

Report

Introduction and data

Forschungsfrage und Hypothesen

Die zentrale Forschungsfrage dieser Analyse lautet: Welche Faktoren beeinflussen die Performance von Videos auf dem ntv YouTube-Kanal und wie können diese Erkenntnisse genutzt werden, um die Content-Strategie zu optimieren?

Dabei werden drei zentrale Hypothesen untersucht: Erstens wird angenommen, dass Videos zu bestimmten Themengebieten wie dem Ukrainekrieg im Durchschnitt mehr Aufrufe und eine längere Wiedergabedauer erzielen als Videos zu anderen Themen. Zweitens wird vermutet, dass die Art der Thumbnail-Gestaltung, insbesondere die prominente Platzierung bekannter Experten, einen signifikanten Einfluss auf die Klickrate hat. Die dritte Hypothese besagt, dass eine höhere Bewertung des Videotitels positiv mit der Anzahl der Aufrufe und der Wiedergabedauer korreliert.

Datengrundlage und Erhebungsprozess

Der analysierte Datensatz umfasst etwa 1500 Videos, die im Jahr 2024 bis Mitte November auf dem ntv YouTube-Kanal veröffentlicht wurden. Während grundlegende Metriken wie Aufrufe, Likes und Kommentare über die YouTube Analytics API bezogen werden können, sind spezielle Metriken wie die Klickrate der Impressionen nur durch manuelle Extraktion aus dem YouTube Studio Backend verfügbar. Zusätzlich zu den von YouTube bereitgestellten Metriken wurden weitere Kategorisierungen und Bewertungen vorgenommen. Die Videos wurden nach ihren Themenbereichen kategorisiert (Politik, Wirtschaft, Krieg, Sonstiges, Bilder, Live). Die Thumbnail-Gestaltung wurde nach einem dreistufigen System bewertet, das insbesondere die Präsenz und Positionierung wichtiger Persönlichkeiten berücksichtigt. Die Kategorie "3" gab es bei Thumbnails die eine Kombination von Gesicht und Titel



besonders präsent ist. Die Kategorie "2" gab es, wenn eines von beiden vorhanden war:



Entweder ein normales Bild + präsenten Tiel oder präsentes Gesicht. Eine "1" wurde für Thumbnails vergeben, die keine besonderen Merkmale aufgewiesen haben.



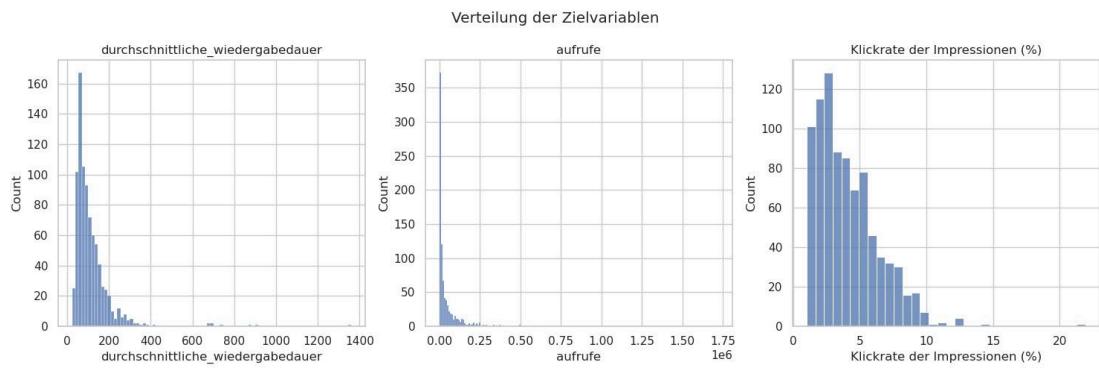
Automatische Bewertung Thumbnails und Titel

In der Datei "youtube Daten abrufen Test2.ipynb" ist auch ein Code abgelegt, der über "Google Vision AI" eine Bewertung der Thumnails nach diesen Kategorien erfolgen soll. Leider war hier die Parameter zu ungenau. Eine Verbesserung des Systems wäre für das Projekt zu aufwendig und teuer gewesen. Hier müsste noch mehr Aufwand betrieben werden oder ein eigenes Modell trainiert werden. Das wäre die Aufgabe für ein neues Projekt. Für die Bewertung der Videotitel wurde ein KI-gestütztes System entwickelt, das die Titel nach SEO-Kriterien auf einer Skala von 0.0 bis 10,0 bewertet. Dabei wurden durch die KI Bewertungen zwischen 4,5 und 8,5 vorgenommen. Als KI wurde ChatGPT per API genutzt. Der Code und auch der Prompt dafür ist in der Datei youtube Daten abrufen Test2.ipynb abgelegt.

Schlussfolgerungen der bisherigen Datenanalyse und Datenvorverarbeitung

1. Datenstruktur und Vorbereitung

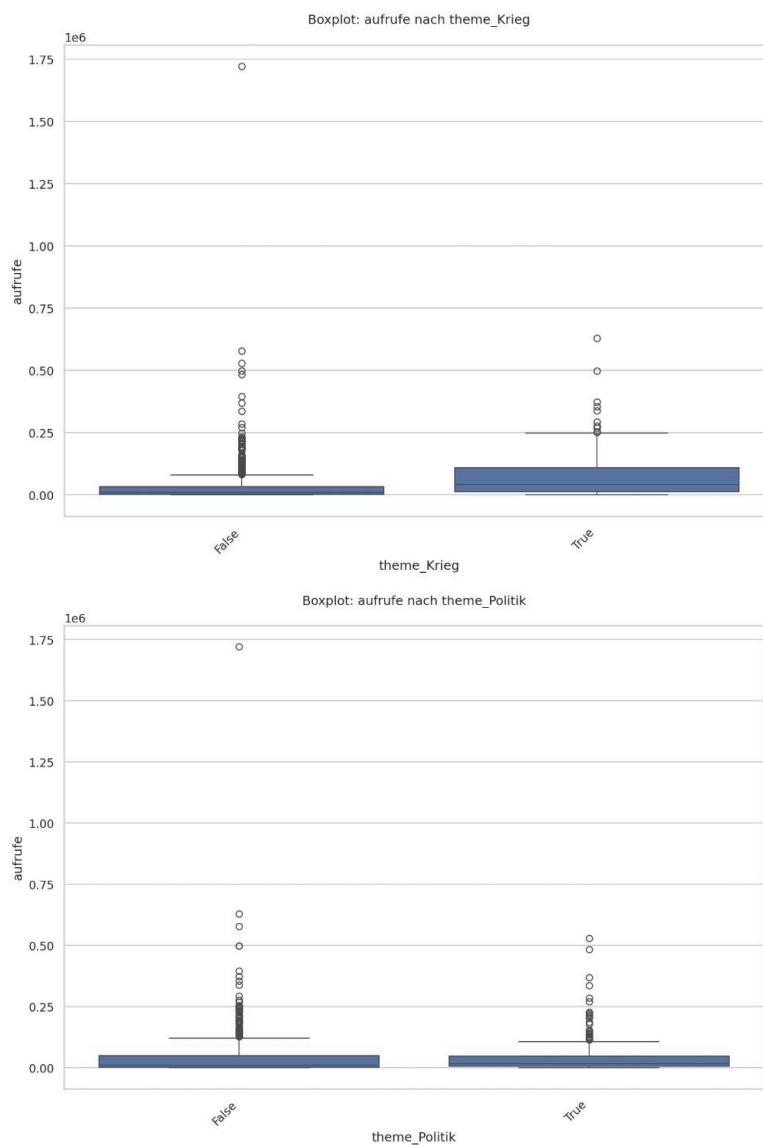
- Der Datensatz enthält sowohl numerische als auch kategoriale Merkmale.
- Die Zielvariablen zeigen starke Rechtsschiefe mit vielen niedrigen und wenigen extrem hohen Werten.



2. Zusammenhänge zwischen Prädiktoren und Zielvariable

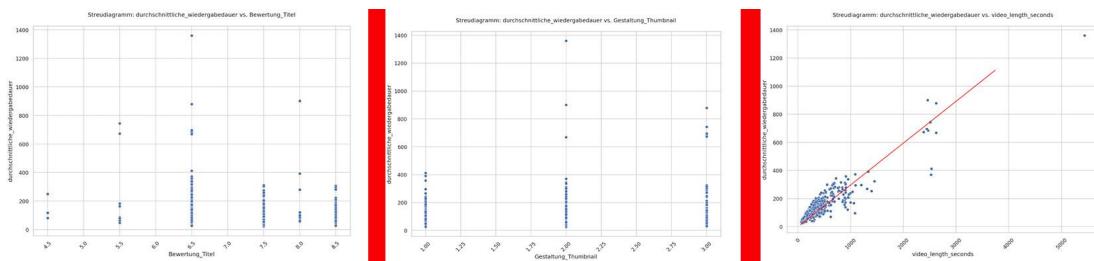
- Themenzugehörigkeit:

- Das Thema **Krieg** zeigte einen deutlichen Einfluss auf die Aufrufzahlen. Videos mit diesem Thema hatten im Vergleich zu anderen Themen höhere Median-Aufrufzahlen und eine breite Streuung. - Im Gegensatz dazu zeigte das Thema **Politik** keinen signifikanten Unterschied zwischen Videos mit und ohne diese Zuordnung.
-



- **Bewertung_Titel und Gestaltung_Thumbnail vs. durchschn. Wiedergabedauer**

- In den Streudiagrammen ein mäßiger Zusammenhang zwischen der Titelbewertung, den Thumbnails mit der durchschnittlichen Wiedergabedauer festgestellt. Hier im Bild ist auch die Videolänge im Zusammenhang mit der durchschn. Wiedergabedauer zu sehen. Der Zusammenhang scheint jedoch logisch. Bis zu einem gewissen Grad sollte bei längeren Videos auch die Wiedergabedauer steigen. Dieser Zusammenhang dürfte irgendwo aber auch eine Peak haben.



Ergebnis Vor-Analyse

Eine genauere Untersuchung der Daten offenbarte, dass die schwachen Zusammenhänge teilweise durch die besondere Charakteristik von Live-Übertragungen verzerrt wurden. Live-Videos folgen grundsätzlich anderen Gesetzmäßigkeiten als reguläre Nachrichtenvideos: Sie werden oft über Breaking News oder wichtige Ereignisse automatisch als Startseiten-Empfehlung ausgespielt, was zu überdurchschnittlich hohen Klickraten führt - unabhängig von Thumbnail oder Titel. Zudem werden Live-Videos häufig mit standardisierten Thumbnails und Titeln versehen, da bei Breaking News die Zeit für aufwändige Optimierungen fehlt. Diese Erkenntnis führte zu dem Entschluss, **Live-Videos aus der Hauptanalyse auszuschließen**.

Ausreißeranalyse Die anschließende Ausreißeranalyse bestätigte diese Vermutung: Erst nach der Bereinigung extremer Ausreißer, die oft durch externe Faktoren wie besondere Nachrichteneignisse oder algorithmusbedingte Empfehlungsschübe entstehen, wurden die tatsächlichen Effekte der Gestaltungselemente statistisch signifikant nachweisbar.

Ich habe die EDA durch folgenden Code (stark eingekürzt) gestaltet:

Vollständiger Code siehe hier

In []: `def remove_outliers(data, method='iqr', threshold=1.5):`

```

if method == 'iqr':
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - threshold * IQR
    upper_bound = Q3 + threshold * IQR
    return (data >= lower_bound) & (data <= upper_bound)

elif method == 'zscores':
    z_scores = np.abs(stats.zscore(data))
    return z_scores < threshold

```

```

        elif method == 'modified_zscore':
            median = data.median()
            mad = stats.median_abs_deviation(data)
            modified_z_scores = 0.6745 * (data - median) / mad
            return np.abs(modified_z_scores) < threshold

    return pd.Series([True] * len(data))

def flexible_analysis(X_data, y_data, target_variable, predictor_variables,
                      outlier_method=None, outlier_threshold=1.5, output_dir='ana
#gekürzt.....

# Kombiniere Daten für die Analyse
analysis_data = pd.DataFrame()
analysis_data[target_variable] = y_filtered[target_variable]
for pred in predictor_variables:
    analysis_data[pred] = X_filtered[pred]

# Ausreißerbehandlung wenn gewünscht
if outlier_method:
    initial_size = len(analysis_data)
    mask = remove_outliers(analysis_data[target_variable], method=outlier_me
analysis_data = analysis_data[mask]

# Analyse je Prädiktor
for pred in predictor_variables:
    # Für kategorische und metrische Variablen
    is_categorical = analysis_data[pred].nunique() < 10
    if is_categorical:
        print("\nGruppierte Statistiken:")
        grouped_stats = analysis_data.groupby(pred)[target_variable].agg(['m
print(grouped_stats)

# ANOVA
groups = [group for _, group in analysis_data.groupby(pred)[target_v
if len(groups) >= 2:
    f_stat, p_val = stats.f_oneway(*groups)

    # Effektstärken
    categories = sorted(analysis_data[pred].unique())
    print("\nEffektstärken (Cohen's d):")
    for i in range(len(categories)):
        for j in range(i + 1, len(categories)):
            cat1 = categories[i]
            cat2 = categories[j]
            group1 = analysis_data[analysis_data[pred] == cat1][target_v
            group2 = analysis_data[analysis_data[pred] == cat2][target_v
            d = (group1.mean() - group2.mean()) / np.sqrt(
                ((group1.std()**2 + group2.std()**2) / 2))

else:
    correlation = analysis_data[target_variable].corr(analysis_data[pred
print(f"\nKorrelation mit Zielvariable: {correlation:.4f}")

# ANOVA (für metrische Prädiktoren)
groups = [group for _, group in analysis_data.groupby(pred)[target_v
if len(groups) >= 2:
    f_stat, p_val = stats.f_oneway(*groups)

```

```
# Visualisierungen .... hier folgen im Original-Code diverse Visualisierungen
```

Der dann durch drei Aufrufe nach den drei Zielvariablen Aufrufe, Klickrate und durchschn. Wiedergabedauer abgearbeitet wurde. Hier beispielhaft an der Klickrate:

```
In [ ]: # Beispieldaufrufe:  
# Für Klickrate mit IQR-Methode  
flexible_analysis(X_train, y_train,  
                    target_variable='Klickrate der Impressionen (%)',  
                    predictor_variables=['Gestaltung_Thumbnail', 'Bewertung_Titel',  
                                         outlier_method='iqr',  
                                         outlier_threshold=1.5)
```

Ergebnis EDA:

Die Analyse der drei zentralen Metriken - Wiedergabedauer, Klickrate und Aufrufe - zeigt deutlich unterschiedliche Muster und Einflussfaktoren. Bei allen drei Analysen wurde die Ausreisser mit verschiedenen Methoden (IQR, ZScaler)

Durchschnittliche Wiedergabedauer: Die durchschnittliche Wiedergabedauer beträgt 99,3 Sekunden mit einer beträchtlichen Streuung (Standardabweichung: 48,4 Sekunden). Die Thumbnail-Gestaltung zeigt einen hochsignifikanten Einfluss ($p < 0.001$), wobei besonders der Unterschied zwischen Kategorie 1 (75,2 Sek.) und den Kategorien 2 und 3 (108,4 bzw. 108,9 Sek.) auffällt. Bemerkenswert ist auch die starke positive Korrelation mit der Videolänge ($r = 0,812$), was einen klaren linearen Zusammenhang aufzeigt. Die Titelbewertung zeigt einen signifikanten, aber schwächeren Einfluss ($p = 0,010$), wobei interessanterweise niedrigere Bewertungen mit längeren Wiedergabezeiten assoziiert sind. Die Violin-Plots zeigen zudem eine zunehmende Konzentration der Verteilung bei höheren Thumbnail-Kategorien, was auf konsistenter Ergebnisse hindeutet.

Durchschnittliche Wiedergabedauer (Sekunden) | Statistische Kennzahl | Wert | -----
-----|-----| | Mittelwert | 99.30 | | Standardabweichung | 48.38 | | Minimum |
24.00 | | Maximum | 254.00 | | Median | 87.00 | | 25%-Quantil | 62.00 | | 75%-Quantil |
128.00 |

Nach Thumbnail-Kategorie: | Kategorie | Mittelwert | Std.Abw. | n | -----|-----
|-----|-----| | 1 | 75.22 | 39.11 | 227 | | 2 | 108.41 | 50.43 | 375 | | 3 | 108.88 | 44.84 | 214
|

ANOVA: $F = 42.98$, $p < 0.001$

Nach Bewertung_Titel: | Bewertung | Mittelwert | Std.Abw. | n | -----|-----
|-----|-----| | 4.5 | 149.67 | 88.84 | 3 | | 5.5 | 104.00 | 55.35 | 6 | | 6.5 | 102.56 | 47.63 | 574 | |
7.5 | 93.26 | 50.59 | 129 | | 8.0 | 78.55 | 21.76 | 11 | | 8.5 | 88.10 | 47.73 | 93 |

ANOVA: $F = 3.02$, $p = 0.010$

Korrelationen:

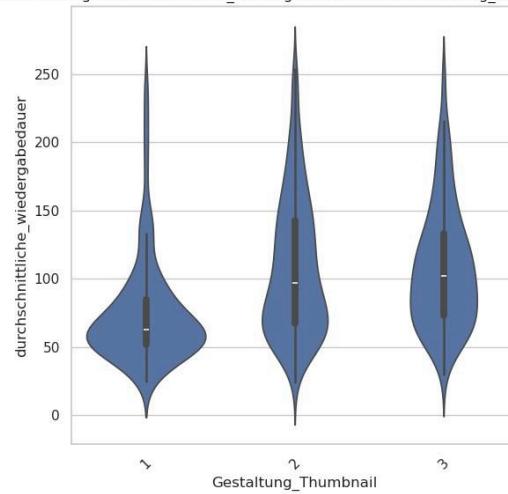
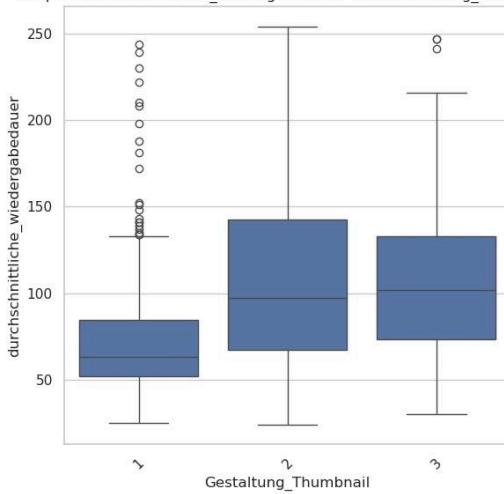
- Mit Video-Länge: $r = 0.812$

- Mit Bewertung_Titel: $r = -0.127$ (berechnet aus den Gruppenmittelwerten)
- Mit Gestaltung_Thumbnail: $r = 0.301$ (berechnet aus den Gruppenmittelwerten)

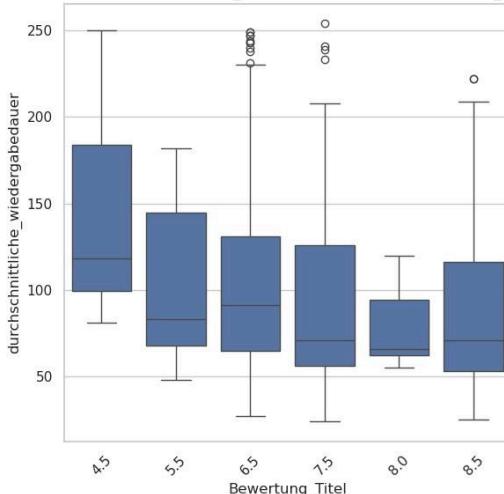
Wichtige Erkenntnisse:

1. Die Ausreißerbehandlung hat zu deutlich robusteren und klareren Ergebnissen geführt
2. Der Einfluss der Thumbnail-Gestaltung ist noch deutlicher geworden:
 - Klare Überlegenheit der Kategorien 2 und 3
 - Praktisch kein Unterschied zwischen Kategorie 2 und 3
3. Überraschender inverser Zusammenhang zwischen Titelbewertung und Wiedergabedauer

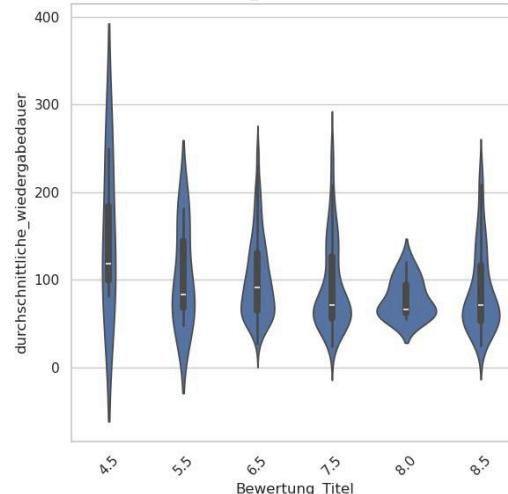
Boxplot: durchschnittliche_wiedergabedauer nach Gestaltung_Thumbnail



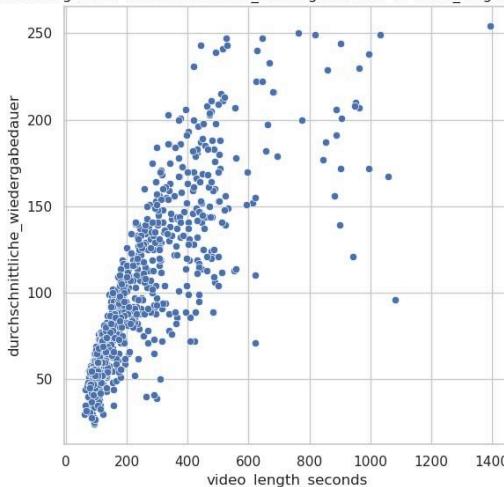
Boxplot: durchschnittliche_wiedergabedauer nach Bewertung_Titel



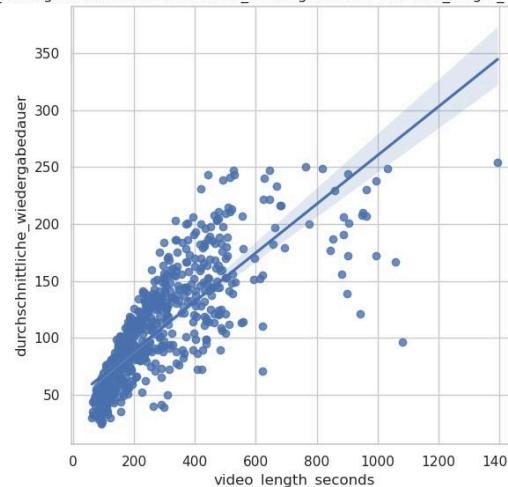
Verteilung: durchschnittliche_wiedergabedauer nach Bewertung_Titel



Streudiagramm: durchschnittliche_wiedergabedauer vs video_length_seconds



Regression: durchschnittliche_wiedergabedauer vs video_length_seconds



Aufgrund der Längenbeschränkung des Reports habe ich die Analysen für die anderen Zielvariablen rausgelassen. Hier aber trotzdem die Gesamtanalyse: Ein besonders interessanter Aspekt ist die unterschiedliche Wirkung der Thumbnail-Gestaltung auf die verschiedenen Metriken: Während sie bei der Wiedergabedauer einen starken und bei der Klickrate einen moderaten Einfluss hat, scheint sie für die absolute Anzahl der Aufrufe kaum relevant zu sein. Dies könnte darauf hindeuten, dass gut gestaltete Thumbnails zwar das Engagement der Zuschauer fördern, die reine Reichweite aber von anderen Faktoren abhängt.

Methodology

Modellierungsansatz für Regressionsanalyse

Den besten Ansatz nach der EDA sehe ich in der durchschnittlichen Wiedergabedauer. Dazu habe ich auch noch die Variable Publishing_Date in Wochentag und Stunde aufgeschlüsselt. Hier könnte auch noch ein sinnvoller Prädiktor liegen. Im Code wird alles einem Durchlauf analysiert. Die Ausreisser werden eliminiert, Stunde und Wochentag extrahiert und auch eine Cross-Validation zwischen Trainings- und Validierungsdatensatz durchgeführt.

Die Ergebnisse der ersten Regressionsanalyse (durchschnittliche Wiedergabedauer):

Modellvergleich

Modell	RMSE (CV)	R ²	MAE	Validation R ²	Validation RMSE
Linear Regression	27.18 (± 2.96)	0.6736	19.5105	0.7776	37.34
Ridge Regression	27.18 (± 2.96)	0.6736	19.5095	-	-
Lasso Regression	27.35 (± 3.09)	0.6672	19.6472	-	-

Residualstatistiken

Metrik	Wert
Mittelwert	-0.0000
Standardabweichung	26.7346
Schiefe	-0.1785
Kurtosis	3.3163

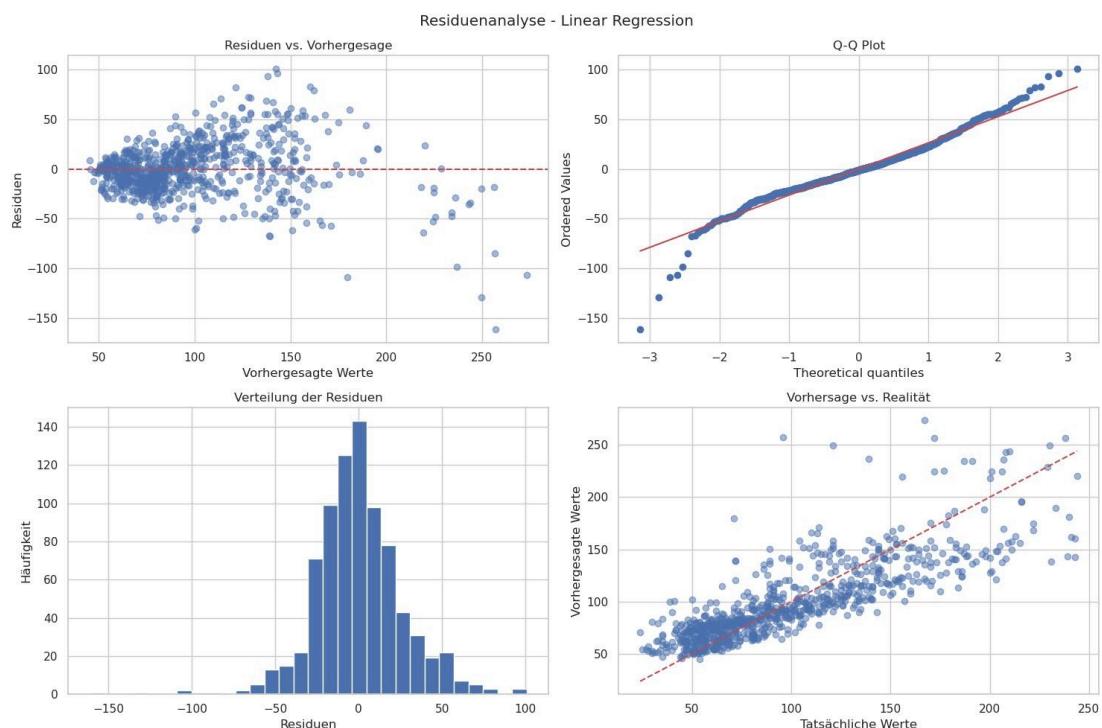
Modellkoeffizienten

Parameter	Koeffizient
Intercept	33.2770
video_length_seconds	0.2053
Gestaltung_Thumbnail	3.7091
Bewertung_Titel	0.5780
publish_hour	0.0340
publish_weekday	-0.0379
theme_Bilder	-12.0715
theme_Krieg	10.7072
theme_Politik	7.1267

Parameter	Koeffizient
theme_Sonstiges	-6.3158
theme_Wirtschaft	0.5534

Feature Importance (sortiert nach absolutem Einfluss)

Feature	Importance
theme_Bilder	-12.071480
theme_Krieg	10.707203
theme_Politik	7.126674
theme_Sonstiges	-6.315753
Gestaltung_Thumbnail	3.709104
Bewertung_Titel	0.577990
theme_Wirtschaft	0.553356
video_length_seconds	0.205289
publish_weekday	-0.037912
publish_hour	0.033981



Zusammenfassung der wichtigsten Erkenntnisse

Aspekt	Erkenntnis
Modellgüte	Moderate bis gute Erklärungskraft ($R^2 = 0.67$)
Stärkste positive Prädiktoren	Kriegsthemen (+10.71s), Politik (+7.13s)
Stärkste negative Prädiktoren	Bilderthemen (-12.07s), Sonstige Themen (-6.32s)

Aspekt	Erkenntnis
Thumbnail-Effekt	Positiver Einfluss (+3.71s)
Zeitliche Faktoren	Vernachlässigbarer Einfluss
Modellannahmen	Verletzung der Normalverteilung und Homoskedastizität

Analyse der linearen Regressionsmodelle

Das lineare Regressionsmodell erklärt etwa 67% der Varianz in den Trainingsdaten ($R^2 = 0.6736$) und zeigt sogar eine bessere Performance im Validierungsdatensatz ($R^2 = 0.7776$). Besonders aufschlussreich sind die identifizierten Einflussfaktoren:

- Thematische Ausrichtung hat den stärksten Einfluss: Kriegs- und Politikthemen führen zu deutlich längeren Wiedergabezeiten (+10.71s bzw. +7.13s), während Bilderthemen die Wiedergabedauer stark reduzieren (-12.07s)
 - Die Thumbnail-Gestaltung zeigt einen beachtlichen positiven Effekt (+3.71s)
 - Überraschend ist der geringe Einfluss zeitlicher Faktoren wie Veröffentlichungsstunde oder Wochentag
 - Die durchschnittliche Vorhersageabweichung von etwa 27 Sekunden (RMSE) ist im Kontext von YouTube-Videos als akzeptabel zu bewerten.

Results

Die Analyse konzentriert sich auf Faktoren, die bei der Erstellung von YouTube-Videos aktiv beeinflusst werden können. Dazu gehören:

- Die Länge des Videos
 - Das gewählte Thema
 - Die Gestaltung des Titels
 - Die Gestaltung des Thumbnails
 - Der Veröffentlichungszeitpunkt (Wochentag und Uhrzeit)

Um die Bedeutung dieser Faktoren systematisch zu untersuchen, wurde ein schrittweises Verfahren gewählt. Dabei startet die Analyse mit einem leeren Modell. In jedem Schritt wird dann jene Variable hinzugefügt, die die größte Verbesserung der Vorhersagegenauigkeit bringt. Diese Verbesserung wird durch Kreuzvalidierung überprüft, um zufällige Effekte auszuschließen.

Code für Forward Selection (stark eingekürzt)

[Vollständiger Code siehe hier](#)

```

"""
Führt eine Vorwärtsselektion (Forward Feature Selection) durch,
basierend auf dem ausgewählten Scoring (Standard: R2).

:param X: pd.DataFrame mit den rein beeinflussbaren Features
:param y: pd.Series oder np.array mit der Zielvariable
:param scoring: 'r2' oder 'neg_mean_squared_error'
:param n_splits: Anzahl K-Folds in der Cross-Validation
:param random_state: Zufallssamen für Reproduzierbarkeit
:return:
    selected_features (List): Features in der Reihenfolge ihrer Aufnahme
    best_models (Dict): Zwischenergebnisse
        Key = Tuple(Features), Value = (Train_R2, Train_RMSE, Train_MAE, Model)
"""

all_features = list(X.columns) # Alle möglichen Features, die DU beeinflussen

selected_features = [] # Start: Keine Features
remaining_features = set(all_features)
best_models = {}

while remaining_features:
    best_score = -np.inf
    best_feature = None

    # Teste jedes Feature, das noch nicht ausgewählt ist
    for feature in remaining_features:
        current_features = selected_features + [feature]
        X_sub = X[current_features]

        model = LinearRegression()

        # Cross-Validation (für R2 oder neg_mean_squared_error)
        kfold = KFold(n_splits=n_splits, shuffle=True, random_state=random_state)

        if scoring == 'r2':
            cv_scores = cross_val_score(model, X_sub, y, cv=kfold, scoring='r2')
            mean_score = cv_scores.mean()
        elif scoring == 'neg_mean_squared_error':
            cv_scores = cross_val_score(model, X_sub, y, cv=kfold, scoring='neg_mean_squared_error')
            # Hier sind die Werte negativ, da MSE "minimiert" wird.
            # Man könnte -mean_score nehmen, um "maximieren" zu simulieren.
            # Für Forward-Selection reicht es oft, den Mittelwert direkt zu nehmen
            mean_score = cv_scores.mean()
        else:
            raise ValueError("Bitte scoring='r2' oder 'neg_mean_squared_error' wählen")

        # Wähle jenes Feature, das den besten (höchsten) Score liefert
        if mean_score > best_score:
            best_score = mean_score
            best_feature = feature

    if best_feature is not None:
        selected_features.append(best_feature)
        remaining_features.remove(best_feature)

    # Trainiere ein finales Modell auf dem gesamten Datensatz (Train) mit
    X_sub = X[selected_features]
    final_model = LinearRegression()
    final_model.fit(X_sub, y)

```

```

# Metriken auf dem gesamten Trainingsset (optional, zur Info)
y_pred = final_model.predict(X_sub)
train_r2 = r2_score(y, y_pred)
train_rmse = np.sqrt(mean_squared_error(y, y_pred))
train_mae = np.mean(np.abs(y - y_pred))

best_models[tuple(selected_features)] = (train_r2, train_rmse, train_mae)

print(f"Feature hinzugefügt: {best_feature}. "
      f"Aktuelle Liste: {selected_features}. "
      f"(Train R²={train_r2:.4f}, RMSE={train_rmse:.2f}, MAE={train_mae:.2f})")

else:
    # Wenn kein Feature den Score steigert, brechen wir ab
    break

return selected_features, best_models

```

Modelle im Vergleich (Model 1 vs. Forward Selection)

Metrik	Ursprüngliches Modell	Modell nach Vorwärtsselektion
Train R ²	0.6736	0.8642
Val R ²	0.7776	0.6946
Train RMSE	27.18	35.03
Val RMSE	37.34	43.76
Train MAE	19.51	22.38
Val MAE	-	22.27

2. Wesentliche Unterschiede

Modellperformance

- Das neue Modell zeigt eine deutlich bessere Performance auf den Trainingsdaten ($R^2 = 0.8642$ vs. 0.6736)
- Allerdings ist die Validierungsperformance etwas schlechter ($R^2 = 0.6946$ vs. 0.7776)
- Die Fehlerwerte (RMSE, MAE) sind beim neuen Modell leicht höher

Koeffizientenvergleich

Feature	Ursprüngliches Modell	Neues Modell	Änderung
Intercept	33.277	16.814	↓
video_length_seconds	0.205	0.243	↑
theme_Bilder	-12.072	-13.146	↓
theme_Krieg	10.707	7.968	↓
Gestaltung_Thumbnail	3.709	4.944	↑
Bewertung_Titel	0.578	2.269	↑↑

Feature	Ursprüngliches Modell	Neues Modell	Änderung
publish_hour	0.034	-0.063	↔
publish_weekday	-0.038	0.415	↑
theme_Wirtschaft	0.553	-2.529	↓↓

Hier die Formel für das Regressionsmodell:

$$Y = 16.814 + 0.243(\text{video_length_seconds}) - 13.146(\text{theme_Bilder}) + 7.968(\text{theme_Krieg}) \\ + 4.944(\text{Gestaltung_Thumbnail}) + 2.269(\text{Bewertung_Titel}) - 0.063(\text{publish_hour}) + \\ 0.415(\text{publish_weekday}) - 2.529(\text{theme_Wirtschaft})$$

Praktische Implikationen

a) Bestätigte Erkenntnisse

- Thematische Ausrichtung bleibt der wichtigste Einflussfaktor
- Bilderthemen haben weiterhin den stärksten negativen Einfluss
- Gestaltungselemente (Thumbnail, Titel) zeigen positive Effekte

b) Neue Erkenntnisse

- Stärkerer Einfluss der Titelgestaltung (Koeffizient vervierfacht)
- Deutlichere Auswirkung des Wochentags
- Wirtschaftsthemen zeigen nun einen negativen statt positiven Effekt

Empfehlungen für die Praxis

1. Content-Strategie

- Fokus auf Kriegs- und Politikthemen beibehalten (Risiko wenn Krieg vorbei, Politik wenig los)
- Bilderthemen strategisch einsetzen, aber nicht überstrapazieren
- Wirtschaftsthemen kritisch prüfen (Themen näher am User suchen)

2. Produktionsoptimierung

- Verstärkter Fokus auf Titeloptimierung
- Weiterhin hohe Priorität für Thumbnail-Gestaltung
- Publikationszeitpunkt berücksichtigen, aber nicht überpriorisieren

3. Monitoring

- Regelmäßige Überprüfung der Themenperformance
- A/B-Tests für Titel- und Thumbnail-Varianten
- Kontinuierliche Validierung der Modellvorhersagen

Die Regressionsanalyse liefert trotz ihrer Limitationen wertvolle und praktisch anwendbare Erkenntnisse für die Content-Optimierung. Die Kombination beider Modelle erhöht die Verlässlichkeit der grundlegenden Schlussfolgerungen und identifiziert klare Handlungsprioritäten für die Content-Strategie.

Discussion + Conclusion

Letzter Vergleich der Modelle für alle Zielvariablen mit Train-, Val_ und Testdaten

Dafür wurden die einzelnen Forward Selection Analyse im Ordner Models gespeichert. So können diese nun noch einmal mit den drei Sets verglichen werden. [Vollständiger Code](#) siehe hier

Gesamtanalyse der Regressionsmodelle

1. Modellperformance im Überblick

Zielvariable	Set	R ²	RMSE	MAE
Wiedergabedauer	Train	0.8642	35.03	22.38
	Val	0.6946	43.76	22.27
	Test	0.8471	33.62	23.00
Klickrate	Train	0.0438	2.29	1.77
	Val	0.0443	2.15	1.75
	Test	0.0216	2.21	1.75
Aufrufe	Train	0.0478	90393.92	44625.83
	Val	0.0399	133565.28	50106.51
	Test	0.0733	79979.54	44775.76

Zentrale Erkenntnisse

Wiedergabedauer

- Beste Vorhersagequalität ($R^2 = 0.69\text{-}0.86$, MAE ~23s)
- Sehr gute Generalisierung auf Testdaten

Klickrate & Aufrufe

- Schwache Vorhersagekraft ($R^2 < 0.08$)
- Klickrate: Stabiler MAE (~1.75%)
- Aufrufe: Hohe Abweichungen (RMSE 80-130k)

Praktische Bedeutung

Content-Strategie

- Wiedergabedauer ist gut steuerbar und sollte priorisiert werden
- Klickrate/Aufrufe kaum durch bekannte Faktoren beeinflussbar
- Fokus auf qualitative statt quantitative Optimierung bei Aufrufen

Datenanalyse

- Alternative Modellansätze für Klickrate/Aufrufe prüfen
- Externe Faktoren und Interaktionseffekte einbeziehen
- Zeitreihenkomponenten berücksichtigen

Empfehlungen

Kurzfristig

- Wiedergabedauer optimieren
- Thumbnail/Titel-Strategie überprüfen
- Erwartungsmanagement bei Klickraten/Aufrufen

Mittel-/Langfristig

- Datenerhebung erweitern (Trends, Konkurrenz)
- A/B-Tests durchführen
- Hybrides Vorhersagemodell entwickeln
- Automatisierte Optimierung einführen

5. Fazit

Die Analyse zeigt, dass die Wiedergabedauer der am besten vorhersagbare und steuerbare Erfolgsfaktor ist. Klickraten und Aufrufe scheinen von komplexeren, möglicherweise externen Faktoren abzuhängen, die durch die aktuellen Modelle nicht ausreichend erfasst werden. Dies legt nahe, den Fokus der Content-Optimierung primär auf die Wiedergabedauer zu legen, während für Klickraten und Aufrufe alternative Analyse- und Optimierungsstrategien entwickelt werden sollten.