TELLING STORIES WITH DATA

Part – 1

Dataset Identification and Analysis

Ankita. V

Date: 24th November 2021

Taylor Swift Spotify Analysis

Introduction:

Taylor Alison Swift is an American singer-songwriter and one of the most successful artists of the 21st century. As a modern musician and cultural icon, Taylor Swift has earned worldwide acclaim via songs that predominantly draw upon the complex dynamics of personal past experiences of love, loss and heartbreak. She first broke into the country scene in 2006 with her self-titled Debut Album, then seamlessly transitioned into pop with the album 'Red' and now has folk and alternative genres under her belt with her latest releases. With a large mix of genres and multiple awards, it is undoubtedly said that she's the quintessential artist of the current generation.

One might wonder what makes this chart-topping artist so widespread? Through the analysis, we aim to establish a better understanding of what makes Taylor's songs so popular.

Dataset:

The dataset "**Taylor Swift Spotify Data**" has been acquired from **Kaggle** that consists of songs from Taylor's Albums. It was created by **Jan Llenzl Dagohoy by extracting Spotify WebAPI data.** It was last updated on 6th November 2021 and consists of 171 rows and 16 columns. Concerts, studio sessions, and features songs have not been included. Taylor's re-recorded Version of the 2008 Studio Album 'Fearless' has been included

instead of the latter. However, between the standard album and the deluxe album, the deluxe version has been used.

Dataset Attributes:

The dataset consists of **14 independent variables** and **1 dependent variable** namely 'popular':

1. **name** - Name of the song
2. **album** - Name of the album
3. **artist** - Name of the artist
4. **release_date** - The release date of the album
5. **length** - The length of the song in milliseconds
6. **popularity** - *Percent* popularity of the song based on Spotify's algorithm
7. **danceability** - How suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat
8. **acousticness** - How acoustic a song is
9. **energy** - A perceptual measure of intensity and activity
10. **instrumentalness** - The amount of vocals in the song
11. **liveness** - Probability that the song was recorded with a live audience
12. **loudness** - Tendency of music to be recorded at steadily higher volumes
13. **speechiness** - Presence of spoken words in a track
14. **valence** - A measure of how happy or sad the song sounds
15. **tempo** - Beats per minute

Note: For more on the calculation of the feature values refer to the following link from Spotify API.

For a better understanding of the dataset, the given snapshot seen below can be comprehended. Consider the first tuple consisting of a single called "Tim Mcgraw" from her debut album "Taylor Swift" released on 24th October 2006. The song shows 49% popularity making it one of her lesser popular songs. The various features are expressed

on a scale from 0 to 1 except for loudness and tempo.

```
df = df.drop(['Unnamed: 0'],axis=1)
df.head()
```

	name	album	artist	release_date	length	popularity	danceability	acousticness	energy	instrumentalness	liveness	loudness	speechiness	valence	
0	Tim McGraw	Taylor Swift	Taylor Swift	2006-10-24	232106	49	0.580	0.575	0.491	0.0	0.1210	-6.462	0.0251	0.425	
1	Picture To Burn	Taylor Swift	Taylor Swift	2006-10-24	173066	54	0.658	0.173	0.877	0.0	0.0962	-2.098	0.0323	0.821	1
2	Teardrops On My Guitar - Radio Single Remix	Taylor Swift	Taylor Swift	2006-10-24	203040	59	0.621	0.288	0.417	0.0	0.1190	-6.941	0.0231	0.289	
3	A Place in this World	Taylor Swift	Taylor Swift	2006-10-24	199200	49	0.576	0.051	0.777	0.0	0.3200	-2.881	0.0324	0.428	
4	Cold As You	Taylor Swift	Taylor Swift	2006-10-24	239013	50	0.418	0.217	0.482	0.0	0.1230	-5.769	0.0266	0.261	1

Dataset Cleaning:

Feature selection is a crucial step in Data Analysis to ensure optimum model performance. The dataset consists of a column named Unnamed: 0 which functions as an Index and is unnecessary and thus can be removed. There are **no null values** in the dataset and thus every tuple is taken into consideration during visualization. Read more about feature selection here.

```
df.isnull().sum() #Finds the null values in each column
```

```
Unnamed: 0          0
name                0
album               0
artist              0
release_date        0
length              0
popularity          0
danceability        0
acousticness        0
energy              0
instrumentalness    0
liveness            0
loudness            0
speechiness         0
valence             0
tempo               0
dtype: int64
```

The artist column consists of **only 1 unique value**, it proves to be redundant and can be dropped as it doesn't provide any valuable information. The features 'album' and 'release_date' both have 9 unique values due to the exclusion of singles and featured songs hence conveying the same information. 'release_date' is dropped to avoid the presence of superfluous columns in the dataset.

```
df.select_dtypes(include=object).nunique()
```

```
name            169
album             9
artist            1
release_date      9
dtype: int64
```

The df.describe() command is used to calculate statistical data like mean, standard deviation, maximum value, etc. While analyzing the statistical quantities for the features in the dataset, it can be noted that 25%, median and, 75% quartile are all zero and the mean is also a very small value. Hence can be dropped to prevent noise during analysis.

```
df.describe()
```

	length	popularity	danceability	acousticness	energy	instrumentalness	liveness	loudness	speechiness	valence	tempo
count	171.000000	171.000000	171.000000	171.000000	171.000000	171.000000	171.000000	171.000000	171.000000	171.000000	171.000000
mean	236663.520468	61.228070	0.588632	0.321634	0.585977	0.002490	0.145927	-7.322111	0.065583	0.422984	124.141415
std	40456.720158	11.904548	0.115067	0.334019	0.189577	0.018766	0.090314	2.878787	0.105956	0.192617	31.484487
min	107133.000000	0.000000	0.292000	0.000191	0.118000	0.000000	0.033500	-17.932000	0.023100	0.049900	68.534000
25%	211833.000000	58.000000	0.527000	0.030450	0.462000	0.000000	0.092950	-8.861500	0.029500	0.277500	96.052000
50%	234000.000000	63.000000	0.593000	0.156000	0.606000	0.000002	0.115000	-6.698000	0.037200	0.416000	121.956000
75%	254447.000000	67.000000	0.655500	0.674000	0.732000	0.000064	0.168000	-5.336500	0.055100	0.545000	146.040500
max	403887.000000	82.000000	0.897000	0.971000	0.944000	0.179000	0.657000	-2.098000	0.912000	0.942000	207.476000

By dropping the unnecessary features, the number of features has been reduced from 15 to 10 independent and 1 dependant feature. The dependent feature 'popularity' is shifted to the rightmost position for convenience during Visualization and Analysis. The dataset is now completely cleaned.

The small size of the dataset might cause issues during further analysis and must be taken into consideration.

```
popularity = df['popularity']
df = df.drop(['instrumentalness','popularity','release_date'],axis=1)
df.insert(loc=len(df.columns), column='popularity', value=popularity)
df
```

	album	length	danceability	acousticness	energy	liveness	loudness	speechiness	valence	tempo	popularity
0	Taylor Swift	232106	0.580	0.575	0.491	0.1210	-6.462	0.0251	0.425	76.009	49
1	Taylor Swift	173066	0.658	0.173	0.877	0.0962	-2.098	0.0323	0.821	105.586	54
2	Taylor Swift	203040	0.621	0.288	0.417	0.1190	-6.941	0.0231	0.289	99.953	59
3	Taylor Swift	199200	0.576	0.051	0.777	0.3200	-2.881	0.0324	0.428	115.028	49
4	Taylor Swift	239013	0.418	0.217	0.482	0.1230	-5.769	0.0266	0.261	175.558	50
...	...	...	...	...	...	...	...	...	...	...	...
166	Fearless (Taylor's Version)	277591	0.660	0.162	0.817	0.0667	-6.269	0.0521	0.714	135.942	74
167	Fearless (Taylor's Version)	244236	0.609	0.849	0.373	0.0779	-8.819	0.0263	0.130	106.007	65
168	Fearless (Taylor's Version)	189495	0.588	0.225	0.608	0.0920	-7.062	0.0365	0.508	90.201	67
169	Fearless (Taylor's Version)	208608	0.563	0.514	0.473	0.1090	-11.548	0.0503	0.405	101.934	66
170	Fearless (Taylor's Version)	242157	0.624	0.334	0.624	0.0995	-7.860	0.0539	0.527	80.132	64

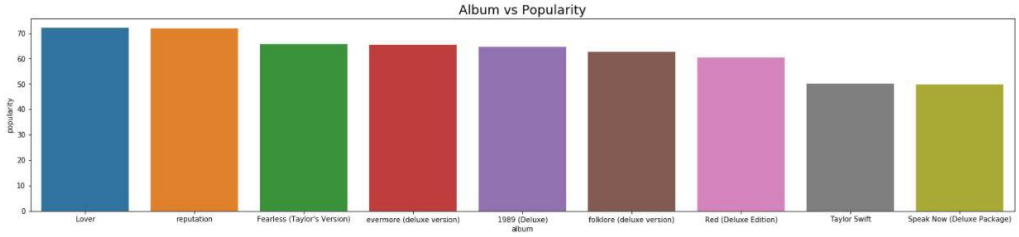
171 rows × 11 columns

Some basic visualizations can be done to further improve our understanding of the dataset. From the below image we can conclude that Lover, Reputation and Fearless(Taylor's Version) are her three most popular albums.

```
pop_data = df.groupby('album').mean().sort_values(['popularity'],ascending=False)
pop_data = pop_data.reset_index()
```

```
plt.figure(figsize=(25, 5))
sns.barplot(x='album', y='popularity', data=pop_data)
plt.title('Album vs Popularity', fontsize=18)
```

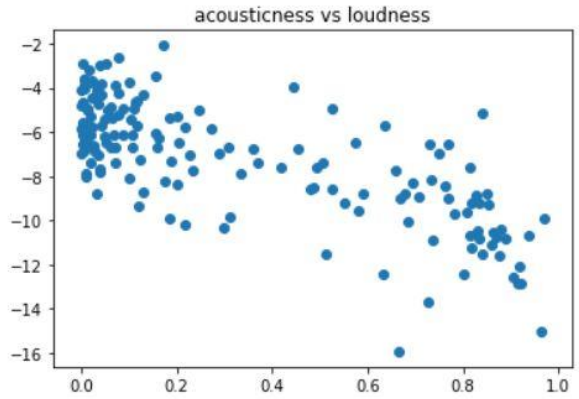
```
Text(0.5, 1.0, 'Album vs Popularity')
```



The scatter plot for the features can be visualized to infer that there exists a negative correlation between acousticness and loudness.

```
plt.scatter(df["acousticness"], df["loudness"])
plt.title("acousticness vs loudness")
```

```
Text(0.5, 1.0, 'acousticness vs loudness')
```



Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance. Features like Tempo and Loudness can be normalized to ensure that all the features lie on the same scale.

```
from sklearn.preprocessing import normalize
import numpy as np
x_array = np.array([df['loudness'],df['tempo']])
print(x_array)
normalized_X = normalize(x_array)
```

```
[[ -6.462  -2.098  -6.941  -2.881  -5.769  -4.055  -4.963  -4.919  -3.771
   -5.28   -4.931  -3.629  -5.723  -5.726  -3.827  -3.863  -2.976  -5.797
   -3.75   -5.378  -3.978  -4.827  -8.829  -3.913  -3.185  -5.295  -2.641
```

```
df.insert(11,'loudness_n',normalized_X[0])
df.insert(12,'tempo_n',normalized_X[1])
```

```
df.head()
```

	album	length	danceability	acousticness	energy	liveness	loudness	speechiness	valence	tempo	popularity	loudness_n	tempo_n
0	Taylor Swift	232106	0.580	0.575	0.491	0.1210	-6.462	0.0251	0.425	76.009	49	-0.065411	0.045691
1	Taylor Swift	173066	0.658	0.173	0.877	0.0962	-2.098	0.0323	0.821	105.586	54	-0.021237	0.063470
2	Taylor Swift	203040	0.621	0.288	0.417	0.1190	-6.941	0.0231	0.289	99.953	59	-0.070260	0.060084
3	Taylor Swift	199200	0.576	0.051	0.777	0.3200	-2.881	0.0324	0.428	115.028	49	-0.029163	0.069146
4	Taylor Swift	239013	0.418	0.217	0.482	0.1230	-5.769	0.0266	0.261	175.558	50	-0.058397	0.105532

Dataset Usage:

Since all of the features that are used in the prediction of the popularity of the songs in the dataset are continuous, methods of **regression analysis** will be primarily used, where the process of performing a regression allows you to confidently determine which factors matter most, which factors can be ignored, and how these factors influence each other. Methods of **normalization**, **correlation** can also be used to visualize the data and get definitive conclusions from the dataset.

There are endless possibilities and lots of ways this dataset can be analyzed. That is why I have chosen it for the final project, hoping to have lots of fun exploring it.