

เปิดโลกดนตรีไทยด้วย Machine Learning: จำแนกเอกลักษณ์เพลงไทย 4 ภาค

(Thai Regional Music Classification By Python)

Introduction

โปรเจกต์ Thai Regional Music Classification เป็นการพัฒนาและใช้งานโมเดล Machine Learning เพื่อจำแนกแนวเพลงจากไฟล์เสียงที่มีต้นกำเนิดจากภาคต่าง ๆ ของประเทศไทย ได้แก่ ภาคกลาง, ภาคเหนือ, ภาคอีสาน และภาคใต้ โดยการนำเทคนิคการแปลงข้อมูลเสียง (Audio) ให้เป็นคุณลักษณะ (Features) เช่น Mel-Frequency Cepstral Coefficients (MFCC) ที่สามารถใช้ในการฝึกโมเดลการเรียนรู้ได้อย่างมีประสิทธิภาพ

ซึ่งการจำแนกแนวเพลงจากภาคต่าง ๆ ของประเทศไทยมีความท้าทาย เนื่องจากลักษณะเสียงของเพลงจากแต่ละภาคมีความแตกต่างกันในด้านจังหวะ, การใช้เครื่องดนตรี การร้องและภาษาถิ่นซึ่งต้องการการแปลงและประมวลผลข้อมูลที่ซับซ้อนเพื่อให้โมเดลสามารถแยกแยะและจำแนกได้อย่างแม่นยำและมีประสิทธิภาพ

ดังนั้นเป้าหมายหลักของโปรเจกต์นี้คือการสร้างโมเดลที่สามารถจำแนกเพลงออกเป็นประเภทตามภาคของไทยได้ อย่างมีประสิทธิภาพ อาทิเช่น เพลงภาคกลาง, เพลงภาคเหนือ, เพลงภาคอีสาน, และเพลงภาคใต้ โดยใช้ Random Forest Classifier ในการฝึกฝนโมเดลและทดสอบกับข้อมูลใหม่ เพื่อประเมินความแม่นยำและประสิทธิภาพของโมเดลในการจำแนกเพลงตามภูมิภาคต่างๆในประเทศไทยและสามารถนำไปใช้สานต่อหรือ พัฒนาโมเดลกับแอปพลิเคชันต่างๆที่ต้องการการจำแนกแนวเพลงไทยได้ในอนาคตได้อย่างมีประสิทธิภาพ

Dataset & Data Preprocessing

แหล่งข้อมูล

ในโปรเจกต์นี้เราได้เตรียม Dataset ที่ประกอบไปด้วยเพลงจากหลากหลายภาคของประเทศไทย ได้แก่ ภาคกลาง, ภาคเหนือ, ภาคอีสาน และภาคใต้ โดยในแต่ละภาคจะถูกเก็บแยกในไฟล์เดอร์ของตัวเองและในแต่ละไฟล์เดอร์จะประกอบไปด้วยเพลงที่มีจำนวนหลายสิบถึงร้อยไฟล์ โดยจะมีองค์ประกอบดังนี้:

จำนวนไฟล์: แต่ละไฟล์เดอร์ประกอบไปด้วยเพลงจำนวนมากที่ถูกดาวน์โหลดจากแหล่งที่ถูกกฎหมายรวมทั้งเพลงยอดฮิตต่างๆจากแพลตฟอร์มยูทูปนำมาใช้เพื่อการศึกษาโปรเจกต์นี้โดยแต่ละแนวเพลงแต่ละภาคจะมีไฟล์เสียงหลายร้อยไฟล์

ประเภทของเพลง: เพลงใน dataset นี้มีทั้งเพลงที่มี เนื้อร้อง (vocal) และเพลงที่เป็น อินสตรูเมนทัล (instrumental) ซึ่งการมีทั้งสองประเภทจะช่วยให้โมเดลสามารถแยกแยะระหว่างลักษณะของเพลงที่มีเสียงร้องและไม่มีเสียงร้องได้

เพลงจากหลายภาค: เพลงใน Dataset นี้จะสะท้อนถึงความหลากหลายของเพลงไทยจากภาคต่าง ๆ ได้แก่ เพลงจากภาคกลาง, ภาคเหนือ, ภาคอีสาน และภาคใต้ ซึ่งจะช่วยให้โมเดลสามารถจำแนกเพลงจากภูมิภาคต่าง ๆ ได้อย่างแม่นยำและเป็นธรรมชาติ

การแปลงเป็นไฟล์ .wav: ไฟล์เสียงทั้งหมดใน dataset จะถูกแปลงจากรูปแบบ .mp3, .ogg หรือ .flac เป็น .wav ซึ่งเป็นรูปแบบที่ AI และเครื่องมือ Machine Learning สามารถประมวลผลได้ดี หลังจากทำการรันโค้ดบน colab นั้นเอง

แปลงไฟล์เสียง (Preprocessing)

ในขั้นตอน Data Preprocessing นั้นเราได้ทำการแปลงไฟล์เสียงจาก .mp3 เป็น .wav ด้วย Pydub เพื่อให้รองรับการประมวลผลในเครื่องมืออย่าง Librosa และ TensorFlow โดยตั้งค่าการแปลงเป็น Mono และ Sample Rate 22,050 Hz ซึ่งเหมาะสำหรับการฝึกโมเดล Machine Learning นั้นเองครับ

```
แปลงไฟล์เสียงเป็น .wav และปรับ Sample Rate & Mono/Stereo

สำหรับการแปลงเสียงจาก MP3 เป็นรูปแบบ WAV

[ ] !apt-get install ffmpeg -y

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ffmpeg is already the newest version (7:4.4.2-0ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.

[ ] !pip install pydub
from pydub import AudioSegment
from pydub.utils import which

AudioSegment.converter = which("ffmpeg") # บอกให้ pydub ใช้นี้ ffmpeg

Collecting pydub
  Downloading pydub-0.25.1-py2.py3-none-any.whl.metadata (1.4 kB)
  Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub
Successfully installed pydub-0.25.1
```

ภาพตัวอย่าง การติดตั้งเครื่องมือเพิ่มเติม

```
from pydub import AudioSegment
import os

base_dir = "/content/drive/My Drive/Project DE242 Music Genre Classification/Dataset"
output_dir = "/content/drive/My Drive/Project DE242 Music Genre Classification/Dataset_WAV"

for genre in genres:
    input_folder = os.path.join(base_dir, genre)
    output_folder = os.path.join(output_dir, genre)
    os.makedirs(output_folder, exist_ok=True)

    for file in os.listdir(input_folder):
        if file.endswith(".mp3"):
            mp3_path = os.path.join(input_folder, file)
            wav_path = os.path.join(output_folder, file.replace(".mp3", ".wav"))

            audio = AudioSegment.from_mp3(mp3_path)
            audio = audio.set_frame_rate(22050).set_channels(1) # ตั้งค่า Sample Rate & Mono
            audio.export(wav_path, format="wav")

    print(f"✅ แปลงไฟล์ (genre) เสร็จแล้ว!")

✅ แปลงไฟล์ Northern song เสร็จแล้ว!
✅ แปลงไฟล์ Northeastern song เสร็จแล้ว!
✅ แปลงไฟล์ Southern song เสร็จแล้ว!
✅ แปลงไฟล์ Central song เสร็จแล้ว!
```

ภาพตัวอย่าง การแปลงไฟล์เสียง

```
import os
from pydub import AudioSegment

# ตั้งค่าพาธไดเรกทอรี
input_dir = "/content/drive/My Drive/Project DE242 Music Genre Classification/Dataset_WAV"
output_dir = "/content/drive/My Drive/Project DE242 Music Genre Classification/Dataset_Segments"

window_size = 30000 # 30 วินาที
step_size = 15000 # 15 วินาที (Overlap 50%)

# สร้างโฟลเดอร์สำหรับเก็บไฟล์ที่ตัดแล้ว
os.makedirs(output_dir, exist_ok=True)

# วนลูปการประมวลผลเพลง
for genre in os.listdir(input_dir):
    genre_folder = os.path.join(input_dir, genre)
    output_genre_folder = os.path.join(output_dir, genre)

    # ตรวจสอบว่าโฟลเดอร์มีหรือไม่
    if not os.path.isdir(genre_folder):
        continue

    os.makedirs(output_genre_folder, exist_ok=True) # สร้างโฟลเดอร์สำหรับเก็บ segment

    # วนลูปผ่านไฟล์เพลงในโฟลเดอร์เพลงแต่ละประเภท
    for file in os.listdir(genre_folder):
        if file.endswith(".wav"):
            file_path = os.path.join(genre_folder, file)
            audio = AudioSegment.from_wav(file_path)

            # ตัดเพลงเป็น segment
            segments = []
            for start in range(0, len(audio) - window_size, step_size):
                segment = audio[start:start + window_size]
                segments.append(segment)

            # บันทึก segment ลงโฟลเดอร์ output
            for i, segment in enumerate(segments):
                segment_path = os.path.join(output_genre_folder, f"{file.replace('.wav', '')}_{seg(i)}.wav")
                segment.export(segment_path, format="wav")

    print(f"✅ ตัดเพลงใน (genre) เสร็จแล้ว!")

✅ ตัดเพลงใน Northern song เสร็จแล้ว!
✅ ตัดเพลงใน Northeastern song เสร็จแล้ว!
✅ ตัดเพลงใน Southern song เสร็จแล้ว!
✅ ตัดเพลงใน Central song เสร็จแล้ว!
```

ภาพตัวอย่าง การตัดเพลงในโฟลเดอร์ให้สั้นลง

```

import librosa
import numpy as np
import os

dataset_dir = "/content/drive/My Drive/Project DE242 Music Genre Classification/Dataset_Segments"
features_list = []

for genre in os.listdir(dataset_dir):
    genre_folder = os.path.join(dataset_dir, genre)
    if os.path.isdir(genre_folder):
        for file in os.listdir(genre_folder):
            if file.endswith(".wav"):
                file_path = os.path.join(genre_folder, file)

                # โหลดไฟล์เสียง
                y, sr = librosa.load(file_path, sr=22050)

                # คำนวณค่า MFCC
                mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13).mean(axis=-1)

                # คำนวณค่า Chroma
                chroma = librosa.feature.chroma_stft(y=y, sr=sr).mean(axis=-1)

                # คำนวณค่า Spectral Contrast
                spectral_contrast = librosa.feature.spectral_contrast(y=y, sr=sr).mean(axis=-1)

                # รวมค่า features
                features = np.hstack([mfcc, chroma, spectral_contrast])

                # เพิ่ม features
                features_list.append(features)

print(f"📌 Extracted Features from {len(features_list)} files!")

/usr/local/lib/python3.11/dist-packages/librosa/core/pitch.py:183: UserWarning: Trying to estimate tuning from empty frequency set.
return pitch_tuning()
📌 Extracted Features from 4918 files!

```

ภาพตัวอย่าง การดึง Chroma Features และ Spectral Contrast

```

import pandas as pd
import os

dataset = []
labels = []
genres = ["Northern song", "Northeastern song", "Southern song", "Central song"] # กำหนด
dataset_path = "/content/drive/My Drive/Project DE242 Music Genre Classification/Dataset_Segments" # กำหนดโฟลเดอร์

for genre in genres:
    genre_folder = os.path.join(dataset_path, genre)
    for file in os.listdir(genre_folder):
        if file.endswith(".wav"):
            file_path = os.path.join(genre_folder, file)
            y, sr = librosa.load(file_path, sr=22050)

            # คำนวณค่า Features
            mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13).mean(axis=-1)
            chroma = librosa.feature.chroma_stft(y=y, sr=sr).mean(axis=-1)
            spectral_contrast = librosa.feature.spectral_contrast(y=y, sr=sr).mean(axis=-1)

            # รวมค่า Features
            features = np.hstack([mfcc, chroma, spectral_contrast])
            dataset.append(features)
            labels.append(genre)

# สร้าง DataFrame
df = pd.DataFrame(dataset)
df["label"] = labels

# บันทึกเป็น CSV
df.to_csv("thai_music_dataset.csv", index=False)
print(f"📌 บันทึก dataset ที่ dataset")

/usr/local/lib/python3.11/dist-packages/librosa/core/pitch.py:183: UserWarning: Trying to estimate tuning from empty frequency set.
return pitch_tuning()
📌 บันทึก dataset ที่ dataset

```

ภาพตัวอย่าง การแปลงข้อมูลเป็น data frame

การสร้างโมเดล Machine Learning

ในขั้นตอน Modeling เราใช้ Random Forest Classifier ซึ่งเป็นอัลกอริทึมที่มีประสิทธิภาพในการจำแนกข้อมูลแบบหลายประเภท (multiclass classification) โดยจะสร้าง ต้นไม้การตัดสินใจ(Decision Trees) หลายๆต้นและรวมผลลัพธ์จากแต่ละต้นไม้เพื่อให้การตัดสินใจแม่นยำยิ่งขึ้น Random Forest ช่วยให้เราโมเดลสามารถจัดการกับข้อมูลที่มีลักษณะซับซ้อนได้ดี โดยไม่เสี่ยงต่อการ overfitting เพราะมันใช้หลายต้นไม้ในการตัดสินใจและนำผลลัพธ์มาประมวลผลร่วมกันอีกทั้งโมเดลนี้ถูกฝึกโดยใช้ข้อมูล MFCC ที่ไดจากการแปลงไฟล์เสียงและใช้ accuracy score ในการประเมินผลการทำนายอีกด้วย

```

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

df = pd.read_csv("thai_music_dataset.csv")

X = df.iloc[:, :-1] # Feature (MFCC)
y = df["label"] # Label (Genre)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = RandomForestClassifier(n_estimators=100)
model.fit(X_train, y_train)

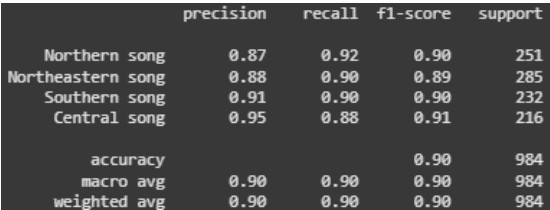
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)

print(f"📌 Accuracy: {accuracy:.2f}")

📌 Accuracy: 0.90

```

โมเดลถูกประเมินด้วย Confusion Matrix, F1-Score, Precision และ Recall เพื่อวัดประสิทธิภาพในการจำแนกแฉเพลง Confusion Matrix แสดงการทำนายที่ถูกต้องและผิดพลาดของแต่ละแฉเพลง Precision วัดความแม่นยำของการทำนายแต่ละคลาส Recall วัดความสามารถของโมเดลในการดึงข้อมูลของแต่ละคลาส F1-Score เป็นค่าเฉลี่ยระหว่าง Precision และ Recall โมเดลถูกทดสอบกับข้อมูลเสียงที่ไม่เคยเห็นมาก่อน โดยผลลัพธ์ที่ได้แสดงให้เห็นถึงความสามารถของโมเดลในการจำแนกแฉเพลงได้อย่างแม่นยำ



วิธีการเชื่อมโยงตัวโมเดลไปที่เว็บ ของเราใช้ Ngrok เพื่อสร้าง Public URL สำหรับ Flask API บน Google Colab โดยนำ Auth Token จาก <https://dashboard.ngrok.com/get-started/your-auth-token> มาเชื่อมกับโค้ด เพื่อให้สามารถเปิดใช้งานลิงก์ได้จากภายนอกได้

```
so.remove(remove_path)

return jsonify({'error': predicted_error, 'status': 'success'})

except Exception as e:
    print(f"❗ Error: {str(e)}")
    return jsonify({'error': str(e)}), 500

# pyngrok tunnel

from pyngrok import ngrok

public_url = ngrok.connect(5000)
print(f"🌐 Public URL: {public_url}")

app.run()
```

```

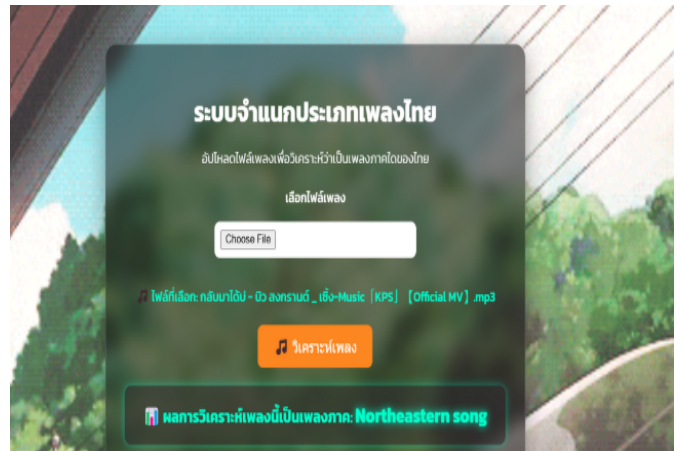
❖ SaveModelDriver: /content/fin/MyPyProject/main_forest_model.pkl
@ Public URL: NgrokURL: "https://0b9636-34-70-373-108.ngrok-free.net" -> "http://localhost:5000"
* Serving Flask app "main"
* Debug mode: off

INFO:werkzeug:127.0.0.1 is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit

/usr/local/lib/python.11/dist-packages/xgboost/ml_utils/validation.py:2720: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(

INFO:werkzeug:127.0.0.1 - [12/Apr/2025 07:44:03] "POST /predict HTTP/1.1" 200 -
```

Web Page



Challenges & Solutions

ปัญหา (Challenges):

1. ข้อมูลเพลงภูมิภาคไทยมีจำกัดการหาข้อมูลเพลงที่ระบุภูมิภาคชัดเจนทำได้ยาก
2. ไฟล์เสียงมีหลายรูปแบบเช่น .mp3, .ogg, .flac ทำให้ประมวลผลยาก
3. ขนาด dataset ไม่ใหญ่มากอาจส่งผลต่อความแม่นยำของโมเดล
4. ต้องการนำโมเดลไปใช้ผ่านเว็บไซต์แต่ Google Colab ไม่มี public endpoint
5. ต้องทำให้เว็บไซต์เข้าใจง่ายและใช้งานได้จริง

แนวทางการแก้ไข (Solutions):

1. ทำการรวบรวมเพลงจากหลายแหล่ง เช่น YouTube, เว็บไซต์เพลงพื้นบ้าน และระบุภูมิภาคด้วยตนเองก่อนสร้าง dataset
2. ทำการแปลงทุกไฟล์เสียงให้อยู่ในรูปแบบ .wav ที่เหมาะกับการประมวลผลด้วย Librosa
3. ใช้เทคนิค feature extraction ที่มีประสิทธิภาพ เช่น MFCC, Chroma และ Spectral Contrast เพื่อให้ได้ข้อมูลที่เพียงพอต่อการเรสู
4. ใช้เครื่องมือ Ngrok เพื่อสร้าง Public URL ชั่วคราว ให้สามารถเชื่อมต่อ API จากเว็บไซต์ได้
5. ออกแบบหน้าเว็บให้เรียบง่าย โดยใช้ HTML+CSS+JavaScript พร้อมเชื่อมต่อกับ Flask API เพื่อให้ผู้ใช้สามารถอัปโหลดไฟล์เสียงและรับผลลัพธ์ได้ทันที

Next Steps

เพิ่มจำนวนเพลงใน Dataset ให้ครอบคลุมมากขึ้น

- เพื่อช่วยให้โมเดลเรียนรู้ลักษณะเฉพาะของแต่ละภูมิภาคได้ดียิ่งขึ้นและเน้นเก็บเพลงจากแหล่งที่มีความน่าเชื่อถือและตรงกับภูมิภาคจริง ๆ

ทดลองใช้โมเดลอื่นๆ เพิ่มเติม เช่น CNN หรือ LSTM

- เพื่อเปรียบเทียบผลลัพธ์กับ Random Forest และเลือกโมเดลที่ให้ความแม่นยำสูงที่สุด

ปรับแต่ง Feature Extraction ให้เหมาะสมยิ่งขึ้น

- โดยอาจจะใช้ทดลองเพิ่มหรือเปลี่ยนชนิดของฟีเจอร์เสียง เช่น Zero-Crossing Rate, Tempo เป็นต้น

พัฒนาเว็บไซต์ให้สมบูรณ์และใช้งานได้หลากหลายขึ้น

- เช่น เพิ่มระบบแสดงผลภาพ Spectrogram หรือประวัติเพลง และเพิ่มความสามารถให้รองรับทั้งมือถือ และเดสก์ท็อปที่มีความสเกลรูปภาพมากขึ้น

Conclusion

โครงการ Thai Regional Music Classification เป็นการประยุกต์ใช้เทคนิค Machine Learning โดยใช้โมเดล Random Forest ร่วมกับการดึงคุณลักษณะของเสียง (เช่น MFCC, Chroma, Spectral Contrast) เพื่อจำแนกแนวเพลงตามภูมิภาคของไทย ได้แก่ เพลงภาคเหนือ ภาคตะวันออกเฉียงเหนือ ภาคกลาง และภาคใต้

แม้ว่าแนวเพลงเหล่านี้มีความคล้ายคลึงกันในด้าน การใช้ฟีเจอร์เสียงและการฝึกโมเดลอย่างเหมาะสมช่วยให้ระบบสามารถจำแนกได้ในระดับที่แม่นยำเพียงพอสำหรับการใช้งานเบื้องต้น นอกจากนี้ยังได้มีการพัฒนาเว็บไซต์เพื่อรองรับการใช้งานจริง โดยสามารถอัปโหลดไฟล์เสียงและดูผลลัพธ์การจำแนกแบบเรียลไทม์ผ่านอินเทอร์เฟซที่ใช้งานง่าย

โดยโครงการนี้ไม่เพียงแสดงให้เห็นถึงความเป็นไปได้ของการใช้ machine learning กับดนตรีไทย แต่ยังสามารถต่อยอดไปสู่การพัฒนาระบบแนะนำเพลง วิเคราะห์เสียงพื้นถิ่น หรือแม้แต่ใช้ในงานอนุรักษ์วัฒนธรรมไทยในอนาคตได้อีกด้วย

แหล่งข้อมูลเพิ่มเติม

📌 Librosa Docs → <https://librosa.org/>

📌 Scikit-learn Docs → <https://scikit-learn.org/>

📌 Youtube → <https://www.youtube.com/>

📌 free music archive → <https://freemusicarchive.org/home>

📌 Colab →

🔗 Project DE242 Music Genre Classification.ipynb

🔗 Model_HTML_Test_5_4_68.ipynb

📌 Slide →

https://www.canva.com/design/DAGidhIUX6Y/3FjfKp5UINgjNY8QNx50QA/view?utm_content=DAGidhIUX6Y&utm_campaign=designshare&utm_medium=link2&utm_source=unique_links&utm_id=h5c42041a2a

สมาชิกผู้จัดทำ

1. พลวัต พงศ์กิติพย์พนัส รหัสนิสิต 66102010249
2. ปันณรร สุวรรณาศรัย รหัสนิสิต 66102010175
3. พลวิชญ์ วนิดา รหัสนิสิต 66102010584

