Gruppe DiKuM

Analysing Facebook posts (Facebook Data)

Einführung in die Programmierung für Nicht-Informatiker I WiSe 2024/25 I 17.02.2025 I Alea Schleier, Lilly-Lu Warnken, Marie Seeger, Abdulaziz Abunjaila, Skofiare Berisha

Gliederung

1. Beschreibung des Datensets
2. Auswertung der Hypothesen
    - 2.1 Hypothese 1
    - 2.2 Hypothese 2
    - 2.3 Hypothese 3
3. Simulation

1. a) Describe the Data

- Facebook-Daten → 500 Beiträge
- 19 Kategorien insgesamt gegeben

- Shapiro Wilk Test → Überprüfung, ob das Datenset eine Normalverteilung aufweist, oder nicht

→ hier keine Normalverteilung, da p-Wert kleiner als 0.05 ist

b) Inhalt des Datensets

- Post-Typ → Foto oder Status
- Kategorie
- Veröffentlichungsmonat, -Wochentag und -uhrzeit
- Paid oder unpaid Post
- Gesamtreichweite
- Engagement → Kommentare, Likes, Shares
- Reichweite und Engagement von Personen, die die Seite bereits geliked haben
- Gesamtinteraktionen

c) Datenlücken und Einschränkungen des Datensets

- Keine Zeitstempel im Detail
- Kein-Content-Inhalt
- Keine Angaben zur Zielgruppe
- Keine Angaben, ob es sich um eine einzelne Facebook Seite handelt oder mehrere

- einzelne Werte (6 insgesamt) fehlen in den Daten

d) Ziele der Datenanalyse

- Performance-Bewertung:

  → Wie effektiv sind die verschiedenen Facebook-Posts in Bezug auf Reichweite, Engagement und Interaktionen?

- Optimierungsstrategien:

  → Wann und wie kann man Posts erstellen, um bessere Ergebnisse zu erzielen?

- Vergleich von paid vs. unpaid Posts:

  → Überprüfung, ob bezahlte Beiträge eine bessere Performance zeigen

e) What can be drawn from the data?

- Nutzerverhalten bei verschiedenen Arten von Posts
- Einfluss von Promotion
- optimalster Zeitpunkt
- Online-Präsenz analysieren
- Verhältnis von Likes, Kommentare, Shares zur Reichweite

f) What cannot be drawn from the data?

- folgen/ liken die Personen, die interagieren, der Seite überhaupt?
- Um welche Seite es sich genau handelt
- Welche Relevanz ergibt sich daraus für uns?
- Was ist das für Content/ Thema
- Warum ein Post erfolgreich war
- Zielgruppen-Insights

2.1 Hypothese 1

“Paid contributions are more likely to be seen by many people. Among other things, more users interact with these posts.”

Vorgehensweise:

- Daten einlesen & bereinigen
- Deskriptive Analyse
- Visualisierung: Balkendiagramm, Boxplot
- Hypothesentest: Mann-Whitney U-Test
- Ergebnisinterpretation
- Fazit

```
# Sicherstellen, dass 'Paid' und die Interaktionen numerisch sind
df['Paid'] = pd.to_numeric(df['Paid'], errors='coerce')
df['comment'] = pd.to_numeric(df['comment'], errors='coerce')
df['like'] = pd.to_numeric(df['like'], errors='coerce')
df['share'] = pd.to_numeric(df['share'], errors='coerce')
df['Lifetime Post Total Reach'] = pd.to_numeric(df['Lifetime Post Total Reach'], errors='coerce')
```

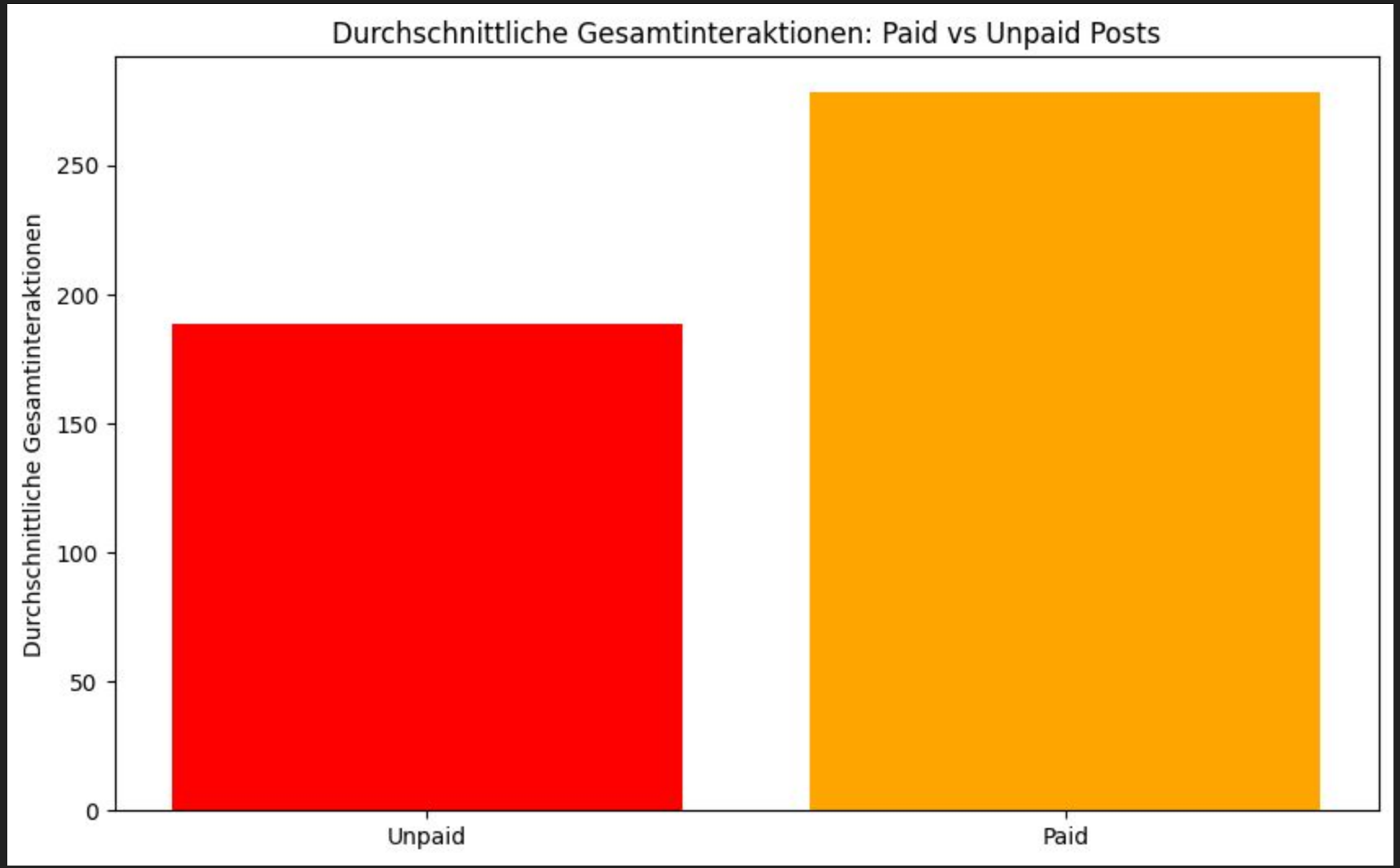
Relevante Spalten:

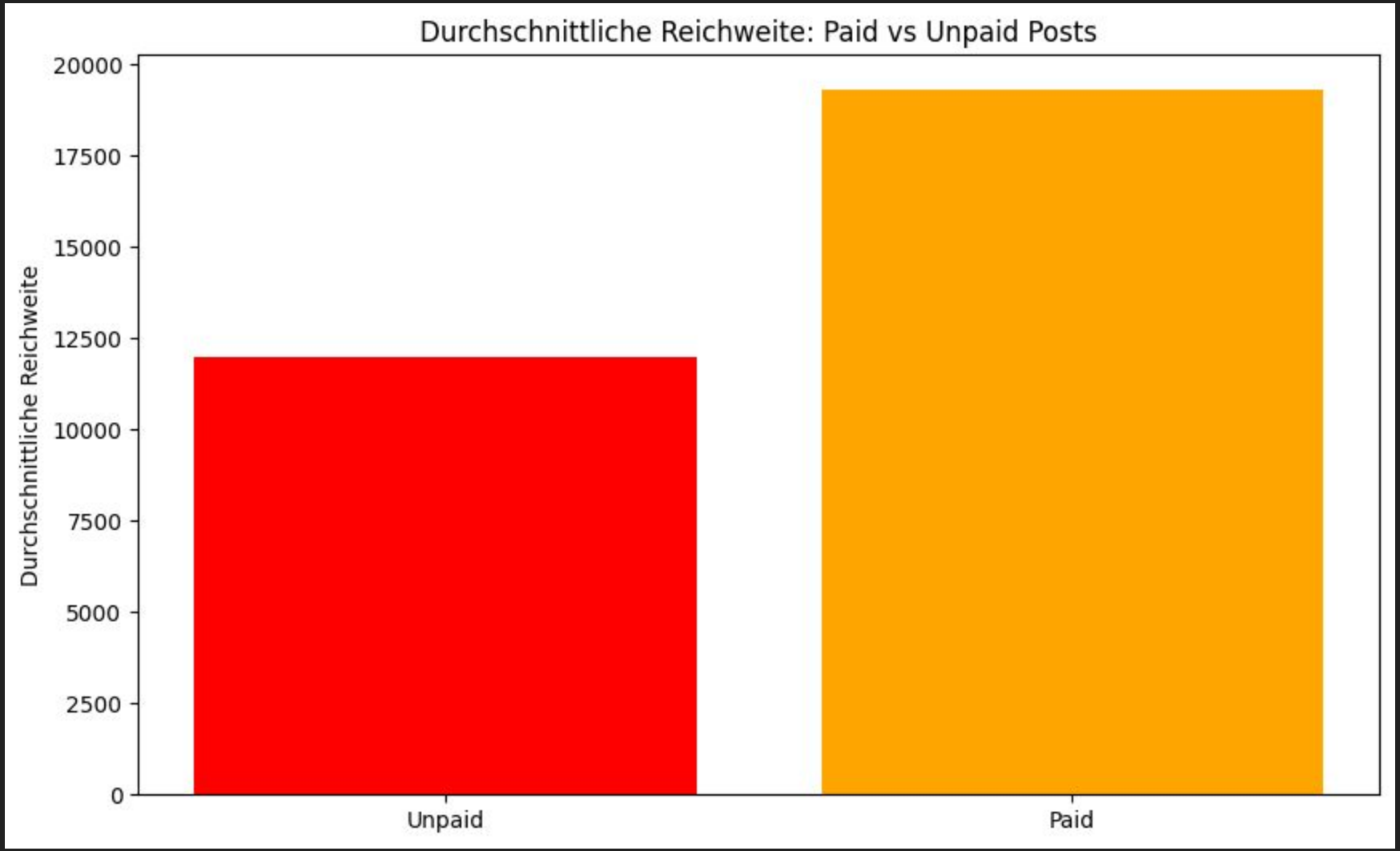
- “Paid”
- “comment”
- “like”
- “share”
- “Lifetime Post Total Reach”

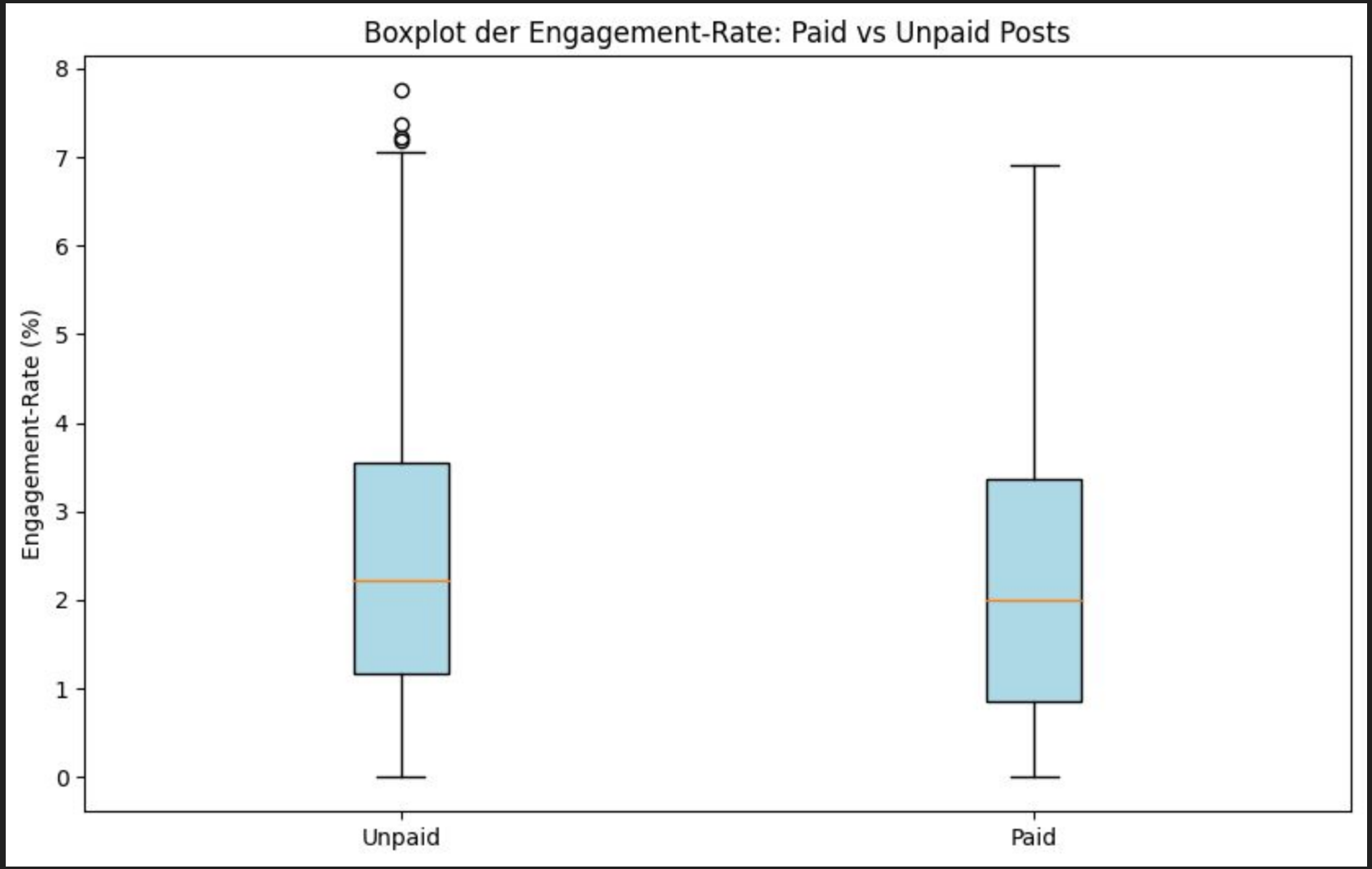
```
# Berechnung der Gesamtinteraktionen
df['Total_Interactions'] = df['comment'] + df['like'] + df['share']

# Berechnung der Engagement-Rate
df['Engagement_Rate'] = (df['Total_Interactions'] / df['Lifetime Post Total Reach']) * 100
```

Berechnung der Mean-Werte

```
mean_paid_interactions = df[df['Paid'] == 1]['Total_Interactions'].mean()
mean_unpaid_interactions = df[df['Paid'] == 0]['Total_Interactions'].mean()
mean_paid_reach = df[df['Paid'] == 1]['Lifetime Post Total Reach'].mean()
mean_unpaid_reach = df[df['Paid'] == 0]['Lifetime Post Total Reach'].mean()
mean_paid_engagement = df[df['Paid'] == 1]['Engagement_Rate'].mean()
mean_unpaid_engagement = df[df['Paid'] == 0]['Engagement_Rate'].mean()
```


Durchschnittliche Gesamtinteraktionen: Paid vs Unpaid Posts
Durchschnittliche Gesamtinteraktionen
0
50
100
150
200
250
Unpaid
Paid


Durchschnittliche Reichweite: Paid vs Unpaid Posts
Durchschnittliche Reichweite
20000
17500
15000
12500
10000
7500
5000
2500
0
Unpaid
Paid


Boxplot der Engagement-Rate: Paid vs Unpaid Posts
Engagement-Rate (%)
0
1
2
3
4
5
6
7
8
Unpaid
Paid

Mann-Whitney U Test (Signifikanz)

```python
# Mann-Whitney-U-Test für Interaktionen
paid_interactions = df[df['Paid'] == 1]['Total_Interactions']
unpaid_interactions = df[df['Paid'] == 0]['Total_Interactions']

if len(paid_interactions) > 0 and len(unpaid_interactions) > 0:
    stat_interactions, p_value_interactions = mannwhitneyu(paid_interactions, unpaid_interactions, alternative='two-sided')
    print(f"\nMann-Whitney U-Test für Interaktionen - p-Wert: {p_value_interactions:.4f}")
    if p_value_interactions < 0.05:
        print("Die Nullhypothese wird verworfen, Paid-Posts erhalten signifikant mehr Interaktionen.")
    else:
        print("Kein signifikanter Unterschied in den Interaktionen festgestellt.")
```

- Daten nicht normalverteilt (somit kein t-Test möglich)
- Stichproben unterschiedlich groß
- Gesamtinteraktionen p-Wert: 0,0003 (<0,05)
- Reichweite p-Wert: 0,00002
- Engagement-Rate p-Wert: 0,1230 (>0,05)

Hypothese 1

Schlussfolgerung:

- Paid-Posts erhalten signifikant mehr Interaktionen als Unpaid-Posts und haben im Schnitt eine höhere Reichweite, was die Hypothese 1 unterstützt
- Doch: Engagement-Rate der bezahlten Posts ist nicht signifikant höher, als die der unbezahlten
- keine klare Evidenz dafür, dass bezahlte Beiträge direkt zu mehr Interaktionen führen

2.2 Hypothese 2

“Videos are paid for more often and are shared more often. So it shows that advertising pays off.”

→ Werden Videoanzeigen häufiger bezahlt als Fotoanzeigen?

→ Haben Videoanzeigen mehr Reichweite, d.h. werden sie häufiger geteilt und erhalten mehr Interaktionen als Fotoanzeigen?

Hypothese 2

Vorgehensweise:

1. Laden der Daten
2. Sicherstellen, dass alle relevanten Spalten vorhanden sind
3. Aufbereitung der Daten

Hypothese 2 - Aufbereitung der Daten

```
df_filtered = df[df['Type'].isin(['Photo', 'Video'])].copy()
df_filtered['Is_Paid'] = pd.to_numeric(df_filtered['Paid'], errors='coerce').fillna(0).astype(int)
df_filtered['Shares'] = pd.to_numeric(df_filtered['share'], errors='coerce').fillna(0).astype(int)
df_filtered['Total_Interactions'] = df_filtered[['like', 'comment', 'share']].sum(axis=1)
```

- Nur der Einbezug von Fotos und Videos ist für die Hypothese relevant
- Die Spalte “Is_Paid” wird bereinigt
- LIkes, Kommentare und Sharen werden in der Spalte “Total_Interactions” summiert

Hypothese 2

Vorgehensweise:

1. Laden der Daten
2. Sicherstellen, dass alle relevanten Spalten vorhanden sind
3. Aufbereitung der Daten
4. Hypothese
5. Vergleich der Paid Rates

Hypothese 2 - Vergleich der Paid Rates

```
video_paid_rate = df_filtered[(df_filtered['Type'] == 'Video') & (df_filtered['Is_Paid'] == 1)].shape[0] / df_filtered[df_filtered['Type'] == 'Video'].shape[0]
image_paid_rate = df_filtered[(df_filtered['Type'] == 'Photo') & (df_filtered['Is_Paid'] == 1)].shape[0] / df_filtered[df_filtered['Type'] == 'Photo'].shape[0]

print(f"\n1. Paid Rates:")
print(f"Video Paid Rate: {video_paid_rate:.2f}")
print(f"Image Paid Rate: {image_paid_rate:.2f}")
```

- Paid Rate bei Videos: **57 %**
- Paid Rate bei Fotos: **28%**

→ Videos werden deutlich öfter bezahlt als Bilder.

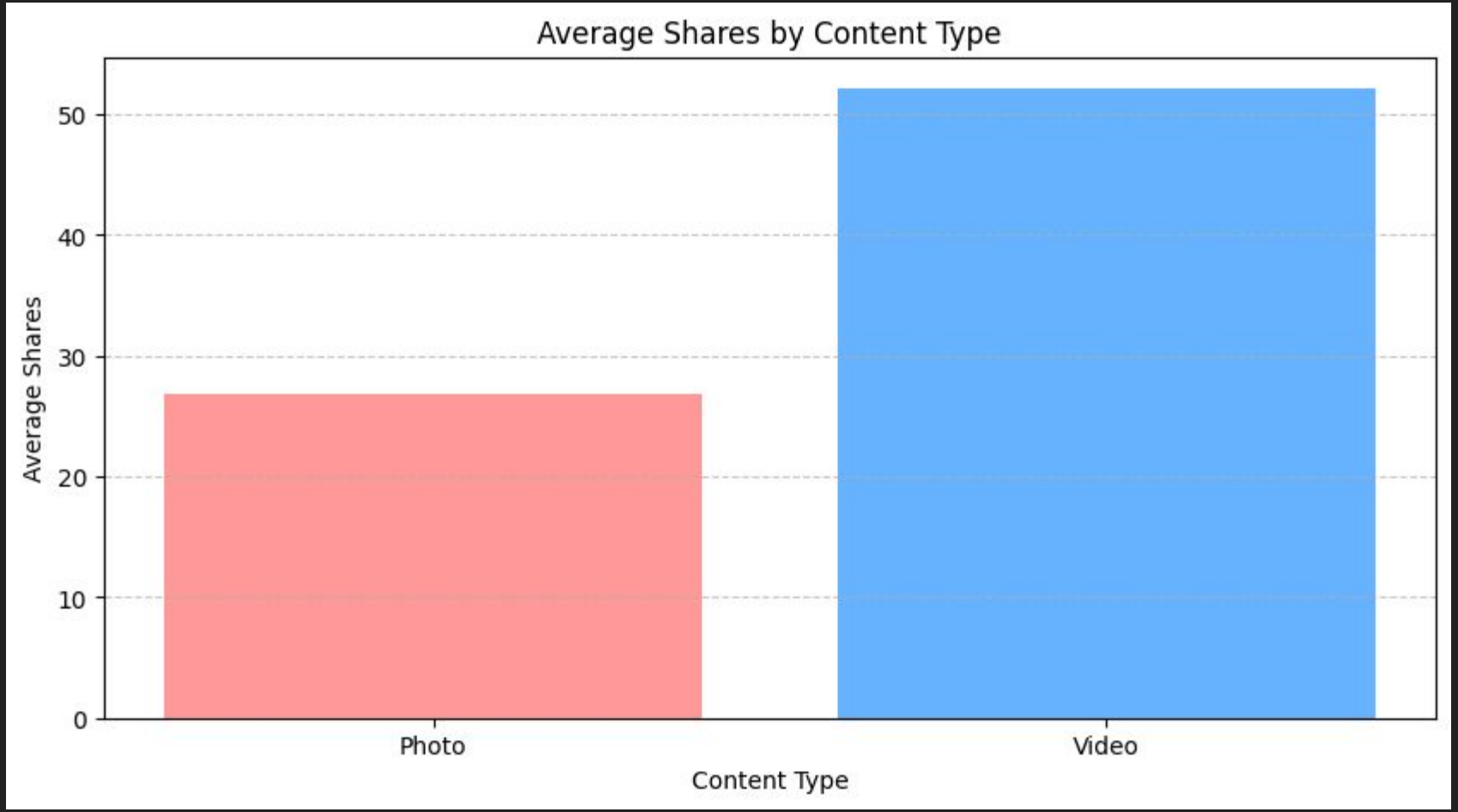
Hypothese 2

Vorgehensweise:

1. Laden der Daten
2. Sicherstellen, dass alle relevanten Spalten vorhanden sind
3. Aufbereitung der Daten
4. Hypothese
5. Vergleich der Paid Rates
6. Vergleich der Shares

Hypothese 2 - Vergleich der Shares

```
video_shares = df_filtered[df_filtered['Type'] == 'Video']['Shares']
image_shares = df_filtered[df_filtered['Type'] == 'Photo']['Shares']

print("\n2. Share Comparison:")
print(f"Video Average Shares: {video_shares.mean():.2f}")
print(f"Image Average Shares: {image_shares.mean():.2f}")

# Mann-Whitney U Test for shares
u_stat, p_value = stats.mannwhitneyu(video_shares, image_shares, alternative='two-sided')
print(f"Mann-Whitney U test p-value for shares: {p_value:.4f}")
```


Average Shares by Content Type
Average Shares
0
10
20
30
40
50
Photo
Video
Content Type

Hypothese 2 - Vergleich der Shares

- Videos: Ø 52,14 Shares
- Bilder: Ø 26,90 Shares
- Durchführung eines Mann Whitney U Tests
- p - Wert: 0,0945 → Es liegt keine Signifikanz vor (> 0,05)

→ ***Videos werden häufiger geteilt als Fotos, der Unterschied ist jedoch nicht signifikant***

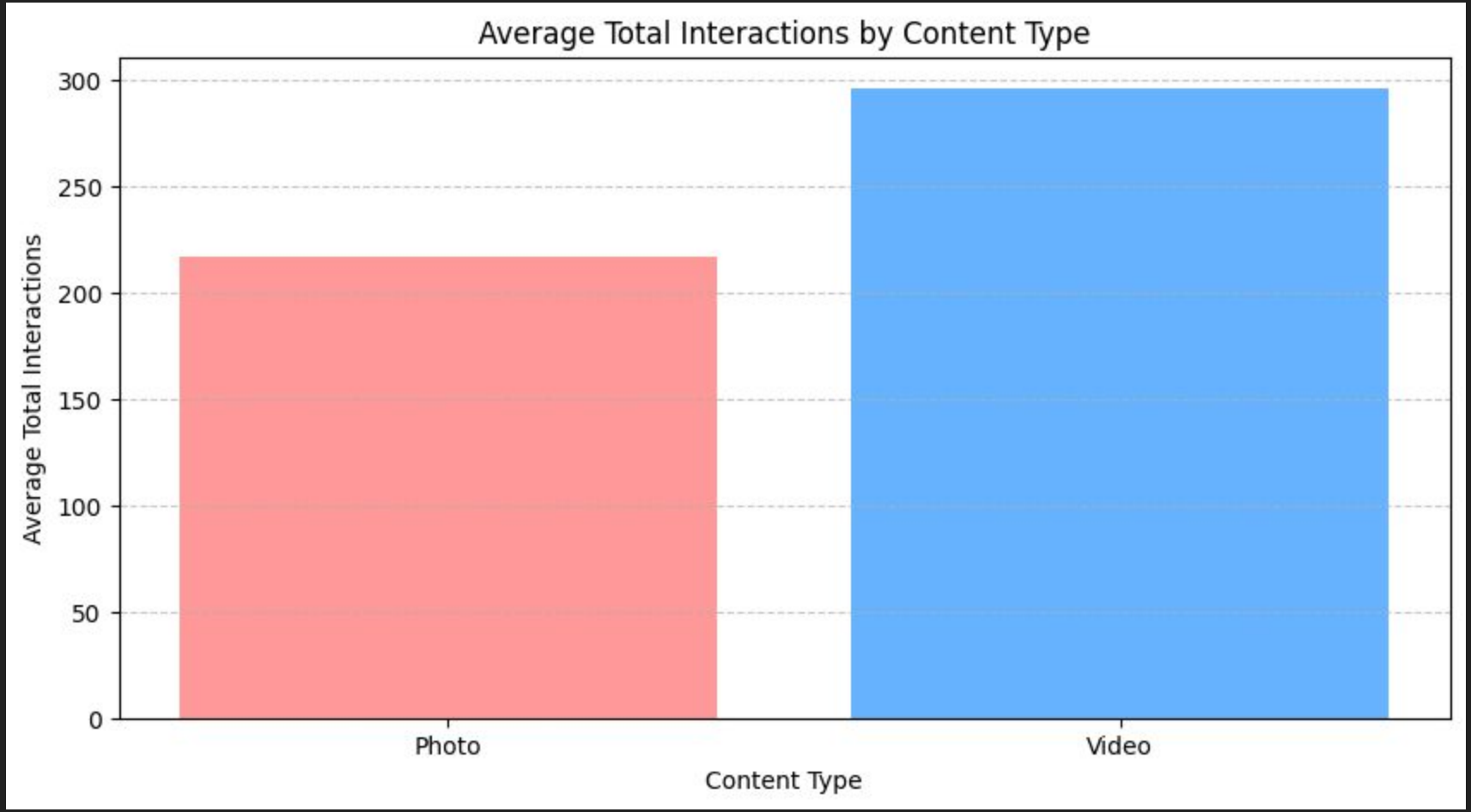
Hypothese 2

Vorgehensweise:

1. Laden der Daten
2. Sicherstellen, dass alle relevanten Spalten vorhanden sind
3. Aufbereitung der Daten
4. Hypothese
5. Vergleich der Paid Rates
6. Vergleich der Shares
7. Vergleich der Gesamtinteraktion

Hypothese 2 - Vergleich der Gesamtinteraktion

```
video_interactions = df_filtered[df_filtered['Type'] == 'Video']['Total_Interactions']
image_interactions = df_filtered[df_filtered['Type'] == 'Photo']['Total_Interactions']

print("\n3. Total Interaction Comparison:")
print(f"Video Average Total Interactions: {video_interactions.mean():.2f}")
print(f"Image Average Total Interactions: {image_interactions.mean():.2f}")

# Mann-Whitney U Test for total interactions
u_stat_interactions, p_value_interactions = stats.mannwhitneyu(video_interactions, image_interactions, alternative='two-sided')
print(f"Mann-Whitney U test p-value for interactions: {p_value_interactions:.4f}")
```


Average Total Interactions by Content Type
Average Total Interactions
300
250
200
150
100
50
0
Photo
Video
Content Type

Hypothese 2 - Vergleich der Gesamtinteraktion

- Videos: Ø 295, 86 Interaktionen
- Bilder: Ø 216,58 Interaktionen
- Durchführung eines Mann Whitney U Tests
- p - Wert: 0,0586 → Es liegt keine Signifikanz vor (> 0,05)

→ ***Videos haben mehr Gesamtinteraktionen, der Unterschied ist jedoch nicht signifikant.***

Hypothese 2 - Fazit

- Videoanzeigen sind häufiger bezahlt
- Videoanzeigen haben höhere Werte für Shares und Interaktionen, jedoch ohne statistische Signifikanz
- Da die Signifikanz fehlt, ist die Hypothese nicht vollständig bestätigt
- Es wird deutlich, dass Werbung für Videos zwar häufiger gekauft wird, aber nicht eindeutig mehr Engagement bringt.

In der Praxis sollte nicht davon ausgegangen werden, dass Video-Werbung mehr Reichweite bringt, da noch mehr Faktoren berücksichtigt werden müssen.

- Es werden simulierte Daten verwendet, reale Daten könnten zu anderen Ergebnissen führen
- Andere Faktoren wie die Qualität des Anzeigeninhalts oder plattformspezifische Besonderheiten werden nicht berücksichtigt
- In der zukünftigen Forschung sollte ein größeres Datenset und mehrere Variablen einbezogen werden

2.3 Hypothese 3

Describe hypothesis

The top 2% of all posts were published in the afternoon on a Thursday. However, users interact with posts most frequently in the evening.

This hypothesis consists of two parts:

The first part: "The top 2% of all posts were published in the afternoon on a Thursday: The top 2% of posts are determined based on a specific criterion, such as the number of likes, shares, comments, or reach

The second part: "Users interact with posts most frequently in the evening." (interactions also include comments, shares and likes)

- **We should focus on using sum instead of mean when analyzing the hypothesis since "most frequently" refers to the total number of interactions**
- **Also we need to be accurate in defining the time periods because, in the dataset, there is only Post Hour and no specific column indicating the exact time period**

In this code, time periods are classified based on the following values in **Post Hour**:

- Morning: from 5:00 to 11:59 ($5 \leq x < 12$)
- Noon: from 12:00 to 14:59 ($12 \leq x < 15$)
- Afternoon: from 15:00 to 17:59 ($15 \leq x < 18$)
- Evening: from 18:00 to 21:59 ($18 \leq x < 22$)
- Night: from 22:00 to 4:59 (outside the previous range)

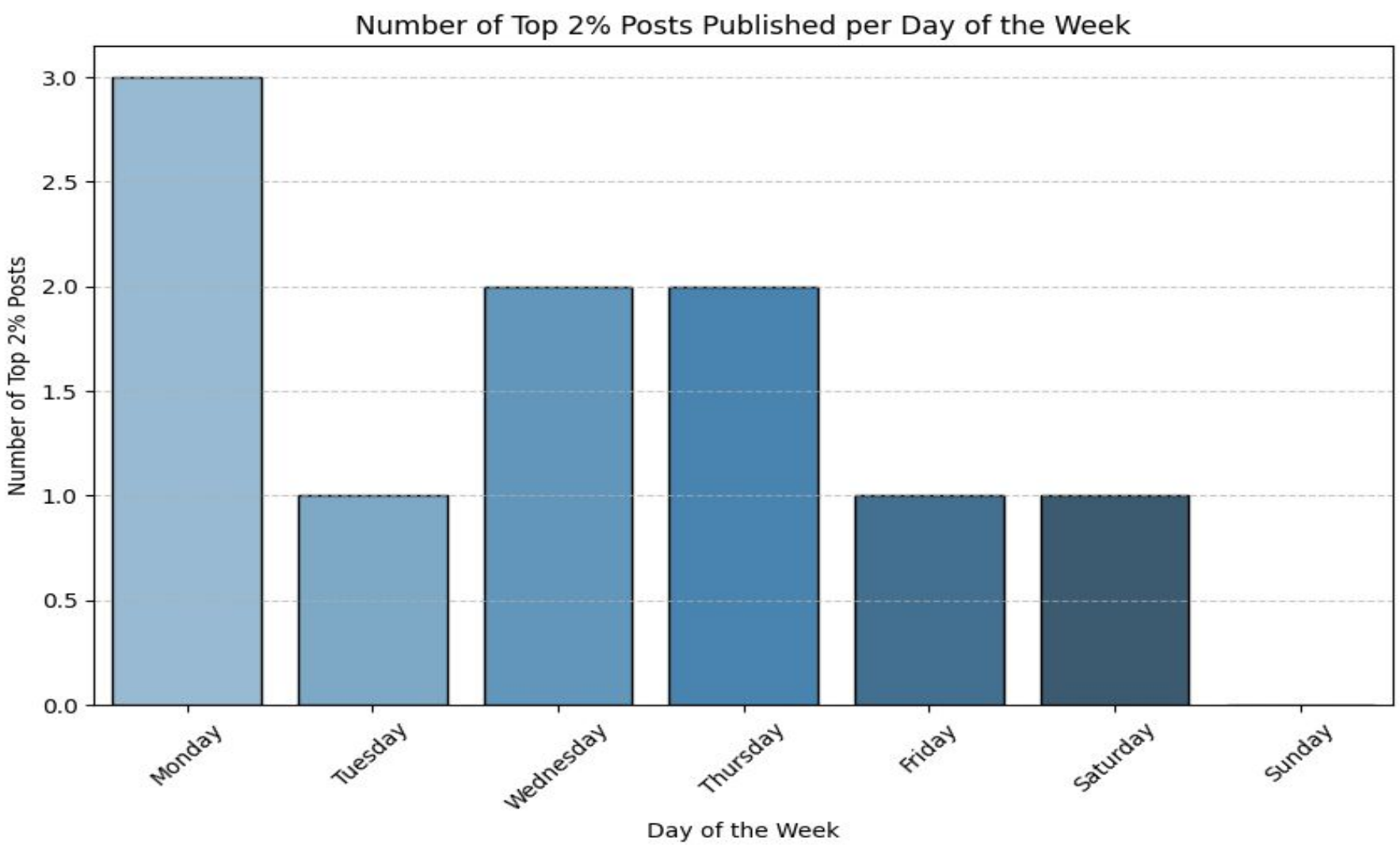
H3 Part 1 Analysis and Testing

```python
# Shapiro-Wilk Test: Checking normality of Total_Interactions
shapiro_stat, shapiro_p = shapiro(df['Total_Interactions'].dropna())

print(f"\nShapiro-Wilk Test Statistic: {shapiro_stat:.4f}, p-value: {shapiro_p:.4f}")

# Interpretation of Shapiro-Wilk Test
if shapiro_p < 0.05:
    print("❌ Data is NOT normally distributed (p < 0.05). Non-parametric tests should be used.")
else:
    print("✅ Data appears to be normally distributed (p ≥ 0.05). Parametric tests can be used.")

# Fisher's Exact Test: Checking if top 2% posts are mostly published on Thursday Afternoon
table_fisher = [[len(top_2_percent_thursday_afternoon), len(top_2_percent) - len(top_2_percent_thursday_afternoon)],
                [len(df[(df['time_of_day'] == 'Afternoon')]), len(df) - len(df[(df['time_of_day'] == 'Afternoon')])]]

odds_ratio, p_value_fisher = fisher_exact(table_fisher)

# Kruskal-Wallis Test: Comparing interactions across different times of the day
morning_interactions = df[df['time_of_day'] == 'Morning']['Total_Interactions']
noon_interactions = df[df['time_of_day'] == 'Noon']['Total_Interactions']
afternoon_interactions = df[df['time_of_day'] == 'Afternoon']['Total_Interactions']
evening_interactions = df[df['time_of_day'] == 'Evening']['Total_Interactions']
night_interactions = df[df['time_of_day'] == 'Night']['Total_Interactions']
```

First, the Shapiro-Wilk test was conducted to check the type of data distribution and the test result showed that the data is **not normally distributed (Non-parametric data)**

Therefore, Fisher's Exact Test was used to analyze the first part, whether the top 2% of posts were significantly associated with being published on Thursday afternoon. The result of Fisher's exact test determines whether the hypothesis is valid or not.

If p-Value < 0.05, the hypothesis is supported, otherwise, the hypothesis is not supported


Number of Top 2% Posts Published per Day of the Week
Number of Top 2% Posts
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day of the Week

H3 | Results of part 1

The attached graph confirms the result of **Fisher's exact test**, showing that the most posts are published on Monday, not Thursday.

```
# Conclusion
print("\nConclusion:")
if p_value_fisher < 0.05:
    print("✅ The top 2% of posts were disproportionately published on Thursday afternoon, supporting the hypothesis.")
else:
    print("❌ The top 2% of posts were NOT disproportionately published on Thursday afternoon.")

if most_interactive_time == 'Evening':
    print("✅ Users interact most frequently in the evening, supporting the hypothesis.")
else:
    print(f"❌ Users do NOT interact most frequently in the evening; the peak interaction time is in the {most_interactive_time}
```

- The result of Fisher's exact test was P-Value = **1.000** > 0.05

Therefore, we cannot confirm the alternative hypothesis, and thus the null hypothesis is accepted

H3 | Part 2

Users interact with posts most frequently in the evening." (interactions also include comments, shares and likes)

Since the data were not normally distributed, the **Kruskal-Wallis test** was used to analyze differences in interactions across different times of the day.

Additionally, the **Mann-Whitney U test** was performed as an extra verification to check if interactions in the evening were significantly different from other times.

Final Conclusion:

Both tests led to the same result: **The most frequent user interactions occurred in the Morning, not in the evening.**
Thus, the **null hypothesis is supported**, and the alternative hypothesis is rejected, especially since there was a significant difference between the afternoon and evening interaction periods.

H3| Results of part 2

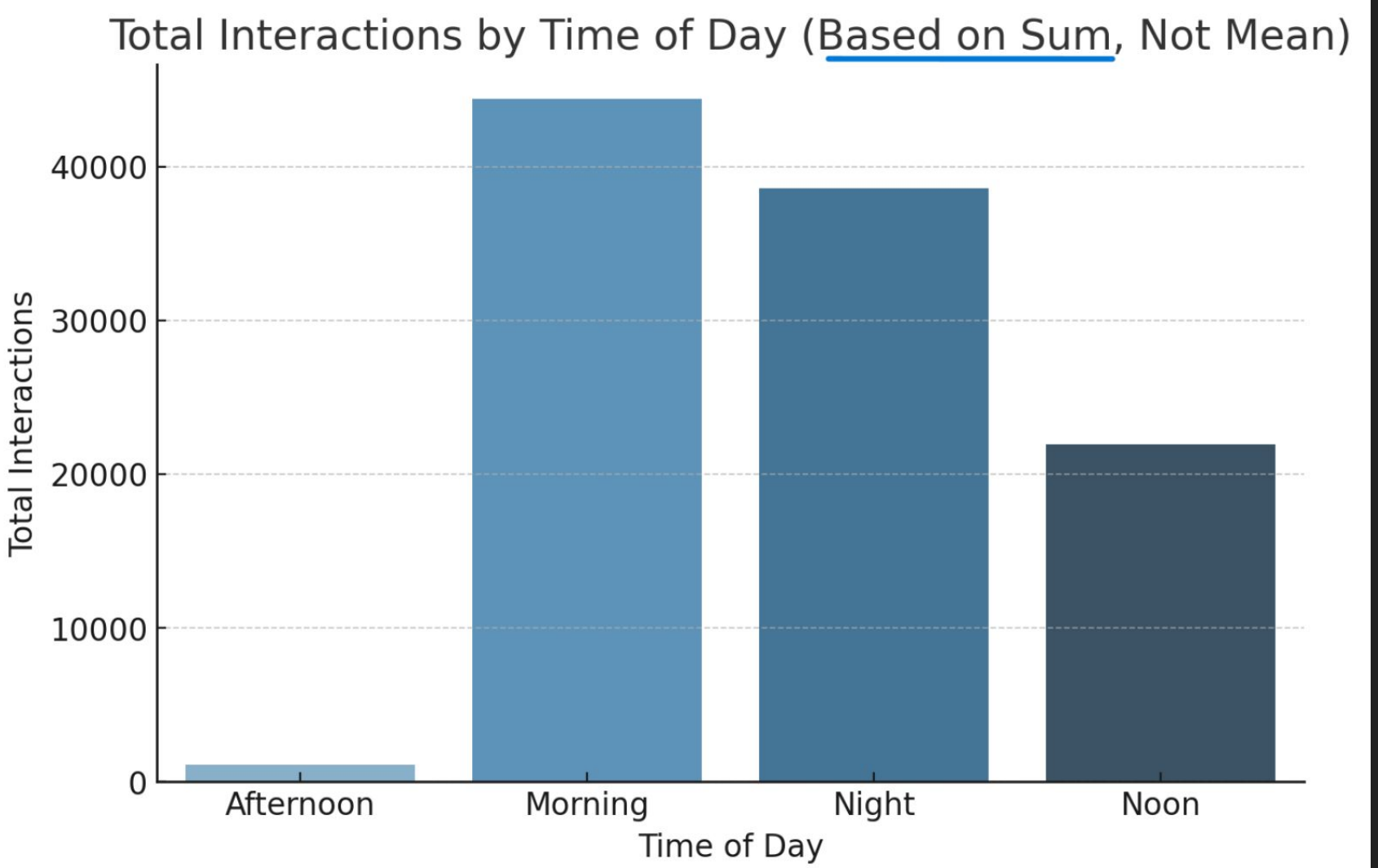
```
# Kruskal-Wallis Test: Comparing interactions across different times of the day
morning_interactions = df[df['time_of_day'] == 'Morning']['Total_Interactions']
noon_interactions = df[df['time_of_day'] == 'Noon']['Total_Interactions']
afternoon_interactions = df[df['time_of_day'] == 'Afternoon']['Total_Interactions']
evening_interactions = df[df['time_of_day'] == 'Evening']['Total_Interactions']
night_interactions = df[df['time_of_day'] == 'Night']['Total_Interactions']

# Cleaning data by dropping NaN values before Kruskal-Wallis Test
h_stat, p_value_kruskal = kruskal(
    morning_interactions.dropna(),
    noon_interactions.dropna(),
    afternoon_interactions.dropna(),
    evening_interactions.dropna(),
    night_interactions.dropna()
)

# Calculate the time of day with the highest average interactions
most_interactive_time_total = df.groupby('time_of_day')['Total_Interactions'].sum().idxmax()
```

The graph confirms the result of the **Mann-Whitney U** Test and the **Kruskal-Wallis Test.**

Kruskal-Wallis Test: **p**-Value = **0.0897 >** 0.05 → The null hypothesis is accepted.

Mann-Whitney U Test: **p**-Value = **0.4676 >** 0.05 → The null hypothesis is accepted.

Both tests led to the same result: The most frequent user interactions occurred in the Morning, not in the evening. Thus, the null hypothesis is supported.


Total Interactions by Time of Day (Based on Sum, Not Mean)
Total Interactions
40000
30000
20000
10000
0
Afternoon
Morning
Night
Noon
Time of Day

H3 | WHY?

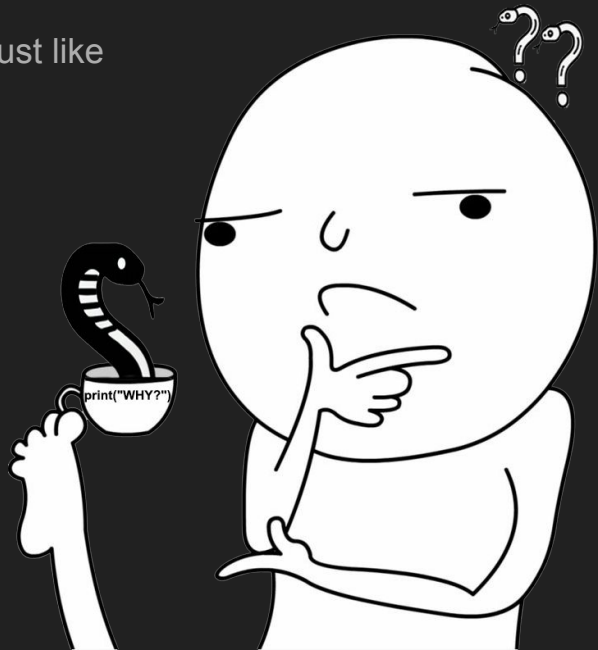
Why were these tests (Kruskal-Wallis test, Mann-Whitney U test and Fisher exact test) chosen?

- Suitable for **non-normally distributed data**
- Effective for **small sample sizes**
- **Not affected by outliers**
- Based on the **median** instead of the mean, making them more reliable in this case

We took into account that all the previous tests were affected by missing values just like

Mr. Shapiro-Wilk, who gets really annoyed by NaNs. So, to keep them all happy

we had to clean up the missing data before running the tests



Simulation - Maximierung der Likes

Aufgabe:

‘Use the available data to apply an accurate simulation. Describe which properties a post must have in order to maximize its likes.’

https://cdn-icons-png.flaticon.com/512/4658/4658740.png

Durchführung I

1. Datenbereinigung und Auswahl der Features (post_time, mentions, hashtags, day_of_week und caption_length)

```
# Wichtige Features für die Analyse auswählen
features = ["post_time", "mentions", "hashtags", "day_of_week", "caption_length"]
target = "likes"
```

Durchführung II

2. Modellierung mit Polynomialer Regression

3. Bewertung der Modellgüte

```
# Polynomial Regression Modell erstellen
X = df[["post_time"]]
y = df["likes"]

poly = PolynomialFeatures(degree=3)
X_poly = poly.fit_transform(X)

model = LinearRegression()
model.fit(X_poly, y)
y_pred = model.predict(X_poly)

# R²-Score berechnen
r2 = r2_score(y, y_pred)
print(f"Polynomial Regression R² Score: {r2:.3f}")
```

Durchführung IV

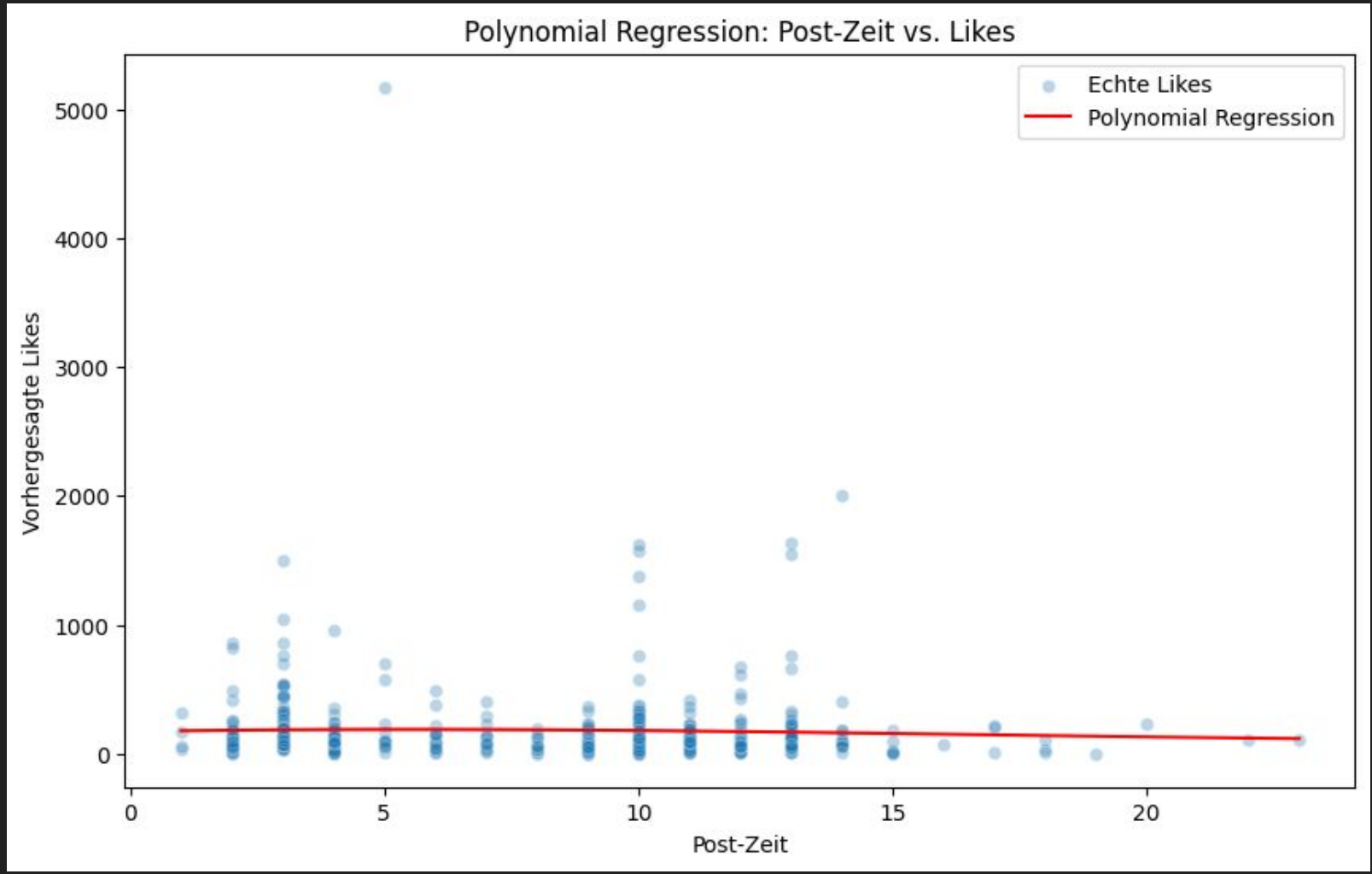
4. Analyse der wichtigsten Einflussfaktoren

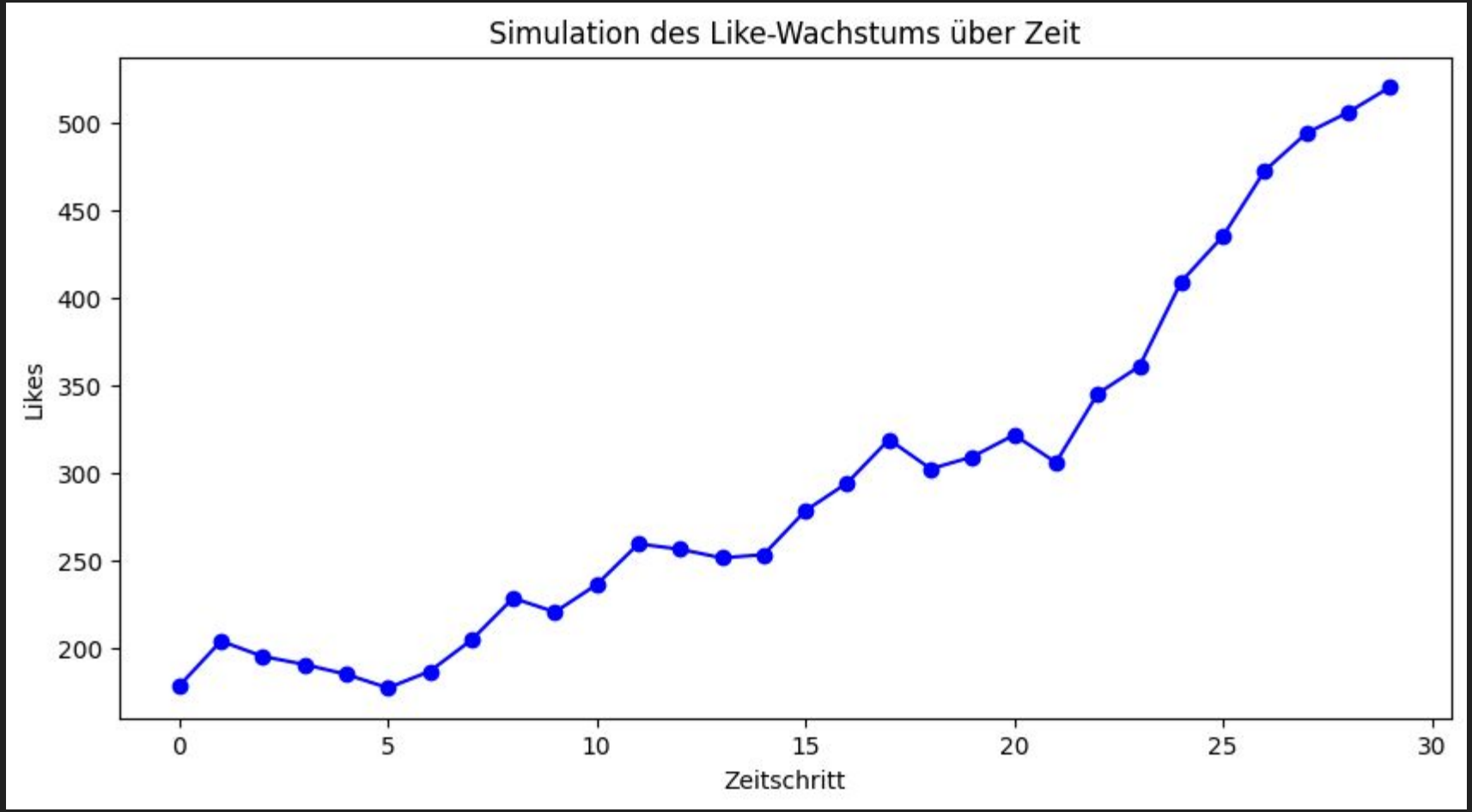
```
# Einflussreiche Features analysieren
coefficients = pd.Series(model.coef_, index=poly.get_feature_names_out(["post_time"]))
print("Top-Einflussfaktoren im Modell:")
print(coefficients.sort_values(ascending=False).head())
```

Durchführung III

5. Simulation des Engagement-Wachstums

```python
#SIMULATION: Engagement-Wachstum über Zeit
def simulate_engagement(initial_likes, growth_rate, time_steps, randomness=0.1):
    """Simuliert das Wachstum von Likes über die Zeit."""
    likes_over_time = [initial_likes]

    for t in range(1, time_steps):
        random_factor = np.random.uniform(1 - randomness, 1 + randomness)  # Zufälligkeit hinzufügen
        new_likes = likes_over_time[-1] * growth_rate * random_factor
        likes_over_time.append(new_likes)

    return likes_over_time

# Beispiel-Simulation für einen typischen Post
initial_likes = df["likes"].mean()  # Durchschnittliche Likes als Ausgangswert
growth_rate = 1.05  # Wachstumsrate (5% pro Zeiteinheit)
time_steps = 30  # 30 Zeitschritte simulieren

simulated_likes = simulate_engagement(initial_likes, growth_rate, time_steps)
```


Polynomial Regression: Post-Zeit vs. Likes
Echte Likes
Polynomial Regression
Vorhergesagte Likes
Post-Zeit
0
1000
2000
3000
4000
5000
0
5
10
15
20


Simulation des Like-Wachstums über Zeit
Likes
Zeitschritt
500
450
400
350
300
250
200
0
5
10
15
20
25
30

Simulation - Fazit

1. Der Zeitpunkt der Veröffentlichung scheint vorerst nur einen minimalen Einfluss zu haben,

   ABER:

- Es können bestimmte Zeitfenster bestehen, in denen mehr Likes erhalten werden können (Simulation, Diagramm 2)
- Optimaler Zeitpunkt ist jedoch nicht direkt linear, sondern kann durch weitere Faktoren beeinflusst werden (z.B. Algorithmus, Zielgruppenpräferenzen, Content…)
- Zeitfenster ist nicht allein entscheidender Faktor

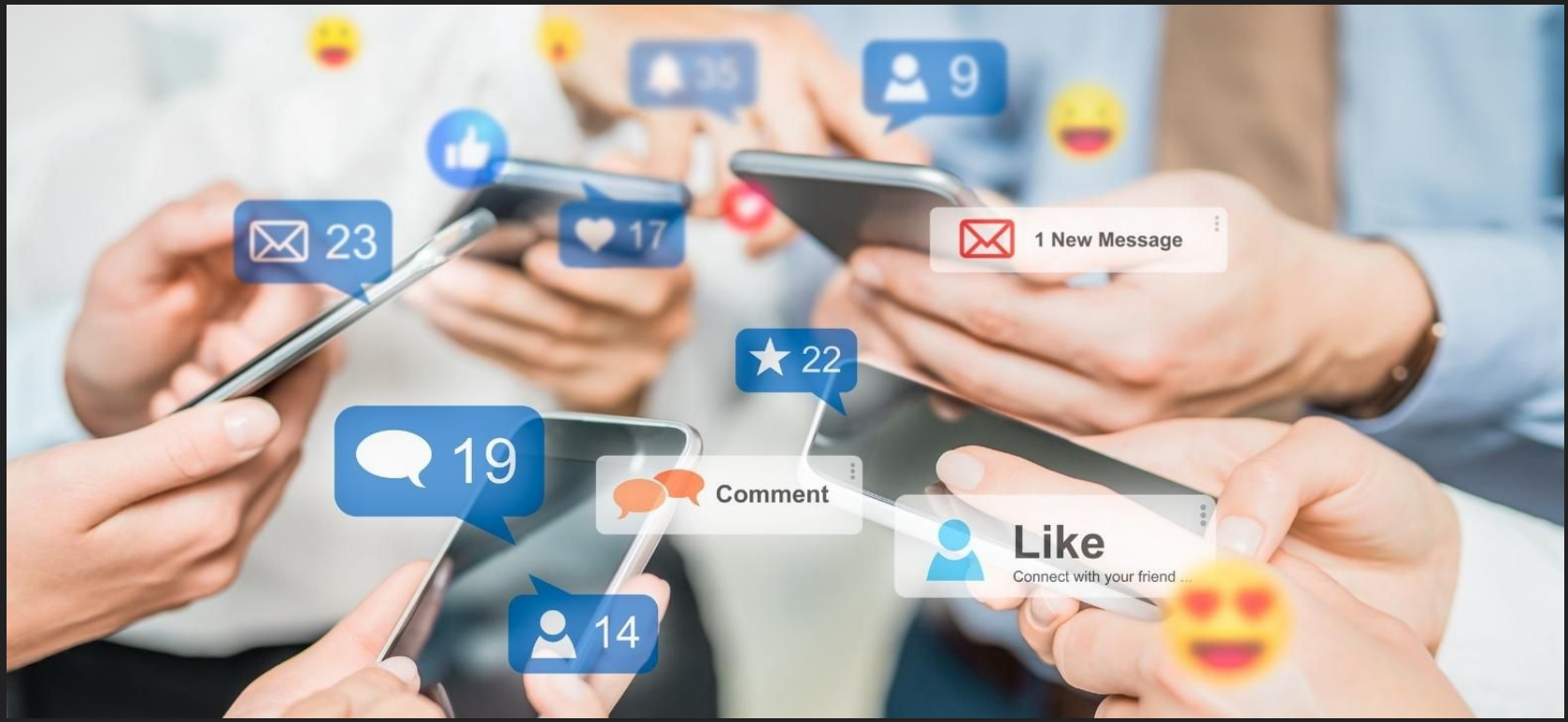
2. Die Simulation bestätigt langfristiges Wachstum durch Engagement

- Kontinuierlicher Anstieg der Likes über die Zeit ist möglich, wenn das Engagement hoch bleibt
- Der Zufallsfaktor *simulate_engagement()* zeigt, dass auch virale Effekte und Algorithmen einen Einfluss haben können.

3. Mentions und Hashtags haben potenziellen Einfluss

Vielen Dank für Ihre Aufmerksamkeit!

Gibt es Fragen/Anmerkungen?



https://ethz.ch/staffnet/de/service/kommunikation/social-media/_jcr_content/par/fullwidthimage/image.imageformat.1286.1004611511.jpg