

Spotify Song Prediction

Link Github : https://github.com/Ravelijn/ML_Process

The Problem

Saat ini kemampuan untuk memprediksi sesuatu yang mungkin akan menjadi populer adalah riset yang penting bagi setiap industri. Terutama sekali bagi pertumbuhan dan kompetisi pada industri music. Sejak meluasnya penggunaan platform musik digital (Spotify, Billboard, Lastfm), data sangatlah mudah dijangkau dan perilaku pendengar bisa dengan mudah diobservasi. Hal ini memberikan kemudahan bagi teknik peramalan dan juga kerap kali digunakan pada sistem rekomendasi.

Goals & Success

Metrik yang digunakan adalah popularity. The Spotify Popularity Index memiliki range dari 0-sampai-100 yang menunjukkan peringkat seberapa populer seorang artis atau lagu relative terhadap artis dan track pada spotify.

Key Solution

Untuk melakukan analisa prediktif, didapatkan dataset dari

Dataset yang digunakan adalah sebagai berikut :

1. **Accousticness** : ukuran dari 0.0 to 1.0 apakah suatu track adalah akusting, dengan 1.0 merepresentasikan angka keyakinan yang tinggi bahwa suatu track adalah akustik.
2. **Danceability** : Mendeskripsikan seberapa cocok suatu track untuk digunakan berdasar berdasarkan kombinasi dari elemen music termasuk tempo, stabilitas ritme, kekuatan beat dan keseluruhan Nilai 0.0

adalah kurang cocok untuk berdansa dan 1.0 adalah sangat cocok

- 3. Duration : Panjang track dalam millisecond
- 4. Energy : Ukuran dari 0.0 sampai dengan 1.0 yang merepresentasikan ukuran intensitas dan aktifitas. Track yang berenergi akan terasa cepat dan riuh.
- 5. Explicit : Apakah suatu track memiliki lirik yang eksplisit
- 6. ID : ID Spotify untuk suatu track
- 7. Instrumentalness : Memprediksi agar suatu track tidak memiliki
- 8. Key : Kunci yang digunakan suatu track, menggunakan notasi standard pitch
- 9. Liveliness : Mendeteksi keberadaan penonton di dalam rekaman. Nilai yang tinggi menunjukkan kemungkinan bahwa track direkam secara langsung.
- 10. Loudness : Tingkat kebisingan track dalam decibel (db)
- 11. Mode : Mode mengindikasikan modality (mayor atau minor) dari suatu track, tipe dari tangga nada yang menurunkan melodi
- 12. Name : Nama
- 13. Popularity : Popularitas dari track dengan nilai antara 0 dan 100, dengan 100 adalah paling populer. Dihitung oleh algoritma dan didasarkan pada total jumlah suatu track diputarkan dan pemutaran terkini.
- 14. Release Date : Tahun perilisan
- 15. Speechiness : Mendeteksi keberadaan kata-kata dalam suatu track.
- 16. Tempo : Tempo dari suatu track dalam ukuran beats per minute (BPM).

Data diambil dari Kaggle dan algoritma yang digunakan untuk prediksi adalah Random Forest.

Key Flows

Project pipeline yang digunakan adalah end to end machine learning hingga deployment model. Struktur data dari Project disesuaikan dengan best practice dalam pembuatan machine learning di mana terdapat beberapa script python utama yang dijalankan yaitu :

1. data_pipeline.py : berisikan modul terkait data collection, data validation
2. preprocessing.py : berisikan modul terkait handling missing values
3. Modeling : modul terkait baseline model, evaluation
4. Api.py dan Streamlit.Py : modul terkait API sebagai penghubung dengan user dan streamlit sebagai user interface.

Untuk service API dan streamlit ditunjang menggunakan docker service sebagai virtual environment

Launch Readiness

Project dari timeline sesuai dengan deadline dari pengerjaan tugas proyek ML Process tanggal 9 November 2024

Artifact

Komponen yang digunakan dalam proyek ini yaitu Visual Studio Code, Jupyter Notebook, Google Chrome User, Git Hub, Git Bash, Docker. Adapun modul python yang digunakan adalah pandas, numpy, sklearn, matplotlib, seaborn, dan seterusnya

References

Spotify Data Analysis and Song Popularity Prediction : Sivasai Bhavanasi, Sahil Malla, V Manichetan, CVNJ Dhanush, Dr B Prakash

<https://www.kaggle.com/datasets/leonardopena/top-spotify-songs-from-20102019-by-year>

Source Code end to end workflow

1. Exploratory Data Analysis

Tahapan ini berada pada file Jupyter Notebook untuk melakukan analisis yang nantinya akan diubah ke dalam bentuk *.py python file

a. Cek ketersediaan predictor

Fitur yang digunakan sebagai predictor adalah acousticness, danceability, duration, energy, loudness, speechiness, valence

b. Cek data types

Saat pengecekan awal, data types untuk semua fitur merupakan float

c. Cek data hilang, NaN, dan Outliers

Tidak ditemukan data yang hilang.

d. Cek Range dari data

Pengecekan range data dilakukan untuk mendapatkan batas bawah dan batas atas dari data.

e. Pengecekan fitur lainnya

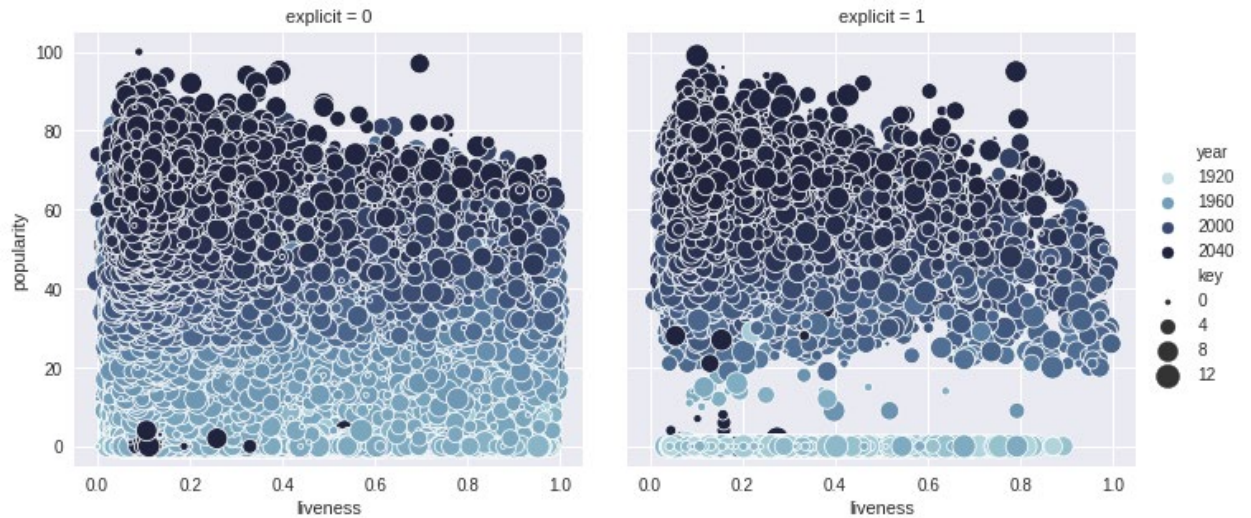
2. Data Preprocessing

a. Analisa Univariate

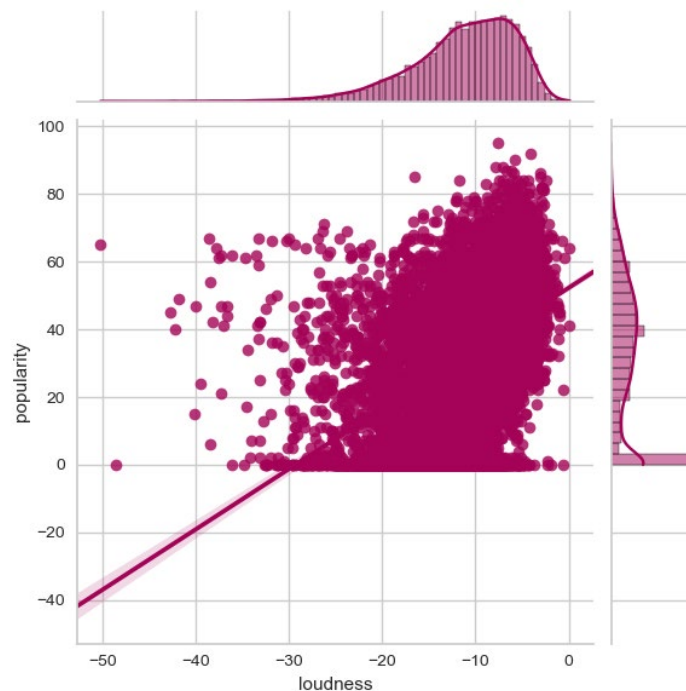
Dilakukan analisa Univariate untuk menghandle outlier. Untuk itu data distandardisasi ke mean 0 dan standard deviasi 1

b. Analisa bivariate

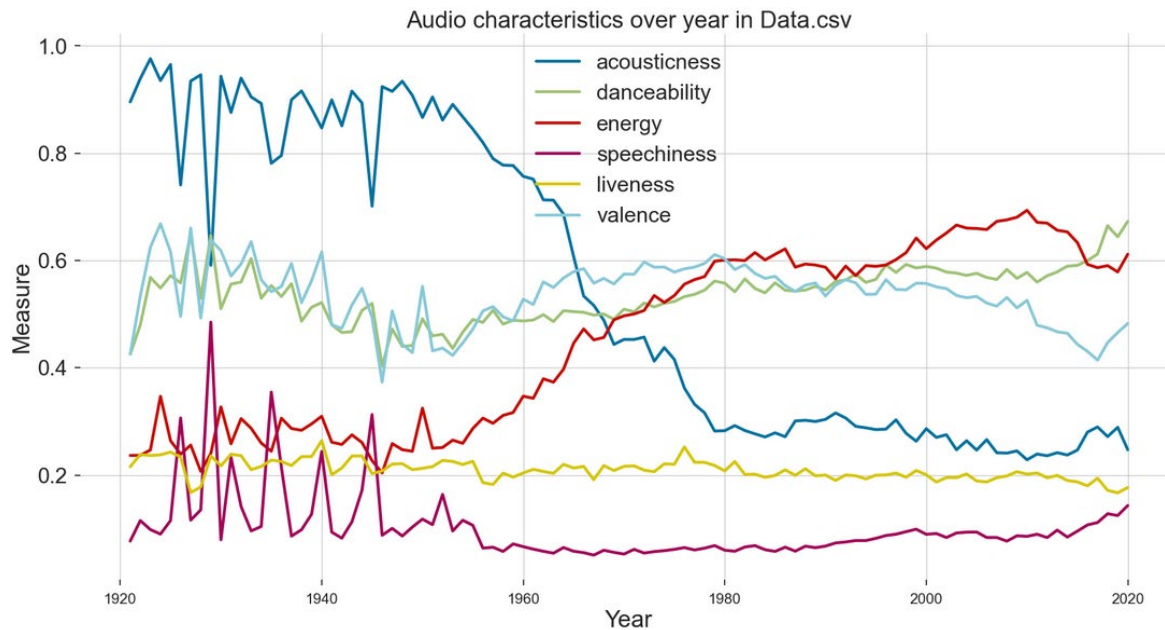
- Didapatkan hasil bahwa loudness memiliki relasi yang baik dengan popularity,
- "classical piano" genres memiliki "acousticness" dan tidak memiliki "speechiness" dan "energy"
- "movie tunes" dan "show Tunes" memiliki karakteristik Audio "valence" yang rendah dan "acousticness" yang tinggi.



loudness memiliki relasi yang baik dengan popularitas, berdasarkan genre, karakteristik audio akan berubah sesuai dengan key yang digunakan. sebagian besar genre "acousticness" memiliki efek lebih "movie tunes" and "show Tunes" terlihat memiliki karakteristik audio "valence" rendah dan acousticness tinggi genre "classical piano" memiliki "acousticness" dan tidak ada "speechiness" and "energy"

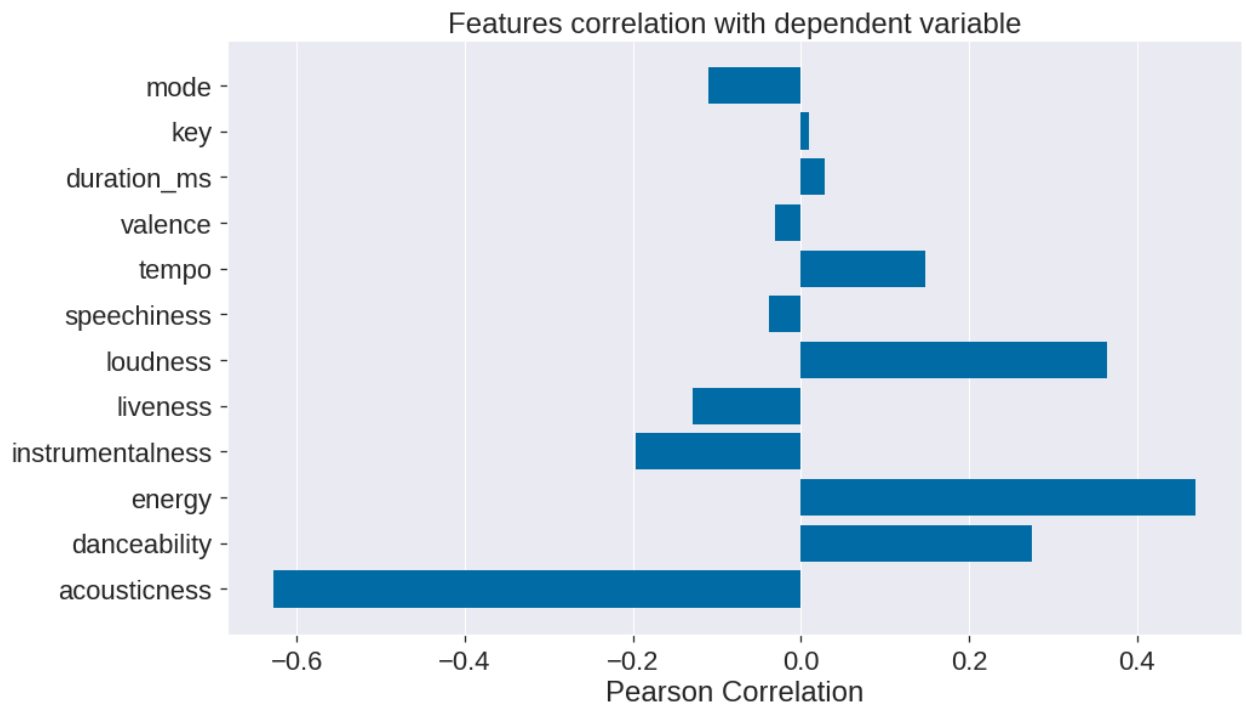


Perubahan karakteristik audio sejalan dengan waktu untuk berbagai variabel



3. Feature Selection

Dari analisa korelasi, yang memiliki korelasi positif dengan dependent variable adalah key, durasi, tempo, loudness, energy dan danceability. Sedangkan yang memiliki korelasi negative adalah mode, valance, speechiness, liveness, serta accousticness.



4. Modeling

Model yang digunakan adalah Random Forest. Hasil dari model dapat dievaluasi dengan menggunakan metrics accuracy adalah sebesar 93.68.

5. Serving

Setelah pipeline dari data pipeline, preprocessing dan modeling didapatkan dan dilakukan pengetesan, maka dilakukan serving deployment model. Ada 2 servis penting yang digunakan dalam deployment model yaitu Fast API dan Streamlit. Fast API digunakan sebagai jalur komunikasi antara Front End dan Back End sehingga data yang didapat dari user bisa digunakan untuk predict data menggunakan Fast API.

File dari main.py berisikan modul seperti Fast API, base model yang merupakan object class dari pydantic yang inherit dari base model, uvicorn untuk web server yang digunakan oleh python dan file python berupa pipeline yang telah dibuat. Setelah pembuatan object FAST API, dibuat struktur input data dan memuat model menggunakan class.

Streamlit digunakan sebagai User Interface agar user dapat memasukkan data. File streamlit.py berisikan modul streamlit, request yang digunakan untuk melakukan request pada web yang mengizinkan untuk mengirimkan HTTP request, dan PIL image untuk memasukkan gambar. Selanjutnya dibuat inputan dan select box menggunakan streamlite dan dibuat submit button berupa "Predict". Hasil input tersebut dimasukkan ke dalam JSON file dan diberikan output akhir berupa prediksi popularitas. Cara untuk melakukan akses adalah mengeksekusi pada terminal untuk file api dan menjalankan streamlit pada terminal secara terpisah.

```
PS C:\Users\PPL2\Documents\Spotify_Customer> streamlit run C:\Users\PPL2\Documents\Spotify_Customer\Spotify_app.py
```

```
You can now view your Streamlit app in your browser.
```

```
Local URL: http://localhost:8501
```

```
Network URL: http://10.111.163.144:8501
```

```
PS C:\Users\PPL2\Documents\Spotify_Customer> c:; cd 'c:\Users\PPL2\Documents\Spotify_Customer'; & 'c:\Users\PPL2\anaconda3\python.exe' 'c:\Users\PPL2\.vscode\extensions\ms-python.debugpy-2024.12.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '59290' '--' '-m' 'uvicorn' 'main:app' '--reload'
INFO: Will watch for changes in these directories: ['C:\Users\PPL2\Documents\Spotify_Customer']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [26028] using WatchFiles
INFO: Started server process [20268]
INFO: Waiting for application startup.
INFO: Application startup complete.
```

Hasil interface dari servis yang digunakan untuk prediksi popularitas lagu pada Spotify:



Spotify®

acousticness

0.2280

danceability

0.368

duration_ms

157840

energy

0.480

loudness

-11.605

speechiness

0.0306

valence

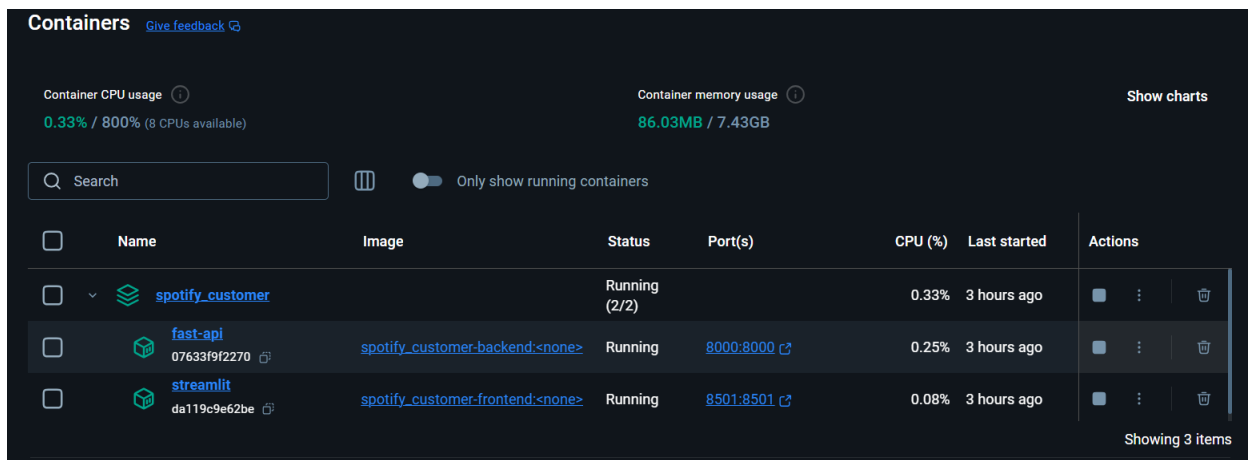
0.338

Predict

















The Popularity of the song is [47.48]

Sebagai langkah pertama, docker di build terlebih dahulu sehingga menghasilkan docker image yang selanjutnya dilakukan run pada wsl/ubuntu. Dua buah docker file tersebut menggunakan python:3.9.15-slim-buster dan terdapat dua docker image untuk masing-masing servis. Docker akan melakukan install terhadap dependensi yang telah ditulis pada requirement.txt.

Agar kedua docker dapat berjalan secara bersama, maka digunakan docker-compose.yml. File ini berisikan services yang digunakan (streamlit dan api), yang terdapat konfigurasi nama terhadap build, image, container_name, ports, dan volume (directory).



The screenshot shows the Docker Desktop interface with a dark theme. At the top, it displays 'Containers' with a 'Give feedback' link. Below this, there are two status bars: 'Container CPU usage' at 0.33% / 800% (8 CPUs available) and 'Container memory usage' at 86.03MB / 7.43GB. A 'Show charts' link is also present. A search bar and a toggle for 'Only show running containers' are located below the status bars. The main area contains a table of running containers. The table has columns for Name, Image, Status, Port(s), CPU (%), Last started, and Actions. Three containers are listed: 'spotify_customer' (Running 2/2), 'fast-api' (Running), and 'streamlit' (Running). Each container has a corresponding icon and a link to its details. The 'fast-api' container is highlighted in blue. At the bottom right, it says 'Showing 3 items'.

	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	 spotify_customer		Running (2/2)		0.33%	3 hours ago	  
<input type="checkbox"/>	 fast-api 07633f9f2270 	spotify_customer-backend<none>	Running	8000:8000 	0.25%	3 hours ago	  
<input type="checkbox"/>	 streamlit da119c9e62be 	spotify_customer-frontend<none>	Running	8501:8501 	0.08%	3 hours ago	  

Sampai tahapan ini, service sudah dapat berjalan pada localhost