

เปิดโลกดูตระห่ำด้วย Machine Learning: จำแนกເວັກລັກບໍນໄພລົງໄທຍ່າງກຳ

(Thai Regional Music Classification By Python)

Introduction

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

ซึ่งการจำแนกແນວເພື່ອຈາກການດໍາເກີນ ຂອງປະເທດໄກຍມີຄວາມກ້າວກາຍ ເນື່ອຈາກລັກບໍນາສີເສີຍຂອງເພື່ອຈາກແຕ່ລະການມີຄວາມແຕກດໍາເກີນໃນດ້ານຈັງຂວະ, ການໃຊ້ເຄື່ອງດົນຕີ ການຮັບອັງການທີ່ມີບໍ່ສິ່ງຕ້ອງການການແປລົງແລະປະນວລຜູ້ບ້ອນມຸລືກໍ່ຫັບຫຼວມເພື່ອໃຫ້ໄນ້ເດືອນສານາຣັກແຍກແຍະແລະຈຳແນກໄດ້ອ່າຍໆແມ່ນຍໍາແລະມີປະສິກີກົວ

ดังนั้นเป้าหมายหลักของໂປຣເຈັກນີ້ຄ້ອງການຮັງໄນ້ເດືອກທີ່ສາມາດຈຳແນວພັບອອກເປັນປະເທດຕາມການຂອງໄທຢ້າງເປົ້າພະສິກີກາວ ຈາກີເຫັນພັບການຄລາງ, ພັບການເໜັ້ນ, ພັບການອໍສານ, ແລະພັບການໃຊ້ໄຕ ໂດຍໃຊ້ Random Forest Classifier ໃນການຝຶກຝັນໄນ້ເດືອນແລະກົດສອບກັບຂ້ອງມູນໃຫຍ່ ເພື່ອປະເບີນຄວາມແມ່ນຢໍາແລະປະສິກີກາພອງໄນ້ເດືອນໃນການຈຳແນວພັບດາມຖຸເກົາຄດ່າງໆໃນປະເທດໄທຢູ່ແລະສາມາດນໍາໄປໃຊ້ສານຕ່ອງຮົ້ວຂໍ ພັນຍາໄນ້ເດືອນກັນແລ້ວພັບເຄື່ອງຕ່າງໆທີ່ຕ້ອງການການຈຳແນວພັບພົວໄທຢ້າງເປົ້ານອາຄາຕີໄດ້ອ່ານຍິນຍຸດ

Dataset & Data Preprocessing

ແກ່ລົງບັວນຸລ

ในโปรดเจคนี้เราได้เติรยบ Dataset กี่ประกอบไปด้วยเพลงຈາກಹລາຍກາຄຂອງປະເທດໄກ ໄດ້ແກ່ ກາຄຄາງ, ກາຄເໜືອ, ກາຄວີສານ ແລະ ກາຄໄຕ ໂດຍໃນແຕ່ລະກາຈະຖຸກເກີບແຍກໃນໄຟເລເດວຮຂອງຕັວເອງແລະໃນແຕ່ລະໄຟເລເດວຮຈະປະກອບໄປດ້ວຍພෙງທີ່ນີ້ຈຳນວນຫລາຍສັບດົງຮ້ອຍໄຟລ໌ ໂດຍຈະນີ້ອັນດີປະກອບດັ່ງນີ້:

ចំណែកផល: តែតែទិន្នន័យភាគចំណែកផលមួយគឺជាប្រព័ន្ធដែលបានរាយការណ៍ដោយសារព័ត៌មានអាជីវកម្មរបស់ខ្លួន។ ការងារនេះមានប្រព័ន្ធទិន្នន័យភាគចំណែកផលដែលបានរាយការណ៍ដោយសារព័ត៌មានអាជីវកម្មរបស់ខ្លួន។

ประเภทของเพลง: เพลงใน dataset นี้มีทั้งเพลงที่มี เนื้อร้อง (vocal) และเพลงที่เป็น อันสติรุเม้นทัล (instrumental) ซึ่งการมีทั้งสองประเภทจะช่วยให้โน้ตเดลสามารถแยกแยะระหว่างลักษณะของเพลงที่มีเสียงร้องและไม่มีเสียงร้องได้

เพลงจาระภาษาคต: เพลงใน Dataset นี้จะสังกัดอันดับความคล้ายคลายของเพลงไทยจากภาคต่าง ๆ ได้แก่ เพลงจากภาคกลาง, ภาคเหนือ, ภาคอีสาน และภาคใต้ ซึ่งจะช่วยให้โน้ตเดลล์สามารถจำแนกเพลงจากภูมิภาคต่าง ๆ ได้อย่างแม่นยำและเป็นธรรมชาติ

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

แปลงไฟล์เสียง (Preprocessing)

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

แปลงไฟล์เสียงเป็น .wav และปรับ Sample Rate & Mono/Stereo

สำหรับการแปลงเสียงจาก MP3 เป็นอูปแบบ 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") # บอกไฟ pydub ใช้ 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

# แปลง wav เป็น mono
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 วินท์
step_size = 15000 # 15 วินท์ (Overlap 50%)
# ตั้งค่า window size และ step size
os.makedirs(output_dir, exist_ok=True)

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

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

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

    # รุบไฟล์ wav ทั้งหมดในโฟลเดอร์
    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)

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

            # บันทึก segment 作为一个 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"✓ ตัดเพลง {genre} เสร็จแล้ว!")

✓ ตัดเพลง Northern song เสร็จแล้ว!
✓ ตัดเพลง Northeastern song เสร็จแล้ว!
✓ ตัดเพลง Southern song เสร็จแล้ว!
✓ ตัดเพลง 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'): # ถ้าเป็นไฟล์ wav
                file_path = os.path.join(genre_folder, file)
                # โหลดข้อมูล
                y, sr = librosa.load(file_path, sr=22050)
                # คำนวณ MFCC
                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)
                # รวมทุกค่าเข้าด้วยกัน
                features = np.hstack([mfcc, chroma, spectral_contrast])
                # เก็บลง
                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 4018 files!

```

ภาพตัวอย่าง การดึง Chroma Features และ Spectral Contrast

```

import pandas as pd
import os
datasets = []
labels = []
genres = ["Northern song", "Northeastern song", "Southern song", "Central song"] # ภูมิศาสตร์
dataset_path = "/content/drive/My Drive/Project DE242 Music Genre Classification/Dataset_Segments" # ตำแหน่งไฟล์เดิม

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

            # MFCC 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)

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

# สร้าง DataFrame
df = pd.DataFrame(dataset)
df['label'] = labels
# บันทึก CSV
df.to_csv("thai_music_dataset.csv", index=False)
print(f"บันทึก dataset เเรียบร้อย!")

```

ภาพตัวอย่าง การแปลงข้อมูลเป็น data frame

การสร้างโมเดล Machine Learning

ในขั้นตอน Modeling เราใช้ Random Forest Classifier ซึ่งเป็นอัลกอริธึมที่มีประสิทธิภาพในการจำแนกข้อมูลแบบหลายประเภท (multiclass classification) โดยจะสร้าง ต้นไม้การตัดสินใจ(Decision Trees) หลายต้นและรวมผลลัพธ์จากแต่ละต้นให้ไปเพื่อให้การตัดสินใจแม่นยำยิ่งขึ้น Random Forest ช่วยให้โน้ตเดลสามารถจัดการกับข้อมูลที่มีลักษณะซับซ้อนได้ดี โดยไม่เสี่ยงต่อการ overfitting เพราะมันใช้หลายต้นไม้ในการตัดสินใจและนำผลลัพธ์มาประมวลผลร่วมกันอีกด้วย สำหรับข้อมูล MFCC ที่ได้จากการแปลงไฟล์เสียงและใช้ accuracy score ในการประเมินผลการจำแนยอีกด้วย

```

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)

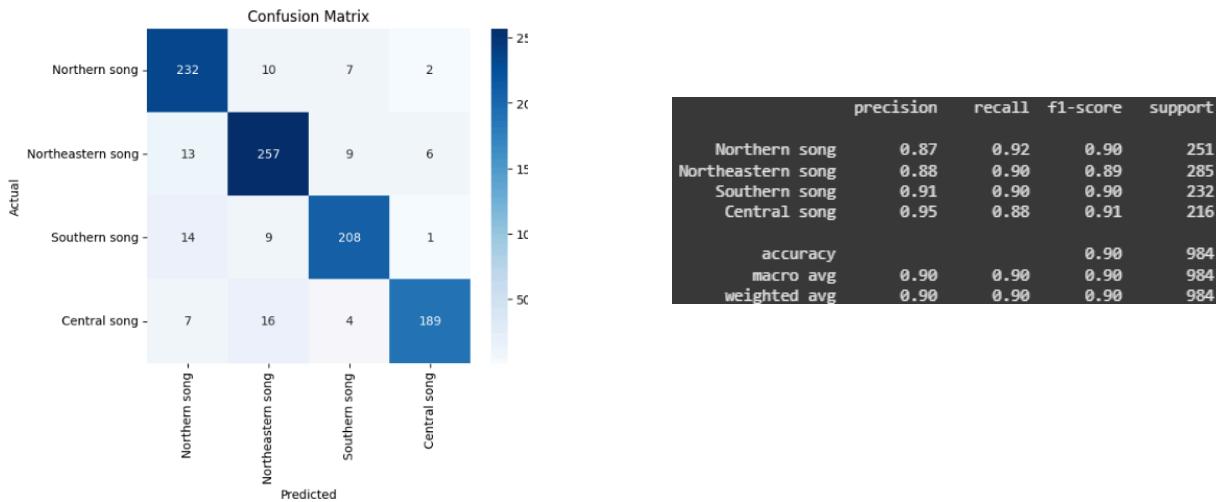
print(f"\n Accuracy: {accuracy:.2f}")

Accuracy: 0.90

```

Evaluation & Results

โนเดลถูกประบেนด์วัย Confusion Matrix, F1-Score, Precision และ Recall เพื่อวัดประสิทธิภาพในการจำแนกแบบเพลง Confusion Matrix และแสดงการกำนายที่ถูกและผิดของแต่ละแบบเพลง Precision วัดความแม่นยำของการกำนายแต่ละคลาส Recall วัดความสามารถของโนเดลในการดึงข้อมูลของแต่ละคลาส F1-Score เป็นค่าตัวเฉลี่ยระหว่าง Precision และ Recall โนเดลถูกทดสอบกับข้อมูลเสียงที่ไม่เคยเห็นมาก่อน โดยผลลัพธ์ที่ได้แสดงให้เห็นถึงความสามารถของโนเดลในการจำแนกแบบเพลงได้อย่างแม่นยำ



LINK WEB API

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

```

# ตั้งค่า Ngrok
ngrok.set_auth_token("2vJ335kP4jXmdC8MhNmWk6n9z0_73uWjPScBpwId2HwiyP") # ใส่ Auth Token ของของคุณ
public_url = ngrok.connect(5000, "public_url")
AudioSegment.converter = which("ffmpeg")

app = Flask(__name__)
CORS(app)

# โหลดโมเดล
model_path = "/content/drive/My Drive/random_forest_model.pkl"
model = joblib.load(model_path)
model = model["RandomForestClassifier"]([model_path])

# คำนวณแนวโน้ม
genres = ["Northern song", "Northeastern song", "Southern song", "Central song"]

# ฟังก์ชันที่จะรับไฟล์
def extract_features(file_path):
    y, sr = librosa.load(file_path, sr=22050)
    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)
    return np.hstack([mfcc, chroma, spectral_contrast])

@app.route('/predict', methods=["POST"])
def predict():
    try:
        file = request.files["file"]
        temp_path = f"/tmp/{file.filename}"
        file.save(temp_path)

        # แปลงไฟล์เป็น wav
        if file.filename.endswith(".mp3"):
            wav_path = temp_path.replace(".mp3", ".wav")
            audio = AudioSegment.from_mp3(temp_path)
            audio.set_frame_rate(22050).set_channels(1)
            audio.export(wav_path, format="WAV")
            os.remove(temp_path)
            temp_path = wav_path

        # คำนวณแนวโน้ม
        genre_prediction = model.predict([extract_features(temp_path)])
        genre_name = genres[genre_prediction[0]]
        print(f"Predicted genre: {genre_name}")

        # ลบไฟล์ที่ไม่จำเป็น
        os.remove(temp_path)

        # ตอบกลับ
        response = {
            "genre": predicted_genre,
            "status": "success"
        }
        return jsonify(response)

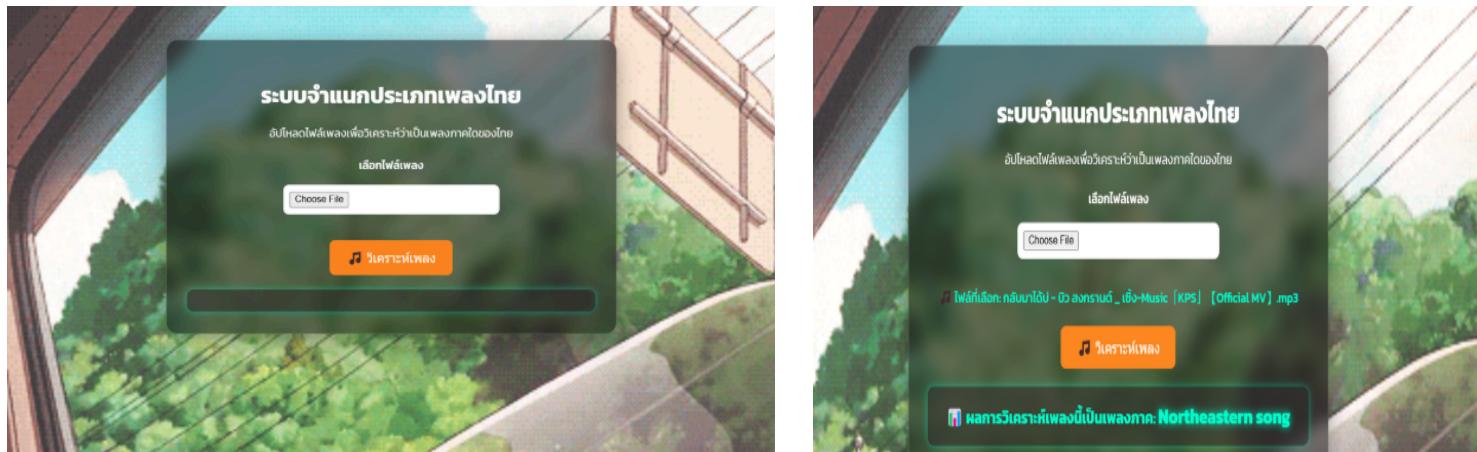
    except Exception as e:
        print(f"Error: {str(e)}")
        return jsonify({"error": str(e)}), 500

# รัน Ngrok
from pyngrok import ngrok
public_url = ngrok.connect(5000)
print(f"Public URL: {public_url}")

app.run()

```

Web Page



Challenges & Solutions

ปัญหา (Challenges):

- ข้อมูลเพลงภูมิภาคไทยมีจำกัดการหาข้อมูลเพลงที่ระบุภูมิภาคชัดเจนทำได้ยาก
- ไฟล์เสียงมีหลายรูปแบบ เช่น .mp3, .ogg, .flac ทำให้ประมวลผลยาก
- ขนาด dataset ไม่ใหญ่มากอาจส่งผลต่อความแม่นยำของโมเดล
- ต้องการนำโมเดลไปใช้ผ่านเว็บไซต์แต่ Google Colab ไม่มี public endpoint
- ต้องทำให้เว็บไซต์เข้าใจง่ายและใช้งานได้จริง

แนวทางการแก้ไข (Solutions):

- ทำการรวบรวมเพลงจากหลายแหล่ง เช่น YouTube, เว็บไซต์เพลงพื้นบ้าน และระบุภูมิภาคด้วยตนเองก่อนสร้าง dataset
- ทำการแปลงทุกไฟล์เสียงให้อยู่ในรูปแบบ .wav ก่อนนำไปใช้ Librosa
- ใช้เทคนิค feature extraction ที่มีประสิทธิภาพ เช่น MFCC, Chroma และ Spectral Contrast เพื่อให้ได้ข้อมูลที่เพียงพอต่อการเรียนรู้
- ใช้เครื่องมือ Ngrok เพื่อสร้าง Public URL ชั่วคราว ให้สามารถเชื่อมต่อ API จากเว็บไซต์ได้
- ออกแบบหน้าเว็บให้เรียบง่าย โดยใช้ HTML+CSS+JavaScript พร้อมเชื่อมต่อกับ Flask API เพื่อให้ผู้ใช้สามารถอัปโหลดไฟล์เสียงและรับผลลัพธ์ได้กันที

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

พัฒนาเว็บไซต์ให้สนับสนุนและใช้งานได้หลากหลายขึ้น

- เช่น เพิ่มระบบแสดงผลภาพ Spectrogram หรือประวัติเพลง และเพิ่มความสามารถให้รองรับกับมือถือ และเดสก์ท็อปที่มีความสามารถในการสแกนภาพมากขึ้น

Conclusion

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

แนวโน้มของการพัฒนาเว็บไซต์ในอนาคต คือการเพิ่มประสิทธิภาพการทำงาน เช่น การปรับปรุงโมเดลให้สามารถจัดจำแนกเพลงตามภูมิภาคได้แม่นยำยิ่งขึ้น หรือการเพิ่มฟีเจอร์ใหม่ เช่น การแนะนำเพลงใหม่ตามความชอบของผู้ใช้งาน หรือการสร้างแพลตฟอร์มสำหรับการแลกเปลี่ยนเพลงระหว่างประเทศ

โดยโครงการนี้ไม่เพียงแค่เป็นเครื่องมือในการจำแนกเพลง แต่ยังสามารถนำไปใช้ในการศึกษาทางวิทยาศาสตร์ อาทิ การวิเคราะห์ความถี่ของโน้ตดนตรีในแต่ละเพลง หรือการใช้โมเดลนี้ในการฝึกอบรมเครื่องจักรเรียนรู้เพื่อการจัดอันดับเพลงในแอปพลิเคชันต่างๆ

ແກລ່ງຂ້ວມູລເພີ່ມເຕີມ

📌 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/DAGidhlUX6Y/3FjfKp5UINgjNY8QNx50QA/view?utm_content=DAGidhlUX6Y&utm_campaign=designshare&utm_medium=link2&utm_source=unique_links&utllid=h5c42041a2a

ສນາບິກຜູ້ຈັດກຳ

1. ພລວັດ ພົງຄົກພຍົມບັນສ ຮຮສະນິສິຕ 66102010249
2. ປັນຍຸຮຣ ສຸວຽບນາຄຮຍ ຮຮສະນິສິຕ 66102010175
3. ພລວັ່ງຍົງ ວົນດາ ຮຮສະນິສິຕ 66102010584

