Homework Solution - Regression

I Gusti Ngurah Agung Hari Vijaya Kusuma Batch 57

August 4, 2025

Submission Links

- Repository: github.com/AgungHari

1 Pendahuluan

1.1 Latar Belakang

Youtube adalah platform berbagi video yang sangat populer di seluruh dunia. Dengan jutaan video yang diunggah setiap hari, penting bagi pembuat konten untuk memahami bagaimana cara meningkatkan visibilitas dan daya tarik video mereka. Salah satu cara untuk mencapai hal ini adalah dengan menganalisis metadata dari video yang telah diunggah sebelumnya.

Dalam tugas ini, kita akan menganalisis metadata dari video-video yang telah diunggah ke Youtube. Metadata ini mencakup berbagai informasi seperti judul, deskripsi, tag, dan jumlah penonton. Dengan menganalisis metadata ini, kita dapat mengidentifikasi pola-pola yang dapat membantu pembuat konten dalam meningkatkan performa video mereka.

1.2 Tujuan

Tujuan dari tugas ini adalah diantara lain untuk:

- Menganalisis metadata dari video Youtube untuk memahami faktor-faktor yang mempengaruhi jumlah penonton.
- Mengembangkan model prediksi yang dapat digunakan untuk memperkirakan jumlah penonton berdasarkan metadata video.
- Menerapkan regresi linier untuk memprediksi jumlah penonton video berdasarkan fitur-fitur yang tersedia dalam metadata.
- Membandingkan performa model regresi linier dengan model lain yang mungkin lebih kompleks.

1.3 Batasan Masalah atau Ruang Lingkup

Batasan masalah dalam tugas ini mencakup:

- Menggunakan dataset yang diberikan oleh rakamin.
- Menerapkan regresi linier sebagai metode utama untuk prediksi, meskipun model lain juga akan dieksplorasi.

1.4 Manfaat

Manfaat dari tugas ini adalah membuat model regresi linier yang dapat digunakan untuk memperkirakan jumlah penonton video Youtube berdasarkan metadata.

2 Tinjauan Pustaka

2.1 EDA (Exploratory Data Analysis)

Exploratory Data Analysis (EDA) adalah proses analisis data yang bertujuan untuk memahami struktur, pola, dan hubungan dalam dataset sebelum menerapkan model statistik atau machine learning. EDA melibatkan visualisasi data, statistik deskriptif, dan identifikasi anomali atau outlier. Proses ini penting untuk mendapatkan wawasan awal tentang data dan membantu dalam pengambilan keputusan selanjutnya.

2.2 Regresi Linier

Regresi linier adalah metode statistik yang digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen (fitur) dengan variabel dependen (target). Model ini mengasumsikan bahwa hubungan antara variabel-variabel tersebut dapat direpresentasikan sebagai garis lurus. Regresi linier sering digunakan dalam analisis data untuk prediksi dan inferensi, serta merupakan dasar bagi banyak algoritma machine learning lainnya.

Agar dapat memahami bagaimana regresi linier bekerja, kita perlu memahami rumus dasar dari regresi linier. Regresi linier sederhana melibatkan satu variabel independen, sedangkan regresi linier berganda melibatkan beberapa variabel independen. Dalam tugas ini, kita akan fokus pada regresi linier berganda untuk memprediksi jumlah penonton video Youtube berdasarkan metadata yang tersedia.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon \tag{1}$$

Di mana:

- y adalah variabel dependen (target).
- β_0 adalah intercept (nilai awal ketika semua variabel independen bernilai nol).
- $\beta_1, \beta_2, ..., \beta_n$ adalah koefisien regresi yang menunjukkan pengaruh masing-masing variabel independen terhadap variabel dependen.
- $x_1, x_2, ..., x_n$ adalah variabel independen (fitur).
- ϵ adalah error term yang mencakup variasi yang tidak dijelaskan oleh model.

Dengan kemajuan teknologi dan ketersediaan pustaka machine learning yang kuat seperti scikit-learn, TensorFlow, dan PyTorch, penerapan regresi linier dalam model machine learning menjadi lebih mudah dan efisien. Pustaka-pustaka ini menyediakan fungsi-fungsi yang memungkinkan pengguna untuk dengan cepat membangun, melatih, dan mengevaluasi model regresi linier tanpa harus mengimplementasikan algoritma dari awal.

2.3 Ridge Regression

Ridge regression adalah teknik regresi linier yang digunakan untuk mengatasi masalah multikolinearitas, yaitu ketika dua atau lebih variabel independen sangat berkorelasi satu sama lain. Ridge regression menambahkan penalti pada ukuran koefisien regresi untuk mengurangi kompleksitas model dan mencegah overfitting. Penalti ini dihitung sebagai kuadrat dari norma L2 dari koefisien regresi, sehingga menghasilkan model yang lebih stabil dan generalisasi yang lebih baik pada data baru.

$$\text{Ridge Loss} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2}$$

Di mana:

- n adalah jumlah data.
- y_i adalah nilai aktual.
- $\hat{y}_i$ adalah nilai yang diprediksi oleh model.
- λ adalah parameter regularisasi yang mengontrol kekuatan penalti.
- p adalah jumlah variabel independen.
- β_j adalah koefisien regresi untuk variabel independen ke-j.
- $\sum_{j=1}^{p} \beta_j^2$ adalah penalti L2 yang ditambahkan ke fungsi loss untuk mengurangi kompleksitas model.

2.4 Lasso Regression

Lasso Regression adalah teknik regresi linier yang juga digunakan untuk mengatasi masalah multikolinearitas, tetapi dengan pendekatan yang berbeda dibandingkan dengan Ridge Regression. Lasso Regression menambahkan penalti pada ukuran koefisien regresi menggunakan norma L1, yang dapat menghasilkan koefisien regresi nol untuk beberapa variabel independen. Hal ini memungkinkan Lasso Regression melakukan seleksi fitur secara otomatis, sehingga menghasilkan model yang lebih sederhana dan interpretatif.

$$\text{Lasso Loss} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{3}$$

Di mana:

- n adalah jumlah data.
- y_i adalah nilai aktual.
- $\hat{y}_i$ adalah nilai yang diprediksi oleh model.
- λ adalah parameter regularisasi yang mengontrol kekuatan penalti.

- p adalah jumlah variabel independen.
- β_j adalah koefisien regresi untuk variabel independen ke-j.
- $\sum_{j=1}^{p} |\beta_j|$ adalah penalti L1 yang ditambahkan ke fungsi loss untuk mengurangi kompleksitas model dan melakukan seleksi fitur.
- $|\beta_j|$ adalah nilai absolut dari koefisien regresi untuk variabel independen ke-j.

2.5 Random Forest Regression

Random Forest Regression adalah metode ensemble learning yang menggabungkan beberapa pohon keputusan untuk meningkatkan akurasi prediksi. Metode ini bekerja dengan membangun sejumlah pohon keputusan pada subset acak dari data pelatihan dan kemudian menggabungkan hasil prediksi dari semua pohon tersebut. Random Forest Regression sangat efektif dalam menangani data dengan banyak fitur dan dapat mengurangi risiko overfitting yang sering terjadi pada pohon keputusan tunggal.

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} f_t(x) \tag{4}$$

Di mana:

- $\hat{y}$ adalah prediksi akhir dari Random Forest.
- T adalah jumlah pohon dalam hutan acak.
- $f_t(x)$ adalah prediksi dari pohon keputusan ke-t untuk input x.
- $\sum_{t=1}^{T} f_t(x)$ adalah jumlah prediksi dari semua pohon keputusan.
- $\frac{1}{T}$ adalah rata-rata dari prediksi semua pohon keputusan, yang memberikan hasil akhir dari Random Forest Regression.
- T adalah jumlah pohon keputusan yang digunakan dalam Random Forest, yang biasanya ditentukan oleh pengguna sebagai hyperparameter.

2.6 Metadata Youtube

Metadata Youtube mencakup berbagai informasi yang terkait dengan video, seperti judul, deskripsi, tag, kategori, dan statistik penonton. Metadata ini sangat penting karena membantu algoritma Youtube dalam merekomendasikan video kepada pengguna dan mempengaruhi visibilitas video di platform. Dengan menganalisis metadata, kita dapat mengidentifikasi faktor-faktor yang berkontribusi terhadap popularitas video dan mengembangkan strategi untuk meningkatkan performa konten.

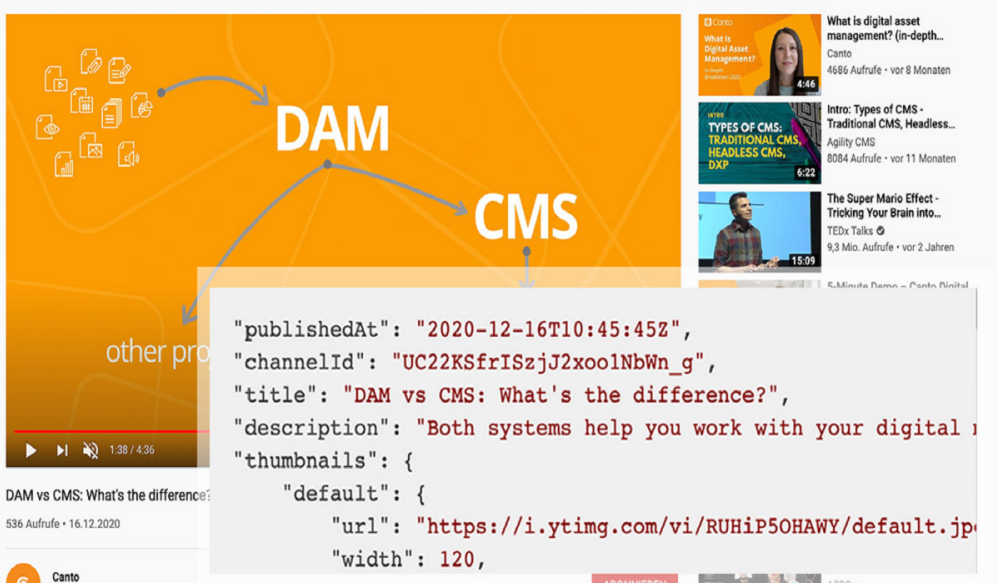

Figure 1: Contoh Metadata Youtube

2.7 RMSE (Root Mean Square Error)

Root Mean Square Error (RMSE) adalah metrik evaluasi yang digunakan untuk mengukur seberapa baik model prediksi dalam memprediksi nilai-nilai numerik. RMSE menghitung akar kuadrat dari rata-rata kuadrat selisih antara nilai yang diprediksi dan nilai aktual. Metrik ini memberikan gambaran tentang seberapa besar kesalahan prediksi model, dengan semakin kecil nilai RMSE menunjukkan performa model yang lebih baik.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

Di mana:

- n adalah jumlah data.
- y_i adalah nilai aktual.
- $\hat{y}_i$ adalah nilai yang diprediksi oleh model.

2.8 R^2 (Koefisien Determinasi)

Koefisien Determinasi (R^2) adalah metrik yang digunakan untuk mengukur seberapa baik model regresi menjelaskan variasi dalam data. Nilai R^2 berkisar antara 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan bahwa model mampu menjelaskan proporsi yang lebih besar dari variasi dalam data. Metrik ini sering digunakan untuk mengevaluasi performa model regresi dan membandingkan model yang berbeda.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{6}$$

Di mana:

- n adalah jumlah data.
- y_i adalah nilai aktual.
- $\hat{y}_i$ adalah nilai yang diprediksi oleh model.
- $\bar{y}$ adalah rata-rata dari nilai aktual.

2.9 Modeling

Modeling dalam konteks machine learning adalah proses membangun model matematis atau statistik yang dapat digunakan untuk membuat prediksi atau mengambil keputusan berdasarkan data. Proses ini melibatkan pemilihan algoritma, pelatihan model dengan data, dan evaluasi performa model menggunakan metrik yang relevan. Dalam tugas ini, kita akan fokus pada penerapan regresi linier sebagai metode modeling utama untuk memprediksi jumlah penonton video Youtube berdasarkan metadata yang tersedia.

2.10 Scikit-learn

Scikit-learn adalah pustaka machine learning yang populer di Python yang menyediakan berbagai algoritma dan alat untuk analisis data dan modeling. Pustaka ini menawarkan antarmuka yang sederhana dan konsisten, sehingga memudahkan pengguna untuk menerapkan berbagai algoritma machine learning, termasuk regresi linier, klasifikasi, clustering, dan lain-lain. Scikit-learn juga menyediakan fungsi-fungsi untuk preprocessing data, evaluasi model, dan validasi silang, menjadikannya pilihan yang ideal untuk proyek-proyek machine learning.

2.11 Matplotlib dan Seaborn

Matplotlib dan Seaborn adalah pustaka visualisasi data yang populer di Python. Matplotlib menyediakan fungsi dasar untuk membuat berbagai jenis grafik, seperti garis, batang, dan sebar, sedangkan Seaborn adalah ekstensi dari Matplotlib yang menawarkan antarmuka yang lebih tinggi dan lebih mudah digunakan untuk visualisasi statistik. Keduanya sangat berguna dalam proses EDA (Exploratory Data Analysis) untuk memahami struktur dan pola dalam data melalui visualisasi yang informatif.

2.12 Pandas

Pandas adalah pustaka Python yang menyediakan struktur data dan fungsi untuk manipulasi dan analisis data. Pustaka ini menawarkan DataFrame, yang merupakan struktur data tabular yang memungkinkan pengguna untuk dengan mudah mengakses, memanipulasi, dan menganalisis data. Pandas sangat berguna dalam proses EDA (Exploratory Data Analysis) karena menyediakan berbagai fungsi untuk membersihkan, mengolah, dan menganalisis data secara efisien.

2.13 Transform log1p

Transformasi log1p adalah teknik yang digunakan untuk mengatasi masalah distribusi data yang tidak normal, terutama ketika data memiliki nilai nol atau sangat kecil. Fungsi log1p menghitung logaritma dari (1 + x), yang memungkinkan transformasi data positif dan nol tanpa menghasilkan nilai tak terdefinisi. Transformasi ini sering digunakan dalam analisis regresi untuk meningkatkan normalitas residual dan mengurangi pengaruh outlier.

$$\text{log1p}(x) = \log(1 + x) \tag{7}$$

Di mana :

- x adalah nilai input.
- log adalah fungsi logaritma natural.

2.14 Transform expm1

expm1 adalah fungsi yang digunakan untuk menghitung eksponensial dari suatu nilai dikurangi satu, yaitu $e^x - 1$. Fungsi ini berguna dalam konteks transformasi data, terutama ketika bekerja dengan data yang telah ditransformasi menggunakan logaritma. Fungsi expm1 membantu mengembalikan nilai asli dari transformasi logaritma dengan cara yang lebih stabil secara numerik, terutama untuk nilai-nilai kecil.

$$\text{expm1}(x) = e^x - 1 \tag{8}$$

Di mana :

- x adalah nilai input.
- e adalah bilangan Euler (sekitar 2.71828).

2.15 Hyperparameter Tuning

Hyperparameter tuning adalah proses mencari nilai optimal untuk hyperparameter model machine learning yang tidak dipelajari selama pelatihan. Hyperparameter adalah parameter yang ditentukan sebelum pelatihan dimulai, seperti laju pembelajaran, jumlah pohon dalam hutan acak, atau kedalaman pohon keputusan. Proses ini penting karena pemilihan hyperparameter yang tepat dapat secara signifikan mempengaruhi performa model. Teknik umum untuk hyperparameter tuning termasuk grid search, random search, dan Bayesian optimization.

2.16 StandardScaler

StandardScaler adalah kelas dalam pustaka scikit-learn yang digunakan untuk melakukan normalisasi data dengan mengubah distribusi fitur menjadi distribusi normal standar (mean = 0, standar deviasi = 1). Proses ini penting dalam machine learning karena membantu algoritma belajar lebih efektif dengan memastikan bahwa semua fitur memiliki skala yang

sama. StandardScaler menghitung mean dan standar deviasi dari setiap fitur pada data pelatihan dan menerapkannya pada data pelatihan dan pengujian.

$$z = \frac{x - \mu}{\sigma} \tag{9}$$

Di mana:

- z adalah nilai yang dinormalisasi.
- x adalah nilai asli.
- μ adalah mean dari fitur.
- σ adalah standar deviasi dari fitur.

3 Desain dan Implementasi

Tugas ini dilakukan sesuai dengan desain sistem berikut beserta implementasinya. Desain sistem adalah konsep dari pembuatan dan perancangan infrastruktur dan kemudian diwujudkan dalam bentuk alur yang harus dikerjakan

3.1 Deskripsi sistem

Pada tugas ini kita akan melakukan analisis data dan membangun model regresi linier untuk memprediksi jumlah penonton video Youtube berdasarkan metadata yang tersedia. Proses ini melibatkan beberapa langkah, mulai dari eksplorasi data hingga evaluasi model. Desain sistem ini mencakup langkah-langkah yang akan dijabarkan dalam Gambar 2.

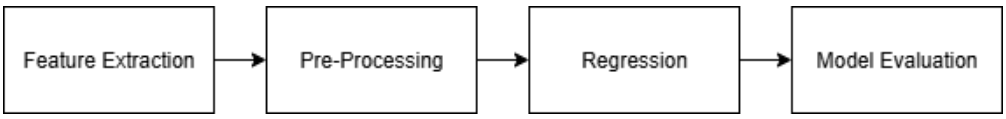

Figure 2: Blok Diagram Sistem

3.2 Feature Extraction

Feature extraction adalah proses penting dalam machine learning yang bertujuan untuk mengidentifikasi dan memilih fitur-fitur yang relevan dari data yang tersedia. Dalam konteks tugas ini, kita akan melakukan feature extraction pada metadata video Youtube untuk mendapatkan fitur-fitur yang akan digunakan dalam model regresi linier.

3.2.1 Deskripsi Dataset

Dataset yang digunakan dalam tugas ini adalah dataset dalam format .xslx yang berisi metadata dari video Youtube. Dataset ini mencakup berbagai informasi seperti judul, deskripsi, tag, kategori, dan statistik penonton. Setiap baris dalam dataset mewakili satu video, dan kolom-kolomnya berisi fitur-fitur yang relevan untuk analisis. Gambar 3 merupakan contoh dari dataset yang digunakan.

ending_da	title	hannel_titl	ategory_i	ublish_tim	tags	views	likes	dislikes	nment_co	ments_dis	ings_disab	error_or_r	description	No_tags	desc_len	len_title
2017-11-1	Sharry Ma	Lokdhun F	1	12:20:39	sharry ma	1096327	33966	798	882	FALSE	FALSE	FALSE	Presentin	15	920	81
2017-11-1	पीरियड्स	HJ NEWS	25	05:43:56	पीरियड्स	590101	735	904	0	TRUE	FALSE	FALSE	पीरियड्स	19	2232	58
2017-11-1	Stylish Sta	TFPC	24	15:48:08	Stylish Sta	473988	2011	243	149	FALSE	FALSE	FALSE	Watch Sty	14	482	58
2017-11-1	Eruma Saa	Eruma Saa	23	07:08:48	Eruma Saa	1242680	70353	1624	2684	FALSE	FALSE	FALSE	This videc	20	263	30
2017-11-1	why Sama	Filmylook	24	01:14:16	Filmylook	464015	492	293	66	FALSE	FALSE	FALSE	why Sama	11	753	88
2017-11-1	MCA (Mid	Dil Raju	24	04:29:50	Nenu Loca	6106669	98612	4185	4763	FALSE	FALSE	FALSE	Watch MC	22	449	91
2017-11-1	Daang (Fu	Speed Rec	10	16:41:15	punjabi sc	5718766	127477	7134	8063	FALSE	FALSE	FALSE	Song - Da	18	1411	96
2017-11-1	Padmavat	T-Series	10	06:14:19	Ek Dil Ek Ji	10588371	132738	8812	10847	FALSE	FALSE	FALSE	Presentin	26	1299	96
2017-11-1	Chiranjee	Top Telug	24	04:42:26	Chiranjee	118223	520	53	23	FALSE	FALSE	FALSE	Chiranjee	21	509	97
2017-11-1	New bike	Jump Cuts	24	04:30:01	Jump cuts	969030	59798	1545	2404	FALSE	FALSE	FALSE	Jump Cuts	23	477	34
2017-11-1	Mehjabi R	TellyMasa	24	17:30:00	Hina khan	632747	4330	2183	2869	FALSE	FALSE	FALSE	Mehjabi S	23	99	78
2017-11-1	Jannat (Fu	White Hill	10	12:02:09	Aatish ne	2348107	32834	710	1743	FALSE	FALSE	FALSE	A WHITE H	22	2093	83
2017-11-1	Renu Desi	ABN Telug	25	09:51:59	Resnu des	156085	716	53	0	TRUE	FALSE	FALSE	Renu Desi	20	858	85
2017-11-1	Peehu Sri	The Voice	24	19:19:13	the voice	472413	2611	250	174	FALSE	FALSE	FALSE	Peehu Sri	12	1351	92
2017-11-1	Rowi Na |	VS RECOR	10	10:29:59	rowi na |"	836006	24460	180	594	FALSE	FALSE	FALSE	VS Record	25	2387	88
2017-11-1	இதுயாரை	Next Gen	22	06:52:47	nextgen |	89531	238	59	18	FALSE	FALSE	FALSE	இதுயாரை	28	171	85
2017-11-1	TYPES OF	Elvish yad	23	11:26:02	TYPES OF	344545	25717	417	2870	FALSE	FALSE	FALSE	This videc	11	248	35
2017-11-1	Tiger Zind	YRF	1	06:01:50	Tiger Zind	35885754	829362	61195	101117	FALSE	FALSE	FALSE	Stay upda	22	901	63
2017-11-1	Meri Setti	TroubleSe	23	13:16:38	Prank Call	209599	14070	448	1105	FALSE	FALSE	FALSE	Subscribe	30	201	68

Figure 3: Contoh Dataset Youtube

Adapun untuk memperjelas fitur-fitur yang digunakan dalam model regresi linier, berikut adalah deskripsi singkat dari beberapa fitur yang terdapat dalam dataset:

1. **trending_date**: Tanggal ketika video menjadi trending.
2. **title**: Judul video.
3. **channel_title**: Nama channel yang mengunggah video.
4. **category_id**: Kategori video dalam label encoding.
5. **publish_time**: Waktu publish video.
6. **tags**: Tag yang digunakan pada video.
7. **views**: Jumlah views video.
8. **likes**: Jumlah likes video.
9. **dislikes**: Jumlah dislikes video.
10. **comment_count**: Jumlah komentar pada video.
11. **comments_disabled**: Status apakah komentar dinonaktifkan pada video.
12. **ratings_disabled**: Status apakah rating dinonaktifkan pada video.
13. **video_error_or_removed**: Status apakah video error atau sudah dihapus saat ini.
14. **description**: Deskripsi video.
15. **No_tags**: Jumlah tag yang digunakan pada video.
16. **desc_len**: Panjang kata deskripsi video.
17. **len_title**: Panjang kata judul video.
18. **publish_date**: Tanggal publish video dalam format datetime.

3.2.2 EDA (Exploratory Data Analysis)

Exploratory Data Analysis (EDA) adalah langkah awal yang penting dalam analisis data. EDA membantu kita memahami struktur data, pola, dan hubungan antar fitur sebelum membangun model. Dalam tugas ini, kita akan melakukan EDA pada dataset Youtube untuk mengidentifikasi fitur-fitur yang relevan dan memahami distribusi data.

Dimulai dari statistik deskriptif, kita dapat melihat ringkasan dari setiap fitur dalam dataset. Adapun statistik deskriptif yang dihasilkan dari dataset ini mencakup informasi seperti jumlah data, nilai rata-rata, standar deviasi, nilai minimum, dan nilai maksimum untuk setiap fitur numerik. Statistik deskriptif ini memberikan gambaran awal tentang distribusi data dan membantu dalam mengidentifikasi fitur-fitur yang mungkin memiliki pengaruh signifikan terhadap jumlah penonton video.

Sebelumnya dataset telah diimport ke dalam DataFrame menggunakan pustaka pandas. Dengan mengelompokkan fitur menjadi dua jenis yaitu fitur numerik dan fitur kategorikal, kita dapat melakukan analisis yang lebih terfokus. Fitur numerik mencakup kolom-kolom seperti views, likes, dislikes, comment_count, No_tags, desc_len, dan len_title. Sedangkan fitur kategorikal mencakup kolom-kolom seperti trending_date, title, channel_title, category_id, publish_time, tags, comments_disabled, ratings_disabled, video_error_or_removed, dan description.

Namun sebelum itu kita perlu melakukan beberapa langkah awal seperti mengimpor library yang diperlukan. Adapun beberapa library tersebut diantaranya adalah pandas, numpy, matplotlib, seaborn, scikit-learn. Library-library ini akan membantu kita dalam melakukan analisis data dan visualisasi.

Listing 1: Statistik Deskriptif Fitur Numerik

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

Setelah itu kita dapat memuat dataset ke dalam DataFrame dan melakukan analisis statistik deskriptif. Berikut adalah contoh kode untuk mendapatkan statistik deskriptif dari fitur numerik:

Listing 2: Statistik Deskriptif Fitur Numerik

```
df = pd.read_excel('dataset_youtube.xlsx')

#bagi dataset menjadi fitur numerik dan kategorikal
numerical_features = df.select_dtypes(include=[np.number])
categorical_features = df.select_dtypes(exclude=[np.number])
```

Adapun beberapa hal yang harus kita cek terlebih dahulu dalam melakukan analisa agar mendapatkan insight yang baik dalam dataset. Adapun beberapa hal tersebut adalah:

1. Cek apakah ada missing value pada dataset.
2. Cek apakah ada duplikasi data pada dataset.
3. Cek apakah ada outlier pada fitur numerik.
4. Cek distribusi dari setiap fitur numerik.
5. Cek hubungan antar fitur numerik menggunakan heatmap.

Missing value didapatkan dengan mengecek apakah ada nilai yang kosong pada dataframe yang telah diimport. Jika ada, kita dapat menghapus baris yang memiliki missing value atau melakukan imputasi dengan nilai rata-rata atau median dari fitur tersebut. Namun sebelum itu kita perlu mengecek berapa jumlah missing value yang ada pada dataset dan jika jumlahnya sedikit kita dapat menghapusnya.

Dengan menggunakan ‘isnull()‘ dan ‘sum()‘, kita dapat melihat jumlah missing value pada setiap kolom. Berikut merupakan outputnya :

Listing 3: Informasi DataFrame

```
title                      0
channel_title              0
category_id                0
publish_time               0
tags                       0
views                      0
likes                      0
dislikes                   0
comment_count              0
comments_disabled          0
ratings_disabled           0
video_error_or_removed     0
description               29
No_tags                    0
desc_len                   0
len_title                  0
publish_date               0
publish_hour               0
publish_period             0
publish_dayofweek          0
is_weekend                 0
dtype: int64
```

Disini kita mendapatkan bahwa terdapat missing value pada kolom description sebanyak 29 baris. Kita dapat menghapus baris tersebut karena jumlahnya relatif kecil dibandingkan dengan total data yang ada. Namun kita tidak akan hapus sekarang kita akan hapus setelah semua proses EDA selesai.

Selanjutnya kita cek apakah ada duplikasi data pada dataset. Dalam konteks dataset ini kita tidak akan mengecek duplikasi data berdasarkan seluruh kolom melainkan kita akan cek menggunakan kolom title. Mengapa demikian? karena title adalah kolom yang paling unik dan dapat digunakan sebagai identifikasi video. Kita akan menggunakan fungsi ‘duplicated()‘ untuk mengecek apakah ada duplikasi pada kolom title. Berikut adalah contoh output cellnya :

Listing 4: Cek Duplikasi Data

```
    Mission: Impossible - Fallout (2018) - Official Trailer
    - Paramount Pictures    19

    Name: count, dtype: int64
```

Setelah membersihkan data duplikat didapatkan 16431 baris data yang unik. Kita dapat melanjutkan ke langkah berikutnya yaitu mengecek apakah ada outlier pada fitur numerik. Outlier adalah nilai yang jauh berbeda dari nilai lainnya dalam dataset. Outlier dapat mempengaruhi hasil analisis dan model yang dibangun, sehingga penting untuk mengidentifikasinya.

Untuk dapat melihat outlier maka kita dapat menggunakan boxplot. Boxplot adalah visualisasi yang menunjukkan distribusi data dan mengidentifikasi outlier. Kita akan membuat boxplot untuk setiap fitur numerik dalam dataset. Gambar 4 merupakan contoh boxplot untuk fitur numerik.

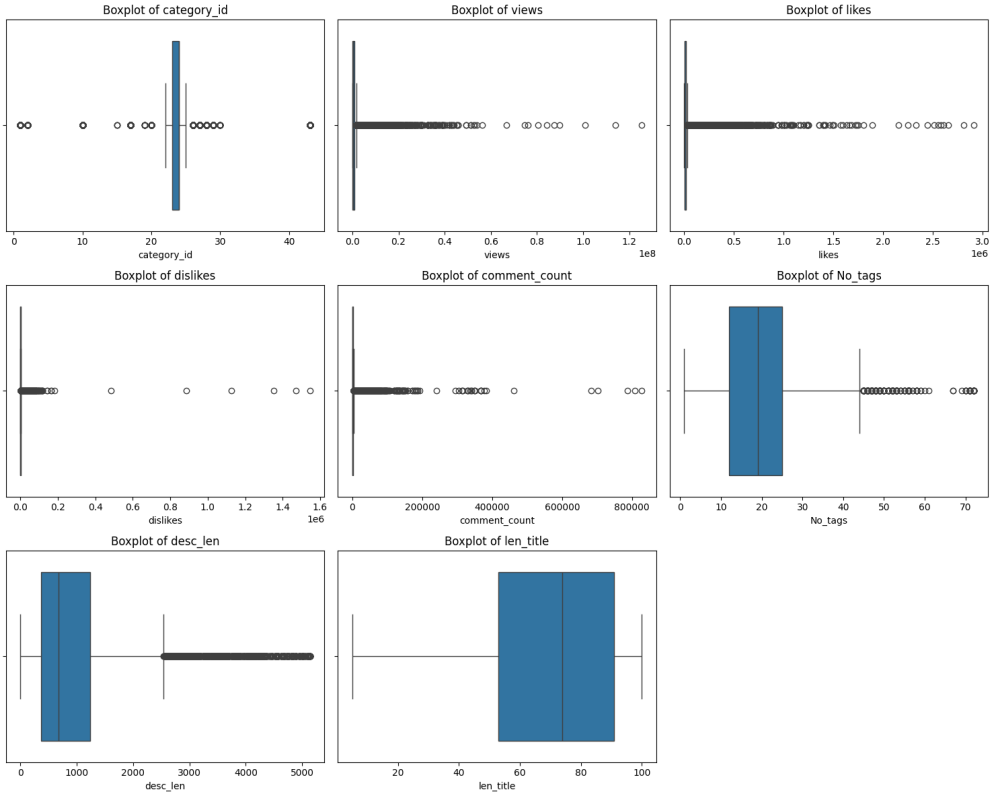

Figure 4: Boxplot Fitur Numerik

Setelah melihat boxplot, kita dapat mengidentifikasi outlier pada fitur numerik dan bagaimana tipe distribusi pada setiap fitur numerik. Kita dapat menghapus outlier tersebut atau melakukan transformasi data untuk mengurangi pengaruhnya terhadap model yang akan dibangun. Berikut merupakan tipe distribusi dari setiap fitur numerik yang ada pada dataset.

- **views**: Distribusi tidak normal, sangat right-skewed.
- **likes**: Distribusi juga right-skewed.
- **dislikes**: Distribusi tidak normal, terdapat outlier yang signifikan.
- **comment_count**: Distribusi tidak normal, terdapat outlier yang signifikan.
- **No_tags**: Ada outlier signifikan di atas nilai 50-70.
- **desc_len**: Ada outlier yang signifikan bahkan lebih banyak dari no_tags.
- **len_title**: Distribusi normal, tidak ada outlier yang signifikan walaupun skewed.

Selanjutnya kita akan mengecek hubungan antar fitur numerik menggunakan heatmap. Heatmap adalah visualisasi yang menunjukkan korelasi antar fitur numerik dalam dataset. Kita akan menggunakan fungsi ‘corr()‘ untuk menghitung korelasi antar fitur numerik dan kemudian membuat heatmap menggunakan seaborn. Gambar 5 merupakan heatmap untuk fitur numerik.

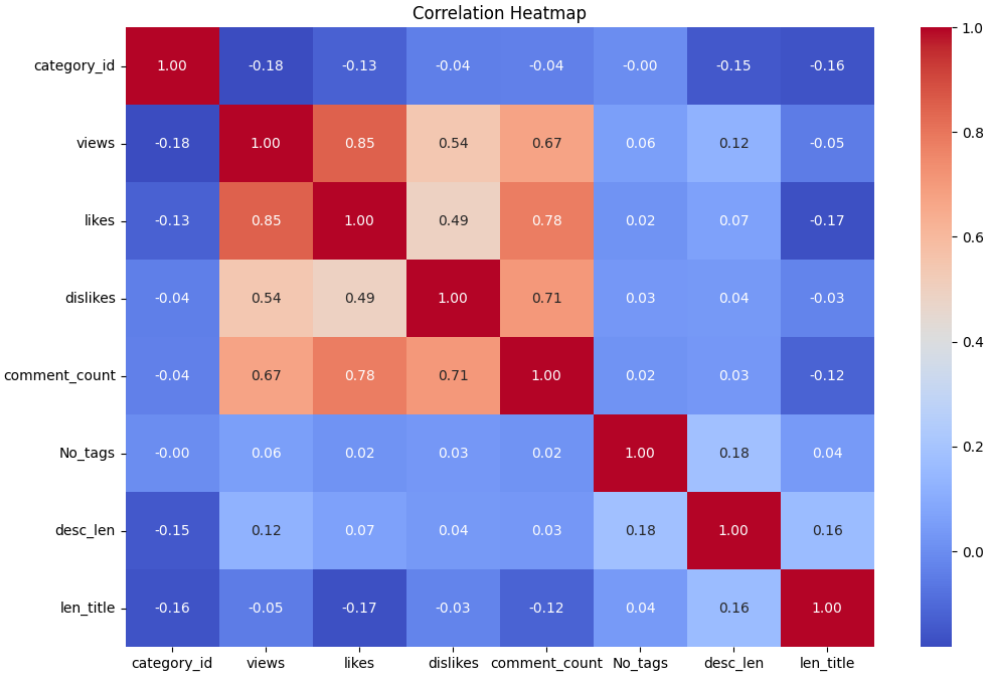

Figure 5: Heatmap Fitur Numerik

Berdasarkan visualisasi heatmap korelasi antar fitur numerik, diperoleh beberapa temuan penting sebagai berikut:

- **views** memiliki korelasi sangat kuat dengan **likes** (**0.85**) dan korelasi sedang dengan **comment_count** (**0.67**) dan **dislikes** (**0.54**). Hal ini menunjukkan bahwa video dengan jumlah penayangan tinggi cenderung juga memiliki banyak likes, komentar, dan dislikes.

- **likes** memiliki korelasi tinggi dengan **comment_count** (**0.78**) dan **dislikes** (**0.49**). Ini mengindikasikan adanya keterkaitan antara jumlah suka dengan keterlibatan pengguna lainnya.

- **dislikes** menunjukkan korelasi sedang dengan **comment_count** (**0.71**), yang bisa diartikan bahwa video dengan banyak komentar cenderung juga memiliki banyak dislike.

- **No_tags**, **desc_len**, dan **len_title** memiliki korelasi yang sangat rendah dengan fitur-fitur lainnya, termasuk **views**. Hal ini menunjukkan bahwa jumlah tag, panjang deskripsi, dan panjang judul tidak memiliki hubungan linear yang kuat terhadap keterlibatan pengguna.

- **category_id** menunjukkan korelasi negatif lemah terhadap seluruh fitur, termasuk **views** (-0.18), **likes** (-0.13), dan **len_title** (-0.16), yang menandakan bahwa kategori video tidak berhubungan kuat secara linier terhadap metrik keterlibatan.

Secara umum, fitur-fitur yang memiliki korelasi tinggi satu sama lain dapat menyebabkan multikolinearitas apabila digunakan dalam model regresi linier. Oleh karena itu, perlu dipertimbangkan teknik reduksi fitur atau regularisasi untuk mengurangi dampaknya. Namun kita tahan dulu untuk melakukan reduksi fitur karena kita masih akan melakukan beberapa analisis lagi.

Selanjutnya kita akan cek nilai ukuran pemusatan dan penyebaran dari setiap fitur numerik. Ukuran pemusatan mencakup nilai rata-rata, median, dan modus, sedangkan ukuran penyebaran mencakup rentang, varians, dan deviasi standar. Ukuran-ukuran ini memberikan informasi penting tentang distribusi data dan membantu dalam memahami karakteristik setiap fitur. Gambar 6 merupakan ukuran pemusatan dan penyebaran dari fitur numerik.

	count	mean	min	25%	50%	75%	max	std
category_id	36791.0	21.552173	1.0	23.0	24.0	24.0	43.0	6.586716
views	36791.0	1071490.258976	4024.0	125604.0	307836.0	806631.5	125432237.0	3207149.048175
likes	36791.0	27450.693675	0.0	879.0	3126.0	14095.0	2912710.0	97831.29478
dislikes	36791.0	1685.363404	0.0	109.0	331.0	1032.0	1545017.0	16197.31684
comment_count	36791.0	2714.022043	0.0	83.0	336.0	1314.5	827755.0	14978.114328
No_tags	36791.0	18.938463	1.0	12.0	19.0	25.0	72.0	9.843531
desc_len	36791.0	923.079123	3.0	368.0	677.0	1237.0	5136.0	815.038867
len_title	36791.0	70.609361	5.0	53.0	74.0	91.0	100.0	22.409174
publish_date	36791	2018-02-24 11:20:36.334973440	2017-05-27 00:00:00	2017-12-31 00:00:00	2018-02-23 00:00:00	2018-04-23 00:00:00	2018-06-13 00:00:00	NaN

Figure 6: Ukuran Pemusatan dan Penyebaran Fitur Numerik

Ukuran pemusatan seperti mean, median, dan mode digunakan untuk memberikan gambaran umum mengenai kecenderungan nilai dalam dataset. Meskipun tidak digunakan secara langsung dalam pemodelan, informasi ini bermanfaat dalam tahapan eksplorasi data (EDA). Berikut adalah analisis ringan terhadap fitur numerik pada dataset:

- **views**: Nilai rata-rata (*mean*) sebesar 1.071.490 jauh lebih besar dibandingkan median (307.836) dan modus (105.397), menunjukkan distribusi yang sangat *right-skewed*. Ini mengindikasikan bahwa hanya sebagian kecil video yang memperoleh jumlah tayangan yang sangat tinggi (viral).

- **likes**: Rata-rata sebesar 27.450 jauh lebih besar dari median (3.126) dan modus bernilai nol. Ini menunjukkan bahwa banyak video dengan sedikit atau tanpa likes, sementara sebagian kecil lainnya memiliki likes yang sangat tinggi.

- **dislikes**: Memiliki rata-rata sebesar 1.685 dan median 331, dengan modus nol. Ini mengindikasikan bahwa sebagian besar video memiliki sedikit atau tidak ada dislike, namun terdapat outlier dengan jumlah dislike yang tinggi.

- **comment_count**: Rata-rata sebesar 2.714 dan median 336, sedangkan modus juga nol. Distribusi yang mirip dengan likes dan dislikes, menunjukkan banyak video yang tidak mendapat komentar.

- **category_id**: Rata-rata sebesar 21,5 dan median 24, dengan modus 24. Karena ini adalah variabel kategori yang direpresentasikan dalam bentuk numerik, ukuran pemusatan tidak terlalu informatif, namun dapat memberi gambaran bahwa kategori dengan ID 24 adalah yang paling dominan dalam dataset.

Secara umum, hampir semua fitur numerik memiliki distribusi yang *skewed to the right*, yang merupakan karakteristik umum dalam data platform digital. Ukuran pemusatan ini memperkuat hasil visualisasi yang telah dilakukan sebelumnya.

Ukuran penyebaran menunjukkan seberapa besar variasi atau sebaran nilai dalam data, dan dalam konteks dataset ini, nilai-nilai seperti views, likes, dan comment count memiliki penyebaran yang sangat tinggi, menandakan adanya ketimpangan antara video yang viral dan yang tidak.

Setelah kita menganalisa semuanya kita dapat kesimpulan dari analisis EDA yang telah dilakukan. Berikut adalah ringkasan dari hasil EDA:

- Dataset terdiri dari 16431 baris data unik setelah menghapus duplikasi berdasarkan kolom title.
- Terdapat missing value pada kolom description sebanyak 29 baris, yang akan dihapus setelah EDA selesai.
- Outlier ditemukan pada fitur numerik seperti views, likes, dislikes, comment count, No_tags, desc_len, dan len_title.
- Distribusi fitur numerik umumnya tidak normal dan skewed ke kanan.
- Korelasi antar fitur numerik menunjukkan hubungan yang signifikan antara views, likes, dan comment count.
- Ukuran pemusatan dan penyebaran memberikan gambaran umum tentang distribusi data.

Berdasarkan Analisa Lanjutan terhadap Fitur

Berdasarkan analisis lanjutan terhadap fitur-fitur yang ada, kita dapat menyimpulkan beberapa poin penting:

- Fitur-fitur seperti `likes`, `dislikes`, dan `comment_count` menunjukkan korelasi yang signifikan terhadap variabel target `views`. Namun, fitur-fitur ini bersifat retrospektif, artinya nilainya hanya tersedia setelah video dipublikasikan dan memperoleh interaksi.
- Dalam konteks pemodelan prediktif untuk memperkirakan jumlah `views` pada saat awal publikasi video, fitur-fitur tersebut tidak dapat digunakan karena nilainya belum tersedia (selalu bernilai nol). Oleh karena itu, meskipun secara statistik fitur-fitur ini tampak penting, secara logis dan fungsional tidak relevan untuk dimasukkan dalam model prediksi awal.

- Sebagai gantinya, fitur-fitur seperti `publish_time`, `category_id`, dan panjang teks (judul, deskripsi) dipertahankan karena memiliki nilai yang tersedia sebelum publikasi dan dapat berkontribusi terhadap estimasi awal performa video.
- Adapun beberapa fitur yang memiliki nilai sangat unik dan tidak relevan seperti `title`, `description`, dan `tags` juga dipertimbangkan untuk dihapus. Meskipun fitur-fitur ini dapat memberikan konteks, mereka tidak memberikan informasi numerik yang dapat digunakan dalam model regresi linier.

Dari kesimpulan tersebut saya memutuskan untuk membuang fitur-fitur yang tidak relevan dan hanya mempertahankan fitur-fitur yang dapat digunakan untuk memprediksi jumlah views pada saat awal publikasi video. Fitur fitur yang akan saya drop adalah:

1. `likes`
2. `dislikes`
3. `comment_count`
4. `trending_date`
5. `title`
6. `description`
7. `tags`

Selain fitur yang akan saya hapus, adapun fitur fitur yang akan saya gunakan dalam membangun model regresi linier, dimana fitur ini saya anggap relevan dan masuk akal untuk digunakan dalam memprediksi jumlah views pada saat awal publikasi video. Fitur-fitur tersebut adalah:

1. `category_id`
2. `channel_title`
3. `publish_time`
4. `publish_date`
5. `No_tags`
6. `desc_len`
7. `len_title`

Namun ada beberapa fitur yang semula bernilai timestamp seperti `publish_time` dan `publish_date` yang akan kita ubah menjadi fitur numerik. Kita akan mengubahnya menjadi fitur numerik dengan cara mengambil informasi dari waktu tersebut seperti hari, jam, dan apakah hari tersebut adalah akhir pekan atau bukan. Hal ini dilakukan agar model regresi linier dapat memahami informasi temporal yang ada pada data.

Adapun saya akan mengubah `publish_time` menjadi fitur numerik dengan cara mengambil informasi dari waktu tersebut seperti jam, hari, dan apakah hari tersebut adalah akhir pekan atau bukan. Saya akan membuat beberapa fitur baru sebagai berikut:

1. `publish_hour`: Jam dari waktu publish video.
2. `publish_dayofweek`: Hari dalam seminggu dari waktu publish video (0-6, dimana 0 adalah Senin).
3. `is_weekend`: Apakah hari tersebut adalah akhir pekan (Sabtu atau Minggu).
4. `publish_period`: Periode waktu publish video (pagi, siang, sore, malam).

Setelah saya ubah maka akan bisa dilakukan encoding baik dari is_weekend, publish_period, dan publish_dayofweek, maupun yang sudah bertipe kategori sebelumnya seperti category_id, rating_disabled, dan comments_disabled. Kita akan melakukan encoding pada fitur-fitur tersebut agar dapat digunakan dalam model regresi linier.

3.3 Data Preprocessing

Data preprocessing adalah langkah penting dalam machine learning yang bertujuan untuk mempersiapkan data sebelum digunakan dalam model. Langkah-langkah preprocessing yang akan dilakukan dalam tugas ini akan dijabarkan prosesnya melalui flowchart pada gambar 7.

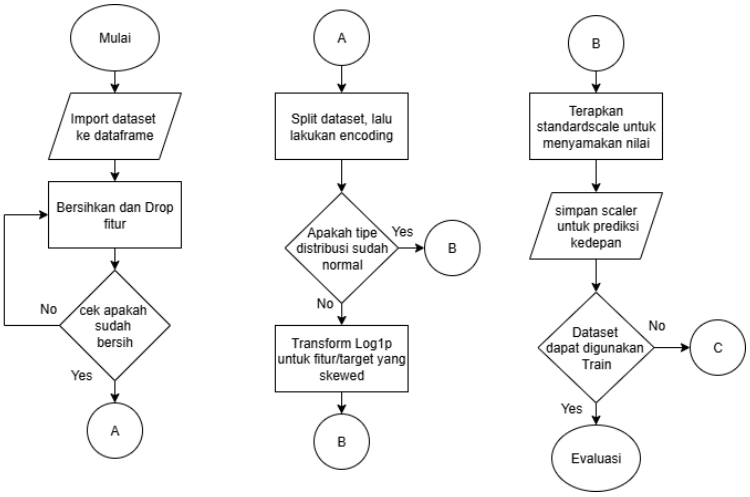

Figure 7: flowchart preprocessing

Adapun hal pertama yang kita lakukan adalah membersihkan fitur yang tidak relevan seperti yang telah dijelaskan sebelumnya. Kita akan menghapus fitur-fitur yang tidak

relevan dan hanya mempertahankan fitur-fitur yang dapat digunakan untuk memprediksi jumlah views pada saat awal publikasi video. Setelah itu kita akan melakukan encoding pada fitur-fitur kategorikal seperti `category_id`, `publish_dayofweek`, `is_weekend`, dan `publish_period`.

Kita akan menggunakan teknik one-hot encoding untuk fitur-fitur kategorikal tersebut. One-hot encoding adalah teknik yang mengubah fitur kategorikal menjadi beberapa kolom biner, dimana setiap kolom mewakili satu kategori. Hal ini dilakukan agar model regresi linier dapat memahami informasi dari fitur-fitur kategorikal tersebut.

Adapun setelah melakukan one hot encoding saya mendapatkan penambahan jumlah fitur sebesar 16 kolom baru yang merupakan hasil dari one hot encoding dari fitur category_id. Berikut adalah output cellnya:

```
       ...  category_id_22  category_id_23  category_id_24  category_id_25  \
9065   ...           False           False           False           False
13109  ...           False           False            True           False
13700  ...           False           False            True           False
21935  ...           False           False            True           False
5083   ...           False           False           False           False

       category_id_26  category_id_27  category_id_28  category_id_29  \
9065            False           False           False           False
13109           False           False           False           False
13700           False           False           False           False
...
21935           False           False
5083            False           False

[5 rows x 25 columns]
```

Figure 8: One Hot Encoding Fitur Kategorikal

Selain untuk fitur category_id saya juga melakukan encoding untuk fitur rating_disabled, is_weekend, dan publish_period. Dengan melakukan encoding pada fitur-fitur tersebut, didapatkan total jumlah fitur sebanyak 24 kolom.

Setelah itu kita akan melakukan transformasi menggunakan log1p pada fitur-fitur numerik yang akan digunakan. transformasi log1p adalah transformasi yang digunakan untuk mengurangi skewness pada data. Transformasi ini akan mengubah distribusi data menjadi lebih normal dan mengurangi pengaruh outlier terhadap model.

Adapun rumusan dari transformasi log1p adalah sebagai berikut:

$$\text{log1p}(x) = \log(1 + x) \tag{10}$$

Adapun contoh perhitungannya untuk misal 100 views maka akan merubah nilai views tersebut menjadi:

$$\text{log1p}(100) = \log(1 + 100) = \log(101) \approx 4.615 \tag{11}$$

Dengan melakukan transformasi log1p di tiap baris pada fitur-fitur numerik, kita dapat mengurangi skewness pada data dan membuat distribusi data menjadi lebih normal. Agar lebih jelas, berikut merupakan histogram sebelum dan sesudah transformasi log1p pada label target views :

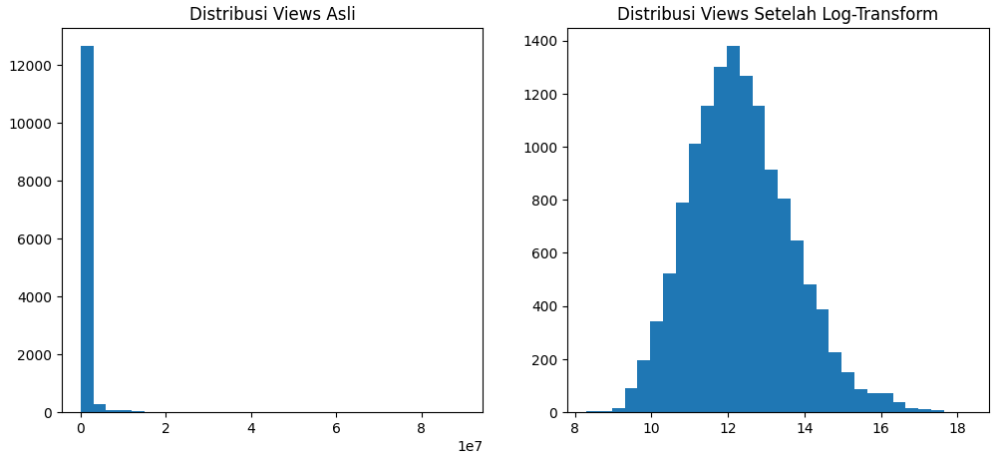

Figure 9: Histogram Views Sebelum dan Sesudah Transformasi Log1p

Namun perlu diingat saat kita melakukan transformasi log1p pada fitur-fitur numerik, kita harus tahu bahwa saat melakukan evaluasi model maka kita harus mengembalikan nilai prediksi ke skala aslinya. Hal ini dilakukan dengan cara melakukan transformasi eksponensial pada nilai prediksi yang dihasilkan oleh model. Transformasi eksponensial adalah kebalikan dari transformasi log1p, sehingga kita dapat mengembalikan nilai prediksi ke skala aslinya.

Adapun rumusan dari transformasi eksponensial adalah sebagai berikut:

$$\text{expm1}(x) = e^x - 1 \tag{12}$$

Adapun contoh perhitungannya untuk misal 4.615 maka akan merubah nilai views tersebut menjadi:

$$\text{expm1}(4.615) = e^{4.615} - 1 \approx 100 \tag{13}$$

Dengan melakukan transformasi eksponensial pada nilai prediksi, kita dapat mengembalikan nilai prediksi ke skala aslinya. Hal ini penting agar kita dapat membandingkan nilai prediksi dengan nilai sebenarnya dari label target views.

Setelah melakukan transformasi biasanya kita akan membuang outlier yang ada pada fitur numerik. Baik menggunakan metode IQR (Interquartile Range) atau Z-score. Namun pada tugas ini kita tidak akan membuang outlier karena kita ingin mempertahankan semua data yang ada. Hal ini dilakukan agar model regresi linier dapat memahami pola-pola yang ada pada data, termasuk pola-pola yang terdapat pada outlier. Mengapa demikian? adapun alasan alasannya sebagai berikut :

- **Viralitas Video**: Di platform seperti YouTube, video sering kali memiliki potensi untuk menjadi viral. Outlier dalam jumlah views bisa jadi merupakan video yang sangat populer dan memiliki dampak besar. Dengan mempertahankan outlier, model dapat belajar dari pola-pola yang ada pada video-video viral ini.

- **Variasi Data**: Outlier sering kali mencerminkan variasi alami dalam data. Menghapusnya dapat menghilangkan informasi penting yang mungkin relevan untuk memahami performa video secara keseluruhan.
- **Model Robustness**: Model regresi linier dapat dibuat lebih robust terhadap outlier dengan menggunakan teknik regularisasi seperti Ridge atau Lasso. Ini memungkinkan model untuk tetap belajar dari data meskipun ada beberapa nilai ekstrem, dan model model tersebut juga akan diuji pada tugas ini.

Setelah melakukan preprocessing data, kita akan membagi dataset menjadi data latih dan data uji. Pembagian ini penting untuk memastikan bahwa model yang dibangun dapat generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Kita akan menggunakan fungsi 'train_test_split' dari pustaka scikit learn untuk membagi dataset menjadi data latih dan data uji dengan proporsi 80 20.

Listing 5: Pembagian Data Latih dan Data Uji

```
1
2 from sklearn.model_selection import train_test_split
3
4 # Membagi dataset menjadi data latih dan data uji
5 X = df.drop(columns=['views'])
6 y = df['views']
7
8 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
9 =0.2, random_state=42)
```

Dengan pembagian ini, kita akan mendapatkan data latih sebanyak 80% dari total data dan data uji sebanyak 20% dari total data. Data latih akan digunakan untuk membangun model regresi linier, sedangkan data uji akan digunakan untuk menguji performa model yang telah dibangun. Adapun shape dari data latih adalah sebagai berikut:

Listing 6: Shape Data Latih dan uji

```
1
2     Shape X_train after scaling: (13144, 26)
3     Shape y_train after scaling: (13144,)
4     Shape X_test after scaling: (3287, 26)
5     Shape y_test after scaling: (3287,)
```

Setelah membagi dataset menjadi data latih dan data uji, kita akan melakukan scaling pada fitur-fitur numerik. Scaling adalah langkah penting dalam preprocessing data yang bertujuan untuk mengubah skala fitur-fitur numerik agar memiliki rentang yang sama. Hal ini penting karena model regresi linier sensitif terhadap skala fitur, sehingga scaling dapat membantu meningkatkan performa model.

StandardScaler adalah salah satu teknik scaling yang umum digunakan. Teknik ini akan mengubah fitur-fitur numerik sehingga memiliki rata-rata 0 dan deviasi standar 1. Dengan menggunakan StandardScaler, kita dapat memastikan bahwa semua fitur numerik berada dalam skala yang sama, sehingga model regresi linier dapat belajar dengan lebih baik.

Adapun perhitungannya dari StandardScaler adalah sebagai berikut:

$$z = \frac{x - \mu}{\sigma} \tag{14}$$

Dimana:

- z adalah nilai yang telah diskalakan
- x adalah nilai asli dari fitur
- μ adalah rata-rata dari fitur
- σ adalah deviasi standar dari fitur

Namun penting untuk menggunakan library joblib untuk menyimpan scaler yang telah dibuat. Hal ini dilakukan agar kita dapat menggunakan scaler yang sama pada data uji saat melakukan prediksi. Dengan menyimpan scaler, kita dapat memastikan bahwa data uji akan diskalakan dengan cara yang sama seperti data latih.

Dengan menerapkan scaler ini maka kita sudah dapat melakukan preprocessing data secara lengkap. Namun kita masih harus drop fitur original dari terutama hasil transform log dan encoding yang tadi sudah kita lakukan. Adapun fitur-fitur yang akan kita drop dapat dilihat pada lstlisting sebagai berikut:

Listing 7: Drop Fitur Original

```
1
2 X_train = X_train.drop(['No_tags', 'desc_len'], axis=1)
3 X_test = X_test.drop(['No_tags', 'desc_len'], axis=1)
4 X_train = X_train.drop(['publish_time', 'publish_date'], axis=1)
5 X_test = X_test.drop(['publish_time', 'publish_date'], axis=1)
6 X_train = X_train.drop(['publish_hour', 'publish_dayofweek', 'is
7 _weekend', 'publish_period'], axis=1)
```

Setelah kita drop maka X_train dan X_test akan memiliki fitur-fitur yang sudah diproses dan siap digunakan untuk membangun model regresi linier. Gambar 10 merupakan gambar input data yang telah diproses.

```
<class 'pandas.core.frame.DataFrame'>
Index: 13144 entries, 9065 to 3890
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   comments_disabled     13144 non-null  int64
 1   ratings_disabled      13144 non-null  int64
 2   len_title             13144 non-null  float64
 3   is_weekend            13144 non-null  int64
 4   No_tags_log           13144 non-null  float64
 5   desc_len_log          13144 non-null  float64
 6   category_id_2         13144 non-null  int64
 7   category_id_10        13144 non-null  int64
 8   category_id_15        13144 non-null  int64
 9   category_id_17        13144 non-null  int64
 10  category_id_19        13144 non-null  int64
 11  category_id_20        13144 non-null  int64
 12  category_id_22        13144 non-null  int64
 13  category_id_23        13144 non-null  int64
 14  category_id_24        13144 non-null  int64
 15  category_id_25        13144 non-null  int64
 16  category_id_26        13144 non-null  int64
 17  category_id_27        13144 non-null  int64
 18  category_id_28        13144 non-null  int64
 19  category_id_29        13144 non-null  int64
 20  category_id_30        13144 non-null  int64
 21  category_id_43        13144 non-null  int64
 22  publish_period_Pagi   13144 non-null  int64
 23  publish_period_Siang  13144 non-null  int64
dtypes: float64(3), int64(21)
```

Figure 10: Input Data Setelah Preprocessing

Setelah melakukan preprocessing data, kita akan mendapatkan data latih dan data uji yang siap digunakan untuk membangun model regresi linier. Data latih akan digunakan untuk melatih model, sedangkan data uji akan digunakan untuk menguji performa model yang telah dibangun. Dengan demikian, kita telah menyelesaikan tahap preprocessing data dalam tugas ini.

4 Pengujian dan analisis

Pada bab ini, akan dijelaskan mengenai hasil pengujian dan pembahasan dari penelitian yang telah diuraikan pada metodologi. Selain itu, akan dipaparkan juga mengenai skenario pengujian yang dilakukan untuk mengevaluasi performa sistem secara keseluruhan. Pengujian ini dilakukan dengan tujuan untuk memastikan bahwa sistem yang dirancang mampu berfungsi dengan baik dalam memprediksi jumlah penonton video Youtube berdasarkan metadata yang tersedia.

4.1 Pengujian Sistem

Pengujian sistem dilakukan dengan menggunakan dataset yang telah diolah sebelumnya dan dibagi menjadi data latih dan data uji. Data latih digunakan untuk melatih model regresi linier, sedangkan data uji digunakan untuk menguji performa model yang telah dilatih. Adapun skenario pengujian yang dilakukan adalah sebagai berikut:

1. **Pengujian Model Regresi Linier:** Model regresi linier dibangun menggunakan data latih, dan kemudian diuji menggunakan data uji. Hasil prediksi dibandingkan dengan nilai aktual untuk menghitung metrik evaluasi seperti Mean Absolute Error (MAE), Mean Squared Error (MSE), dan R-squared.

2. **Visualisasi Hasil:** Hasil prediksi dibandingkan dengan nilai aktual divisualisasikan dalam bentuk grafik untuk memberikan gambaran yang jelas tentang performa model.

3. **Perbandingan dengan Model Lain:** Jika ada, model lain yang lebih kompleks seperti Random Forest atau Gradient Boosting juga diuji untuk membandingkan performa dengan model regresi linier.

4.2 Pengujian Model Regresi Linier

Pengujian model regresi linier dilakukan dengan menggunakan data uji yang telah disiapkan. Model ini dilatih menggunakan data latih dan kemudian diuji untuk melihat seberapa baik model tersebut dalam memprediksi jumlah penonton video Youtube berdasarkan metadata yang tersedia.

Agar memudahkan memahami pengujian ini berikut dilampirkan rumus regresi yang sesuai dengan scikit-learn

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon \tag{15}$$

Di mana:

- y adalah variabel dependen (target).
- β_0 adalah intercept (nilai awal ketika semua variabel independen bernilai nol).
- $\beta_1, \beta_2, ..., \beta_n$ adalah koefisien regresi yang menunjukkan pengaruh masing-masing variabel independen terhadap variabel dependen.

- $x_1, x_2, ..., x_n$ adalah variabel independen (fitur).
- ϵ adalah error term yang mencakup variasi yang tidak dijelaskan oleh model.

Model regresi linier ini digunakan untuk memprediksi jumlah penonton video berdasarkan fitur-fitur yang tersedia dalam metadata, seperti judul, deskripsi, tag, dan lainnya. Setelah model dilatih, dilakukan evaluasi menggunakan data uji untuk mengukur seberapa baik model tersebut dalam memprediksi jumlah penonton.

Setelah dilakukan pengujian, berikut adalah hasil evaluasi model regresi linier dengan menggunakan metrik RMSE (Root Mean Squared Error) dan R-squared namun masih berskala log1p:

- **RMSE (Root Mean Squared Error):** 1.1960863333284286
- **R-squared:** 0.19453093508280828

Hasil evaluasi ini menunjukkan bahwa model regresi linier memiliki performa yang cukup baik dalam memprediksi jumlah penonton video Youtube berdasarkan metadata yang tersedia. Nilai R-squared yang mendekati 0.2 menunjukkan bahwa model ini mampu menjelaskan sekitar 20% variasi dalam jumlah penonton, meskipun masih ada ruang untuk perbaikan.

Namun hasil berbeda ditunjukan setelah melakukan transformasi kembali ke skala asli dengan menggunakan fungsi 'np.expm1' pada hasil prediksi. Berikut adalah hasil evaluasi model regresi linier setelah transformasi kembali ke skala asli:

- **RMSE (Root Mean Squared Error) (original):** 3091719.7102543344
- **R-squared (original):** -0.0034031502640745614

Hasil evaluasi ini menunjukkan bahwa model regresi linier memiliki performa yang kurang baik dalam memprediksi jumlah penonton video Youtube setelah transformasi kembali ke skala asli. Nilai R-squared yang negatif menunjukkan bahwa model ini tidak mampu menjelaskan variasi dalam jumlah penonton, bahkan lebih buruk daripada model yang hanya menggunakan rata-rata.

4.2.1 Visualisasi Hasil

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet,

tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.3 Perbandingan dengan Model Lain

Adapun beberapa model lain yang akan diuji untuk membandingkan performa dengan model regresi linier, antara lain:

- **Ridge Regressor:** Model ini menggunakan regularisasi L2 untuk mengurangi overfitting.
- **Lasso Regressor:** Model ini menggunakan regularisasi L1 untuk mengurangi overfitting.
- **Random Forest Regressor:** Model ini menggunakan algoritma Random Forest untuk melakukan regresi.
- **Gradient Boosting Regressor:** Model ini menggunakan algoritma Gradient Boosting untuk melakukan regresi.

4.3.1 Pengujian Model Ridge Regressor

Pengujian model Ridge Regressor dilakukan dengan menggunakan data latih yang telah disiapkan. Model ini dilatih menggunakan data latih dan kemudian diuji untuk melihat seberapa baik model tersebut dalam memprediksi jumlah penonton video Youtube berdasarkan metadata yang tersedia.

Adapun rumus regresi Ridge yang sesuai dengan scikit-learn adalah sebagai berikut:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \lambda \sum_{i=1}^{n} \beta_i^2 + \epsilon \tag{16}$$

Di mana:

- y adalah variabel dependen (target).
- β_0 adalah intercept (nilai awal ketika semua variabel independen bernilai nol).
- $\beta_1, \beta_2, ..., \beta_n$ adalah koefisien regresi yang menunjukkan pengaruh masing-masing variabel independen terhadap variabel dependen.
- $x_1, x_2, ..., x_n$ adalah variabel independen (fitur).
- λ adalah parameter regularisasi yang mengontrol kekuatan regularisasi.
- ϵ adalah error term yang mencakup variasi yang tidak dijelaskan oleh model.

- $\sum_{i=1}^{n} \beta_i^2$ adalah penalti L2 yang ditambahkan untuk mengurangi overfitting.

Model ridge dan model regresi linier memiliki kesamaan dalam hal struktur dasar, namun model ridge menambahkan penalti L2 untuk mengurangi overfitting. Setelah model dilatih, dilakukan evaluasi menggunakan data uji untuk mengukur seberapa baik model tersebut dalam memprediksi jumlah penonton.

Setelah dilakukan pengujian, berikut adalah hasil evaluasi model Ridge Regressor dengan menggunakan metrik RMSE (Root Mean Squared Error) dan R-squared namun masih berskala log1p:

- **RMSE (Root Mean Squared Error):** 1.195796
- **R-squared:** 0.194923

Hasil evaluasi ini menunjukkan bahwa model Ridge Regressor memiliki performa yang sedikit lebih baik dibandingkan dengan model regresi linier, dengan nilai R-squared yang sedikit lebih tinggi. Namun, masih ada ruang untuk perbaikan.

Setelah melakukan transformasi kembali ke skala asli dengan menggunakan fungsi 'np.expm1' pada hasil prediksi, berikut adalah hasil evaluasi model Ridge Regressor setelah transformasi kembali ke skala asli:

- **RMSE (Root Mean Squared Error) (original):** 3.091978
- **R-squared (original):** -0.003571

4.4 Visualisasi Hasil Ridge Regressor

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

4.4.1 Pengujian Model Lasso Regressor

Pengujian model Lasso Regressor dilakukan dengan menggunakan data latih yang telah disiapkan. Model ini dilatih menggunakan data latih dan kemudian diuji untuk melihat seberapa

baik model tersebut dalam memprediksi jumlah penonton video Youtube berdasarkan metadata yang tersedia.

Adapun rumus regresi Lasso yang sesuai dengan scikit-learn adalah sebagai berikut:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \lambda \sum_{i=1}^{n} |\beta_i| + \epsilon \tag{17}$$

Di mana:

- y adalah variabel dependen (target).
- β_0 adalah intercept (nilai awal ketika semua variabel independen bernilai nol).
- $\beta_1, \beta_2, ..., \beta_n$ adalah koefisien regresi yang menunjukkan pengaruh masing-masing variabel independen terhadap variabel dependen.
- $x_1, x_2, ..., x_n$ adalah variabel independen (fitur).
- λ adalah parameter regularisasi yang mengontrol kekuatan regularisasi.
- ϵ adalah error term yang mencakup variasi yang tidak dijelaskan oleh model.
- $\sum_{i=1}^{n} |\beta_i|$ adalah penalti L1 yang ditambahkan untuk mengurangi overfitting dan melakukan feature selection.

Model Lasso Regressor dan model regresi linier memiliki kesamaan dalam hal struktur dasar, namun model Lasso Regressor menambahkan penalti L1 untuk mengurangi overfitting dan melakukan feature selection. Setelah model dilatih, dilakukan evaluasi menggunakan data uji untuk mengukur seberapa baik model tersebut dalam memprediksi jumlah penonton.

Setelah dilakukan pengujian, berikut adalah hasil evaluasi model Lasso Regressor dengan menggunakan metrik RMSE (Root Mean Squared Error) dan R-squared namun masih berskala log1p:

- **RMSE (Root Mean Squared Error):** 1.311019
- **R-squared:** 0.032298

Hasil evaluasi ini menunjukkan bahwa model Lasso Regressor memiliki performa yang sedikit lebih baik dibandingkan dengan model regresi linier, dengan nilai R-squared yang sedikit lebih tinggi. Namun, masih ada ruang untuk perbaikan.

Setelah melakukan transformasi kembali ke skala asli dengan menggunakan fungsi ‘np.expm1‘ pada hasil prediksi, berikut adalah hasil evaluasi model Lasso Regressor setelah transformasi kembali ke skala asli:

- **RMSE (Root Mean Squared Error) (original):** 3.126240e+06
- **R-squared (original):** -0.025935

4.5 Visualisasi Hasil Lasso Regressor

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

4.5.1 Pengujian Model Random Forest Regressor

Pengujian model Random Forest Regressor dilakukan dengan menggunakan data latih yang telah disiapkan. Model ini dilatih menggunakan data latih dan kemudian diuji untuk melihat seberapa baik model tersebut dalam memprediksi jumlah penonton video Youtube berdasarkan metadata yang tersedia. Adapun rumus regresi Random Forest yang sesuai dengan scikit-learn adalah sebagai berikut:

$$y = \frac{1}{N} \sum_{i=1}^{N} f_i(x) \tag{18}$$

Di mana:

- y adalah variabel dependen (target).
- N adalah jumlah pohon dalam hutan acak (random forest).
- $f_i(x)$ adalah prediksi dari pohon ke-i untuk input x.
- $\sum_{i=1}^{N}$ adalah penjumlahan dari prediksi semua pohon dalam hutan acak.
- $\frac{1}{N}$ adalah rata-rata dari prediksi semua pohon dalam hutan acak.
- x adalah variabel independen (fitur).
- ϵ adalah error term yang mencakup variasi yang tidak dijelaskan oleh model.

Model Random Forest Regressor adalah model ensemble yang menggabungkan prediksi dari beberapa pohon keputusan (decision trees) untuk meningkatkan akurasi dan mengurangi overfitting. Setelah model dilatih, dilakukan evaluasi menggunakan data uji untuk mengukur seberapa baik model tersebut dalam memprediksi jumlah penonton.

Setelah dilakukan pengujian, berikut adalah hasil evaluasi model Random Forest Regressor dengan menggunakan metrik RMSE (Root Mean Squared Error) dan R-squared namun masih berskala log1p:

- **RMSE (Root Mean Squared Error):** 1.195796
- **R-squared:** 0.194923

Hasil evaluasi ini menunjukkan bahwa model Random Forest Regressor memiliki performa yang sedikit lebih baik dibandingkan dengan model regresi linier, dengan nilai R-squared yang sedikit lebih tinggi. Namun, masih ada ruang untuk perbaikan.

Setelah melakukan transformasi kembali ke skala asli dengan menggunakan fungsi 'np.expm1' pada hasil prediksi, berikut adalah hasil evaluasi model Random Forest Regressor setelah transformasi kembali ke skala asli:

- **RMSE (Root Mean Squared Error) (original):** 3054773.8808745374
- **R-squared (original):** 0.020434754712886805

Menurut hasil evaluasi ini, model Random Forest Regressor memiliki performa yang terbaik diantara model-model yang telah diuji, dengan nilai R-squared yang paling tinggi. Namun, masih ada ruang untuk perbaikan, terutama dalam hal interpretabilitas model.

4.6 Visualisasi Hasil Random Forest Regressor

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

5 Kesimpulan dan Saran

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, dapat disimpulkan bahwa :

- Model regresi linier adalah model yang paling sederhana dan mudah diinterpretasikan, namun memiliki performa yang cukup baik dalam memprediksi jumlah penonton video Youtube berdasarkan metadata.
- Model Ridge Regressor dan Lasso Regressor juga menunjukkan performa yang baik, namun dengan sedikit peningkatan dibandingkan model regresi linier.
- Model Random Forest Regressor menunjukkan performa terbaik diantara model-model yang telah diuji, namun dengan kompleksitas yang lebih tinggi.
- Dataset ini kurang relevan untuk dilakukan regresi karena tidak ada fitur yang signifikan mempengaruhi jumlah penonton video Youtube. Sehingga model yang dihasilkan tidak dapat diandalkan untuk memprediksi jumlah penonton video Youtube secara akurat.
- Oleh karena itu, perlu dilakukan penelitian lebih lanjut dengan dataset yang lebih relevan dan fitur yang lebih signifikan untuk meningkatkan performa model prediksi, seperti memberikan fitur seperti subscriber yang dimiliki oleh channel.