

Hit My Beatles

โดย

64010552 นายพนธกร วัลลานนท์

64010670 นางสาวภาพิชญ์ พงศ์พัฒนานวุฒิ

64010757 นายวรโชติ ใจเร็ว

64010761 นายวรพล รังษี

โครงการนี้เป็นส่วนหนึ่งของการศึกษา

วิชา 01076032 ELEMENTARY DIFFERENTIAL EQUATIONS AND

LINEAR ALGEBRA

ปีการศึกษา 2565

บทคัดย่อ

โครงการเรื่อง Hit My Beatles จัดทำขึ้นเพื่อแนะนำเพลงของวง The Beatles ให้ผู้
ใช้ได้ฟัง โดยตัวแอปพลิเคชันนี้จะให้กรอการมณของเพลงที่ผู้ใช้อยากฟังโดยผ่านการตอบ
คำถามเป็นจำนวน 6 ข้อ แล้วแอปพลิเคชันก็จะทำการประมวลผลโดยการนำความรู้ใน
เรื่องของ Matrix และ Vector มาประยุกต์ใช้ผ่านกระบวนการต่างๆ เพื่อแสดงผลแนะนำ
เพลงที่ใกล้เคียงกับความชอบของผู้ใช้ออกมาเพื่อให้ได้ลองฟัง บางทีอาจจะถูกใจจนต้อง
เก็บเข้า playlist อย่างแน่นอน

คำสำคัญ: The Beatles, แนะนำเพลง, Matrix, Vector

สารบัญ

บทคัดย่อ.....	(1)
สารบัญ.....	(2)
บทที่ 1 บทนำ.....	1
1.1 ที่มาของโครงการ	1
1.2 จุดประสงค์โครงการ	2
บทที่ 2 ภาพรวมการออกแบบระบบ	3
2.1 ภาพรวมขั้นตอนการทำงานของระบบ	3
2.2 รายละเอียดข้อมูลที่เกี่ยวข้อง	4
2.3 อธิบายขั้นตอนย่อยแต่ละขั้น	14
บทที่ 3 การประยุกต์ใช้ทฤษฎี	25
3.1 การประยุกต์ใช้ทฤษฎีเวกเตอร์	25
3.2 การประยุกต์ใช้ทฤษฎีเมทริกซ์	27
บทที่ 4 ผลการทดลอง.....	29
.1 Correlations	29
4.2 Cosine Similarity	30
4.3 Model Hit My Beatles	31
4.4 ตัวอย่างการทำงานของ Mode Hit My Beatles	32

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ	33
5.1 สรุปผลการทดลอง	33
5.2 ข้อเสนอแนะ	33
รายการอ้างอิง.....	35
ภาคผนวก	36
ภาคผนวก ก	37
ข้อมูลโครงการ	37
ภาคผนวก ข	38
วิดีโอและสไลด์นำเสนอโครงการ	38

บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

ไม่ว่าจะผ่านมาก็ยุคที่สมัยสื่อบันเทิงที่ยังอยู่ควบคู่กับเรามาตลอดอย่างหนึ่งเลยก็คือเสียงเพลงหรือดนตรีที่คอยบรรเทาให้จิตใจของเราสงบขึ้น ได้ผ่อนคลายออกมาจากสิ่งต่างๆ อีกทั้งยังเป็นการเสริมสร้างจินตนาการให้เราอีกด้วย ปฏิเสธไม่ได้เลยว่าเสียงเพลงอยู่ในทุกช่วงอายุของคนเรามากจริงๆ ซึ่งในแต่ละเพลงนั้นมีจังหวะ ทำนอง คำร้อง ความหมายของเพลงที่แตกต่างกันออกไป ทำให้ผู้คนสามารถเลือกเสพแนวเพลงตามแบบที่ตนเองชื่นชอบได้ ไม่ว่าจะเป็นเพลงป๊อป เพลงร็อก เพลงเศร้าหรือจะเป็นเพลงซึ้งๆสบายๆไว้เปิดฟังตอนทำงาน ด้วยความทันสมัยของเทคโนโลยีสมัยนี้นั้นทำให้เราสามารถรวบรวมเพลงที่ชื่นชอบไว้แล้วนำมาสร้างเป็นเพลย์ลิสต์สำหรับฟังเป็นระยะเวลานานๆได้แล้ว ซึ่งสำหรับบางคนหากฟังเพลย์ลิสต์เดิมวนเพลงเดิมๆซ้ำๆเป็นระยะเวลานานก็อาจจะทำให้เกิดอาการเบื่อได้ ดังนั้นคงจะเป็นเรื่องที่ดีและมีความสุขมากถ้าหากได้ค้นพบเพลงใหม่ๆที่ตรงกับแนวเพลงที่เราชอบฟังหรือสามารถเลือกเพลงได้ตามอารมณ์ความรู้สึกของเราในวันนั้นๆแล้วมาเพิ่มลงในเพลย์ลิสต์เพลงของเราเพื่อไม่ให้เกิดความจำเจและยังสามารถเป็นการเปลี่ยนบรรยากาศใหม่ๆในการฟังเพลงได้อีกด้วย

จากแนวความคิดข้างต้นที่ว่าการค้นพบเพลงใหม่ๆที่ตรงกับแนวเพลงที่เราชอบฟังหรือการที่สามารถเลือกฟังเพลงได้ตามอารมณ์ความรู้สึกของเราในวันหนึ่งวันได้นั้น เป็นเรื่องที่น่าสนใจมากและสามารถทำได้โดยนำชุดข้อมูลที่ผู้ใช้กรอกตอบคำถามเกี่ยวกับการฟังเพลงเข้ามาแล้วนำไปเทียบชุดข้อมูลที่เรามีอยู่แล้ว โดยเมื่อผ่านการคำนวณเสร็จแล้วจะได้ Model การแนะนำเพลงที่ใกล้เคียงที่สุดออกมาได้ ด้วยเหตุนี้ทางคณะผู้จัดทำจึงได้ตัดสินใจที่จะทำโครงการเรื่องนี้เพื่อให้เป็นประโยชน์แก่ผู้ที่ต้องการอยากรู้จัก อยากฟังเพลงตามอารมณ์ความรู้สึกของตนเองหรือต้องการค้นหาเพลงใหม่ๆเพื่อนำไปเพิ่มในเพลย์ลิสต์ให้ได้เปลี่ยนบรรยากาศในการฟังเพลง

1.2 จุดประสงค์โครงการ

- 1.2.1. เป็นตัวช่วยสำหรับมือใหม่ที่ต้องการฟังเพลงของวง The Beatles
- 1.2.2. ลดปัญหาในการตัดสินใจเลือกเพลงฟังไม่ได้
- 1.2.3. เพื่อนำความรู้ที่ได้จากการเรียน Matrix และ Vector มาใช้ในการทำงานจริง

บทที่ 2

ภาพรวมการออกแบบระบบ

2.1 ภาพรวมขั้นตอนการทำงานของระบบ

ระบบจะวิเคราะห์ข้อมูลที่นำเข้ามาโดยใช้การเขียนโปรแกรมภาษา Python โดยจะทำงานผ่านแพลตฟอร์ม Google Colab

2.1.1 การนำเข้าข้อมูลสู่ระบบ

นำข้อมูลที่เป็นไฟล์ CSV ที่ทำการ clean แล้วเข้าสู่ระบบโดยการอัปโหลดไฟล์ไว้ใน GitHub จากนั้นจึงทำการ import ข้อมูลเข้าสู่ระบบ

2.1.2 การปรับแต่งข้อมูล

ข้อมูลที่ได้มานั้นยังไม่สามารถนำไปวิเคราะห์ได้เพราะว่าอาจมีข้อมูลบางตัวที่ผิดพลาด ขาดหายหรือตกหล่นรวมไปถึงข้อมูลบางชุดข้อมูลที่ไม่ได้นำมาวิเคราะห์ ดังนั้นจึงทำการปรับแต่งข้อมูลก่อน

2.1.3 การวิเคราะห์ชุดข้อมูล

นำข้อมูลที่ได้หลังจากการปรับแต่งมาวิเคราะห์ข้อมูลในแต่ละชุดข้อมูล

2.1.4 Model Hit My Beatles

เมื่อทำการวิเคราะห์ชุดข้อมูลสำเร็จแล้วจะได้ model ที่ใช้ในการแนะนำเพลงของวง The Beatles โดยเมื่อนำค่าที่ได้จากการตอบคำถามของผู้ใช้จะนำมาหาค่าความใกล้เคียงโดยใช้ Cosine Similarity เพื่อแนะนำเพลงที่ใกล้เคียงที่สุดออกมา

2.2 รายละเอียดข้อมูลที่เกี่ยวข้อง

ชุดข้อมูลที่ใช้ใช้นั้นมาจากเว็บไซต์ Kaggle ซึ่งมีจำนวนชุดข้อมูลทั้งหมด 193 ชุด โดยที่ชุดข้อมูลประกอบไปด้วยข้อมูลดังต่อไปนี้

1. Id	คือ หมายเลขลำดับของข้อมูล
2. Year	คือ ปีที่ปล่อยเพลงนั้นๆ
3. Album	คือ ชื่ออัลบั้ม
4. Song	คือ ชื่อเพลง
5. Danceability	คือ ค่าการแปลงเป็นตัวเลขของ 0 ถึง 1 ของเพลงนั้นๆว่าสามารถเต้นกับเพลงนั้นๆได้หรือไม่โดยพิจารณาจาก จังหวะ เครื่องดนตรี
6. Energy	คือ ค่าการแปลงเป็นตัวเลขของ 0 ถึง 1 ของเพลงนั้นๆว่ามีความหนักแน่นของเพลงมากน้อยแค่ไหน
7. Speechiness	คือ ค่าการแปลงเป็นตัวเลขของ 0 ถึง 1 ของเพลงนั้นๆว่าเพลงนั้นมีคำร้องมากน้อยแค่ไหน
8. Acousticness	คือ ค่าการแปลงเป็นตัวเลขของ 0 ถึง 1 ของเพลงนั้นๆว่าเพลงนั้นๆมีการใช้เครื่องดนตรี Accoustic มากน้อยแค่ไหน
9. Liveness	คือ ค่าการแปลงเป็นตัวเลขของ 0 ถึง 1 ของเพลงนั้นๆว่าเพลงนั้นๆมีผู้ชมมีส่วนร่วมกับการเพลงมากน้อยแค่ไหน
10. Valence	คือ ค่าการแปลงเป็นตัวเลขของ 0 ถึง 1 ของเพลงนั้นๆว่าเพลงมีเนื้อหาเศร้าหรือความสุขโดยพิจารณาจากคำที่ใช้ในเพลงนั้นๆ
11. duration_ms	คือ ระยะเวลาของเพลงนั้นๆโดยหน่วยเป็น millisecond

2.2.1 ID

```
[ ] data.drop(columns = ["id"])
    data["ids"]=[i for i in range(0,data.shape[0])]
```

```
[ ] data.count()
```

```
[ ] data.count()

id          193
year        193
album       193
song        193
danceability 193
energy       193
speechiness  193
acousticness 193
liveness     193
valence      193
duration_ms  193
cbname       193
ids          193
dtype: int64
```

ทำการนับจำนวนข้อมูลแต่ละชุดข้อมูลทั้งหมด

2.2.2 Year

```
data["year"].value_counts()

1968    30
1969    30
1963    28
1965    28
1964    27
1967    24
1966    14
1970    12
Name: year, dtype: int64
```

2.2.3 Album

```
[ ] data["album"].value_counts()

The Beatles (white album)      30
Abbey Road                     17
Please Please Me                14
With The Beatles               14
Beatles for Sale               14
Help!                          14
Rubber Soul                    14
Revolver                       14
A Hard Day's Night             13
Sgt. Pepper's Lonely Hearts Club Band 13
Yellow Submarine               13
Let It Be                     12
Magical Mystery Tour           11
Name: album, dtype: int64
```

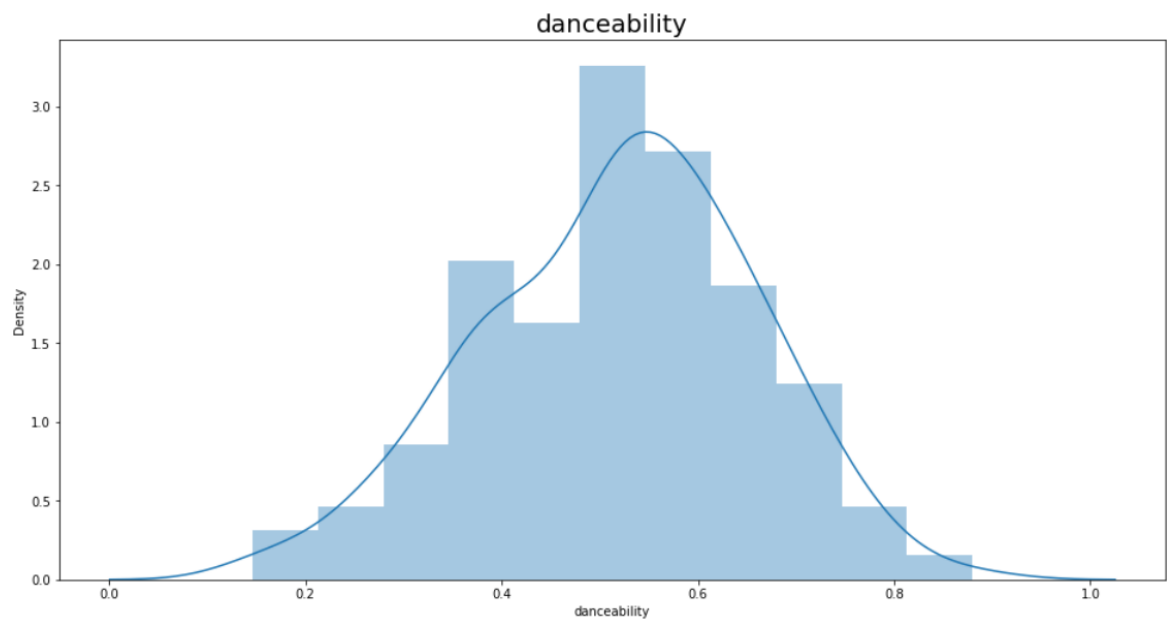
2.2.4 Song

```
[ ] data.song

0      I Saw Her Standing There
1                      Misery
2                      Anna
3                      Chains
4                      Boys
...
188          ive got a feeling
189          one after 909
190      the long and winding road
191          for you blue
192          get back
Name: song, Length: 193, dtype: object
```

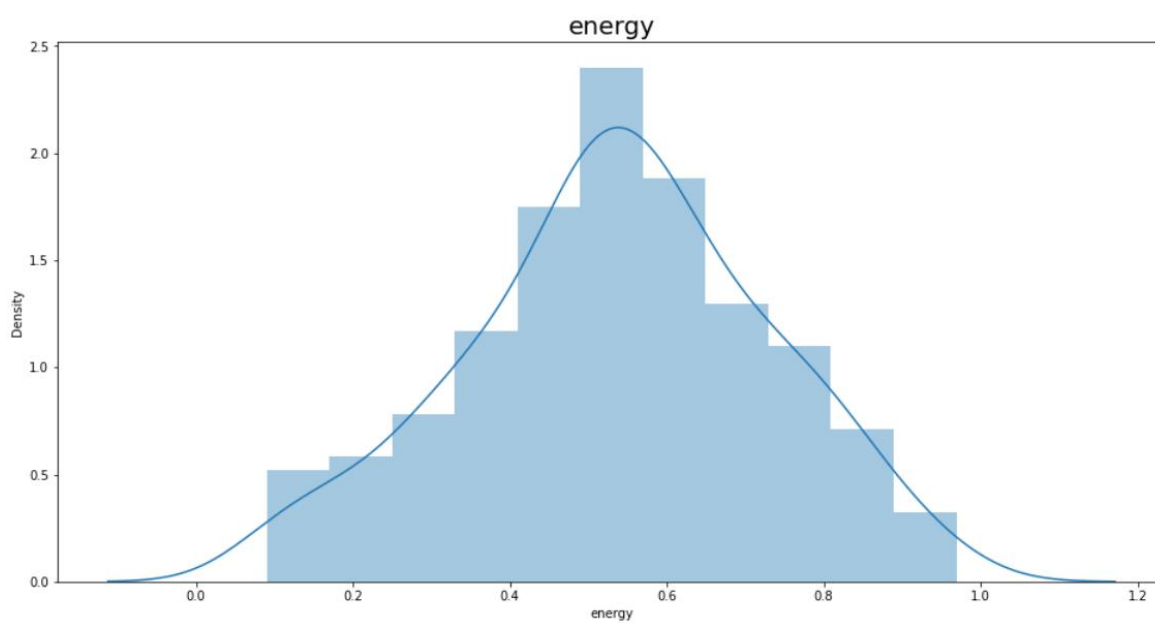
2.2.5 Danceability

```
[ ] plt.figure(figsize = (16, 8))  
    plt.title('danceability', fontsize = 20)  
    sns.distplot(data['danceability'])  
    plt.show()
```



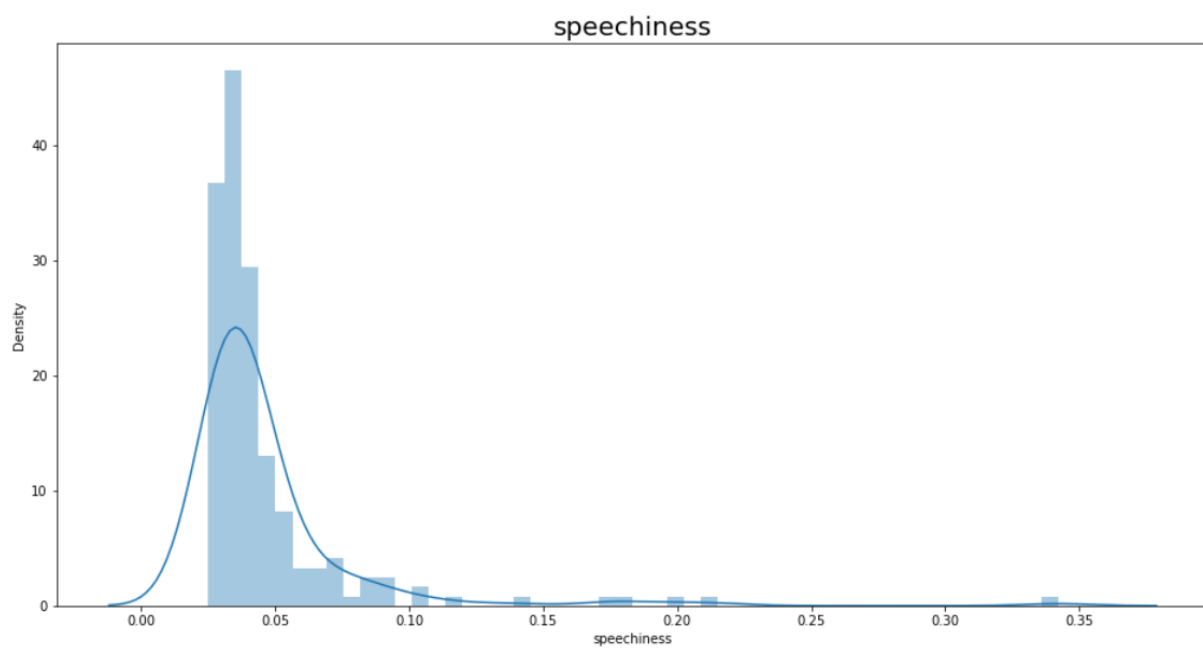
2.2.6 Energy

```
[ ] plt.figure(figsize = (16, 8))  
    plt.title('energy',fontsize = 20)  
    sns.distplot(data['energy'])  
    plt.show()
```



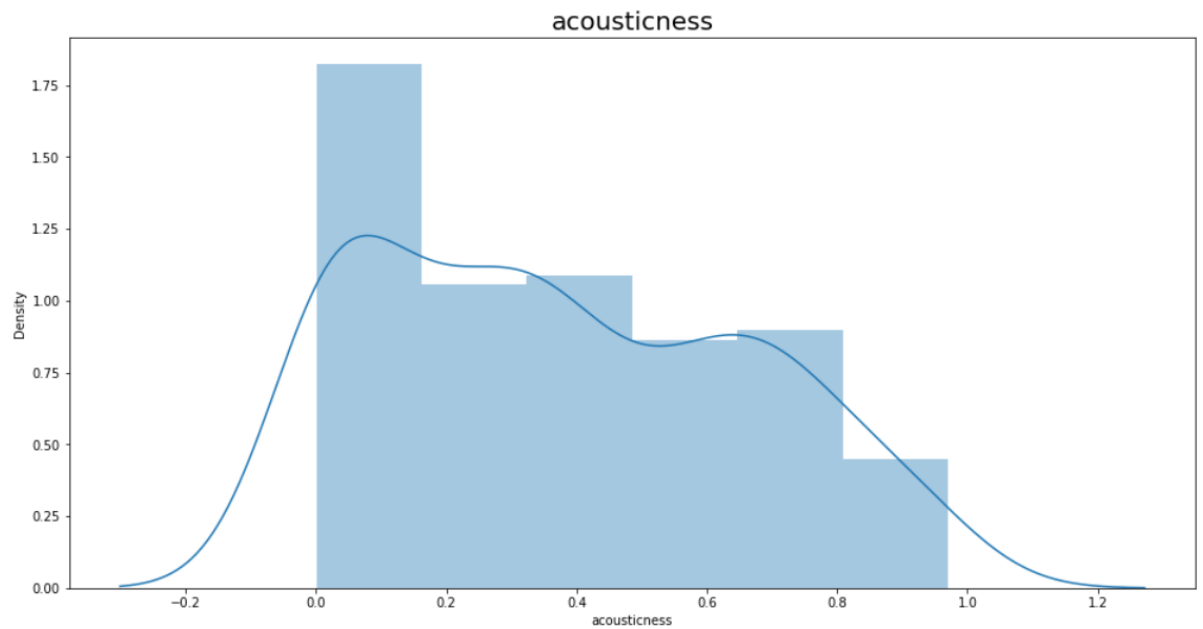
2.2.7 Speechiness

```
[ ] plt.figure(figsize = (16, 8))  
    plt.title('speechiness',fontsize = 20)  
    sns.distplot(data['speechiness'])  
    plt.show()
```



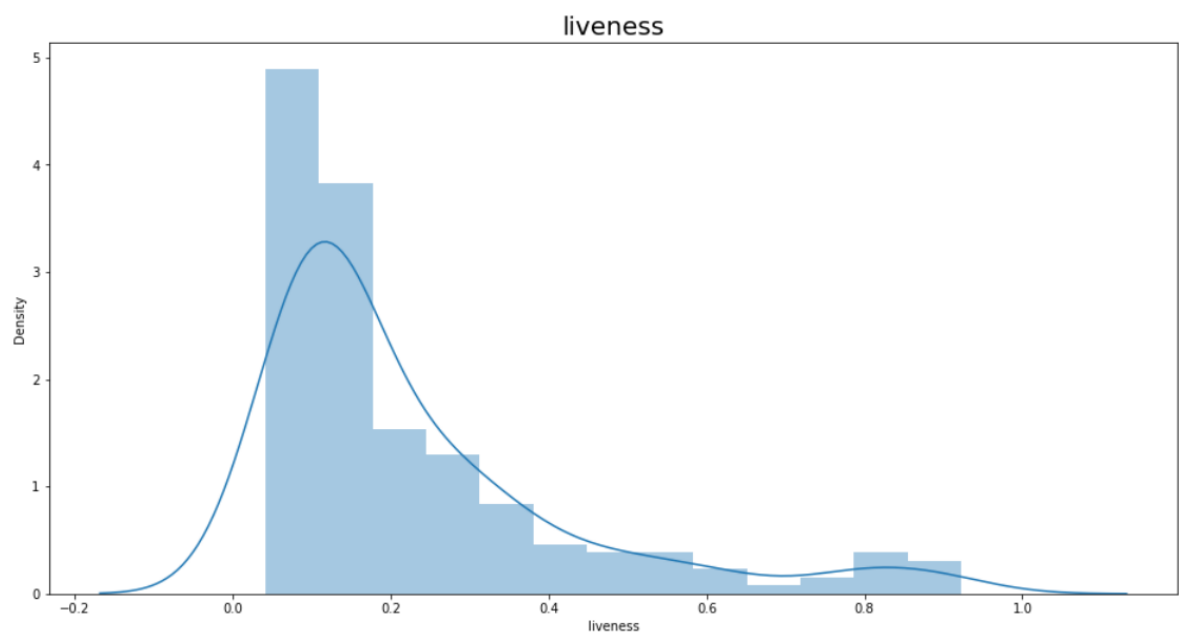
2.2.8 Acousticness

```
[ ] plt.figure(figsize = (16, 8))  
    plt.title('acousticness',fontsize = 20)  
    sns.distplot(data['acousticness'])  
    plt.show()
```



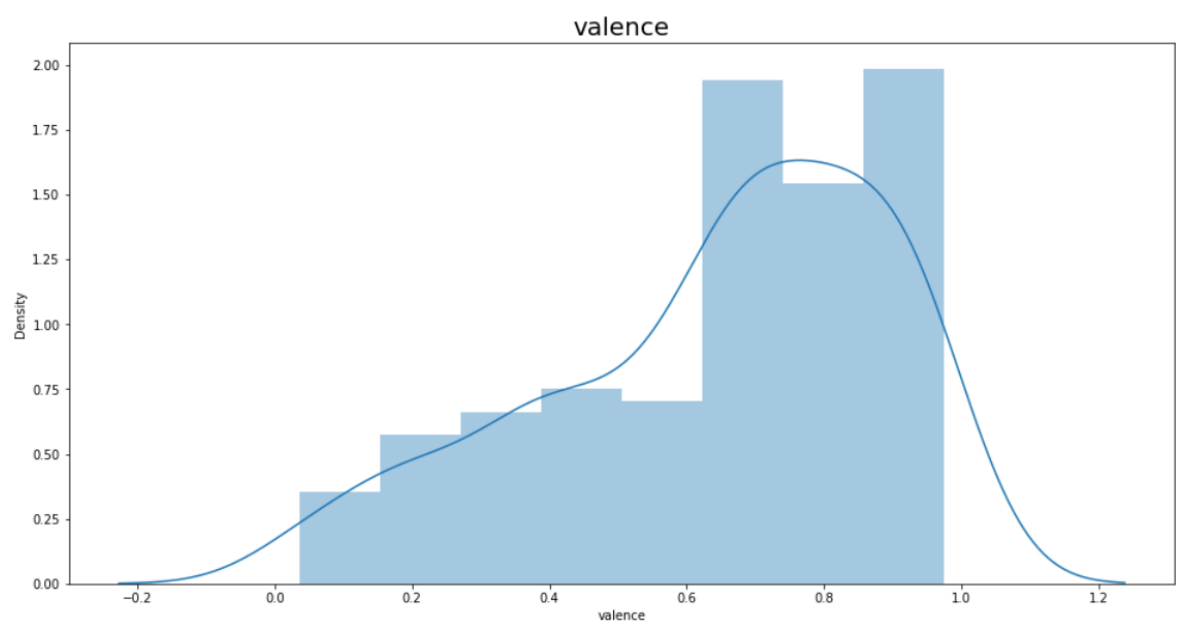
2.2.9 Liveness

```
[ ] plt.figure(figsize = (16, 8))  
    plt.title('liveness',fontsize = 20)  
    sns.distplot(data['liveness'])  
    plt.show()
```



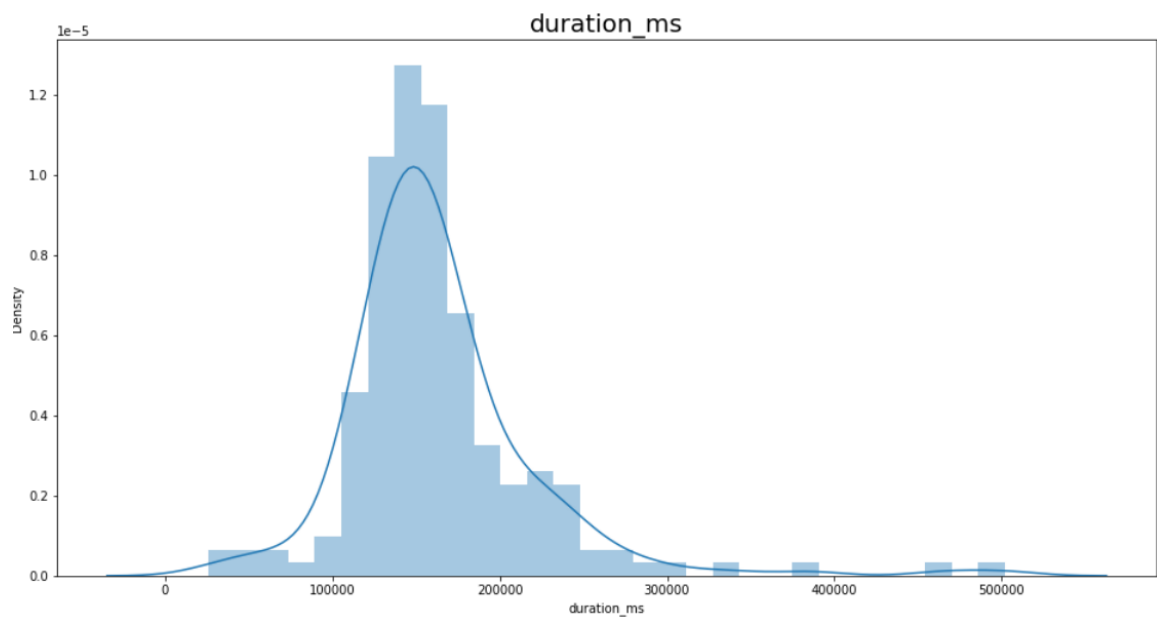
2.2.10 Valence

```
[ ] plt.figure(figsize = (16, 8))  
    plt.title('valence',fontsize = 20)  
    sns.distplot(data['valence'])  
    plt.show()
```



2.2.11 duration_ms

```
plt.figure(figsize = (16, 8))  
plt.title('duration_ms', fontsize = 20)  
sns.distplot(data['duration_ms'])  
plt.show()
```



2.3 อธิบายขั้นตอนย่อยแต่ละขั้น

2.3.1 การนำเข้าข้อมูลสู่ระบบ

2.3.1.1 การประมวลผลข้อมูลเบื้องต้น

สถิติข้อมูลในแต่ละรูปแบบ เช่น ค่าเฉลี่ย ค่าสูงสุด ค่าต่ำสุด จากข้อมูลทั้งหมด

	danceability	energy	speechiness	acousticness	liveness	valence	duration_ms
ค่าเฉลี่ย	0.519	0.537	0.046	0.376	0.229	0.642	163644.155
ค่าต่ำสุด	0.146	0.090	0.025	0.000	0.041	0.036	25987.000
ค่าสูงสุด	0.880	0.969	0.342	0.971	0.922	0.975	502013.000

2.3.2 การปรับแต่งข้อมูล

2.3.2.1 การ clean ข้อมูล

ก่อนการแก้ไขหรือปรับแต่งข้อมูล

มีจำนวนชุดข้อมูลทั้งหมด 193 แถว 17 คอลัมน์ มีรูปแบบข้อมูลดังนี้

1. Id 2. Year 3. Album 4. Song 5. Danceability 6. Energy
7. Key 8. Loudness 9. mode 10. Speechiness
11. Acousticness 12. Instrumentalness 13. Liveness
14. Valence 15. Tempo 16. Spotify_id 17. duration_ms

หลังการแก้ไขหรือปรับแต่งข้อมูล

มีจำนวนชุดข้อมูลทั้งหมด 193 แถว 11 คอลัมน์ ทำการตัด key/loudness/
mode/instrumentalness/tempo/spotify_id ออกไป ทำให้มีรูปแบบข้อมูลดังนี้

1. Id 2. Year 3. Album 4. Song 5. Danceability 6. Energy
7. Speechiness 8. Acousticness 9. Liveness 10. Valence
11. duration_ms

ข้อมูลที่สมบูรณ์/ต้องแก้ไข/ไม่สมบูรณ์ต้องตัดทิ้ง

ข้อมูลก่อนการแก้ไข : ข้อมูลแถวที่ 111 คอลัมน์ที่ 6 (energy) ค่าในช่องใส่มาเป็น

).613 ซึ่งทำให้ไม่สามารถนำไปคำนวณต่อได้

id	year	album	song	danceability	energy	speechiness	acousticness	liveness	valence	duration_ms
111	1967	Magical Mystery Tour	Magical mystery tour	0.297).613	0.0931	0.0385	0.0734	0.338	170107

ข้อมูลหลังการแก้ไข : ทำการเปลี่ยน) ไปเป็นเลข 0

id	year	album	song	danceability	energy	speechiness	acousticness	liveness	valence	duration_ms
111	1967	Magical Mystery Tour	Magical mystery tour	0.297	0.613	0.0931	0.0385	0.0734	0.338	170107

2.3.3 การวิเคราะห์ชุดข้อมูล

2.3.3.1 การนำข้อมูลเข้า google colab

ขั้นตอนที่ 1 : เพิ่ม library ต่างๆที่ใช้ในการคำนวณระบบและนำข้อมูลที่ทำการ clean แล้ว

เข้ามา

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from tqdm import tqdm
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import RobustScaler
import math

import warnings
warnings.filterwarnings('ignore')
```

url=https://raw.githubusercontent.com/PlasticxPlastic/project_linear_algebra/main/TheBeatlesCleaned.csv

[eba/main/TheBeatlesCleaned.csv](https://raw.githubusercontent.com/PlasticxPlastic/project_linear_algebra/main/TheBeatlesCleaned.csv)

```
[ ] url = "https://raw.githubusercontent.com/PlasticxPlastic/project_linear_algebra/main/TheBeatlesCleaned.csv"
dataset = pd.read_csv(url)
dataset.head()
```

	id	year	album	song	danceability	energy	speechiness	acousticness	liveness	valence	duration_ms
0	1	1963	Please Please Me	I Saw Her Standing There	0.491	0.801	0.0361	0.270	0.0665	0.971	173947
1	2	1963	Please Please Me	Misery	0.591	0.605	0.0413	0.707	0.3090	0.882	108547
2	3	1963	Please Please Me	Anna	0.608	0.565	0.0336	0.635	0.0601	0.835	177133
3	4	1963	Please Please Me	Chains	0.654	0.561	0.0304	0.608	0.1290	0.929	145080
4	5	1963	Please Please Me	Boys	0.402	0.860	0.0504	0.607	0.7360	0.822	146440

ขั้นตอนที่ 2 : หาขนาดของ dataset ซึ่งมีทั้งหมด 193 rows 11 columns

```
[ ] dataset.shape

(193, 11)
```

ขั้นตอนที่ 3 : แสดงข้อมูลของdatasetที่นำเข้ามา

```
[ ] dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 193 entries, 0 to 192
Data columns (total 11 columns):
#   Column              Non-Null Count  Dtype  
---  -
0   id                   193 non-null   int64  
1   year                 193 non-null   int64  
2   album                193 non-null   object  
3   song                 193 non-null   object  
4   danceability         193 non-null   float64 
5   energy               193 non-null   float64 
6   speechiness          193 non-null   float64 
7   acousticness         193 non-null   float64 
8   liveness             193 non-null   float64 
9   valence              193 non-null   float64 
10  duration_ms          193 non-null   int64  
dtypes: float64(6), int64(3), object(2)
memory usage: 16.7+ KB
```

ขั้นตอนที่ 4 : จัดระเบียบข้อมูล

```
[ ] def cname(dataset):
    dt = dataset.copy()
    for i in range(0, dataset.shape[0]):
        dt["cname"] = dt["song"] + ' : ' + dt["album"]
    return dt

[ ] data = cname(dataset)

[ ] data.head()

   id  year  album  song  danceability  energy  speechiness  acousticness  liveness  valence  duration_ms  cname
0    1  1963  Please Please Me  I Saw Her Standing There  0.491  0.801  0.0361  0.270  0.0665  0.971  173947  I Saw Her Standing There : Please Please Me
1    2  1963  Please Please Me  Misery  0.591  0.605  0.0413  0.707  0.3090  0.882  108547  Misery : Please Please Me
2    3  1963  Please Please Me  Anna  0.608  0.565  0.0336  0.635  0.0601  0.835  177133  Anna : Please Please Me
3    4  1963  Please Please Me  Chains  0.654  0.561  0.0304  0.608  0.1290  0.929  145080  Chains : Please Please Me
4    5  1963  Please Please Me  Boys  0.402  0.860  0.0504  0.607  0.7360  0.822  146440  Boys : Please Please Me

[ ] data.drop(columns = ["id"])
data["ids"]=[i for i in range(0,data.shape[0])]
```

ขั้นตอนที่ 5 : นับจำนวนข้อมูลทั้งหมด

```
[ ] data.count()

id          193
year        193
album       193
song        193
danceability 193
energy       193
speechiness  193
acousticness 193
liveness     193
valence      193
duration_ms  193
cname        193
ids          193
dtype: int64
```

```
[ ] data.song

0      I Saw Her Standing There
1                      Misery
2                      Anna
3                      Chains
4                      Boys
...
188      ive got a feeling
189      one after 909
190      the long and winding road
191      for you blue
192      get back
Name: song, Length: 193, dtype: object
```

ขั้นตอนที่ 6 : เช็คว่ามีข้อมูลตัวไหนที่ขาดหายหรือไม่สมบูรณ์หรือไม่

```
[ ] data.isnull().any()
```

```
id          False
year        False
album       False
song        False
danceability False
energy      False
speechiness False
acousticness False
liveness    False
valence     False
duration_ms False
curname     False
ids         False
dtype: bool
```

ขั้นตอนที่ 7 : นับจำนวนเพลงทั้งหมดในแต่ละ album

```
[ ] data["album"].value_counts()
```

```
The Beatles (white album)    30
Abbey Road                   17
Please Please Me              14
With The Beatles             14
Beatles for Sale             14
Help!                        14
Rubber Soul                  14
Revolver                     14
A Hard Day's Night           13
Sgt. Pepper's Lonely Hearts Club Band 13
Yellow Submarine             13
Let It Be                    12
Magical Mystery Tour         11
Name: album, dtype: int64
```

Drop column ID, year, album, song, duration_ms off

ขั้นตอนที่ 8 : ใส่ column ID, year, album, song, duration_ms และจัดข้อมูลในรูปแบบ

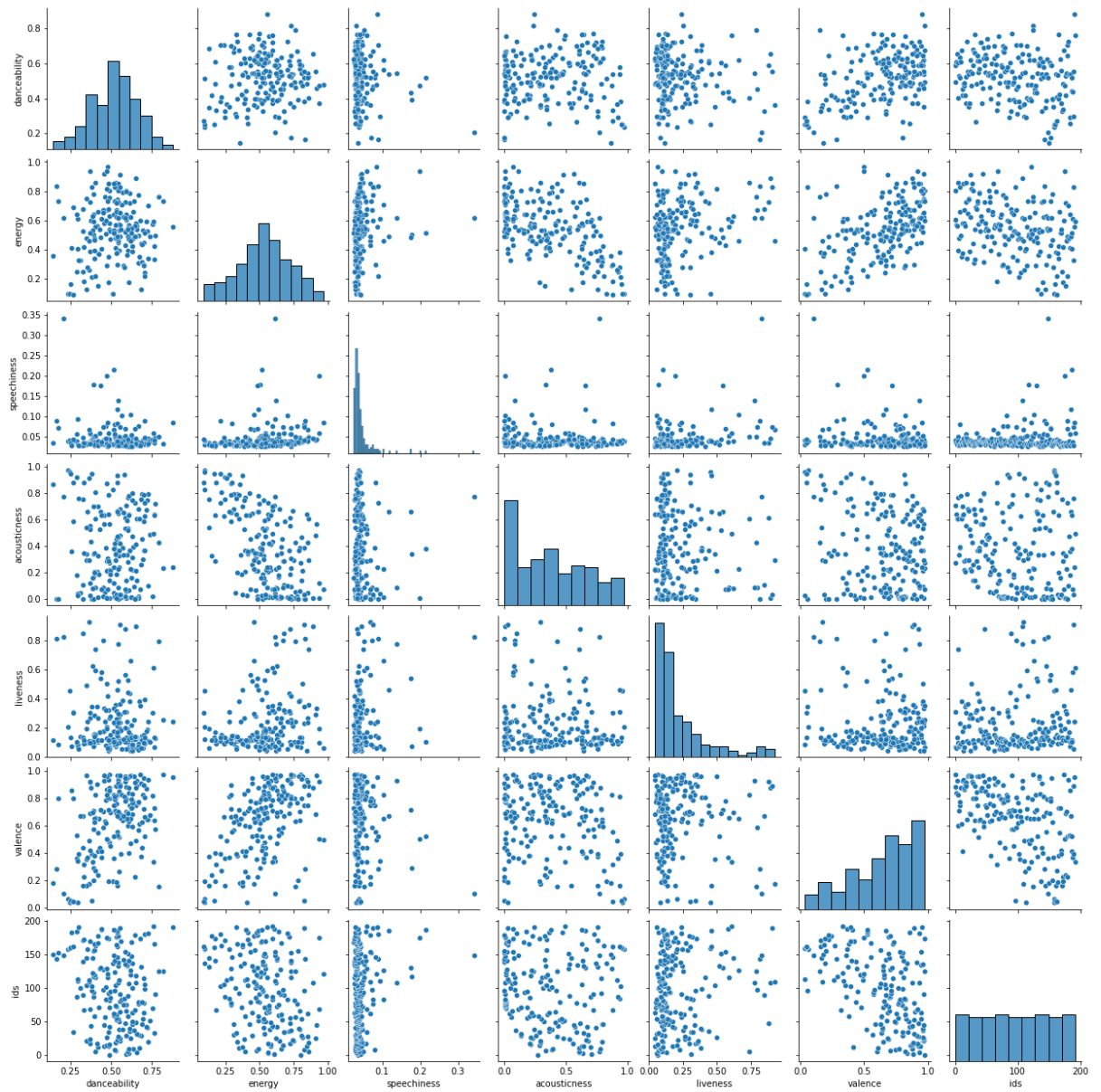
เมทริกซ์ก่อนการประมวลผล

```
[ ] u = data.copy()
    df = data.drop(columns=['id', 'year', 'album', 'song', 'duration_ms', 'ids', 'curname'])
```

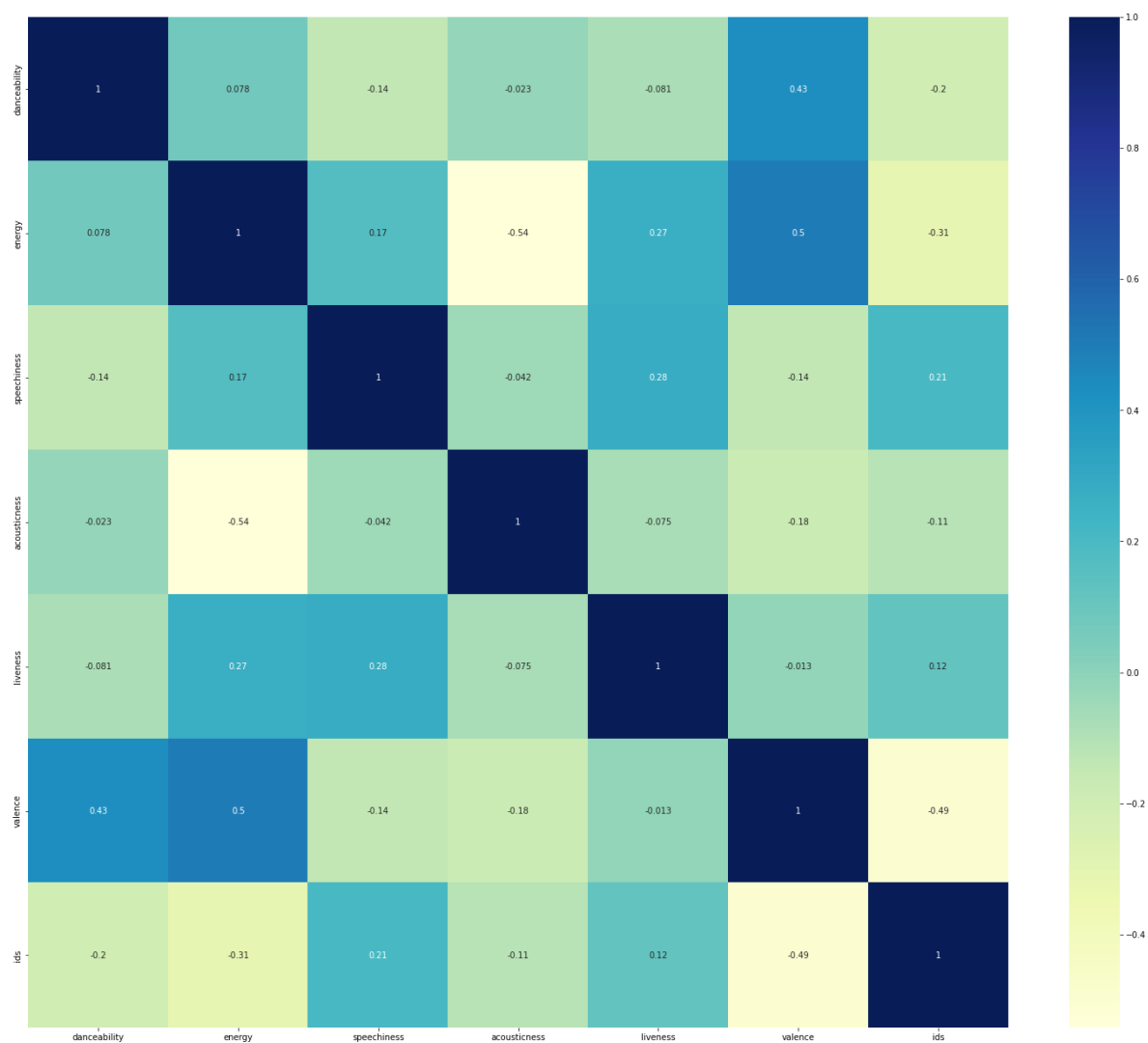
```
[ ] np.matrix(df)
```

```
matrix([[0.491 , 0.801 , 0.0361, 0.27 , 0.0665, 0.971 ],
        [0.591 , 0.605 , 0.0413, 0.707 , 0.309 , 0.882 ],
        [0.608 , 0.565 , 0.0336, 0.635 , 0.0601, 0.835 ],
        ...,
        [0.299 , 0.329 , 0.0279, 0.756 , 0.0559, 0.392 ],
        [0.88 , 0.556 , 0.0855, 0.24 , 0.24 , 0.955 ],
        [0.761 , 0.592 , 0.0586, 0.492 , 0.61 , 0.332 ]])
```

2.3.3.2 การวิเคราะห์ชุดข้อมูล



การเปรียบเทียบข้อมูลแต่ละตัวในชุดข้อมูล



การนำชุดข้อมูลมาสร้าง correlation graph และแสดงอยู่ในรูปของ Matrix


```
[ ] from sklearn.feature_extraction.text import TfidfVectorizer
vec=TfidfVectorizer()
test_data = u.drop(columns=['id', 'year', 'album','song','duration_ms',])
```

```
[ ] test_data
```

	danceability	energy	speechiness	acousticness	liveness	valence	cbname	ids
0	0.491	0.801	0.0361	0.2700	0.0665	0.971	I Saw Her Standing There : Please Please Me	0
1	0.591	0.605	0.0413	0.7070	0.3090	0.882	Misery : Please Please Me	1
2	0.608	0.565	0.0336	0.6350	0.0601	0.835	Anna : Please Please Me	2
3	0.654	0.561	0.0304	0.6080	0.1290	0.929	Chains : Please Please Me	3
4	0.402	0.860	0.0504	0.6070	0.7360	0.822	Boys : Please Please Me	4
...
188	0.440	0.609	0.0358	0.0715	0.5820	0.364	ive got a feeling : Let It Be	188
189	0.554	0.828	0.0739	0.0307	0.9070	0.888	one after 909 : Let It Be	189
190	0.299	0.329	0.0279	0.7560	0.0559	0.392	the long and winding road : Let It Be	190
191	0.880	0.556	0.0855	0.2400	0.2400	0.955	for you blue : Let It Be	191
192	0.761	0.592	0.0586	0.4920	0.6100	0.332	get back : Let It Be	192

193 rows × 8 columns

สร้าง vector ของข้อมูลแต่ละชุด

```
[ ] float(test_data['danceability'][0])
```

0.491

```
[ ] from numpy.linalg import norm
```

```
[ ] def recommend(dcbt, eng, sphn, acct, livn, val):
    test_array = np.array([dcbt, eng, sphn, acct, livn, val])
    score_lst = []
    for i in range(193):
        train_array = np.array([float(test_data['danceability'][i]),float(test_data['energy'][i]),
                                float(test_data['speechiness'][i]),float(test_data['acousticness'][i]),
                                float(test_data['liveness'][i]),float(test_data['valence'][i])])
        cosine = np.dot(test_array,train_array)/(norm(test_array)*norm(train_array))
        score_lst.append(cosine)
    return score_lst
```

สร้างฟังก์ชันแนะนำเพลงออกมา

```

def user_input():
    print("This is the beatle recommendation system we can recommend you the beatle song !!! ")
    print()
    print("we just need to know some information")
    print()
    print("1) How would you rate your taste with instrumental song (1 to 10 or 0 if you dont mind) : ")
    speechiness = int(input())
    if speechiness != 0:
        speechiness = TFnumber(speechiness)
        speechiness = float(0.3173 / speechiness) #0.3173 is max(speechiness) - min(speechiness)
    else:
        speechiness = 0.04 #mean(speechiness)
    print("2) How would you rate your taste with danceable song (1 to 10 or 0 if you dont mind) : ")
    danceability = int(input())
    if danceability != 0:
        danceability = TFnumber(danceability)
        danceability = float(0.734 / danceability) #0.734 is max(danceability) - min(danceability)
    else:
        danceability = 0.51 #mean(danceability)
    print("3) How would you rate your taste with live record song (1 to 10 or 0 if you dont mind) : ")
    liveness = int(input())
    if liveness != 0:
        liveness = TFnumber(liveness)
        liveness = float(0.88 / liveness) #0.88 is max(liveness) - min(liveness)
    else:
        liveness = 0.22 #mean(liveness)
    print("4) How would you rate your taste with accoustic song (1 to 10 or 0 if you dont mind) : ")
    accousticness = int(input())
    if accousticness != 0:
        accousticness = TFnumber(accousticness)
        accousticness = float(0.97 / accousticness) #0.88 is max(accousticness) - min(accousticness)
    else:
        accousticness = 0.37 #mean(accousticness)
    print("5) How would you rate your taste with positive song (1 to 10 or 0 if you dont mind) : ")
    valence = int(input())
    if valence != 0:
        valence = TFnumber(valence)
        valence = float(0.9393 / valence) #0.9393 is max(valence) - min(valence)
    else:
        valence = 0.64 #mean(valence)
    print("6) How would you rate your taste with high energy song (1 to 10 or 0 if you dont mind) : ")
    energy = int(input())
    if energy != 0:
        energy = TFnumber(energy)
        energy = float(0.8788 / energy) #0.8788 is max(energy) - min(energy)
    else:
        energy = 0.53 #mean(energy)
    return speechiness,danceability,liveness,accousticness,valence,energy

```

```

def TFnumber(number):
    if number == 1:
        return int(10)
    if number == 2:
        return int(9)
    if number == 3:
        return int(8)
    if number == 4:
        return int(7)
    if number == 5:
        return int(6)
    if number == 6:
        return int(5)
    if number == 7:
        return int(4)
    if number == 8:
        return int(3)
    if number == 9:
        return int(2)
    if number == 10:
        return int(1)

```

สร้างฟังก์ชันให้ผู้ใช้กรอกข้อมูลเบื้องต้นเพื่อนำไปคำนวณหา vector Cosine similarity

```


    @@markdown This is the beatle recommendation system we can recommend you the beatle song !!!
    @@markdown we just need to know some information
    speechiness = 3 #@param {type:"slider",min:1,max:10,step:1}
    speechiness = TFnumber(speechiness)
    speechiness = float(0.3173 / speechiness) #0.3173 is max(speechiness) - min(speechiness)
    dmspeech = False #@param {type:"boolean"}
    if dmspeech == True:
        speechiness = 0.04 #mean(speechiness)

    @@markdown hello
    danceability = 2 #@param {type:"slider",min:1,max:10,step:1}
    danceability = TFnumber(danceability)
    danceability = float(0.734 / danceability) #0.734 is max(danceability) - min(danceability)
    dmdance = False #@param {type:"boolean"}
    if dmdance == True:
        danceability = 0.51 #mean(danceability)

    @@markdown hello
    liveness = 4 #@param {type:"slider",min:1,max:10,step:1}
    liveness = TFnumber(liveness)
    liveness = float(0.88 / liveness) #0.88 is max(liveness) - min(liveness)
    dmlive = False #@param {type:"boolean"}
    if dmlive == True:
        liveness = 0.22 #mean(liveness)

    @@markdown hello
    accousticness = 7 #@param {type:"slider",min:1,max:10,step:1}
    accousticness = TFnumber(accousticness)
    accousticness = float(0.97 / accousticness) #0.88 is max(accousticness) - min(accousticness)
    dmaccousticness = False #@param {type:"boolean"}
    if dmaccousticness == True:
        accousticness = 0.37 #mean(accousticness)

    @@markdown hello
    valence = 8 #@param {type:"slider",min:1,max:10,step:1}
    valence = TFnumber(valence)
    valence = float(0.9393 / valence) #0.9393 is max(valence) - min(valence)
    dmvalence = False #@param {type:"boolean"}
    if dmvalence == True:
        valence = 0.64 #mean(valence)

    @@markdown hello
    energy = 5 #@param {type:"slider",min:1,max:10,step:1}
    energy = TFnumber(energy)
    energy = float(0.8788 / energy) #0.8788 is max(energy) - min(energy)
    dmenenergy = False #@param {type:"boolean"}
    if dmenenergy == True:
        energy = 0.53 #mean(energy)

```

```

x = recommend(danceability,energy,speechiness,accousticness,liveness,valence)
y = max(x)
for i in range(len(x)):
    if x[i] == y:
        highest = i
data['cname'][highest]

```

ฟังก์ชันการแสดงผล

2.3.4 Model Hit My Beatles

The Beatles Recommendation

This is the beatle recommendation system we can recommend you the beatle song !!!

we just need to know some information

1) How would you rate your taste with instrumental song (1 to 10 or tick box below if you dont mind)

speechiness:

dspeech: ☐

2) How would you rate your taste with danceable song (1 to 10 or tick box below if you dont mind)

danceability:

dmance: ☐

3) How would you rate your taste with live record song (1 to 10 or tick box below if you dont mind)

liveness:

dlive: ☐

4) How would you rate your taste with accoustic song (1 to 10 or tick box below if you dont mind)

acousticness:

dacousticness: ☐

5) How would you rate your taste with positive song (1 to 10 or tick box below if you dont mind)

valence:

dvalence: ☐

6) How would you rate your taste with high energy song (1 to 10 or tick box below if you dont mind)

energy:

denergy: ☐

'when I get home album : A Hard Day's Night'

การรับข้อมูลจากผู้ใช้งานมาสู่ Model การแนะนำเพลงและแสดงผลการแนะนำเพลง

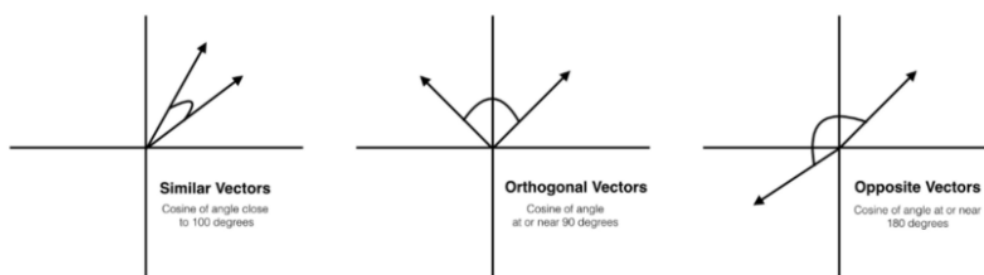
บทที่ 3

การประยุกต์ใช้ทฤษฎี

3.1 การประยุกต์ใช้ทฤษฎีเวกเตอร์

3.1.1 Cosine Similarity

นำ Cosine Similarity มาใช้เพื่อแนะนำเพลงของวง The Beatles ว่าข้อมูลที่ใช้กรอกเข้ามามีค่าใกล้เคียงกับค่าชุดข้อมูลของแต่ละเพลงไหนมากที่สุด



วิธีนี้คือการหาความคล้ายคลึงด้วยองศา เป็นการวัดความเหมือนของ Vector 2 อันว่าไปในทิศทางเดียวกันหรือไม่ โดยที่เป็นการตัดขนาด หรือ Magnitude ของ Vector ออกไป

โดยการคำนวณ cosine similarity จะใช้สูตรดังนี้

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

จากสมการการคำนวณหา Cosine Similarity ที่ได้กล่าวมาเมื่อนำมาใช้ในระบบมีการเขียนโปรแกรมคำนวณดังนี้

```
[ ] from sklearn.feature_extraction.text import TfidfVectorizer
vec=TfidfVectorizer()
test_data = u.drop(columns=['id', 'year', 'album','song','duration_ms',])
```

```
[ ] test_data
```

	danceability	energy	speechiness	acousticness	liveness	valence	cbname	ids
0	0.491	0.801	0.0361	0.2700	0.0665	0.971	I Saw Her Standing There : Please Please Me	0
1	0.591	0.605	0.0413	0.7070	0.3090	0.882	Misery : Please Please Me	1
2	0.608	0.565	0.0336	0.6350	0.0601	0.835	Anna : Please Please Me	2
3	0.654	0.561	0.0304	0.6080	0.1290	0.929	Chains : Please Please Me	3
4	0.402	0.860	0.0504	0.6070	0.7360	0.822	Boys : Please Please Me	4
...
188	0.440	0.609	0.0358	0.0715	0.5820	0.364	ive got a feeling : Let It Be	188
189	0.554	0.828	0.0739	0.0307	0.9070	0.888	one after 909 : Let It Be	189
190	0.299	0.329	0.0279	0.7560	0.0559	0.392	the long and winding road : Let It Be	190
191	0.880	0.556	0.0855	0.2400	0.2400	0.955	for you blue : Let It Be	191
192	0.761	0.592	0.0506	0.4920	0.6100	0.332	get back : Let It Be	192

193 rows x 8 columns

สร้าง vector ของข้อมูลแต่ละชุด

```
[ ] float(test_data["danceability"][0])
```

```
0.491
```

```
[ ] from numpy.linalg import norm
```

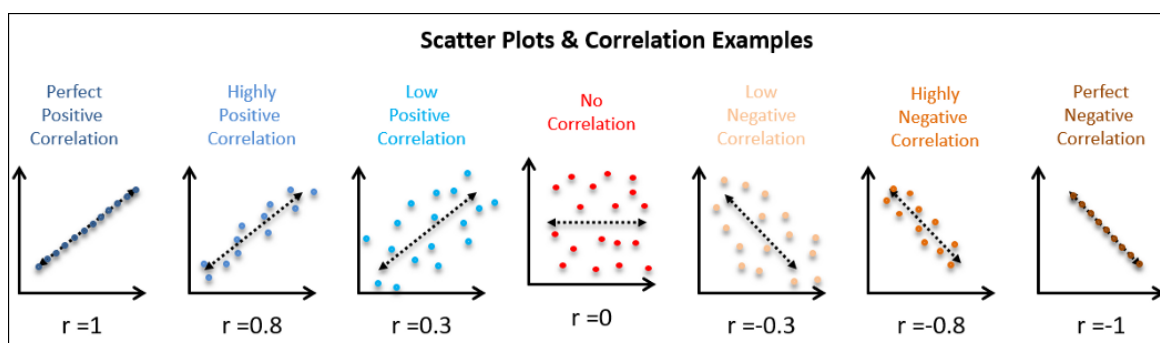
```
[ ] def recommend(dcbt, eng, sphn, acct, livn, val):
test_array = np.array([dcbt, eng, sphn, acct, livn, val])
score_lst = []
for i in range(193):
train_array = np.array((float(test_data["danceability"][i]),float(test_data["energy"][i]),
float(test_data["speechiness"][i]),float(test_data["acousticness"][i]),
float(test_data["liveness"][i]),float(test_data["valence"][i])))
cosine = np.dot(test_array,train_array)/(norm(test_array)*norm(train_array))
score_lst.append(cosine)
return score_lst
```

สร้างฟังก์ชันแนะนำเพลงออกมาด้วยทฤษฎี Cosine similarity

3.2 การประยุกต์ใช้ทฤษฎีเมทริกซ์

3.2.1 Correlation Matrix

คือการหาความสัมพันธ์แบบเส้นตรงของตัวแปร 2 ตัวหรือมากกว่า โดยค่าที่ได้จะเรียกว่าค่า สัมประสิทธิ์สหสัมพันธ์ (Correlation Coefficient) ถ้าหากค่า Correlation Coefficient มีค่ามาก ก็ยิ่งมีความสัมพันธ์กันมาก โดยค่าทางด้านบวกคือมีความสัมพันธ์กันแบบแปรผัน (Positive Correlation) และค่าลบคือมีความสัมพันธ์กันแบบผกผัน (Negative Correlation)



จะเห็นว่า ถ้าหากค่า Correlation มีค่าเท่ากับ 1 หมายความว่า 2 ตัวแปรนั้นมีความสัมพันธ์กันอย่างสมบูรณ์ เมื่อนำทั้ง 2 ตัวแปรมาสร้าง Scatterplot จะเห็นว่าสามารถสร้างเส้นตรงได้พอดี

Pearson Correlation — ใช้หาความสัมพันธ์แบบเส้นตรงของตัวแปรต่อเนื่อง (Continuous Variables)

สูตรของ Pearson Correlation คือ

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Where,

r = Pearson Correlation Coefficient

x_i = x variable samples

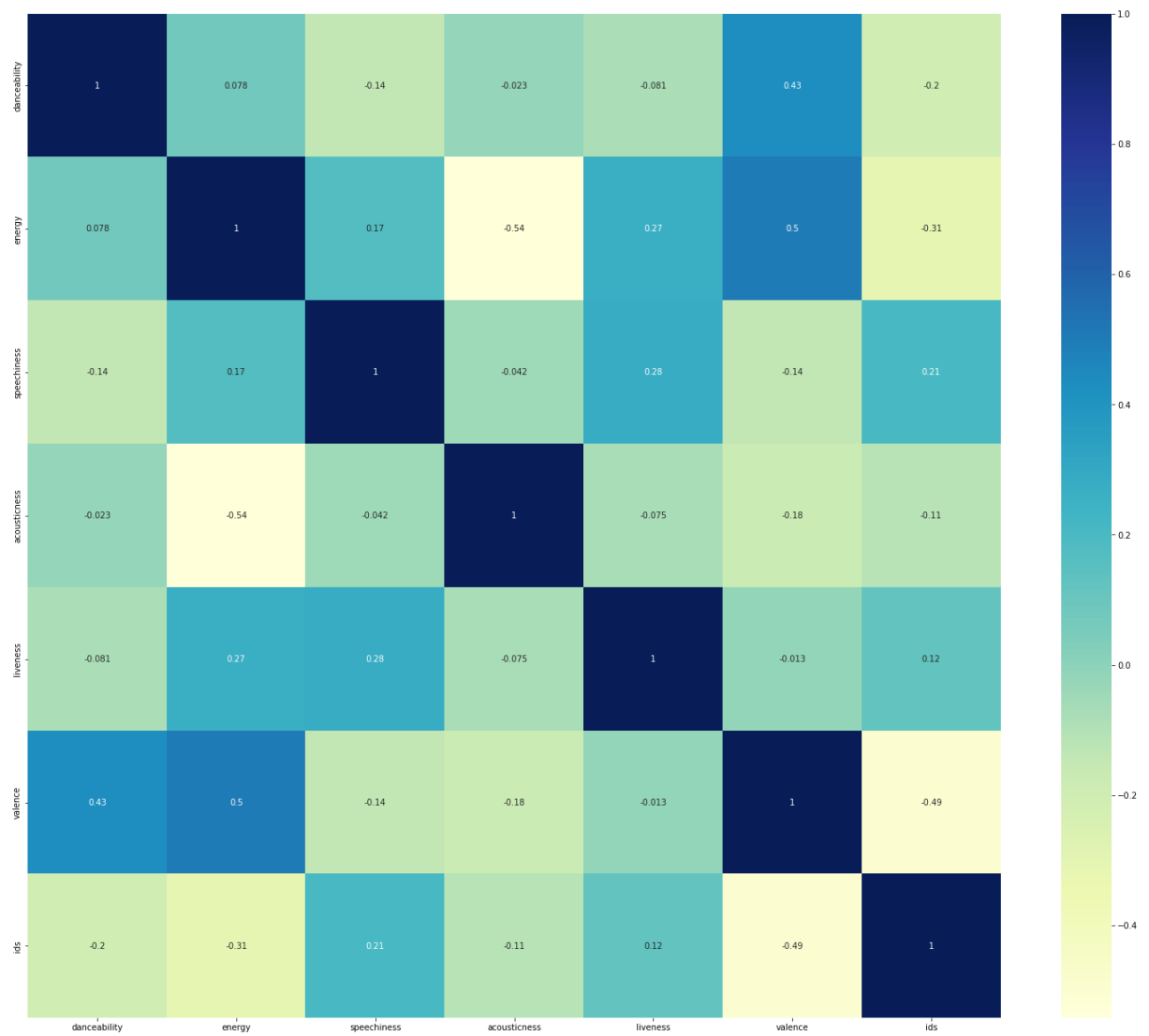
y_i = y variable sample

\bar{x} = mean of values in x variable

\bar{y} = mean of values in y variable

เมื่อนำมาใช้ในระบบจะใช้ในการหาความสัมพันธ์ของข้อมูลที่แต่ละตัวในชุดข้อมูลว่ามีความสัมพันธ์กันมากน้อยแค่ไหน โดยจะแสดงอยู่ในรูปแบบของ Matrix ดังรูปต่อไปนี้

```
[ ] corr = df.corr()
    indx = corr.index
    plt.figure(figsize=(26,22))
    sns.heatmap(df[indx].corr(),annot=True,cmap="YlGnBu")
```

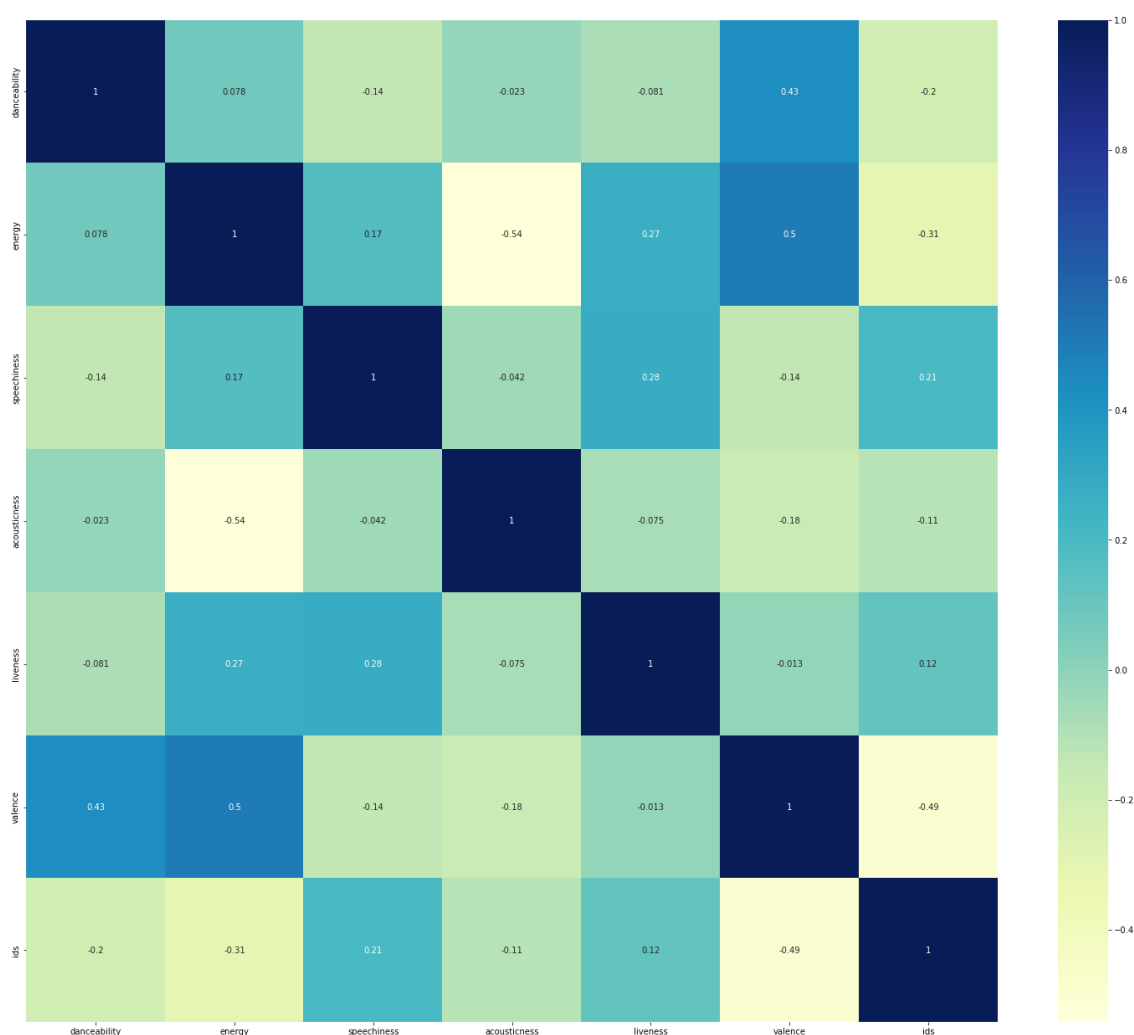


บทที่ 4

ผลการทดลอง

จากขั้นตอนการทำงานทั้งหมด ตั้งแต่การนำเข้าข้อมูลไปจนถึงการ Training ชุดข้อมูลเพื่อสร้าง Model การแนะนำเพลง มีผลการทดลองทั้งหมดดังนี้

4.1 Correlations



ผลการทดลอง จากรูป Matrix ข้างต้นจะเห็นได้ว่าค่าของ Correlations ที่แต่ละในชุดข้อมูลมาเทียบกันนั้นมีค่าที่ใกล้เคียงกันมาก ดังนั้นชุดข้อมูลนี้จึงสามารถนำไปสร้าง Model ที่มีค่าความผิดพลาดที่น้อยมากๆได้

4.2 Cosine Similarity

```

@@title The Beatles Recommendation

@markdown This is the beatle recommendation system we can recommend you the beatle song !!!

@markdown we just need to know some information

@markdown 1) How would you rate your taste with instrumental song (1 to 10 or tick box below if you dont mind)
speechiness = 6 #@param {type:"slider",min:1,max:10,step:1}
speechiness = TFnumber(speechiness)
speechiness = float(0.3173 / speechiness) #0.3173 is max(speechiness) - min(speechiness)
dmspeech = False #@param {type:"boolean"}
if dmspeech == True:
    speechiness = 0.04 #mean(speechiness)

@markdown 2) How would you rate your taste with danceable song (1 to 10 or tick box below if you dont mind)
danceability = 7 #@param {type:"slider",min:1,max:10,step:1}
danceability = TFnumber(danceability)
danceability = float(0.734 / danceability) #0.734 is max(danceability) - min(danceability)
dmdance = False #@param {type:"boolean"}
if dmdance == True:
    danceability = 0.51 #mean(danceability)

@markdown 3) How would you rate your taste with live record song (1 to 10 or tick box below if you dont mind)
liveness = 5 #@param {type:"slider",min:1,max:10,step:1}
liveness = TFnumber(liveness)
liveness = float(0.88 / liveness) #0.88 is max(liveness) - min(liveness)
dmlive = False #@param {type:"boolean"}
if dmlive == True:
    liveness = 0.22 #mean(liveness)

@markdown 4)How would you rate your taste with accoustic song (1 to 10 or tick box below if you dont mind)
accousticness = 5 #@param {type:"slider",min:1,max:10,step:1}
accousticness = TFnumber(accousticness)
accousticness = float(0.97 / accousticness) #0.88 is max(accousticness) - min(accousticness)
dmacousticness = False #@param {type:"boolean"}
if dmacousticness == True:
    accousticness = 0.37 #mean(accousticness)

@markdown 5)How would you rate your taste with positive song (1 to 10 or tick box below if you dont mind)
valence = 5 #@param {type:"slider",min:1,max:10,step:1}
valence = TFnumber(valence)
valence = float(0.9393 / valence) #0.9393 is max(valence) - min(valence)
dmvalence = False #@param {type:"boolean"}
if dmvalence == True:
    valence = 0.64 #mean(valence)

@markdown 6)How would you rate your taste with high energy song (1 to 10 or tick box below if you donit mind)
energy = 7 #@param {type:"slider",min:1,max:10,step:1}
energy = TFnumber(energy)
energy = float(0.8788 / energy) #0.8788 is max(energy) - min(energy)
dmenergy = False #@param {type:"boolean"}
if dmenergy == True:
    energy = 0.53 #mean(energy)

```

```

x = recommend(danceability,energy,speechiness,accousticness,liveness,valence)
y = max(x)
for i in range(len(x)):
    if x[i] == y:
        highest = i
data['cbyname'][highest]

```

ผลการทดลอง จากการใช้ทฤษฎี Cosine Similarity ในการเขียนฟังก์ชันแนะนำ

เพลงข้างต้นแล้วจะเห็นได้ว่าการเปรียบเทียบชุดข้อมูลที่มีอยู่กับชุดข้อมูลที่ได้รับมาจากผู้ใช้จะ

ทำให้เราได้ผลลัพธ์เป็นเพลงที่ใกล้เคียงกับความต้องการของผู้ใช้ออกมา 1 เพลง

4.3 Model Hit My Beatles

The Beatles Recommendation

This is the beattle recommendation system we can recommend you the beattle song !!!
we just need to know some information

1) How would you rate your taste with instrumental song (1 to 10 or tick box below if you dont mind)

speechiness: 4
dspeech: ☐

2) How would you rate your taste with danceable song (1 to 10 or tick box below if you dont mind)

danceability: 8
dm dance: ☐

3) How would you rate your taste with live record song (1 to 10 or tick box below if you dont mind)

liveness: 3
dalive: ☐

4) How would you rate your taste with accoustic song (1 to 10 or tick box below if you dont mind)

accousticness: 6
dmaccousticness: ☐

5) How would you rate your taste with positive song (1 to 10 or tick box below if you dont mind)

valence: 4
dmvalence: ☐

6) How would you rate your taste with high energy song (1 to 10 or tick box below if you dont mind)

energy: 10
denergy: ☐

[แสดงได้](#)

📁 'its all too much' album : Yellow Submarine'

จากการประยุกต์ใช้ทฤษฎี Correlation Matrix และ Cosine Similarity แล้วจะทำให้
ได้ผลลัพธ์ออกมาเป็นการแนะนำเพลงของวง The Beatles ออกมาอยู่ในรูปแบบ
ชื่อเพลง : ชื่ออัลบั้ม

4.4 ตัวอย่างการทำงานของ Mode Hit My Beatles

เริ่มจากการให้ผู้ใช้เลื่อน slides box ตอบคำถามเป็นจำนวน 6 ข้อโดยให้คะแนนตามอารมณ์ความรู้สึก ความชอบได้ตั้งแต่ 1 – 10 (หากไม่ชอบเลยสามารถเลือกลงในช่อง checkbox) ดังนี้

- 1) How would you rate your taste with instrumental song
- 2) How would you rate your taste with danceable song
- 3) How would you rate your taste with live record song
- 4)How would you rate your taste with accoustic song
- 5)How would you rate your taste with positive song
- 6)How would you rate your taste with high energy song

แล้วหลังจากนั้นระบบจะทำการคำนวณแล้วแสดงผลออกมาดังรูป

The Beatles Recommendation

This is the beatle recommendation system we can recommend you the beatle song !!!

we just need to know some information

1) How would you rate your taste with instrumental song (1 to 10 or tick box below if you dont mind)

speechiness: 5

dmspeech: ☐

2) How would you rate your taste with danceable song (1 to 10 or tick box below if you dont mind)

danceability: 7

dmdance: ☐

3) How would you rate your taste with live record song (1 to 10 or tick box below if you dont mind)

liveness: 1

dmlive: ☒

4)How would you rate your taste with accoustic song (1 to 10 or tick box below if you dont mind)

acousticness: 10

dmacousticness: ☐

5)How would you rate your taste with positive song (1 to 10 or tick box below if you dont mind)

valence: 10

dvalence: ☐

6)How would you rate your taste with high energy song (1 to 10 or tick box below if you don't mind)

energy: 5

dmenenergy: ☐

uassoได้

'martha my dear' album : The Beatles (white album)'

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการดำเนินการทั้งหมดที่ได้กล่าวมา ทางคณะผู้จัดทำได้ประยุกต์ความรู้ที่ได้ศึกษาในด้าน Vector และ Matrix มาใช้ให้ก่อนประโยชน์ได้อย่างเต็มที่จนเกิดเป็นโครงงานเรื่องนี้ขึ้นมา

การหาข้อมูล นำเข้าข้อมูล ปรับแต่งชุดข้อมูล รวมไปถึงการประมวลผลข้อมูล เมื่อผ่านการดำเนินงานทั้งหมดที่กล่าวมาทางคณะผู้จัดทำจึงได้ โปรแกรมแนะนำเพลงของวง The beatles โดยอิงจากความต้องการของผู้ฟังที่สามารถนำไปใช้ได้จริง ทางคณะผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงงานเรื่องนี้จะก่อประโยชน์ให้กับบุคคลที่ต้องการฟังเพลงของวง The beatles รวมไปถึงการต่อยอดความรู้ที่ได้จากโครงงานชิ้นนี้ไปใช้กับ โปรแกรม recommendation อื่นๆ ได้อีก

5.2 ข้อเสนอแนะ

5.2.1 ปัญหาที่พบ

1. Dataset ที่หาได้มีข้อมูลที่เยอะมากจนเกินไป หากใช้จะทำให้ตัวแอปที่สร้างเกิดการประมวลผลนานเกินที่ควรจะเป็น

แก้ไขโดย เปลี่ยนมาใช้ Dataset เพลงที่เจาะจงเฉพาะไปที่วง The Beatles ซึ่งมีจำนวนข้อมูลทั้งหมด 193 ข้อมูล

2. ไม่คุ้นชินกับเครื่องมือที่ใช้

แก้ไขโดย 1.ดูคลิปที่มีคนสอนวิธีใช้เครื่องมือใน Youtube
2.แบ่งเวลามาทดลองใช้เครื่องมือจริงให้คุ้นชิน

5.2.2 ข้อเสนอแนะในการพัฒนาหรือปรับปรุง เพื่อย่อยอดโครงการ

1. การใช้ชุดข้อมูลที่มีมากกว่า 193 ชุดข้อมูลจะทำให้ได้ผลลัพธ์ที่กว้างและหลากหลายมากกว่านี้
2. อาจพัฒนาตัวโปรแกรมให้ออกมาในรูปของ web app จะทำให้ดูมีความน่าสนใจมากยิ่งขึ้น

รายการอ้างอิง

สื่ออิเล็กทรอนิกส์

Ashwin Prasad. (22 ตุลาคม 2563). Simple Movie Recommender System with Correlation Coefficient with Python. เข้าถึงได้จาก medium.com : [Simple Movie Recommender System with Correlation Coefficient with Python | by Ashwin Prasad | Analytics Vidhya | Medium](#)

CHAD WAMBLES. (). The Beatles Spotify Song Data. เข้าถึงได้จาก Kaggle.com : <https://www.kaggle.com/datasets/chadwambles/allbeatlesspotifysongdata2009remaster>

Nattawat Srikong. (18 สิงหาคม 2563). Cosine similarity. เข้าถึงได้จาก medium.com : https://medium.com/@srikong_n/cosine-similarity-f1f9a962ddc5

Pasith Thanapatpisarn. (22 สิงหาคม 2564). [Fundamental Data Analytics & Data Scientist EP.19] พื้นฐานสถิติที่ต้องรู้ ก่อนจะเริ่ม Data Science Part 5. เข้าถึงได้จาก medium.com : <https://datascihaeng.medium.com/basic-stats-for-data-science-05-ef7aa838b429>

ภาคผนวก

ภาคผนวก ก

ข้อมูลโครงการ

[1] ข้อมูลที่ใช้

[The Beatles Spotify Song Data | Kaggle](#)

[2] Source code หรือ File ที่ใช้ในการคำนวณ

[Hit My Beatles.ipynb | Colaboratory\(google.com\)](#)

[3] ไฟล์ประกอบอื่นๆ

[แผนการดำเนินการ. xlsx | Google Drive](#)

ภาคผนวก ข

วิดีโอและสไลด์นำเสนอโครงการ

[1] วิดีโอนำเสนอ

[วิดีโอนำเสนอความคืบหน้าครั้งที่ 2 | Google Drive](#)

[วิดีโอนำเสนอ Hit My Beatles | Google Drive](#)

[2] สไลด์นำเสนอ

[นำเสนอความคืบหน้าครั้งที่ 1 | Google Drive](#)

[นำเสนอความคืบหน้าครั้งที่ 2 | Google Drive](#)

[นำเสนอ Hit My Beatles | Google Drive](#)

สมาชิก

โท ยากจ้ง



64010552 นายพนธกร วัลลำนนท์

(โอม)



64010670 นางสาวภาพิษฐ์ พงศ์พัฒน์วุฒิ

(พิชญ์)

9 A.M.



64010757 นายวรโชติ ใจเร็ว

(ไทม์)



64010761 นายวรพล รังษี

(นาย)