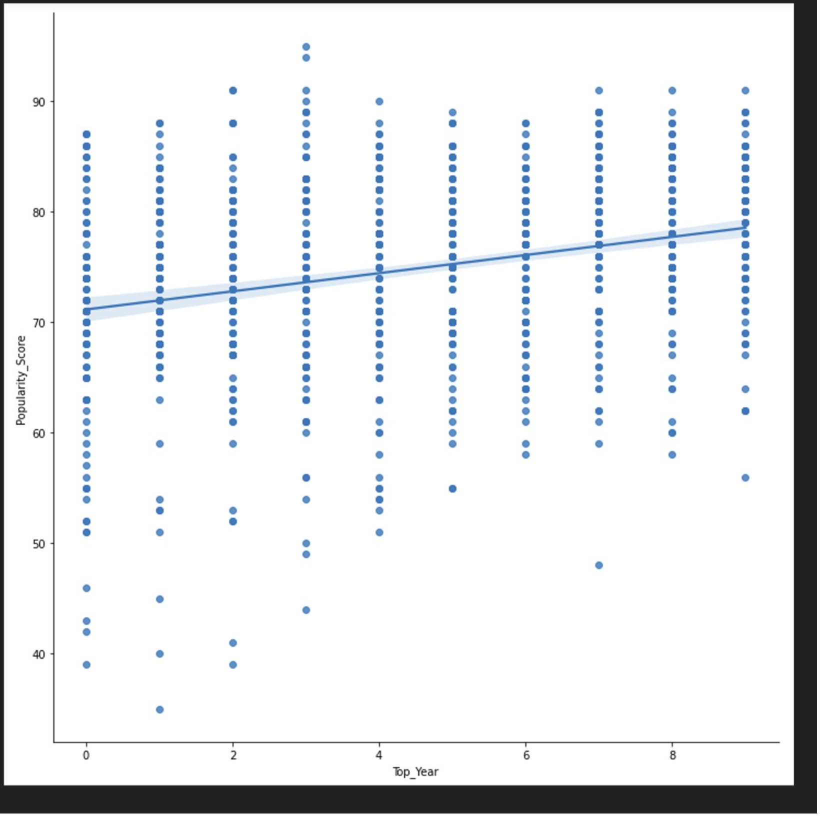
Features Correlating with Popularity_Score

	Popularity_Score
Popularity_Score	1
Top_Year	0.27
Acousticness	0.13
Danceability	0.093
Speechness	0.061
Duration	0.0092
Artist_Type	-0.0027
Valence	-0.0038
BPM	-0.025
Liveness	-0.14
Decibel	-0.15
Energy	-0.23

Dataset Exploration

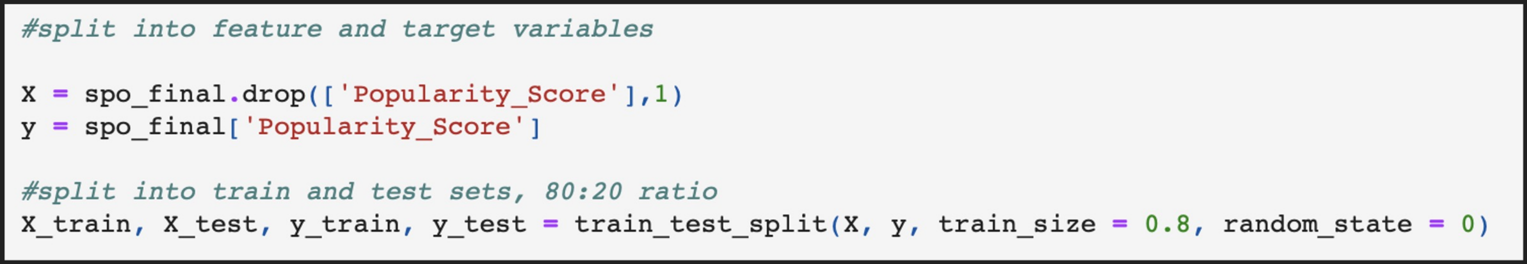






Dataset Analysis (Machine Learning)

PRE

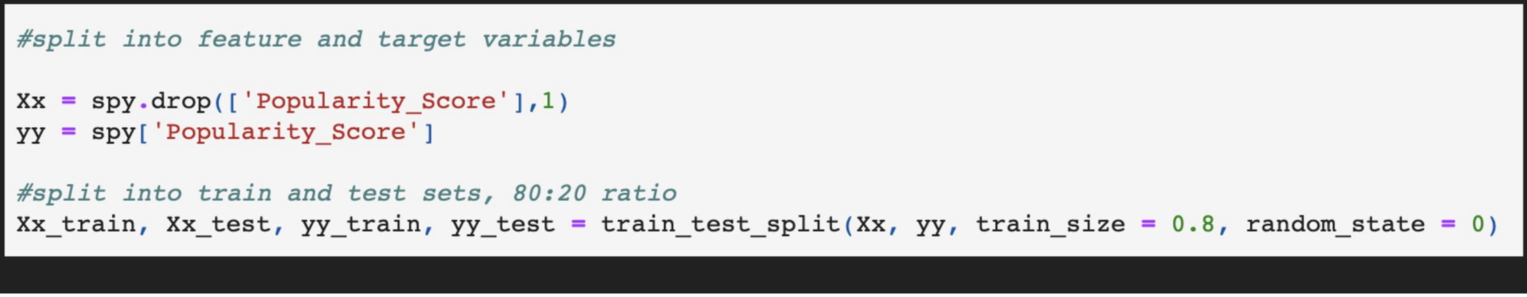


```
#split into feature and target variables

X = spo_final.drop(['Popularity_Score'],1)
y = spo_final['Popularity_Score']

#split into train and test sets, 80:20 ratio
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.8, random_state = 0)
```

POST



```
#split into feature and target variables

Xx = spy.drop(['Popularity_Score'],1)
yy = spy['Popularity_Score']

#split into train and test sets, 80:20 ratio
Xx_train, Xx_test, yy_train, yy_test = train_test_split(Xx, yy, train_size = 0.8, random_state = 0)
```

Dataset Analysis (Machine Learning)

	PRE	POST
Linear Regression	Cross Validation Score: [-17.10 -4.29 -4.37] Average CVS: -14.58 Mean Squared Error: 76.07	Cross Validation Score = [-17.82 -3.36 -4.48] Average CVS: -14.17 Mean Squared Error = 79.67
Logistic Regression	Cross Validation Score: [0.057 0.063 039] Average CVS: 0.046 Mean Squared Error = 108.47 Accuracy = 0.065	Cross Validation Score = [0.054 0.06 0.05] Average CVS: 0.056 Mean Squared Error = 92.19 Accuracy = 0.085
Decision Tree	Cross Validation Score = [0.056 0.06 0.069] Average CVS: 0.072 Mean Squared Error = 129.155 Accuracy = 0.045	Cross Validation Score = [0.042 0.057 0.03] Average CVS: 0.068 Mean Squared Error = 136.73 Accuracy = 0.065

Dataset Analysis (Machine Learning)

	PRE	POST
KNN	Cross Validation Score = [0.012 0.045 0.039] Average CVS: 0.036 Mean Squared Error = 201.07 Accuracy = 0.075	Cross Validation Score = [0.045 0.06 0.036 Average CVS: 0.039 Mean Squared Error = 239.62 Accuracy = 0.03
SVC	Cross Validation Score = [0.066 0.063 0.079] Average CVS: 0.069 Mean Squared Error = 98.24 Accuracy = 0.05	Cross Validation Score = [0.069 0.067 0.075] Average CVS: 0.064 Mean Squared Error = 98.24 Accuracy = 0.05
Neural Network	Cross Validation Score = [0.045 0.039 0.072] Average CVS: 0.057 Mean Squared Error = 106.801 Accuracy = 0.055	Cross Validation Score = [0.06 0.039 0.045] Average CVS: 0.053 Mean Squared Error = 124.61 Accuracy = 0.06

Dataset Analysis (Machine Learning)

	PRE	POST
Logistic Regression	Cross Validation Score: [0.057 0.063 039] Average CVS: 0.046 Mean Squared Error = 108.47 Accuracy = 0.065	Cross Validation Score = [0.054 0.06 0.05] Average CVS: 0.056 Mean Squared Error = 92.19 Accuracy = 0.085
SVC	Cross Validation Score = [0.066 0.063 0.079] Average CVS: 0.069 Mean Squared Error = 98.24 Accuracy = 0.05	Cross Validation Score = [0.069 0.067 0.075] Average CVS: 0.064 Mean Squared Error = 98.24 Accuracy = 0.05

Conclusion

- Recommended model after dropping variables: Logistic regression
- What are the important variables in creating a song?
  - Song Genre, Danceability, Duration, Acousticness, Speechness, Top Year
- Why are our indicators have low numbers?
  - Dropped variables
  - Other qualitative variables
- Business Recommendation:
  - A song producer should consider the above variables in creating a popular song