Regression - YouTube Views Prediction

Team:
- Akhmad Yuzfa Salvian Idris
- Arief Rahman Hakim
- Bernardus Valentino
- Milenia Nadia Afifah Puspitasari
- Sean Frederic Wijaya

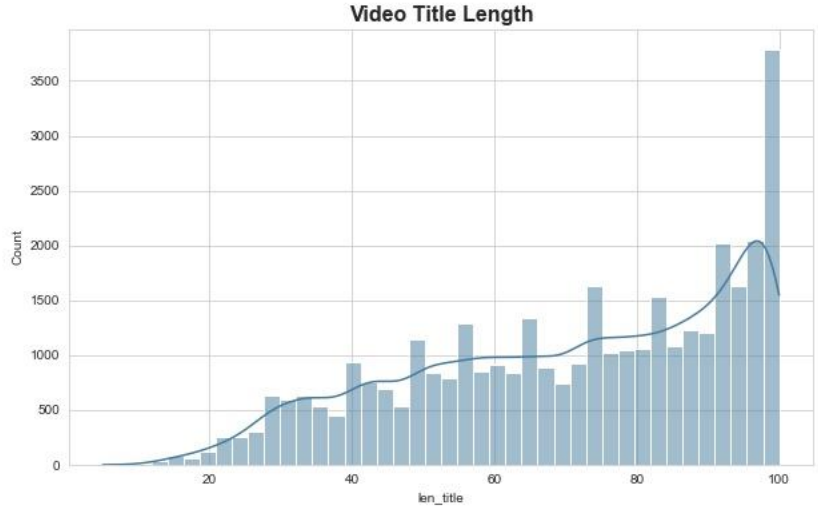
Data: youtube_statistics.xlsx

Columns:
- trending_date: tanggal ketika video trending
- title: judul video
- channel_title: nama channel
- category_id: kategori video dalam label encoding
- publish_time: waktu publish video
- tags: tag yang digunakan pada video
- views: jumlah views video
- likes: jumlah likes video
- dislikes: jumlah dislikes video
- comment_count: jumlah komentar pada video
- comments_disabled: apakah status komentar dinonaktifkan pada video
- ratings_disabled: apakah rating dinonaktifkan pada video
- video_error_or_removed: apakah video error atau sudah dihapus saat ini
- description: deskripsi video
- No_tags: jumlah tags yang digunakan
- desc_len: panjang kata deskripsi video
- len_title: panjang kata judul video
- publish_date: tanggal publish video

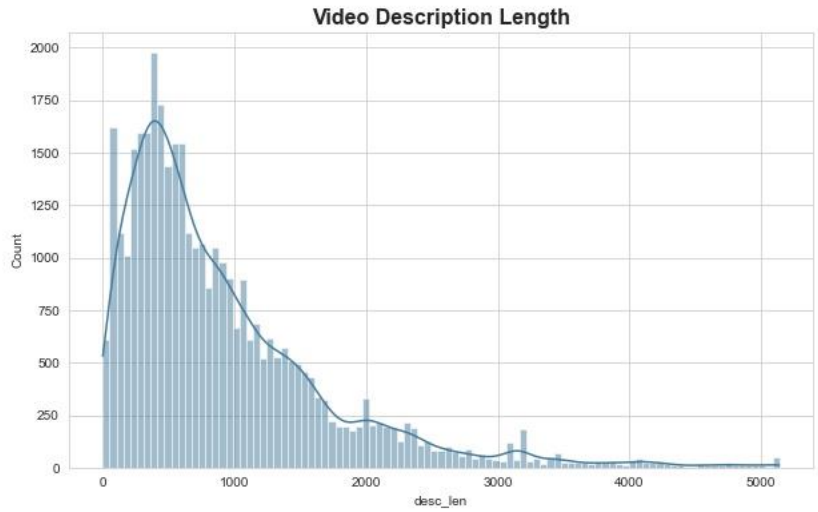
Target feature: views

Exploratory Data Analysis

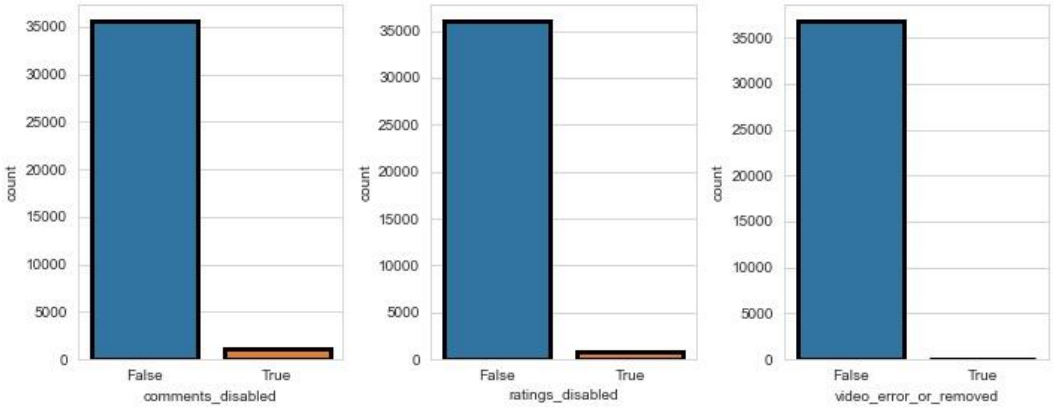
- Dataframe memiliki total 36791 baris dan 18 kolom
- Terdapat missing values di kolom description
- Semua feature memiliki outlier kecuali feature len_title
- Jumlah unik dari setiap feature hanya 2 karena setiap fiturnya memiliki tipe data boolean
- Keseluruhan feature didominasi oleh satu value, yakni FALSE
- Terdapat top 3 feature yang berkorelasi kuat dengan target yakni likes, dislikes, dan comment_count
- Insights:



- Rata-rata panjang judul video: 70.61 karakter
- Content creator cenderung memaksimalkan limit untuk membuat judul video



- Rata-rata panjang judul video: 923.08 karakter
- Kebanyakan content creator tidak terlalu banyak memberikan deskripsi pada videonya



- Kebanyakan content creator YouTube tidak menutup fitur comments, ratings, dan memiliki video error atau menghapus video

Data Preprocessing

1. *Datatype*
    a. Pertama - tama kami mengubah *datatype* kolom ‘category_id’ menjadi *string / object* karena kami ingin melihat statistika deskriptif dari kolom ini karena kolom ini bersifat kategorikal dan bukan numerik.
    b. Data: (36.791, 18)
2. *Missing Value*
    a. Kemudian kami memeriksa apakah ada *missing value* di setiap kolom yang ada dan menemukan bahwa kolom ‘description’ mempunyai 45 baris yang kosong. Kami memutuskan untuk men-*drop row-row* ini.
    b. Data: (36.746, 18)
3. *Duplicated Data*
    a. Selanjutnya kami mengecek apakah ada data duplikat di dataset ini dan menemukan bahwa ada 4.228 baris yang duplikat dan men-*drop* baris-baris tersebut.
    b. Data: (32.518, 18)
4. *Outliers*
    a. Selanjutnya kami men-drop *outlier* menggunakan z-score pada kolom-kolom numerikal: ‘likes’, ’dislikes’, 'comment_count' , 'No_tags' , 'desc_len'.
    b. Baris-baris yang di-*drop* adalah baris yang mempunyai z-score sama dengan atau lebih dari 3.
    c. Kami menggunakan metode z-score karena jika menggunakan IQR, kami rasa terlalu banyak baris/data yang dihilangkan.
    d. Ada sebanyak 1.477 baris yang di-*drop*
    e. Data: (31.041, 18)
5. *Standardization*
    a. Selanjutnya kami menstandarisasi kolom-kolom ‘likes’, ’dislikes’, 'comment_count' , 'No_tags' , 'desc_len' agar data ini siap di-train ke model
6. *Label Encoding*
    a. *Label encoding* dilakukan kepada kolom yang mempunyai tipe data *boolean:* ‘comments_disable’, ‘ratings_disabled’, ‘video_error_or_removed’
7. *One Hot Encoding*
    a. Kami melakukan *one hot encoding* kepada kolom ‘category_id’
    b. Dalam dataset ini terdapat 17 kategori video
    c. Data: (31.041, 34)

Feature Engineering

1. Prime time for uploading YouTube video
    a. Buat fitur baru Hour yang merupakan bagian hour dari fitur publish_time (waktu publish video).
    Code:
    df['Hour'] = df['publish_time'].str[:2]
    b. Buat fungsi daypart untuk mendefinisikan apakah hour tersebut merupakan prime time untuk mengupload video YouTube atau tidak. Lalu apply fungsi tersebut untuk membuat kolom baru is_prime yang berisi 0 dan 1. 0 berarti hour tersebut bukan prime time untuk mengupload video dan sebaliknya untuk prime_time bernilai 1. Ketentuan nya adalah sebagai berikut:
    - prime time: 2 pm - 4 pm, 9 am - 11 am.
    - bukan prime time: selain waktu tersebut di atas.
    Source: https://influencermarketinghub.com/best-times-to-publish-youtube-videos/
    Asumsi: setiap hari memiliki prime time yang sama
    Code:
    ```
    def daypart(Hour):
       if Hour in ['9','10','11','14','15','16']:
          return 1
       else: return 0
    df['is_prime'] = df['Hour'].apply(daypart)
    df.drop(columns=['Hour'], inplace=True)
    ```

2. Delay between uploading YouTube videos and videos becoming trending (day(s))
    a. Ubah tipe fitur trending_date menjadi datetime64.
    Code:
    df['trending_date'] = df['trending_date'].astype('datetime64')
    b. Kurangkan fitur publish_date dari trending_date untuk mendapatkan selisih hari antara upload video dan video tersebut menjadi trending.
    Code:
    df['into_trending'] = df['trending_date'] - df['publish_date']
    c. Untuk memudahkan komputasi nantinya, ambil bagian angka dari fitur into_trending dan ubah tipe manjadi integer (sebelumnya adalah timedelta64[ns]).
    Code:
    ```
    df['into_trending'] = df['into_trending'].astype('str')
    df['into_trending'] = df['into_trending'].str.split().str[0]
    df['into_trending'] = df['into_trending'].astype('int')
    ```
    d. Update kolom into_trending dengan ketentuan sebagai berikut:
    - bernilai 0 jika into_trending>2, yang berarti video trending cukup lama dari waktu video diupload.
    - bernilai 1 jika into_trending<=2, yang berarti video trending dengan cepat

setelah video diupload.
Code:

```
lag = []
for index, column in df.iterrows():
    if column['into_trending'] <=2:
        a = 1
    else:
        a = 0
    lag.append(a)
df['into_trending'] = lag
```

Data sebelum feature engineering: (31.041, 34)
Data setelah feature engineering: (31.041, 36)

Modeling

Feature Selection

Feature: 28 columns

‘likes_std', 'dislikes_std',
'comment_count_std', 'No_tags_std', 'desc_len_std', 'len_title_std',
'comments_disabled_label', 'ratings_disabled_label',
'video_error_or_removed_label', 'category_id_1', 'category_id_10',
'category_id_15', 'category_id_17', 'category_id_19', 'category_id_2',
'category_id_20', 'category_id_22', 'category_id_23', 'category_id_24',
'category_id_25', 'category_id_26', 'category_id_27', 'category_id_28',
'category_id_29', 'category_id_30', 'category_id_43','is_prime','into_trending'

Target: 1 column

‘views’

Data Train dan Data Test

Data awal: (31.041, 29)
Data Train: (24.832, 29)
Data Test: (6.209, 29)

Model Selection

Pada tahap ini, dilakukan percobaan modeling menggunakan algoritma-algoritma yang sudah dipelajari selama bootcamp, yaitu : Linear Regression(Ridge, Lasso, dan Elastic Net), Decision Tree Regressor, Random Forest, dan SVR. Target dari algoritma-algoritma tersebut adalah untuk mencari algoritma dengan nilai R^2 nya terbesar dan juga bestfit, sedangkan untuk nilai RMSE nya terkecil. Karena nilai dari R^2 menunjukkan seberapa kuatnya pengaruh feature terhadap target, semakin besar nilainya maka semakin kuat pengaruhnya. Sedangkan untuk RMSE menunjukkan nilai skor kuadrat yang mengukur besaran error rata-rata yang nantinya digunakan untuk mengukur tingkat akurasi dari hasil perkiraan model, semakin kecil nilainya maka akan semakin akurat suatu model.

Setelah dilakukan percobaan, diperoleh untuk setiap nilai RMSE dan R^2 dari masing-masing algoritma pada tabel berikut:

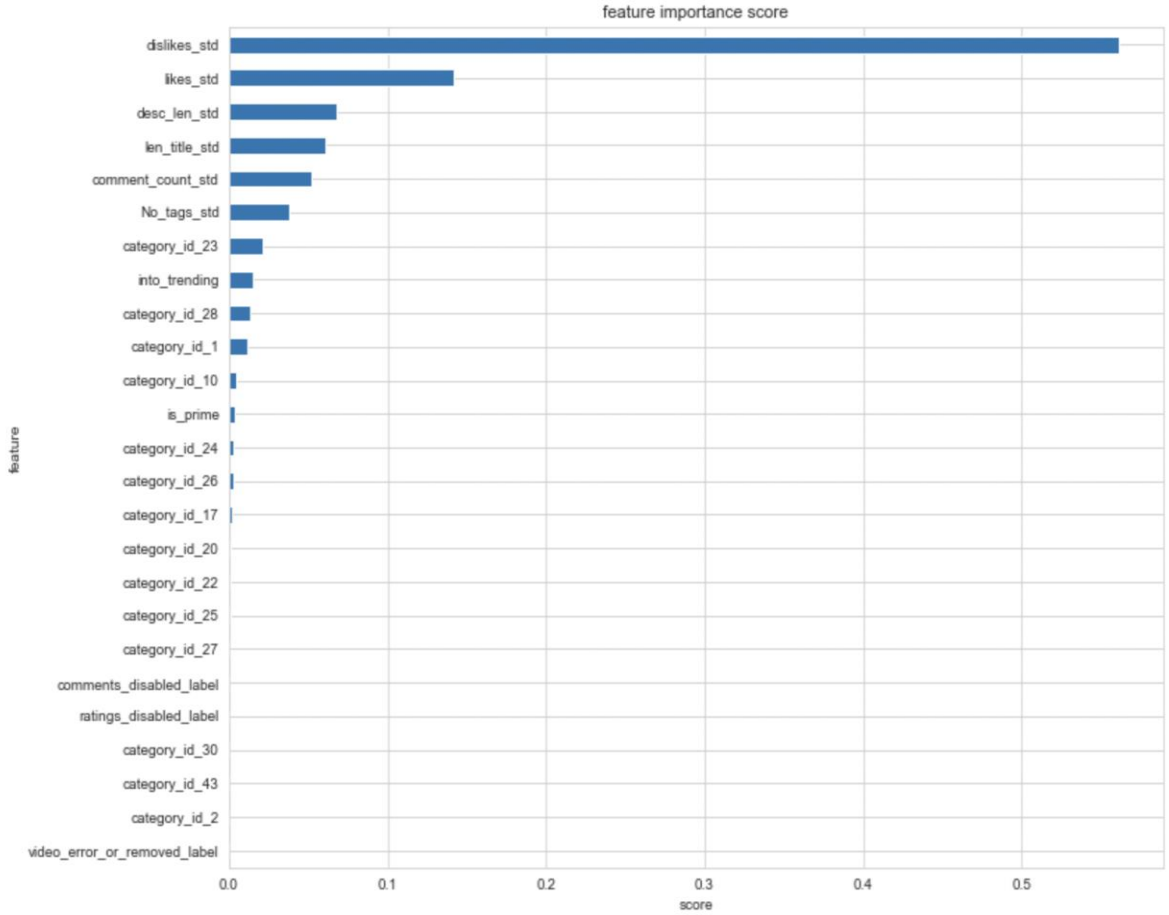
No	Algorithm	R^2 score (Train Set)	R^2 score (Test Set)	RMSE (Train Set)	RMSE (Test Set)	Result
1.	Linear Regression	0.64	0.62	934646.65	1042959.5	Bestfit tetapi nilai masih rendah
	Ridge	0.64→0.64	0.62→0.62	934666.3→934699.81	1042940.5→1042957.53	
	Lasso	0.64→0.64	0.62→0.62	934646.67→934646.66	1042958.64→1042959.47	

	Elastic Net	0.58→0.64	0.54→0.62	1015363.88→934646.95	1146232.18→1042954.96	
2.	Decision Tree Regressor	1.0→0.78	0.83→0.72	4522.53→732181.36	704696.2→891422.55	Overfit
3.	Random Forest	0.99	0.9	190188.45	539586.77	Bestfit
4.	SVR	-0.09	-0.09	1627707.42	1758135.5	Bad Model

Analisa:

1. Pada algoritma **Linear Regression** baik itu menggunakan Regularization tipe Ridge, dan Lasso menghasilkan nilai yang sama pada R^2 score pada training set dan test set nya, yaitu sebesar 0,64 pada training set dan 0,62 pada tes set, sedangkan pada Elastic Net nilai R^2 score pada training set dan test set nya adalah 0,58 dan 0,54. Setelah dilakukan Tuning Hyperparameter, nilai R^2 score pada training set dan test set nya tetap sama, sedangkan pada Elastic Net nilai R^2 score pada training set dan test set nya menjadi 0,64 dan 0,62. Hasil ini sudah bestfit namun nilainya masih rendah untuk dapat dijadikan model. Untuk nilai RMSE nya masih cukup besar sehingga algoritma ini belum bisa digunakan dan harus dilakukan percobaan pada algoritma lain.
2. Pada algoritma **Decision Tree Regressor**, nilai R^2 score pada training set dan test set nya adalah 1,0 dan 0,82. Hasil ini masih **Overfit** karena nilai R^2 score pada test set lebih kecil daripada training set nya. Setelah dilakukan Tuning Hyperparameter, nilai R^2 score pada training set dan test set nya turun menjadi 0,78 dan 0,72. Hasil ini menjadi bestfit, namun nilainya masih rendah untuk dijadikan model. Lalu untuk nilai RMSE nya masih cukup besar sehingga algoritma ini belum bisa digunakan dan harus dilakukan percobaan pada algoritma lain.
3. Pada algoritma **Random Forest**, nilai R^2 score pada training set yaitu 0,99 dan pada test set nya yaitu 0,9. Hasil ini sudah baik dan **Best Fit** sehingga bisa dijadikan sebagai model. Lalu untuk nilai RMSE nya sudah cukup kecil jika dibandingkan dengan algoritma lainnya yang menandakan **algoritma ini lebih akurat**.
4. Pada algoritma **SVR**, nilai R^2 score pada training set dan test set nya sama-sama bernilai -0,09. Hasil ini sangat jauh dari target yang dicari dan nilainya pun negatif yang menandakan model ini tidak bagus untuk digunakan **(Bad Model)**. Lalu untuk nilai RMSE nya masih sangat besar sehingga algoritma ini belum bisa digunakan dan harus dilakukan percobaan pada algoritma lain

Feature Importance



Setelah mendapatkan model yang terbaik, yakni **Random Forest**, maka selanjutnya dapat diperoleh feature importance yang mana merujuk pada urutan seberapa besar feature data tersebut mempengaruhi data target (views). Dari gambar di atas dapat disimpulkan bahwa top 5 feature importance nya adalah sebagai berikut:

1. dislikes_std
2. likes_std
3. desc_len_std
4. len_title_std
5. comment_count_std

Best Model

Random Forest

- R^2 score (Test Set): 0.9
- R^2 score (Train Set): 0.99
- RMSE (Test Set): 539586.77
- RMSE (Train Set): 190188.45
- Result: Bestfit

Summary

Berdasarkan seluruh algoritma yang telah digunakan, algoritma **Random Forest** merupakan algoritma terbaik dari yang lainnya, dengan nilai R^2 score pada training set sebesar 0,99 dan pada test set nya sebesar 0,9. Hasil ini sudah baik dan Best Fit, begitu juga dengan nilai RMSE nya sudah cukup kecil jika dibandingkan dengan algoritma lainnya yang memperkuat bahwa algoritma Random Forest lebih akurat.

5 Feature importance yang paling berpengaruh terhadap views di YouTube adalah dislike, like, panjang kata dalam deskripsi video, panjang kata dalam judul video, dan jumlah komentar. Berdasarkan data yang kita miliki, masing-masing feature importance dapat memberikan insight berikut:

- Dislike sangat berpengaruh dengan banyaknya views karena sangat jauh nilai score-nya dengan feature lainnya, artinya apabila ingin mendapatkan views yang banyak kita bisa membuat video yang kontroversial, yang akan mengundang banyak haters sehingga akan banyak orang yang penasaran dan menonton video yang kita buat.
- Likes dan jumlah komen juga berpengaruh dengan jumlah views yang didapat, untuk para content creator kedepannya agar dapat mengajak para viewersnya untuk memberikan like dan comment pada videonya, karena hal tersebut dapat memberikan jumlah views yang lebih banyak.
- Panjang kata dalam deskripsi video juga berpengaruh dengan jumlah views, para content creator juga sebaiknya membuat deskripsi video sejelas mungkin agar penonton YouTube dapat memahami isi konten video yang kita buat. Kemudian memberikan banyak hashtag pada deskripsi videonya agar videonya dapat tersebar luas kepada penonton lain.
- Panjang kata dalam judul berpengaruh dengan jumlah views, buatlah judul yang menarik dan jangan terlalu pendek agar penonton tau apa yang akan dia lihat. Gunakan judul dengan panjang antara 81-100 karakter. Judul dengan panjang seperti ini memiliki peringkat terbaik di YouTube menurut penelitian *Hubspot.*

Penelitian *Hubspot*
https://cdn2.hubspot.net/hub/53/file-2505556912-pdf/Data_Driven_Strategies_For_Writing_Effective_Titles_and_Headlines.pdf