

## TEMPLATE TUGAS ANALISIS DATA MINING

Mata Kuliah	: Data Mining
Topik	: Implementasi Algoritma Data Mining Menggunakan Python (Contoh)
Nama Mahasiswa	: Erich Christian
NPM	: 231510008
Dosen Pengampu	: Erlin Elisa, S.Kom., M.Kom.

---

### 1 Judul Tugas

Prediksi Popularitas Lagu Menggunakan Regresi Linier Berbasis Fitur Audio dan Genre Playlist

---

### 2 Latar Belakang Masalah

Fenomena popularitas lagu di industri musik modern merupakan subjek yang kompleks dan sangat dinamis. Dengan jutaan lagu yang dirilis setiap tahun, memahami faktor-faktor yang mendorong sebuah lagu menjadi populer menjadi krusial bagi artis, label rekaman, dan platform streaming musik. Dataset ini, yang berisi berbagai karakteristik audio (seperti danceability, energy, loudness, dan tempo) serta informasi genre, mencerminkan kerumitan ini. Tantangannya adalah mengidentifikasi pola tersembunyi dan hubungan antara fitur-fitur intrinsik sebuah lagu dan tingkat popularitasnya di kalangan pendengar. Tanpa pemahaman yang jelas, upaya promosi dan produksi seringkali menjadi tidak efisien, hanya mengandalkan intuisi atau tren yang cepat berlalu.

Dalam konteks ini, penggunaan *data mining* atau *machine learning* menjadi sangat diperlukan. Metode analisis tradisional kesulitan untuk menangani volume data yang besar dan interaksi non-linear yang kompleks antar variabel. Dengan teknik data mining, kita dapat menggali wawasan dari dataset ini, mengungkap fitur-fitur kunci yang secara signifikan berkorelasi dengan popularitas lagu, bahkan yang mungkin tidak terlihat secara langsung. Pendekatan ini memungkinkan kita untuk membangun model prediktif yang tidak hanya menjelaskan mengapa beberapa lagu menjadi hit, tetapi juga berpotensi untuk mengidentifikasi lagu-lagu dengan potensi popularitas tinggi di masa depan, memberikan keunggulan kompetitif di pasar musik yang ketat.

### 3 Rumusan Masalah

Contoh format:

1. Bagaimana proses preprocessing data pada dataset songs.csv untuk mempersiapkan data dalam membangun model prediksi popularitas lagu?
  2. Bagaimana implementasi algoritma Linear Regression pada dataset songs.csv untuk memprediksi track\_popularity?
  3. Bagaimana performa model Linear Regression dalam memprediksi track\_popularity berdasarkan metrik evaluasi seperti MSE, R-squared, dan MAE?
- 

### 4 Tujuan Penelitian

Contoh format:

1. Melakukan eksplorasi data (Exploratory Data Analysis) dan pra-pemrosesan pada dataset songs.csv untuk memahami karakteristik data dan mempersiapkannya untuk pemodelan.
2. Membangun model regresi linier untuk memprediksi track\_popularity berdasarkan fitur-fitur yang relevan dari dataset songs.csv.

3. Mengukur dan menganalisis performa model regresi linier menggunakan metrik evaluasi seperti Mean Squared Error (MSE), R-squared (R<sup>2</sup>), dan Mean Absolute Error (MAE).
- 

## 5 Dataset

- Sumber(Kaggle): <https://www.kaggle.com/datasets/rishabhpancholi1302/spotify-most-popular-songs-dataset>
- Jumlah record Adalah 900 & jumlah atribut dataset awal 28, dan setelah pra-pemrosesan menjadi 17 atribut
- Karakteristik data
  - Label: track\_popularity
  - Variabel Input:  
danceability, energy, key, loudness, mode, speechiness, acousticness, liveness, valence, tempo, duration\_ms, dan playlist\_genre.
- Format tabel singkat (opsional)

	danceability	energy	key	loudness	mode	speechiness	acousticness	liveness	valence	tempo	duration_ms	track_popularity	playlist_genre_Latin	
0	0.632680	0.667346	0.727273	0.680129	1.0	0.010572	0.015932	0.086004	0.391370	0.290605	228267	0.500000	False	
1	0.602614	0.425904	0.000000	0.504094	1.0	0.326045	0.261120	0.102930	0.687634	0.508374	194840	0.318182	False	
2	0.498039	0.628716	0.636364	0.821136	1.0	0.118212	0.079871	0.090236	0.656505	0.314439	174444	0.318182	False	
3	0.601307	0.799335	0.727273	0.753612	1.0	0.135031	0.277700	0.041997	0.538429	0.332383	201072	0.227273	False	
4	0.733333	0.886254	0.727273	0.781608	1.0	0.043969	0.201014	0.313445	0.370975	0.262872	223523	0.409091	False	
	playlist_genre_Pop	playlist_genre_R&b	playlist_genre_Rap	playlist_genre_Rock										
	True	False	False	False										
	True	False	False	False										
	True	False	False	False										
	True	False	False	False										
	True	False	False	False										

## 6 Metodologi

Deskripsikan langkah-langkah analisis:

- **Data Understanding**

Pada tahap ini, data songs.csv dieksplorasi untuk memahami struktur, karakteristik, dan kualitasnya. Langkah-langkah yang dilakukan meliputi:

  - **Pemeriksaan Informasi Data:** Menggunakan metode .info() untuk memeriksa tipe data, jumlah nilai non-null, dan penggunaan memori setiap kolom. Ini memastikan tidak ada nilai yang hilang secara implisit dan memahami jenis setiap atribut.
  - **Statistik Deskriptif:** Menampilkan ringkasan statistik (.describe()) untuk fitur-fitur numerik, guna mendapatkan pemahaman tentang distribusi, tendensi sentral, dan penyebaran data.
  - **Identifikasi Nilai Hilang:** Melakukan pengecekan (.isnull().sum()) untuk memastikan tidak ada nilai yang hilang (missing values) dalam dataset, yang mempermudah langkah pra-pemrosesan selanjutnya.
  - **Visualisasi Distribusi:** Membuat histogram untuk setiap fitur numerik untuk memvisualisasikan distribusinya, mengidentifikasi skewness, atau adanya outlier.
- **Data Preprocessing** (Handling Missing Value, Encoding, Scaling, dsb)

Tahap pra-pemrosesan data bertujuan untuk mempersiapkan dataset agar sesuai untuk pembangunan model. Langkah-langkah kunci meliputi:

- **Penanganan Nilai Hilang:** Berdasarkan analisis Data Understanding, tidak ditemukan nilai hilang sehingga tidak diperlukan penanganan khusus.
  - **One-Hot Encoding:** Kolom kategorikal playlist\_genre diubah menjadi format numerik menggunakan teknik one-hot encoding. Ini menghasilkan beberapa kolom biner baru (misalnya, playlist\_genre\_Pop, playlist\_genre\_Latin) yang memungkinkan model regresi untuk memproses informasi kategorikal.
  - **Seleksi Fitur Awal:** Pemilihan fitur numerik yang relevan (danceability, energy, key, loudness, mode, speechiness, acousticness, liveness, valence, tempo, duration\_ms) dan variabel target (track\_popularity). Beberapa fitur dalam dataset ini telah dinormalisasi atau distandardisasi sebelumnya (berada dalam skala 0-1).
- **Feature Selection** (jika diperlukan)
 

Fitur-fitur yang dipilih untuk pemodelan adalah kombinasi dari karakteristik audio numerik dan representasi one-hot encoded dari genre playlist. Variabel 'track\_popularity' ditetapkan sebagai variabel target (dependen), sementara fitur-fitur lainnya (X) bertindak sebagai variabel independen.
- **Pemodelan** (algoritma yang digunakan)
 

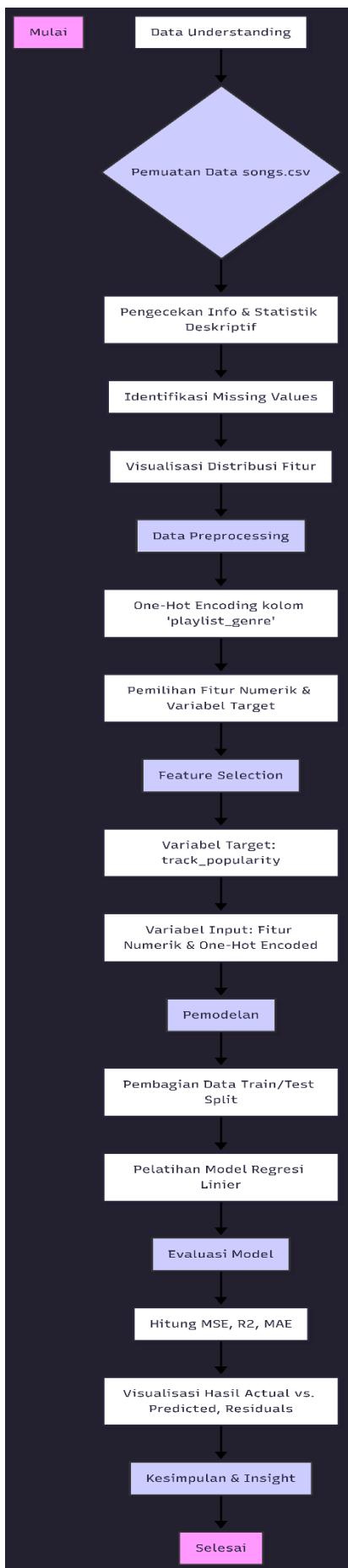
Model prediksi dibangun menggunakan algoritma **Regresi Linier (Linear Regression)**. Model ini dipilih karena kemampuannya untuk mengidentifikasi hubungan linier antara fitur input dan variabel target. Proses pemodelan meliputi:

  - Pembagian Data: Dataset dibagi menjadi set pelatihan (80%) dan set pengujian (20%) menggunakan train\_test\_split untuk memastikan model dievaluasi pada data yang belum pernah dilihat sebelumnya.
  - Pelatihan Model: Model Regresi Linier dilatih menggunakan set data pelatihan ( $X_{train}$  dan  $y_{train}$ ).
- **Evaluasi model**

Kinerja model Regresi Linier dievaluasi menggunakan metrik berikut pada set pengujian ( $X_{test}$  dan  $y_{test}$ ):

- **Mean Squared Error (MSE):** Mengukur rata-rata kuadrat perbedaan antara nilai aktual dan prediksi. Nilai yang lebih rendah menunjukkan akurasi yang lebih tinggi.
- **R-squared (R<sup>2</sup>):** Menunjukkan proporsi varians dalam variabel dependen yang dapat dijelaskan oleh model. Nilai mendekati 1 menunjukkan model yang sangat baik, sementara nilai rendah menunjukkan daya jelas model yang terbatas.
- **Mean Absolute Error (MAE):** Mengukur rata-rata magnitudo kesalahan prediksi, memberikan pemahaman langsung tentang seberapa besar rata-rata penyimpangan prediksi dari nilai sebenarnya.

Tambahkan diagram alur jika ada.



---

## 7 Implementasi Python

Berisi **kode program** per bagian:

✓ Import Library

```
▶ import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

print("Libraries pandas, numpy, matplotlib.pyplot, and seaborn imported successfully.")

... Libraries pandas, numpy, matplotlib.pyplot, and seaborn imported successfully.
```

✓ Load Dataset

```
df_songs = pd.read_csv('/content/songs.csv')
print("Dataset 'songs.csv' loaded successfully. Displaying the first 5 rows :")
display(df_songs.head().T)
```

Output hasilnya:

Dataset 'songs.csv' loaded successfully. Displaying the first 5 rows :			
	0	1	2
Unnamed: 0	0	1	2
track_id	6oJ6le65B3SEqPwMRNXWjY	3yNZ5r3LKfdmjoS3gkhUCT	0qc4QlcCxVTGyShurEv1UU
track_name	higher love	bad guy (with justin bieber)	post malone (feat. rani)
track_artist	Kygo	BillieEilish	Samfeldt
track_popularity	0.5	0.318182	0.318182
track_album_release_date	2019-06-28	2019-07-11	2019-05-24
playlist_genre	Pop	Pop	Pop
danceability	0.63268	0.602614	0.498039
energy	0.667346	0.425904	0.628716
key	0.727273	0.0	0.636364
loudness	0.680129	0.504094	0.821136
mode	1.0	1.0	1.0
speechiness	0.010572	0.326045	0.118212
acousticness	0.015932	0.26112	0.079871
liveness	0.086004	0.10293	0.090236
valence	0.39137	0.687634	0.656505
tempo	0.290605	0.508374	0.314439
duration_ms	228267	194840	174444
track_artist_merged	kygo	billie eilish	sam feldt
lyrics	['bring', 'me', 'higher', 'love', 'love', "..."]	['yeah', 'yeah', 'oh', 'ah', 'white']	['one', 'more', 'drink', 'got', 'one', 'more']
artist_name	Kygo	Billie Eilish	Sam Feldt

3	4
3	4
4PkIDTPGedm0enzdviILNd	5PYQUBXc7NYel1obMKSJK0
sixteen	never really over
Elliegoulding	Katyperry
0.227273	0.409091
2019-04-12	2019-05-31
Pop	Pop
0.601307	0.733333
0.799335	0.886254
0.727273	0.727273
0.753612	0.781608
1.0	1.0
0.135031	0.043969
0.2777	0.201014
0.041997	0.313445
0.538429	0.370975
0.332383	0.262872
201072	223523
ellie goulding	katy perry
['(sixteen)', '(sixteen)', 'do', 'you'...]	["i'm", 'losing', 'my', 'self', 'control', "..."
Ellie Goulding	Katy Perry

### ✓ Exploratory Data Analysis (EDA)

```
print("Displaying DataFrame Info:")
df_songs.info()

print("\nDisplaying Descriptive Statistics:")
df_songs.describe()

print("\nChecking for Missing Values:")
df_songs.isnull().sum()
```

Output yang dihasilkan:

```

Displaying DataFrame Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 900 entries, 0 to 899
Data columns (total 28 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Unnamed: 0        900 non-null    int64  
 1   track_id          900 non-null    object  
 2   track_name         900 non-null    object  
 3   track_artist       900 non-null    object  
 4   track_popularity   900 non-null    float64 
 5   track_album_release_date 900 non-null    object  
 6   playlist_genre     900 non-null    object  
 7   danceability       900 non-null    float64 
 8   energy             900 non-null    float64 
 9   key                900 non-null    float64 
 10  loudness           900 non-null    float64 
 11  mode               900 non-null    float64 
 12  speechiness        900 non-null    float64 
 13  acousticness       900 non-null    float64 
 14  liveness            900 non-null    float64 
 15  valence            900 non-null    float64 
 16  tempo               900 non-null    float64 
 17  duration_ms        900 non-null    int64  
 18  track_artist_merged 900 non-null    object  
 19  lyrics              900 non-null    object  
 20  artist_name         900 non-null    object  
 21  tags                900 non-null    object  
 22  tags_tokenized      900 non-null    object  
 23  doc_vector          900 non-null    object  
 24  combined_vector     900 non-null    object  
 25  cluster              900 non-null    int64  
 26  image_url           900 non-null    object  
 27  spotify_url          900 non-null    object  
dtypes: float64(11), int64(3), object(14)
memory usage: 197.0+ KB

```

Displaying Descriptive Statistics:	
Checking for Missing Values:	
	0
Unnamed: 0	0
track_id	0
track_name	0
track_artist	0
track_popularity	0
track_album_release_date	0
playlist_genre	0
danceability	0
energy	0
key	0
loudness	0
mode	0
speechiness	0
acousticness	0
liveness	0
valence	0
tempo	0
duration_ms	0
track_artist_merged	0
lyrics	0
	0
artist_name	0
tags	0
tags_tokenized	0
doc_vector	0
combined_vector	0
cluster	0
image_url	0
spotify_url	0
lyrics	0

Ada pula untuk fitur numerik

Codenya:

```

numerical_features = [
    'track_popularity', 'danceability', 'energy', 'key', 'loudness',
    'mode', 'speechiness', 'acousticness', 'liveness', 'valence',
    'tempo', 'duration_ms'
]

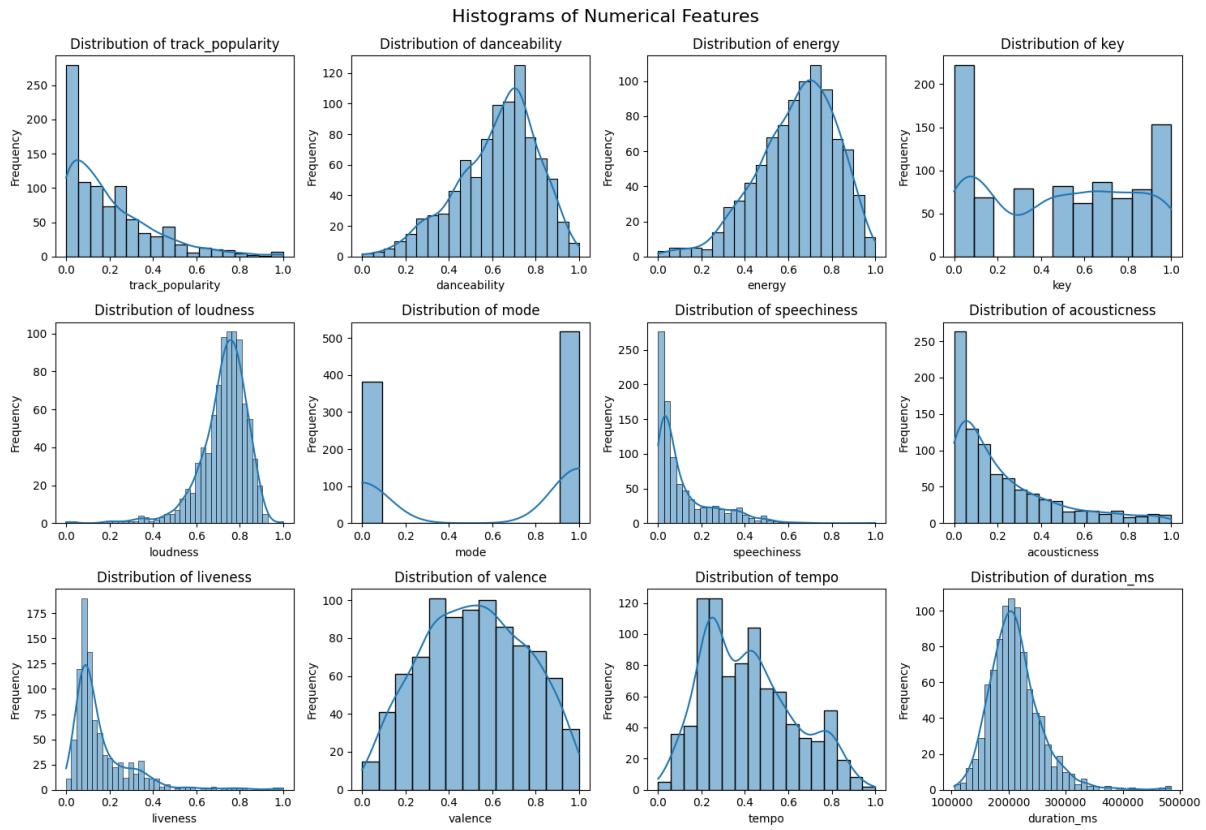
plt.figure(figsize=(15, 10))
for i, feature in enumerate(numerical_features):
    plt.subplot(3, 4, i + 1)
    sns.histplot(df_songs[feature], kde=True)
    plt.title(f'Distribution of {feature}')
    plt.xlabel(feature)
    plt.ylabel('Frequency')

plt.tight_layout()
plt.suptitle('Histograms of Numerical Features', y=1.02, fontsize=16)
plt.show()

print("Histograms for numerical features displayed successfully.")

```

Output yang dihasilkan:



## ✓ Preprocessing

```

numerical_features = [
    'danceability', 'energy', 'key', 'loudness',
    'mode', 'speechiness', 'acousticness', 'liveness', 'valence',
    'tempo', 'duration_ms'
]
target_variable = 'track_popularity'

# Create a new DataFrame with selected features and the target variable
df_model_features = df_songs[numerical_features + ['playlist_genre', target_variable]].copy()

# Apply one-hot encoding to 'playlist_genre'
df_encoded = pd.get_dummies(df_model_features, columns=['playlist_genre'], drop_first=True)

print("Processed DataFrame head after feature selection and one-hot encoding:")
print(df_encoded.head())
print(f"\nShape of the processed DataFrame: {df_encoded.shape}")

```

Output yang akan keluar:

```
Processed DataFrame head after feature selection and one-hot encoding:
   danceability    energy     key  loudness  mode speechiness \
0      0.632680  0.667346  0.727273  0.680129  1.0      0.010572
1      0.602614  0.425984  0.000000  0.504094  1.0      0.326045
2      0.498039  0.628716  0.636364  0.821136  1.0      0.118212
3      0.601307  0.799335  0.727273  0.753612  1.0      0.135031
4      0.733333  0.886254  0.727273  0.781608  1.0      0.043969

   acousticness  liveness  valence    tempo duration_ms track_popularity \
0      0.015932  0.086004  0.391370  0.290605      228267      0.500000
1      0.261120  0.182930  0.687634  0.508374      194840      0.318182
2      0.079871  0.090236  0.656505  0.314439      174444      0.318182
3      0.277700  0.041997  0.538429  0.332383      201072      0.227273
4      0.201014  0.313445  0.370975  0.262872      223523      0.409091

   playlist_genre_Latin  playlist_genre_Pop  playlist_genre_R&b \
0            False        True           False
1            False        True           False
2            False        True           False
3            False        True           False
4            False        True           False

   playlist_genre_Rap  playlist_genre_Rock
0            False           False
1            False           False
2            False           False
3            False           False
4            False           False

Shape of the processed DataFrame: (900, 17)
```

### ✓ Split Data (Train/Test)

Berikut code dan outputnya:

```
X = df_encoded.drop('track_popularity', axis=1)
y = df_encoded['track_popularity']

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

print(f"Shape of X_train: {X_train.shape}")
print(f"Shape of X_test: {X_test.shape}")
print(f"Shape of y_train: {y_train.shape}")
print(f"Shape of y_test: {y_test.shape}")
print("Data split into training and testing sets successfully.")

Shape of X_train: (720, 16)
Shape of X_test: (180, 16)
Shape of y_train: (720,)
Shape of y_test: (180,)
Data split into training and testing sets successfully.
```

### ✓ Model Training

```
model = LinearRegression()
model.fit(X_train, y_train)

print("Linear Regression model trained successfully.")

Linear Regression model trained successfully.
```

### ✓ Evaluasi Model

```

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error (MSE): {mse:.4f}")
print(f"R-squared (R2): {r2:.4f}")
print("Model evaluation complete.")

Mean Squared Error (MSE): 0.0396
R-squared (R2): 0.0730
Model evaluation complete.

```

✓ Visualisasi Hasil (confusion matrix, charts, dsb)

```

import seaborn as sns

# 1. Create a scatter plot of actual vs. predicted values
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=y_pred)
plt.title('Actual vs. Predicted Track Popularity')
plt.xlabel('Actual Track Popularity')
plt.ylabel('Predicted Track Popularity')
plt.grid(True)
plt.show()

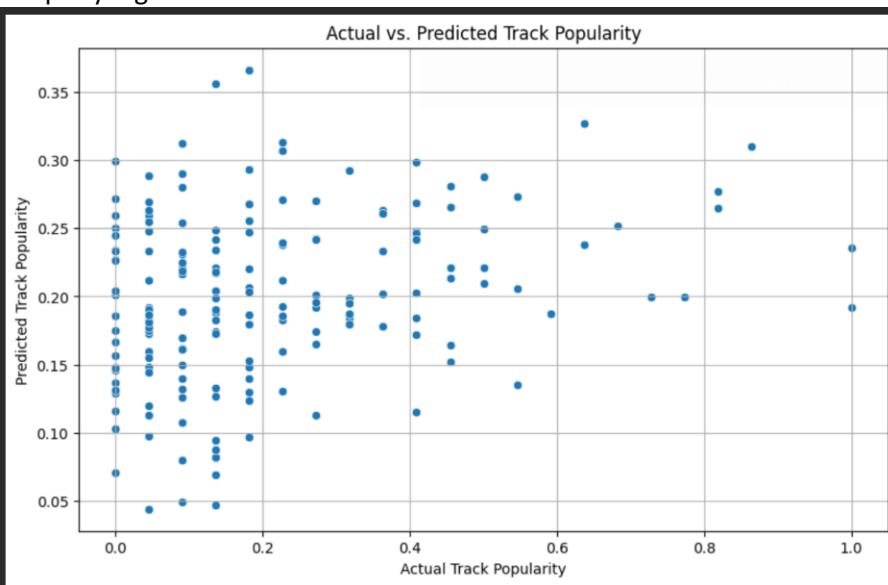
# 2. Calculate residuals
residuals = y_test - y_pred

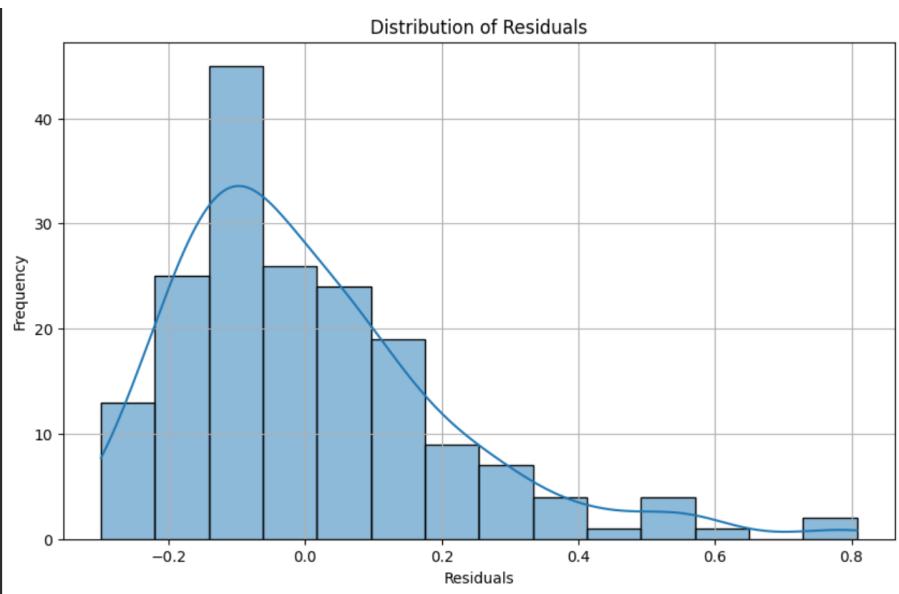
# 3. Create a histogram of residuals
plt.figure(figsize=(10, 6))
sns.histplot(residuals, kde=True)
plt.title('Distribution of Residuals')
plt.xlabel('Residuals')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

print("Scatter plot of actual vs. predicted values and histogram of residuals displayed successfully.")

```

Output yang akan dihasilkan:





Gunakan Google Colab (disarankan)

<https://colab.research.google.com/drive/1neIHs62K7C720QXKOn6pMjVG60ztQk6P#scrollTo=d4867bdc>

## 8 Hasil dan Pembahasan

- Tampilkan hasil evaluasi (Accuracy, F1-score, MAE, MSE, Silhouette Score, dll)
- Jelaskan interpretasi hasil model
- Jika menggunakan lebih dari satu algoritma → bandingkan performanya

Contoh tabel hasil evaluasi:

Algoritma	Mean Squared Error (MSE)	R-squared (R2)	Mean Absolute Error (MAE)
Linear Regression	0.0396	0.0730	0.1506

## 9 Kesimpulan

1. Proses pra-pemrosesan data berhasil dilakukan dengan mengidentifikasi 11 fitur numerik audio yang relevan dan menerapkan one-hot encoding pada kolom playlist\_genre. Hasilnya adalah dataset yang bersih dan terstruktur dengan 900 record dan 17 atribut, siap untuk tahap pemodelan.
2. Model regresi linier berhasil diimplementasikan dan dilatih menggunakan data yang telah diproses. Dataset dibagi menjadi 80% data pelatihan dan 20% data pengujian untuk memastikan evaluasi model yang objektif.
3. Model regresi linier menunjukkan kinerja dengan Mean Squared Error (MSE) sebesar 0.0396, Mean Absolute Error (MAE) sebesar 0.1506, dan R-squared (R2) sebesar 0.0730 pada set pengujian. Nilai R2 yang rendah mengindikasikan bahwa model regresi linier dengan fitur yang ada hanya mampu menjelaskan sekitar 7.3% dari variabilitas popularitas lagu, menyiratkan bahwa masih banyak faktor lain yang mempengaruhi popularitas lagu yang belum tertangkap oleh model atau bahwa hubungan antar variabel mungkin bersifat non-linier.

## 10 Saran

- Pengembangan model berikutnya Mengingat nilai R-squared yang masih rendah, disarankan untuk melakukan eksplorasi fitur (Feature Engineering) yang lebih mendalam. Ini bisa melibatkan pembuatan fitur baru dari data yang sudah ada atau mencari sumber data eksternal yang relevan (misalnya, data tren streaming, engagement media sosial, atau informasi tentang artis). Analisis korelasi yang lebih cermat atau teknik pemilihan fitur yang lebih canggih (seperti Recursive Feature Elimination) juga dapat diterapkan untuk mengidentifikasi fitur yang paling prediktif.
- Penggunaan data lebih besar / algoritma lain
  - **Peningkatan Ukuran Dataset:** Menggunakan dataset yang lebih besar dan lebih beragam (misalnya, mencakup periode waktu yang lebih panjang, lebih banyak genre, atau berbagai platform) dapat membantu model menangkap pola yang lebih kompleks dan meningkatkan kemampuan generalisasi. Jika memungkinkan, penggunaan data real-time atau data historis yang lebih kaya dapat sangat bermanfaat.
  - **Eksplorasi Algoritma Prediktif Alternatif:** Selain regresi linier, disarankan untuk menguji algoritma machine learning lain yang mampu menangani hubungan non-linier dengan lebih baik. Contoh algoritma yang bisa dicoba antara lain: Random Forest Regressor, Gradient Boosting Regressor (seperti XGBoost atau LightGBM), Support Vector Regressor (SVR), atau bahkan pendekatan neural network sederhana. Perbandingan kinerja antar model-model ini dapat membantu mengidentifikasi algoritma terbaik untuk tugas prediksi popularitas lagu.

---

## 1 1 Daftar Pustaka

<https://www.kaggle.com/datasets/rishabhpancholi1302/spotify-most-popular-songs-dataset>

---

### Alternatif Topik Algoritma Data Mining

Mahasiswa boleh memilih satu dari daftar ini:

- Klasifikasi (Decision Tree, Naive Bayes, Random Forest, SVM)
  - Clustering (K-Means, DBSCAN, Hierarchical)
  - Regresi (Linear Regression, Random Forest Regression)
  - Association Rule (Apriori, FP-Growth)
  - Deep Learning Dasar (ANN untuk klasifikasi)
- 

### Rubrik Penilaian

Komponen Penilaian	Bobot	
Pemilihan dan pemahaman dataset	15%	
Preprocessing & EDA	20%	
Implementasi Model Data Mining	25%	
Evaluasi dan Analisis Hasil	25%	
Penulisan laporan dan kerapian	15%	

