

Music Classifier ML

Deskripsi Proyek

Proyek Music Classifier ML bertujuan untuk mengklasifikasikan genre musik berdasarkan berbagai fitur lagu, termasuk durasi lagu. Proyek ini melibatkan beberapa tahap, mulai dari preprocessing data, ekstraksi fitur, pelatihan model, hingga evaluasi model, dengan harapan membangun model klasifikasi yang andal dengan akurasi tinggi.

Tujuan Proyek

Preprocessing Data:

1. Memuat dan menampilkan dataset menggunakan Pandas.
2. Memberi label pada lagu berdasarkan durasi (Short, Medium, Long).
3. Menangani fitur numerik dan kategorikal dengan menormalisasi data numerik dan encoding data kategorikal.

Ekstraksi dan Rekayasa Fitur:

1. Mengekstraksi fitur-fitur relevan dari dataset untuk digunakan dalam pelatihan model.
2. Membuat fitur baru (misalnya, `duration_label`) untuk meningkatkan kinerja model.

Pelatihan Model:

1. Melatih model klasifikasi untuk memprediksi genre lagu.
2. Menggunakan algoritma machine learning seperti `RandomForestClassifier` dari `scikit-learn` untuk membangun model.

Evaluasi Model:

1. Mengevaluasi kinerja model yang sudah dilatih menggunakan metrik seperti `classification report`.
2. Memuat model yang sudah dilatih dan membuat prediksi untuk menilai akurasi.

Penyimpanan Model:

1. Menyimpan model yang sudah dilatih menggunakan `joblib` untuk digunakan di masa depan.
2. Memuat model yang sudah disimpan untuk membuat prediksi dan mengevaluasi kinerja.

Komponen Utama dan Analisis Kode

Preprocessing Data:

Menampilkan Dataset:

```
import pandas as pd
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)
pd.set_option('display.max_colwidth', None)
df = pd.read_csv('songs_normalize.csv')
print(df)
```

Memberi Label pada Durasi:

```
def label_duration(row):
    if row['duration_ms'] < 180000:
        return 'Short'
    elif row['duration_ms'] < 240000:
        return 'Medium'
    else:
        return 'Long'

df['duration_label'] = df.apply(label_duration, axis=1)
print(df.head())
```

Pelatihan dan Evaluasi Model:

Pelatihan:

```
from training import train_model
from evaluation import evaluate_model
```

```
def main():
    train_model()
    evaluate_model()

if __name__ == "__main__":
    main()
```

Evaluasi:

```
import joblib
from sklearn.metrics import classification_report
from preprocessing import load_and_preprocess_data
```

```
def evaluate_model():
    X, y = load_and_preprocess_data('songs_normalize.csv')
    model = joblib.load('trained_model.pkl')
    predictions = model.predict(X)
    report = classification_report(y, predictions, zero_division=0)
    print(report)

if __name__ == "__main__":
    evaluate_model()
```

Memuat dan Preprocessing Data:

```
import pandas as pd
from sklearn.preprocessing import StandardScaler, LabelEncoder

def load_and_preprocess_data(filepath):
    df = pd.read_csv(filepath)
    X = df.drop('genre', axis=1)
    y = df['genre']
    numeric_features = X.select_dtypes(include=['float64', 'int64'])
    categorical_features = X.select_dtypes(include=['object'])
    scaler = StandardScaler()
    numeric_features = pd.DataFrame(scaler.fit_transform(numeric_features),
    columns=numeric_features.columns)
    label_encoder = LabelEncoder()
    for column in categorical_features.columns:
        categorical_features[column] = label_encoder.fit_transform(categorical_features[column])
    X_processed = pd.concat([numeric_features, categorical_features], axis=1)
    return X_processed, y

if __name__ == "__main__":
    X, y = load_and_preprocess_data('songs_normalize.csv')
    print(X.head())
```

Alat dan Pustaka yang Digunakan

1. Pandas: Untuk memuat, memanipulasi, dan preprocessing data.
2. scikit-learn: Untuk pelatihan, preprocessing, dan evaluasi model machine learning.
3. matplotlib: Potensial untuk visualisasi (meskipun tidak secara eksplisit digunakan dalam kode yang diberikan).
4. joblib: Untuk menyimpan dan memuat model yang sudah dilatih.

Kesimpulan

Proyek Music Classifier ML bertujuan untuk mengklasifikasikan genre musik berdasarkan berbagai fitur lagu, termasuk durasi. Proyek ini melibatkan preprocessing data yang ekstensif, ekstraksi fitur, pelatihan dan evaluasi model, serta penyimpanan model menggunakan berbagai pustaka Python. Tujuannya adalah untuk membangun model klasifikasi yang andal dengan akurasi tinggi.

Contributors

Haekal Setiawan (@haekalsetiawan)