Term Project Report

組別: 7

組員: 107062333 湯睿哲 107062334 黃杰

主題: Recommendation System

- 摘要:

  這次期末 project，我們這組主要是研究現今已 Well-Defined 的 Recommendation System，而我們主要 Focus 在推薦熱門歌曲以及對於使用者收聽的某歌曲，推薦於該歌曲曲風相似的作法與實作呈現。

- 什麼是 Recommendation System:

  一種訊息過濾系統，界過濾的資料，進行諸多操作，如:

  利用其他 item 與目標 item 相關程度來預測其受歡迎程度，抑或是利用使用者之間的相近程度，來推薦其他 item 給某位使用者等。

- Based Technic – Similar Matric:

  本次實作內容主要會利用到相似度的運算，而方法有很多種，我們這次實作上主要使用到了 Cosine-Similarity 以及 Euclidean-Similarity，來估算每個 object 之間的相似值，計算公式如下圖:

- Cosine:

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

- Euclidean:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

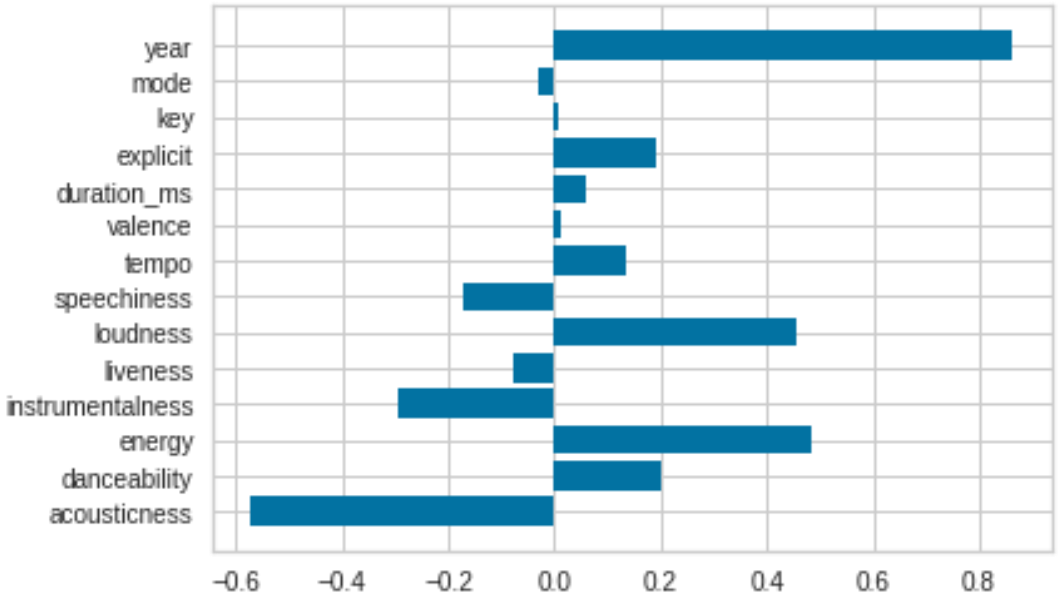
- Dataset:

資料及我們主要採用 Spotify 在 Kaggle 上所提供的 2018 音樂資料，內容大約 17 萬首歌，每首歌會多個 attributes 來描述他的特徵，節錄如下圖:

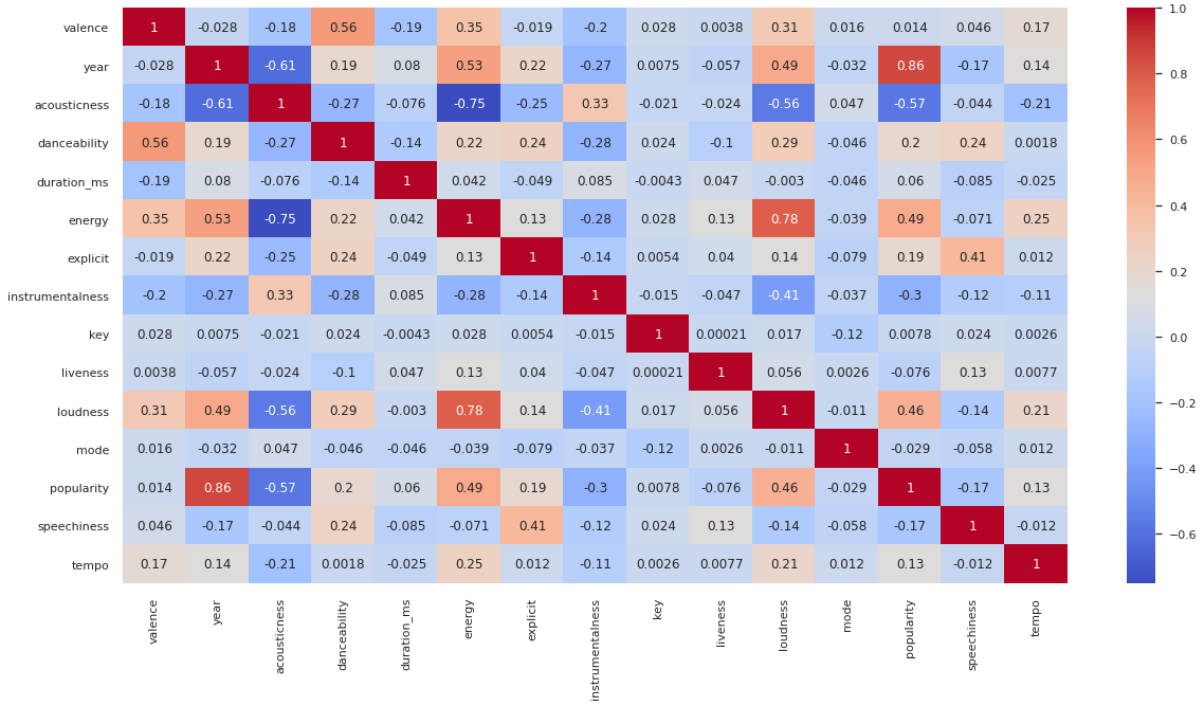
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	valence	year	acousticne	artists	danceabili	duration_r	energy	explicit	id	instrumen	key	liveness	loudness	mode	name	popularity	release_da	speechine	tempo
2	0.0594	1921	0.982	['Sergei R	0.279	831667	0.211	0	4BJqT0Pr	0.878	10	0.665	-20.096	1	Piano Con	4	1921	0.0366	80.954
3	0.963	1921	0.732	['Dennis D	0.819	180533	0.341	0	7xPhfUan	0	7	0.16	-12.441	1	Clancy Lo	5	1921	0.415	60.936
4	0.0394	1921	0.961	['KHP Kri	0.328	500062	0.166	0	1o6I8BglA	0.913	3	0.101	-14.85	1	Gati Bali	5	1921	0.0339	110.339
5	0.165	1921	0.967	['Frank Pa	0.275	210000	0.309	0	3ftBPsC5	2.77E-05	5	0.381	-9.316	1	Danny Bo	3	1921	0.0354	100.109
6	0.253	1921	0.957	['Phil Reg	0.418	166693	0.193	0	4d6HGyG	1.68E-06	3	0.229	-10.096	1	When Iris	2	1921	0.038	101.665
7	0.196	1921	0.579	['KHP Kri	0.697	395076	0.346	0	4pyw9DV	0.168	2	0.13	-12.506	1	Gati Mard	6	1921	0.07	119.824
8	0.406	1921	0.996	['John Mc	0.518	159507	0.203	0	5uNZnElq	0	0	0.115	-10.589	1	The Wear	4	1921	0.0615	66.221
9	0.0731	1921	0.993	['Sergei R	0.389	218773	0.088	0	02GDntO2	0.527	1	0.363	-21.091	0	Morceaux	2	1921	0.0456	92.867
10	0.721	1921	0.996	['Ignacio (	0.485	161520	0.13	0	05xDjWH	0.151	5	0.104	-21.508	0	La Ma簽a	0	1921/3/20	0.0483	64.678
11	0.771	1921	0.982	['Fortug矇	0.684	196560	0.257	0	08zfJvRL	0	8	0.504	-16.415	1	Il Etait Sy	0	1921	0.399	109.378
12	0.826	1921	0.995	['Maurice	0.463	147133	0.26	0	0BMkRpC	0	9	0.258	-16.894	1	Dans La V	0	1921	0.0557	85.146
13	0.578	1921	0.994	['Ignacio (	0.378	155413	0.115	0	0F30WM8	0.906	10	0.11	-27.039	0	Por Que M	0	1921/3/20	0.0414	70.37
14	0.493	1921	0.99	['Georgel'	0.315	190800	0.363	0	0H3k2CvJ	0	5	0.292	-12.562	0	La Vip癡t	0	1921	0.0546	174.532
15	0.212	1921	0.912	['Mehmet	0.415	184973	0.42	0	0LcXzAB	0.89	8	0.108	-10.766	0	Ud Taksin	0	1921	0.114	70.758
16	0.493	1921	0.0175	['Zay Gats	0.527	205072	0.691	1	0MJZ4hht	0.384	7	0.358	-7.298	1	Power Is I	0	1921/3/27	0.0326	159.935
17	0.282	1921	0.989	['Sergei R	0.384	221013	0.171	0	0NFeJgm	0.82	7	0.116	-20.476	0	10 Pr矇lu	4	1921	0.0319	107.698
18	0.218	1921	0.957	['Phil Reg	0.259	186467	0.212	0	0Nk5f07H	0.000222	2	0.236	-13.3	1	Come Bac	1	1921	0.0358	85.726
19	0.664	1921	0.996	['Hector B	0.541	250747	0.283	0	0POO8Xa	0.898	9	0.393	-14.808	1	R癩k籀cz	0	1921	0.0477	108.986
20	0.0778	1921	0.148	['THE GU	0.604	204957	0.418	1	0QQmUf4	0.0382	4	0.102	-11.566	0	When We	0	1921/9/11	0.0417	80.073

https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks

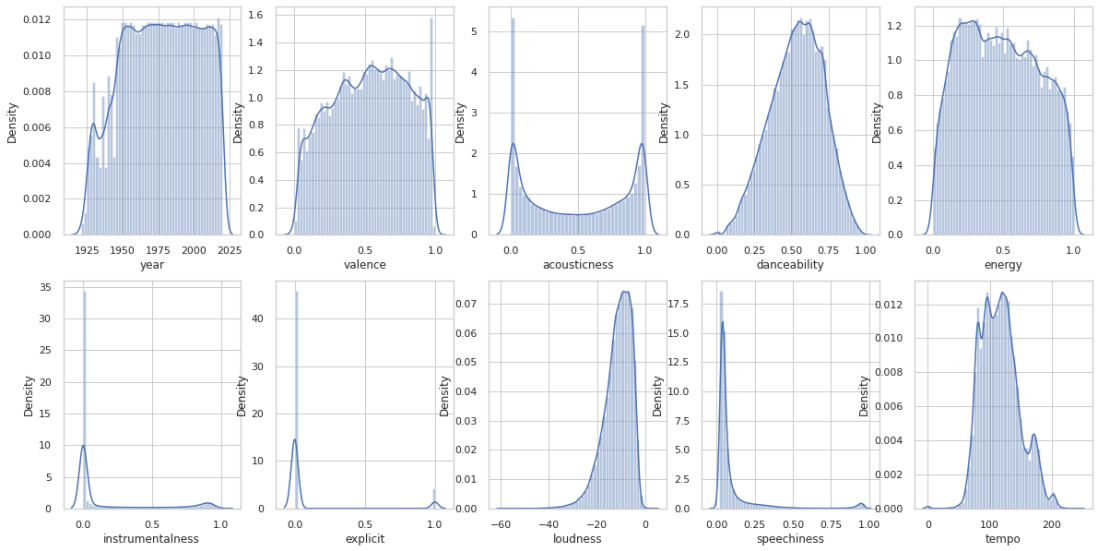
- Each Attributes V.S popularity:

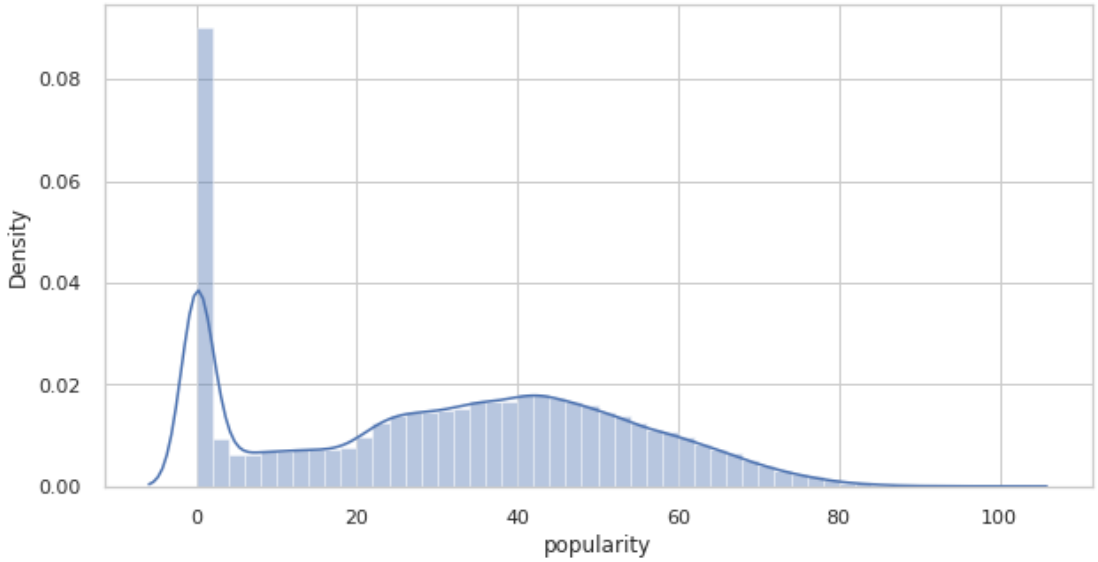


- Heat Map



- Attributes Distribution:





- Popularity Prediction – Naiive Method:

  對於如何預測一首歌的受歡迎程度，我們第一個想到的方法就是計算目標歌曲與資料及所有歌曲的相似值，並將該受比較歌曲的 popularity 乘上該相似值，並加總起來，計算式子如下:

  POP_k = Sigma(sim(k,i)*POP_i)

  k 代表我們要預測的歌曲，I 代表資料其所有個取，不包含 k

- 但不難發現，這種方式很顯然光是預測一首歌的 pop.就會花上不少時間，更何況是應用在多少歌單上。

- Popularity Prediction – Machine Learning:

  也因次，我們改採用分類 Model 的方式訓練出一個預測模型，對於 features selection，有了上圖資料分部以及 heatmap 的輔助，我們選取與 popularity 相關係數大的 attributes 做為要使用的 features。另外，由於 label 為連續分部的值(0~100)，我們使用 RandomForestRegressor 以及 ExtraTreeRegressor 模型，結果 Evaluation 如下圖:

				Our Result	
	Decision Tree Regressor	Decision Tree with Grid Search CV	Random Forest Regressor	Random Forest Regressor	Extra Trees Regressor
R2-Score	0.748	0.829	0.746	0.8036	0.8017
MAE	7.92	7.3	7.56	6.8211	6.856

白色區塊為 Kaggle 其他團隊的結果。

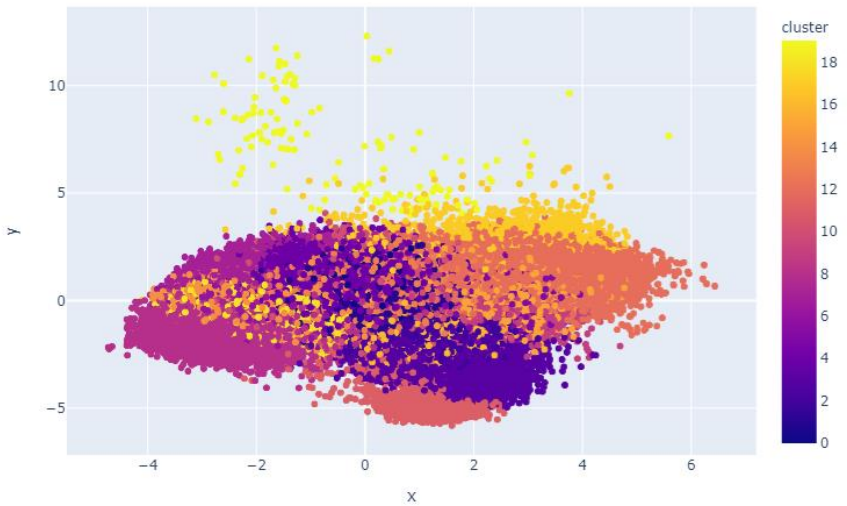
- Similar Song Recommendation – Our Method:

  對於推薦相似歌曲的作法，我們是對目標歌曲，直接計算它與資料及所有歌曲的相似值，並挑選前幾首與該歌曲相近(Euclidean 相似值高)的歌，作為我們 System 的 Output，但此種作法光一首歌的 input，就會花上約 10~15 秒的時間進行計算。

- Similar Song Recommendation – Clustering + Similarity:

  為解決時間上的問題，我們從 Kaggle 上找到某位高手的做法，是先對整個資料集(已 features selection)，先進行分群(Kmeans)，有了一分群 Model，對於任一 input song

  就能有快速的方式，先歸類 input 為哪一群體，再從該群體找出前幾首與 input 相似(cosine similarity)的歌曲，以下是經 Kmeans 分群後，資料集歌曲的分布: N = 18



- Evaluation

在評估方面，我們主要拿上述兩種方式進行推薦出來的歌曲的相似程度比對。

- Input: Drake - In My Feelings

  Our System:

	artists	name
18850	['Drake', 'WizKid', 'Kyla']	One Dance
17621	['Trey Songz', 'Nicki Minaj']	Bottoms Up (feat. Nicki Minaj)
91803	['Eminem']	Greatest
170507	['Eminem', 'Young M.A']	Unaccommodating (feat. Young M.A)
18625	['Big Sean', 'E-40']	I Don't Fuck With You

  System on Kaggle user:

	artists	name
170507	['Eminem', 'Young M.A']	Unaccommodating (feat. Young M.A)
17621	['Trey Songz', 'Nicki Minaj']	Bottoms Up (feat. Nicki Minaj)
91803	['Eminem']	Greatest
38383	['DaBaby', 'Offset']	Baby Sitter (feat. Offset)
18625	['Big Sean', 'E-40']	I Don't Fuck With You

  可以發現兩種方式出來的結果，5 首有 4 首一樣，而對於那不一樣的曲子，入實際搜尋並收聽，相似曲風也會蠻相似的，屬於 rap、hiphop 曲。

- 結論:

  綜以上這次 project 實作的 popularity prediction，可以發現除了 features selection 的重要，optimization 常扮演著重要的腳色，能有效的提升整個 accuracy。

而在 similar song recommender 的實作中，我們除了追求正確性以為，另外重要的是在 Time Complexity 上的掌握，不能總是以淺顯易懂而耗時的方式去解決。必須在兩者之間取得平衡。