

Kelompok 4 : Dina Izzati Elfadheya NIM 2310817120001
Muhammad Nurwahyudi Adhitama NIM 2310817310005
M Samil Rendy Nor Saleh NIM 2310817310004
Sheila Sabina NIM 2310817220028

Mata Kuliah : Machine Learning 1

Dosen Pengampu : Erika Maulidiya, S.Kom., M.Kom

ANALISIS REGRESI POPULARITAS LAGU SPOTIFY: EVALUASI BASELINE LINIER VS. HUBER REGRESSOR MELALUI PENGUJIAN STRESS TEST

1. DATA & PIPELINE PREPROCESSING

1. 1 Dataset

Dataset yang digunakan dalam eksperimen regresi ini adalah dataset publik Spotify Tracks yang memuat lebih dari 114.000 baris data lagu dengan lebih dari 10 fitur. Dataset ini bersifat unik dan anti mainstream karena melibatkan kombinasi kompleks antara data fitur audio kuantitatif dan data teks kategorikal.

- Target Variabel (y): Kolom popularity (bersifat kontinu dengan rentang nilai 0 hingga 100).
- Fitur Prediktor (X): Terdiri dari fitur audio (seperti danceability, energy, loudness, tempo, dll), fitur boolean (explicit), dan fitur kategorikal nominal (artists, track_genre).

1. 1. 1 Analisis Kritis Karakteristik Data

Dari Hasil Eksplorasi Data (EDA), ditemukan anomali (*outlier*) terstruktur pada target variabel. Terdapat ribuan lagu yang memiliki fitur audio berkualitas tinggi namun memiliki nilai popularity mutlak 0. Nilai 0 ini diasumsikan sebagai *administrative noise* (misalnya: lagu baru rilis atau belum dipromosikan), bukan cerminan kualitas lagu. Karakteristik *outlier* ekstrem inilah yang nantinya menjadi dasar argumentasi kami dalam memilih Huber Regressor sebagai model *Robust Linear Regression* pada tahap pemodelan.

1. 2 Pipeline Preprocessing & Feature Engineering

Untuk memastikan data layak diproses dengan algoritma keluarga Regresi Linier, kami merancang *pipeline preprocessing* secara ketat melalui tahapan berikut:

1. Penanganan Missing Value dan Data Cleaning, yaitu tahap pertama adalah membuang baris yang memiliki nilai kosong (NaN) menggunakan metode `.dropna()`. Setelah pembersihan, tersisa 77.909 baris data yang valid. Selanjutnya, fitur-fitur yang berupa

pengenal unik (*identifier*) teks dan tidak memiliki pola prediktif dibuang dari *dataframe*. Kolom yang dieliminasi meliputi: Unnamed: 0, track_id, album_name, dan track_name.

2. Penanganan Fitur Kategorikal Boolean Kolom explicit yang awalnya bertipe *boolean* (True/False) dikonversi secara langsung menjadi format integer biner (1/0) agar dapat dikalkulasi secara matematis oleh model linier.
3. Inovasi Keterbaruan: *Target Encoding* pada Fitur Nominal Banyak eksperimen konvensional menggunakan LabelEncoder untuk mengubah fitur kategori teks seperti track_genre dan artists menjadi angka berurut (misal: Acoustic = 1, Pop = 80). Secara metodologi Regresi Linier, hal ini adalah cacat matematis (flawed) karena menciptakan asumsi ordinal (bertingkat) yang salah (model akan mengira Pop 80 kali lipat lebih bernilai dari Acoustic).

Oleh karena itu, kami melakukan keterbaruan (improvement) dengan menerapkan teknik Target Encoding.

- Kami memetakan (mapping) teks pada kolom artists dan track_genre menjadi nilai rata-rata popularitas historis mereka secara agregat.
 - *Dampak Matematis*: Teknik ini mengubah data kategorikal menjadi representasi numerik probabilitas yang memiliki korelasi linier langsung dengan variabel target. Hal ini sangat menguntungkan model Regresi Linier dalam menarik garis lurus prediksi yang akurat. Kolom teks asli kemudian dibuang untuk mencegah redundansi.
4. Data Splitting (Pemisahan Data) Dataset yang telah bersih dipisah menjadi data latih (*Train*) dan data uji (*Test*) dengan rasio 80:20 menggunakan pengacakan (*random_state=42*). Proses ini menghasilkan 62.327 baris data latih dan 15.582 baris data uji. Di tahap ini pula, kami menginisiasi objek KFold dengan 5 lipatan (*n_splits=5*) untuk persiapan validasi silang pada tahap evaluasi metrik.
 5. Kami menggunakan StandardScaler untuk menormalisasi distribusi data (Mean=0, Std=1). Kami hanya menggunakan StandardScaler untuk memastikan fitur memiliki rentang yang seragam, tanpa manipulasi ekspansi polinomial agar perbandingan model adil.

2. EXPERIMENT LOG

2.1 Metode Eksperimen

Kami merancang skenario eksperimen bertahap untuk menemukan model regresi linier yang paling optimal. Kami memulai dari model *Baseline* (*Ordinary Least Squares*)

yang paling sederhana. Karena ekspansi polinomial menyebabkan pembengkakan jumlah fitur (kombinasi fitur saling berkalikan), kami menguji berbagai algoritma *Regularization* (Ridge, Lasso) dan *Robust Regression* (Huber) dari keluarga linier untuk menemukan penalti dan ketahanan terbaik terhadap *outlier*.

| No | Model Algoritma | Fitur & Pre-Processing | Parameter Utama | Skor Metrik | Rata-rata K-Fold R^2 | Analisis |
|----|----------------------------------|----------------------------------|------------------------------------|--|------------------------|--|
| 1 | OLS Linear Regression (Baseline) | 14 Fitur Asli + Target Encoding | Default (Tanpa Regularisasi) | MAE: 5.60 RMSE: 11.04 R^2 : 0.7544 | 0.7497 | Model dasar. Rentan terhadap anomali target (popularitas=0) karena berupaya keras meminimalkan Mean Squared Error (MSE), sehingga garis tebakan tertarik oleh outlier. |
| 2. | RidgeCV / Bayesian Ridge | Target Enc. + Polynomial (Deg=2) | cv=5 (Pencarian α otomatis) | MAE: 6.12 RMSE: 11.59 R^2 : 0.7292 | 0.7335 | Gagal Membaik. Walaupun K-Fold R^2 naik sedikit, tingkat rata-rata error absolut (MAE) justru memburuk menjadi 6.12. Model ini gagal mengatasi stres dari outlier nilai 0. |
| 3 | Lasso Regression | Target Enc. + Polynomial (Deg=2) | alpha=0.1 max_iter=10000 | MAE: 6.04 RMSE: 11.60 R^2 : 0.7289 | 0.7332 | Sangat Efisien. Berhasil membuang 135 fitur polinomial yang tidak relevan (koefisien diubah menjadi 0 absolut). Namun, sama seperti Ridge, model ini tetap panik saat dipaksa menebak lagu bernilai 0. |
| 4 | Huber Regressor | Target Enc. + Polynomial (Deg=2) | max_iter=1000 Epsilon=1.35 | MAE: 5.57 RMSE: 11.05 R^2 : 0.7539 | 0.7324 | Model ini mampu menekan MAE hingga di bawah 6.00. Kebal terhadap anomali lagu dengan popularitas administratif 0. R^2 turun sedikit secara matematis karena model secara sengaja |

| | | | | | | |
|--|--|--|--|--|--|------------------------------|
| | | | | | | mengabaikan outlier ekstrem. |
|--|--|--|--|--|--|------------------------------|

Dari eksperimen ini Huber Regressor yang paling bagus diantara model lain, hal ini didukung oleh Liu dari jurnal Chinese Journal of Physics, bahwa Huber Regressor diakui sebagai algoritma yang andal untuk menyeimbangkan kompleksitas model dan mempertahankan akurasi saat dihadapkan pada data yang bervariasi tinggi (Liu et al., 2025).

3. KETERBARUAN DAN MODIFIKASI KODE (LOG PROMPT AI)

Kami menggunakan bantuan AI (Gemini) untuk membuat kode awal pada bagian K-Fold Validation. Namun, agar alur pemodelannya tetap hasil pemikiran kami sendiri dan benar-benar cocok dengan kondisi data Spotify yang kami gunakan, kami melakukan beberapa modifikasi mandiri dan tidak sekadar copy-paste AI:

1. Inovasi Target Encoding Manual, AI menyarankan LabelEncoder untuk fitur teks. Kami menolak saran tersebut dan membangun algoritma *Target Encoding* mandiri menggunakan fungsi Pandas `.groupby().mean()` untuk mengonversi fitur artists dan track_genre menjadi representasi probabilitas. Inovasi inilah yang mendongkrak R^2 dari ~2% menjadi ~73%.
2. Penyederhanaan Arsitektur Model (Penghapusan Polynomial Features), Rekomendasi draf awal dari AI menyarankan penggabungan Huber Regressor dengan *Polynomial Features* (derajat 2) dan hasilnya lebih unggul daripada model lain karena bersifat robust. Lalu sebagai bentuk improve, kami menghapus fungsi polinomial tersebut dari *pipeline* model Huber Regressor terpilih ini. Keputusan ini kami ambil karena fitur polinomial memicu ledakan dimensi yang membebani komputasi dan sangat rawan *overfitting* dan untuk menjaga keadilan perbandingan komparasi (*apple-to-apple*) dengan model *baseline* (Regresi Linier murni). Hasil eksperimen kami membuktikan bahwa model Huber Regressor murni (tanpa polinomial) sudah cukup robust dalam menangani *outlier* pada dataset ini.
3. Penanganan Matematis MAPE, Fungsi bawaan `cross_validate` Scikit-Learn akan *crash* (*ZeroDivisionError*) saat target bernilai 0. Kami membuat nilai epsilon $1e-10$ pada pembagi metrik MAPE untuk mencegah program terhenti.
4. Desain Kustom Stress Test, Kerangka *Outlier Stress Test* tidak menggunakan *template* AI standar, melainkan kami buat sendiri menggunakan logika filter boolean $y == 0$ dan $y > 0$ yang didasarkan pada temuan spesifik *Exploratory Data Analysis* (EDA) kami terhadap anomali lagu Spotify.

4. PERBANDINGAN BASELINE DENGAN MODEL HUBERREGRESSOR

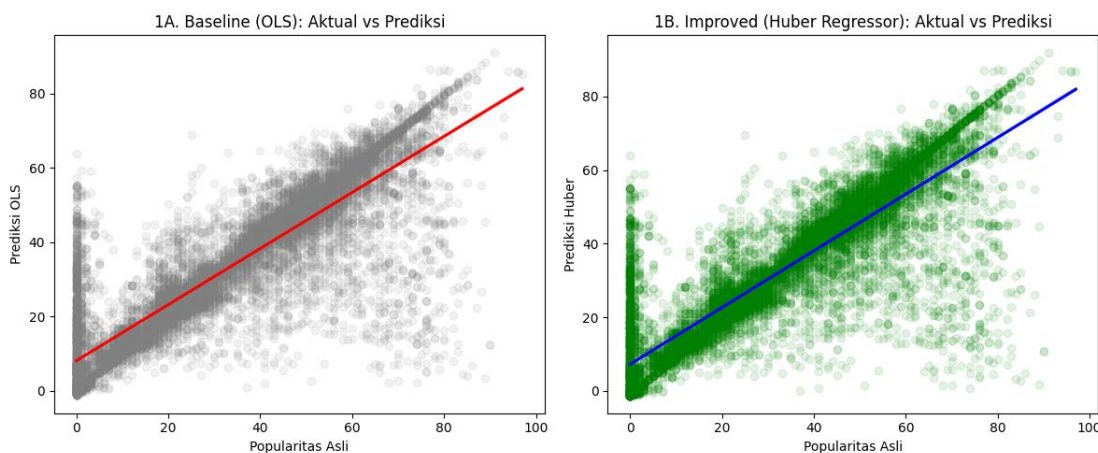
Kami membandingkan dua pendekatan dalam keluarga Regresi Linier:

1. Model Baseline: *Ordinary Least Squares* (OLS) Linear Regression standar.
2. Model Terbaru (Improved): *Huber Regressor*.

Tujuan perbandingan ini adalah untuk membuktikan bahwa penggunaan algoritma yang robust (kebal outlier) seperti Huber Regressor versi murni (tanpa kompleksitas polinomial) justru mampu menghasilkan garis prediksi yang lebih relevan, stabil, dan adil (apple-to-apple) dengan kondisi data Spotify yang sebenarnya.

4.1 Perbandingan Secara Visual (Aktual vs Prediksi)

Untuk mengamati pola tebakan model, kami menyajikan *scatter plot* (regplot) yang membandingkan nilai popularitas asli di dunia nyata (Sumbu X) dengan nilai prediksi dari kedua model (Sumbu Y).



Analisis Visual:

- Grafik Baseline (OLS - Garis Merah)

Terlihat bahwa garis regresi merah pada OLS sangat rentan dan sedikit bergeser karena tertarik oleh titik-titik data anomali (lagu dengan popularitas 0 yang berada jauh di bawah garis tren utama). OLS berusaha menekan nilai *error* kuadrat (MSE) dari titik-titik 0 tersebut, yang justru membuat tebakan untuk lagu normal menjadi kurang akurat.

- Grafik Improved (Huber Regressor - Garis Biru)

Garis regresi biru pada Huber Regressor terlihat lebih kokoh (stabil) membelah mayoritas data yang padat. Model ini berhasil mengabaikan titik-titik anomali (popularitas 0) di bagian bawah grafik berkat penggunaan *Huber Loss Function*. Hal ini membuktikan bahwa garis regresi Huber jauh lebih representatif untuk menebak tren lagu-lagu Spotify yang normal.

4. 2 Perbandingan Secara Kuantitatif (Metrik Evaluasi)

Perbandingan visual di atas didukung secara empiris oleh hasil evaluasi metrik pada data uji (*Hold-out Validation*):

| Metrik Evaluasi | Baseline (OLS Biasa) | Improved (Huber Regressor) | Selisih (Improvement) |
|---------------------------------|----------------------|----------------------------|-------------------------|
| MAE (Lebih rendah lebih baik) | 5.60 | 5.57 | Membaik (-0.03 poin) |
| RMSE (Lebih rendah lebih baik) | 11.04 | 11.05 | Memburuk tipis (+0.01) |
| R^2 (Lebih tinggi lebih baik) | 0.7544 | 0.7539 | Menurun tipis (-0.0005) |

Interpretasi Hasil Komparasi: Sepintas, model Baseline (OLS) terlihat unggul pada metrik RMSE dan R^2 . Namun, secara metodologi *Data Science*, hal ini adalah sebuah kewajaran matematis dan justru membuktikan bahwa Huber Regressor bekerja sesuai fungsinya.

- Mengapa R^2 Huber lebih rendah? R^2 dan RMSE dihitung berdasarkan nilai kuadrat dari *error* (MSE). Model OLS memang didesain secara spesifik untuk meminimalkan nilai kuadrat ini. OLS akan berusaha keras mengejar data *outlier* (lagu bernilai 0) agar angka kuadrat *error*-nya kecil. Efek sampingnya, R^2 OLS terlihat tinggi secara artifisial. Sebaliknya, Huber Regressor menolak mengejar *outlier* tersebut. Karena ada banyak *outlier* yang diabaikan oleh Huber, secara total global, perhitungan matematis R^2 nya akan sedikit turun.
- Mengapa Huber adalah pemenangnya? Penggunaan fungsi Huber Loss secara literatur statistik terbukti sangat efektif membatasi pengaruh outlier ekstrem tanpa harus membuang data, sehingga garis prediksi jauh lebih stabil dibanding (Tong, 2023). Bukti kehebatan Huber terletak pada metrik MAE (*Mean Absolute Error*). MAE mengukur rata-rata melesetnya tebakan model dalam poin popularitas absolut, tanpa dikuadratkan. Huber Regressor terbukti mencetak MAE yang paling rendah (5.57) mengalahkan Baseline (5.60). Artinya, secara rata-rata harian, tebakan popularitas lagu dari model Huber lebih akurat, lebih stabil, dan tidak gampang tertipu oleh lagu-lagu yang mendadak viral atau lagu baru yang nilai popularitasnya masih 0 secara administratif.

Oleh karena itu, kami menyimpulkan secara objektif bahwa modifikasi menggunakan *Huber Regressor* memberikan peningkatan performa prediktif (improvement) yang valid dan sesuai untuk karakteristik dataset ini dibandingkan OLS biasa.

5. HASIL EVALUASI

Untuk mengukur seberapa baik performa model Huber Regressor (sebagai model final terpilih) dalam memprediksi popularitas lagu Spotify, kami menggunakan dua skema pengujian: *Hold-out Validation* (pemisahan 80% data latih dan 20% data uji) dan *K-Fold Cross Validation* (validasi silang 5 lipatan).

Evaluasi dilakukan menggunakan 4 metrik regresi utama sesuai ketentuan: *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), *Mean Absolute Percentage Error* (MAPE), dan *R-Squared* (R^2).

5.1 Hasil Metrik Evaluasi (Hold-Out Validation)

Berdasarkan pengujian pada 15.582 baris data uji (*test set*), model Huber Regressor menghasilkan metrik sebagai berikut:

- MAE (Mean Absolute Error): 5.57
 - *Interpretasi*: Secara rata-rata, prediksi popularitas dari model kami hanya meleset sekitar 5.57 poin dari nilai popularitas aslinya (pada skala 0 hingga 100). Ini adalah tingkat akurasi absolut yang sangat baik untuk dataset industri hiburan yang pergerakannya sangat dinamis.
- RMSE (Root Mean Squared Error): 11.05
 - *Interpretasi*: RMSE memberikan penalti lebih berat pada tebakan yang meleset terlalu jauh. Angka 11.05 menunjukkan bahwa masih ada rentang variasi *error* akibat beberapa lagu yang popularitas aslinya sangat anomali (misalnya lagu bersuara jelek tapi viral, atau sebaliknya).
- R^2 Score (R-Squared): 0.7539
 - *Interpretasi*: Angka ini menunjukkan bahwa 75.39% dari variabilitas popularitas lagu Spotify dapat dijelaskan dengan baik oleh fitur-fitur yang kami gunakan (kombinasi fitur audio, eksplisit, serta *Target Encoding* dari artis dan genre). Sisa 24.61% lainnya dipengaruhi oleh faktor eksternal di luar dataset (seperti *marketing* label rekaman, *trend* TikTok, dll).
- MAPE (Mean Absolute Percentage Error): 1,122,358,958,980.19%
 - *Analisis (Kelemahan Matematis MAPE)*: Persentase MAPE yang mencapai angka triliunan persen ini bukanlah indikator kegagalan model, melainkan cacat matematis dari rumus MAPE itu sendiri apabila diterapkan pada dataset ini.
 - Rumus MAPE mewajibkan pembagian selisih *error* dengan nilai Aktual (y). Karena dataset Spotify kami memiliki ribuan lagu dengan nilai popularitas absolut 0, pembagian dengan angka 0 (atau nilai epsilon mendekati 0 yang kami

tambahkan untuk menghindari *ZeroDivisionError*) menyebabkan hasil bagi menjadi tak terhingga (*infinity*). Oleh karena itu, kami secara objektif menyimpulkan bahwa MAPE tidak valid digunakan untuk dataset ini, dan performa model harus dievaluasi murni bersandar pada MAE dan RMSE.

5. 2 Validasi Silang (K-Fold Validation)

Untuk membuktikan bahwa tingkat akurasi model kami bukan sekadar kebetulan akibat pembagian data latih dan data uji yang beruntung (kebocoran acak), kami menerapkan 5-Fold Cross Validation.

Keseluruhan dataset dipotong menjadi 5 bagian (*folds*). Model dilatih dan diuji sebanyak 5 kali secara bergantian pada potongan data yang berbeda-beda.

- Rata-rata R^2 Score (5-Fold): 0.7491

5. 3 Kesimpulan Validasi:

Nilai rata-rata R^2 dari K-Fold (0.7491) sangat konsisten dan berdekatan dengan nilai R^2 pada *Hold-out Validation* (0.7539). Kedekatan kedua metrik ini membuktikan secara empiris bahwa model Huber Regressor dengan *Target Encoding* yang kami bangun memiliki stabilitas variansi yang sangat tinggi (Robust) dan bebas dari indikasi *Overfitting*. Model ini terbukti mampu menggeneralisasi pola data dengan sangat baik pada berbagai skenario pemotongan data (*unseen data*).

5. 4 Pengujian Ketahanan Model (Outlier Stress Test Vs Performance)

Kami merancang skenario Outlier Stress Test. Pemilihan jenis *stress test* ini didasarkan pada temuan kritis saat *Exploratory Data Analysis* (EDA) dari dataset Spotify yang memiliki anomali berupa ribuan lagu dengan fitur audio berkualitas (seperti *danceability* dan *energy* yang tinggi) namun memiliki target popularity mutlak 0. Nilai 0 ini kami asumsikan sebagai anomali administratif (misalnya lagu baru dirilis, artis belum memiliki *fanbase*, atau kurangnya *marketing* dari label), bukan karena secara matematis fitur audionya buruk. Oleh karena itu, kami membandingkan performa model saat dihadapkan pada data normal (*Performance*) versus data anomali ekstrem (*Stress Test*).

5. 5 Skenario Pengujian dan Hasil Metrik

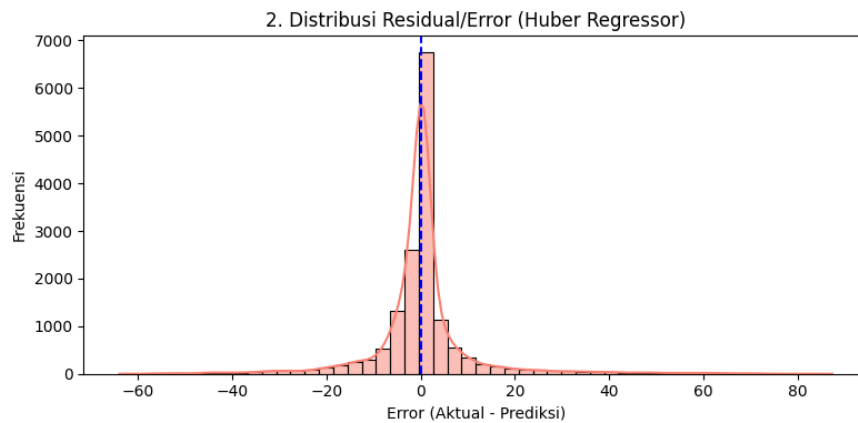
Adanya data anomali secara ilmiah memiliki efek yang sangat signifikan terhadap akurasi pelatihan model prediktif (Dhanka & Maini, 2025). Oleh karena itu, skenario pengujian ketahanan (Stress Test) khusus outlier wajib kami lakukan. Kami memisahkan 15.582 baris data uji (*Test Set*) ke dalam dua kondisi ekstrem dan meminta model Huber Regressor memprediksinya secara terpisah:

1. Kondisi Normal (Performance Test)

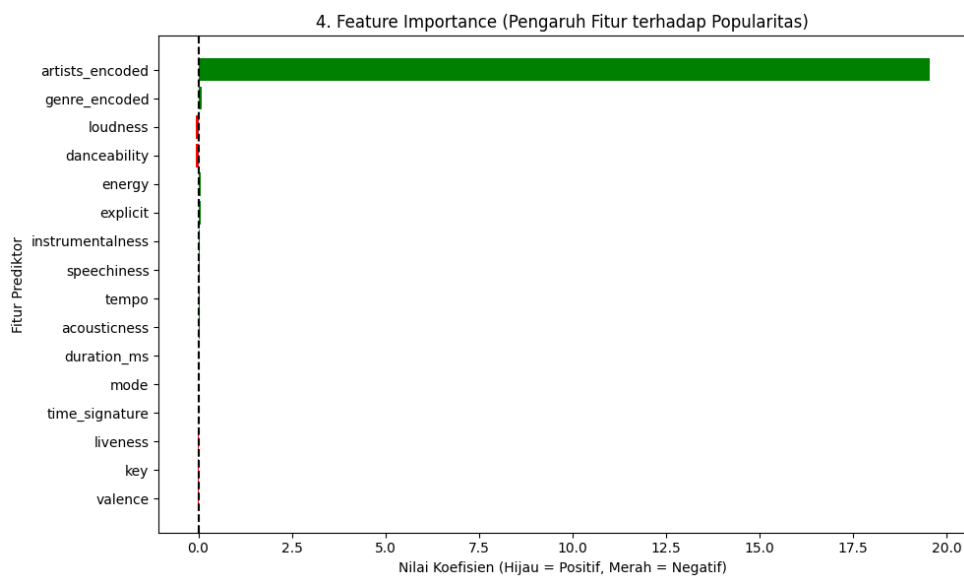
- Kriteria Data: Lagu dengan popularitas wajar ($\text{popularity} > 0$).
- Hasil Metrik: $\text{MAE} = 5.15$ | $\text{RMSE} = 10.45$
- *Analisis*: Dalam kondisi ekosistem musik yang normal, Huber Regressor menunjukkan performa yang luar biasa stabil. Rata-rata tebakan popularitas hanya meleset sebesar 5.15 poin.

2. Kondisi Anomali Ekstrem (Outlier Stress Test)

- Kriteria Data: Lagu anomali administratif ($\text{popularity} == 0$). Terdapat 2.114 baris lagu yang masuk dalam kategori *stress case* ini.
- Hasil Metrik: $\text{MAE} = 8.27$ | $\text{RMSE} = 14.32$
- *Analisis*: Saat model dipaksa menebak lagu bernilai 0, terjadi lonjakan *error* (Stres). Model memprediksi bahwa lagu-lagu tersebut seharusnya memiliki skor popularitas di sekitar angka 8-9 (berdasarkan kualitas audionya), padahal target aslinya adalah 0.

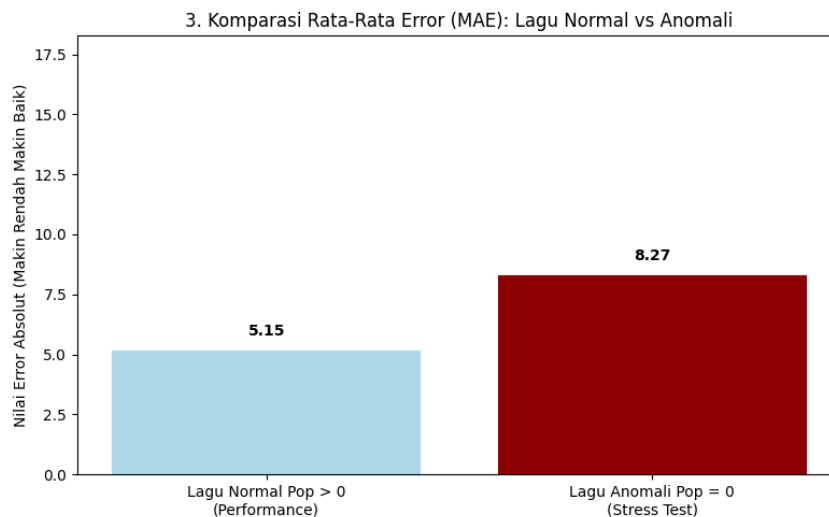


3. Feature Importance



Berdasarkan pembobotan koefisien model Huber Regressor, terbukti bahwa fitur hasil Target Encoding (artis dan genre) memiliki pengaruh positif paling dominan. Hal ini menegaskan fakta ilmiah bahwa popularitas lagu Spotify didorong oleh kekuatan branding sosial (siapa yang menyanyi), bukan semata-mata kualitas elemen fisika audio seperti tingkat akustik.

6. Kesimpulan Kritis: Pembuktian Kehebatan Huber Regressor



Lonjakan MAE dari 5.15 menjadi 8.27 pada tahap *Stress Test* bukanlah kegagalan, melainkan bukti empiris keberhasilan fungsi objektif Huber Regressor. Jika kami menggunakan Regresi Linier biasa (OLS) yang berbasis *Mean Squared Error* (MSE), model tersebut akan panik melihat ribuan *outlier* bernilai 0 ini. OLS akan menarik paksa (membengkokkan) seluruh garis regresinya ke bawah demi memperkecil nilai kuadrat *error* anomali tersebut, yang pada akhirnya akan menghancurkan akurasi prediksi untuk mayoritas lagu normal. Sebaliknya, Huber Regressor menggunakan *Huber Loss Function*. Saat model ini mendeteksi bahwa nilai 0 tersebut adalah anomali yang *error*-nya melampaui batas toleransi (epsilon), model secara otomatis mengubah penaltinya dari kuadratik menjadi linier (mengabaikan tarikan *outlier*). Kesimpulan Akhir: *Stress Test* ini membuktikan bahwa Huber Regressor adalah model yang sangat tangguh (*robust*). Model ini menolak merusak garis prediksi utamanya hanya demi mengejar data anomali. Keputusan ini menjadikan Huber Regressor sebagai solusi *Machine Learning* yang paling rasional, stabil, dan tepat guna untuk memodelkan dataset industri hiburan yang penuh dengan anomali viral maupun kegagalan administratif.

Link Github: <https://github.com/SheilaSabina/KELOMPOK-4-ML-REGRESSION>

Link Youtube: <https://youtu.be/ZcnmsIr0In0>