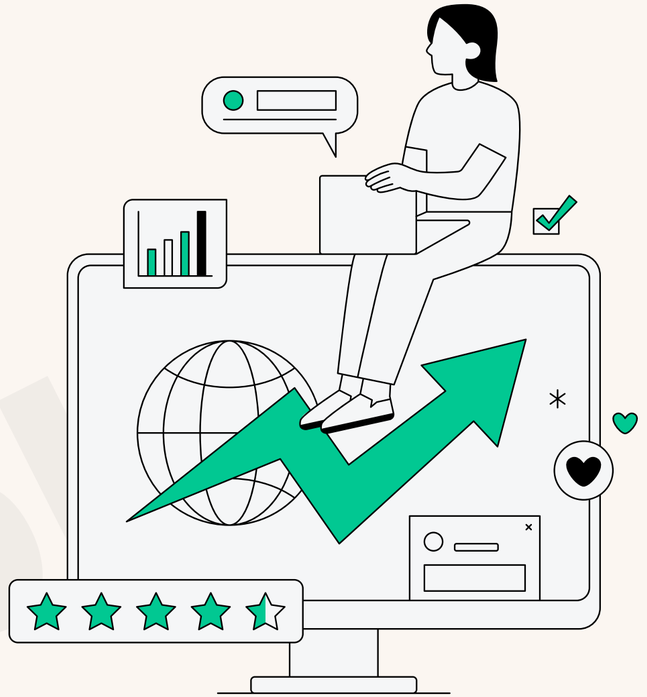
Presented by Mehdi Bohloul

Instagram Engagement Analysis



https://github.com/eddie-bhl

https://linkedin.com/in/mehdi-bohloul

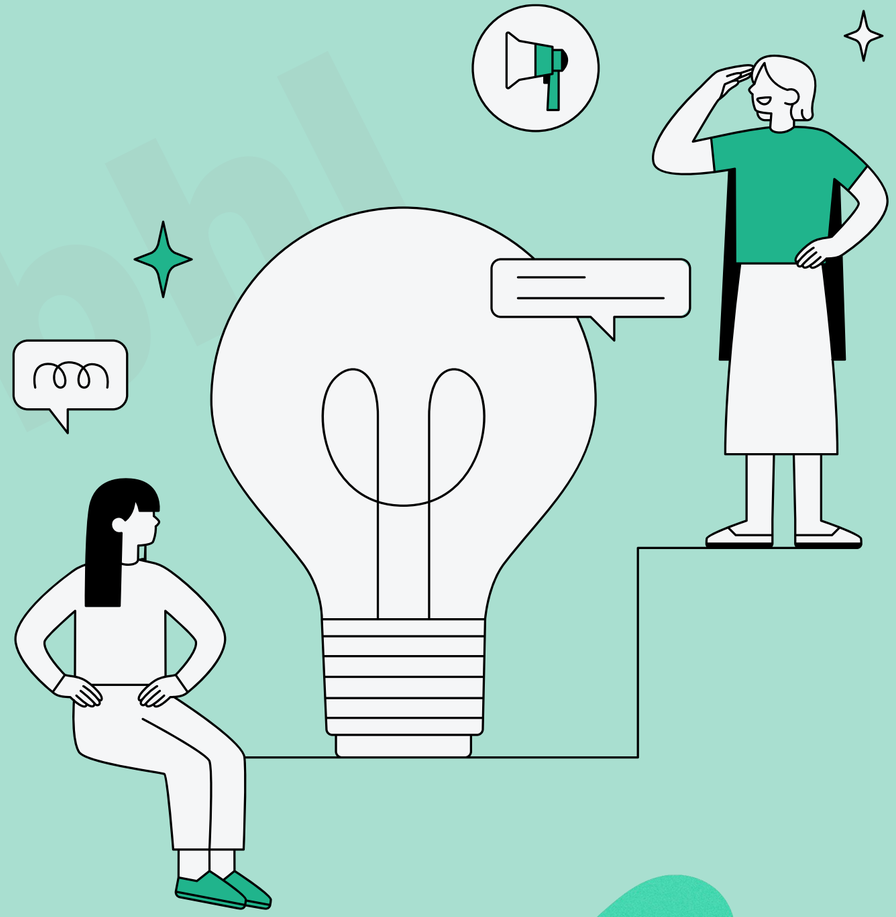
Table of Contents

Introduction

Description

- This analysis examines **post metrics** of an Instagram profile, and their influence on the overall **engagement rate** for each post within different categories.

- By identifying which factors contribute most significantly to audience interaction, we deliver actionable insights that enable social media strategists to refine content strategy and improve overall profile performance.



Introduction

- **Objectives**

By analyzing the Instagram data , we are trying to find out:

- What are the key drivers for engagement rate (what factors contribute most) ?
- Is the relationship linear or non-linear (like classification) ?
- Which categories had the highest engagement rates?
- Did time in the year or posts per week influence engagement rate?

Dataset & Tools

Dataset

- **Instagram Analytics Dataset:** Includes **30,000 Instagram posts** with analytics designed to simulate authentic Instagram Insights in a one-year period from Nov 2024 to Nov 2025. It provides performance metrics like likes, comments, shares, saves, reach, impressions, and engagement rate.

- Link: https://www.kaggle.com/datasets/kundanbedmutha/instagram-analytics-dataset/

Tools

- **Statistical Modelling:** Pandas, NumPy, SciPy

- **Data Visualization:** Seaborn, Matplotlib, Plotly Dash

- **Language & IDE:** Python - VS Code via Jupyter Notebook

Dataset Description

- **Post_ID**
  - A unique identifier for each Instagram post.
- **Upload_Date**
  - The date the post was uploaded.
- **Media_Type**
  - Type of content: Photo, Video, Reel, or Carousel.
- **Likes**
  - Total number of likes the post received.
- **Comments**
  - Number of comments on the post.
- **Shares**
  - How many times users shared the post.
- **Saves**
  - Number of saves
- **Reach**
  - Unique accounts that viewed the post.
- **Impressions**
  - Total views from all sources (can exceed reach).
- **Caption_Length**
  - Length of the caption in characters.
- **Hashtags_Count**
  - Number of hashtags used in the post.
- **Followers_Gained**
  - Number of followers gained directly because of this post.
- **Traffic_Source**
  - Origin of the viewers: Explore, Home Feed, Hashtags, Profile, Reels Feed, or External.
- **Engagement_Rate**
  - Engagement percentage relative to impressions
  - **Formula = (likes+comments+shares+saves) / impressions**

Methodology used in the analysis

1. Data Loading
- Importing libraries
- loading csv file

2. Data Preprocessing
- Data cleaning
- Feature engineering
- Fixing outliers

3. Exploratory Data Analysis (EDA)
- Univariate Analysis
- Multivariate Analysis
- Correlation Matrix

4. Key Insights
- Conclusions and insights driven from the data

5. Future Recommendation
- Recommended future work to increase accuracy of the results

Data Preprocessing

- Loading the initial csv file into VS Code
- Check for redundancies

In our dataset, there is no null values, otherwise we would have to remove or impute them based on importance and data types.

```
# Load Dataset from a CSV file
path = "C:\\Users\\mahdi\\Desktop\\M2MTECH\\Capstone 1\\Instagram_Analytics.csv"
df = pd.read_csv(path)

# Display first 5 rows
display(df.head())

# Check data types and missing values
df.info()
```

	post_id	upload_date	media_type	likes
0	IG0000001	2024-11-30 09:25:22.954916	Reel	31627
1	IG0000002	2025-08-15 09:25:22.954916	Photo	63206
2	IG0000003	2025-09-11 09:25:22.954916	Reel	94373
3	IG0000004	2025-09-18 09:25:22.954916	Reel	172053
4	IG0000005	2025-03-21 09:25:22.954916	Video	99646

5 rows x 15 cols 10 per page Page 1 of 1

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29999 entries, 0 to 29998
Data columns (total 15 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   post_id           29999 non-null  object
 1   upload_date       29999 non-null  object
 2   media_type        29999 non-null  object
 3   likes             29999 non-null  int64
 4   comments          29999 non-null  int64
 5   shares            29999 non-null  int64
 6   saves             29999 non-null  int64
 7   reach             29999 non-null  int64
 8   impressions       29999 non-null  int64
 9   caption_length    29999 non-null  int64
 10  hashtags_count    29999 non-null  int64
 11  followers_gained  29999 non-null  int64
 12  traffic_source    29999 non-null  object
 13  engagement_rate   29999 non-null  float64
 14  content_category  29999 non-null  object
dtypes: float64(1), int64(9), object(5)
memory usage: 3.4+ MB
```

Data Preprocessing

Converting data types

```
# Convert appropriate columns to 'category' dtype
categorical_columns = ['media_type', 'traffic_source', 'content_category']
for col in categorical_columns:
    df[col] = df[col].astype('category')

# Convert date column to datetime dtype
df['upload_date'] = pd.to_datetime(df['upload_date'], format='%Y-%m-%d %H:%M:%S.%f')
# Also we will remove seconds and miliseconds in datetime for simplicity and efficiency.
df['upload_date'] = df['upload_date'].dt.floor('min')
```

```
print(df.nunique().sort_values())
```

✓ 0.1s

```
media_type              4
traffic_source          6
content_category       10
hashtags_count         31
upload_date           366
followers_gained     1001
caption_length       2201
engagement_rate      4942
shares               4991
comments             9495
saves               12943
likes               27858
impressions         29781
reach               29787
post_id             29999
dtype: int64
```

No duplicates in post IDs [Primary Key]

# likes	# comments	# shares	# saves	# reach	...
156126	9259	668	426	3228	
121134	8597	3004	2164	10321	
185597	8539	2025	2581	6398	
167914	8084	2445	3215	20106	
192824	8704	731	6774	2713	
164260	2640	3108	3380	9018	

- **Dropping ‘post ID’ column since it doesn’t contribute to analysis**
- **Dropping ‘reach’ feature because it is not consistent with its definition.**

```
df_new = df.drop(columns=['post_id', 'reach'])
```

Data Preprocessing

Checking Balance

```
counts = {col: df[col].value_counts() for col in ['media_type','traffic_source','content_category']}
print(counts['media_type'])
print(counts['traffic_source'])
print(counts['content_category'])
```

✓ 0.0s

```
media_type
Carousel    7526
Video       7523
Reel        7515
Photo       7435
Name: count, dtype: int64
traffic_source
Home Feed     5069
Hashtags      5063
Reels Feed    5026
External      5005
Profile       4962
Explore       4874
Name: count, dtype: int64
content_category
Photography    3035
Fashion        3034
Technology     3025
Lifestyle      3017
Food           3010
Fitness        3004
Music          3003
Travel         2968
Beauty         2953
Comedy         2950
Name: count, dtype: int64
```

Categories are highly balanced.

Data Preprocessing

Feature Engineering

```
df_clean['weekday'] = df_clean['upload_date'].dt.day_name()
df_clean['week'] = df_clean['upload_date'].dt.isocalendar().week
df_clean['weekday'] = df_clean['weekday'].astype('category')
df_clean['week'] = df_clean['week'].astype('int64')
```

In reality, engagement metrics do not carry the same weights. For example, a comment is much more impactful than a like, and a share is even more impactful than comments.

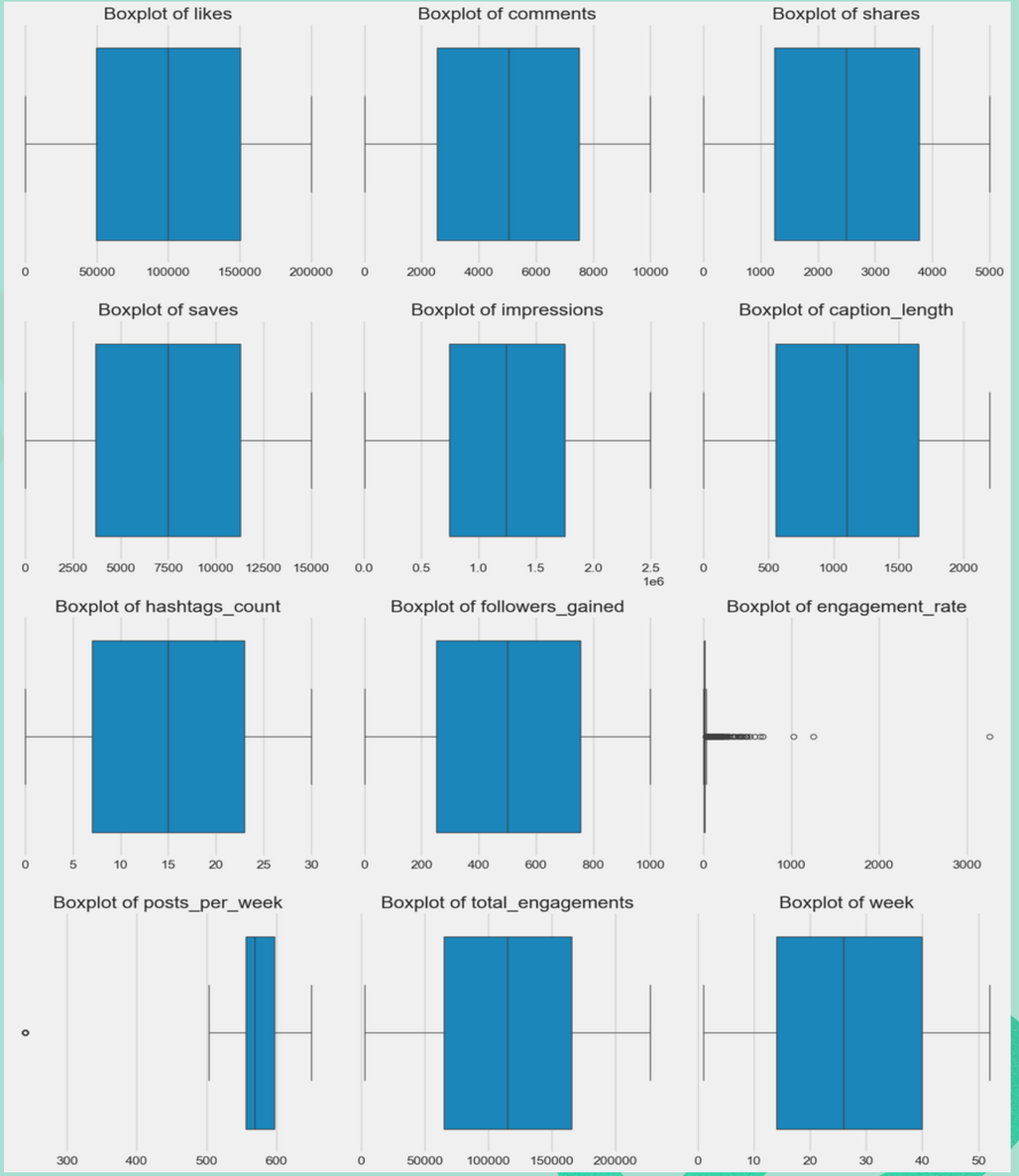
For EDA stage, it is not necessary to apply the weighting, since in a linear correlation analysis, the weights of the features are not important.

```
# Adding Post Per Week feature as an additional indicator
df_new['posts_per_week'] = df_new.groupby(df_new['upload_date'].dt.to_period('W'))['upload_date'].transform('count')
# Adding Total Engagements feature which takes into account all forms of engagement
df_new['total_engagements'] = df_new['likes'] + df_new['comments'] + df_new['shares'] + df_new['saves'] + df_new['followers_gained']
```

Exploratory Data Analysis (EDA)

- All of the features except **engagement_rate** and **Posts_per_week** are normally distributed.
- **Engagement_rate** is our target value and should be handled carefully.
- To take an extra step in EDA, we can create a new dataset **without outliers** and compare the results with the normal dataset.

Checking Outliers for numeric values



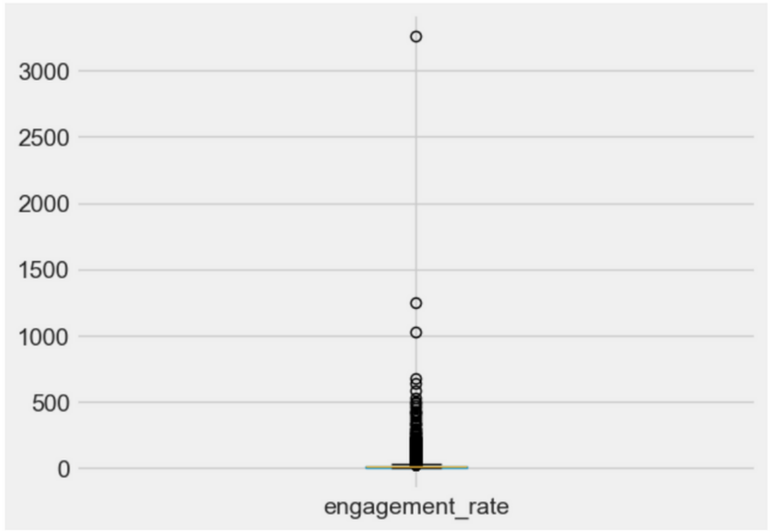
Exploratory Data Analysis (EDA)

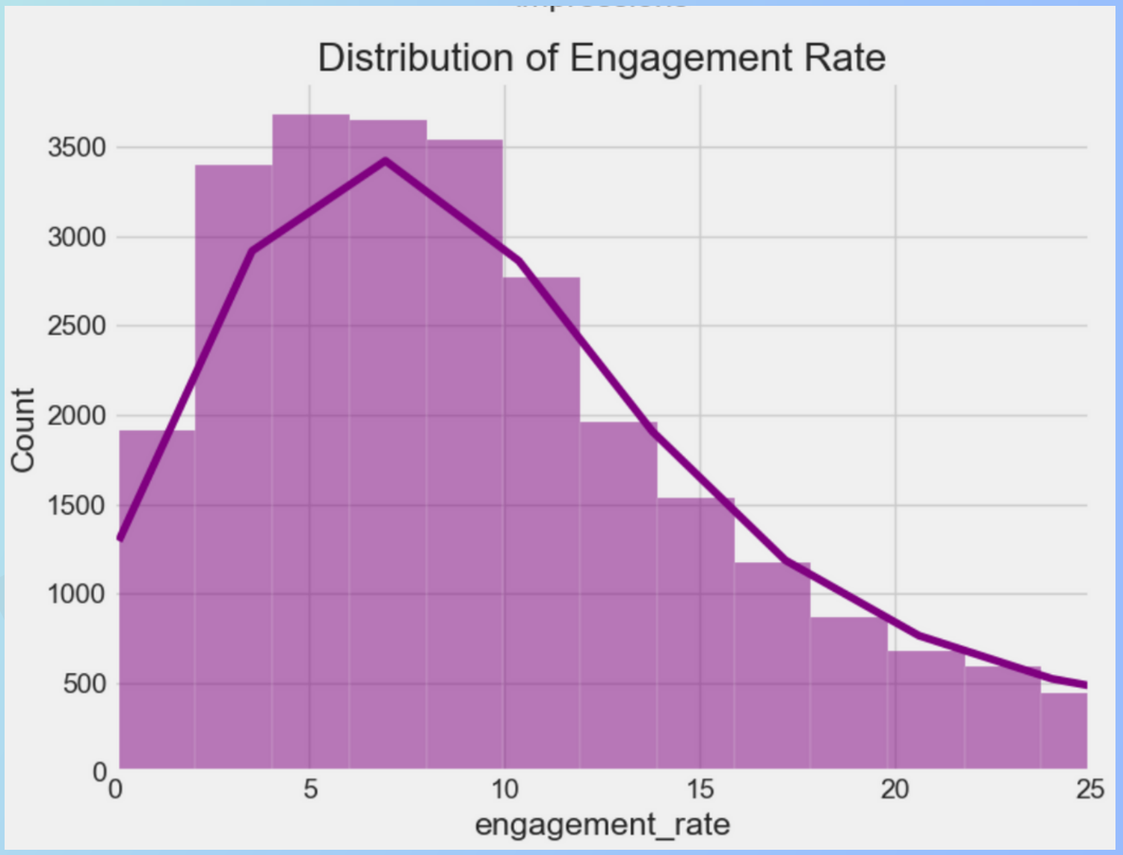
We do not need to do anything on **hashtag_count** or **posts_per_week** at this stage. Normalization and standardization is necessary when performing predictive or classification analysis.

Exploratory Data Analysis (EDA)

```
df_new[['engagement_rate']].boxplot(figsize=(15,10), rot=0, grid=True, ax=plt.gca())
```

```
<Axes: >
```





When analysing the target variable, we have to be very careful. We shouldn't normally drop any feature from the target value unless we are philosophically sure about it. In this case, we can see that we have three distinguishable outliers in the **engagement_rate** column. Therefore we can remove the entire rows safely. Furthermore, although the histogram of *engagement_rate* shows skewness, we do not need to perform any transformation such as logarithmic transformation at this stage.

Exploratory Data Analysis (EDA)

After removing top three outliers

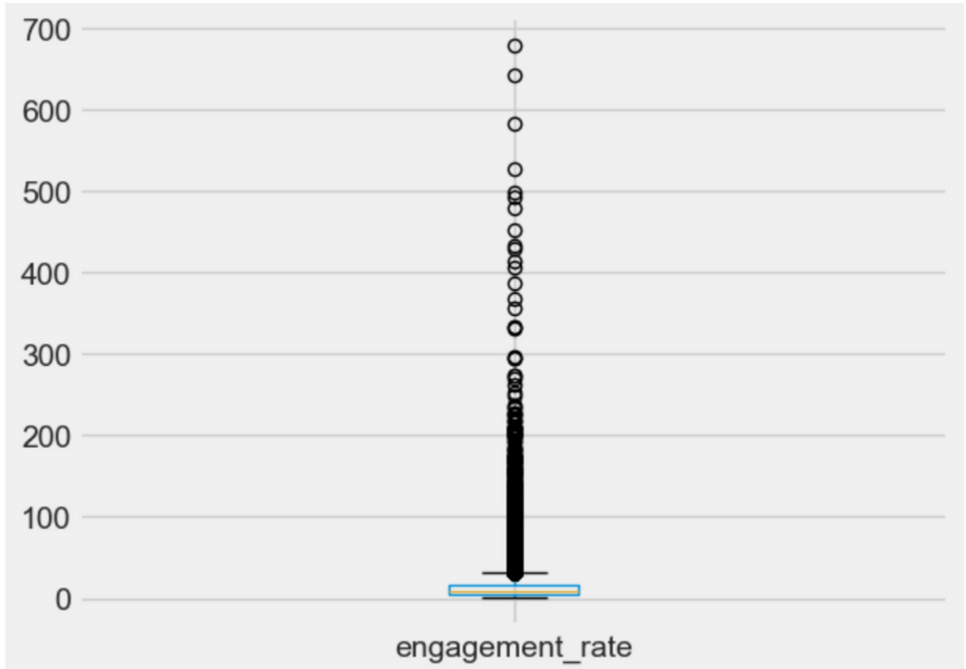
```
df_clean = []
df_clean = df_new[(df_new['engagement_rate'] < 900)]
```

```
df_clean.shape
```

```
(29996, 13)
```

```
df_clean[['engagement_rate']].boxplot(figsize=(15,10), grid=True, ax=plt.gca())
```

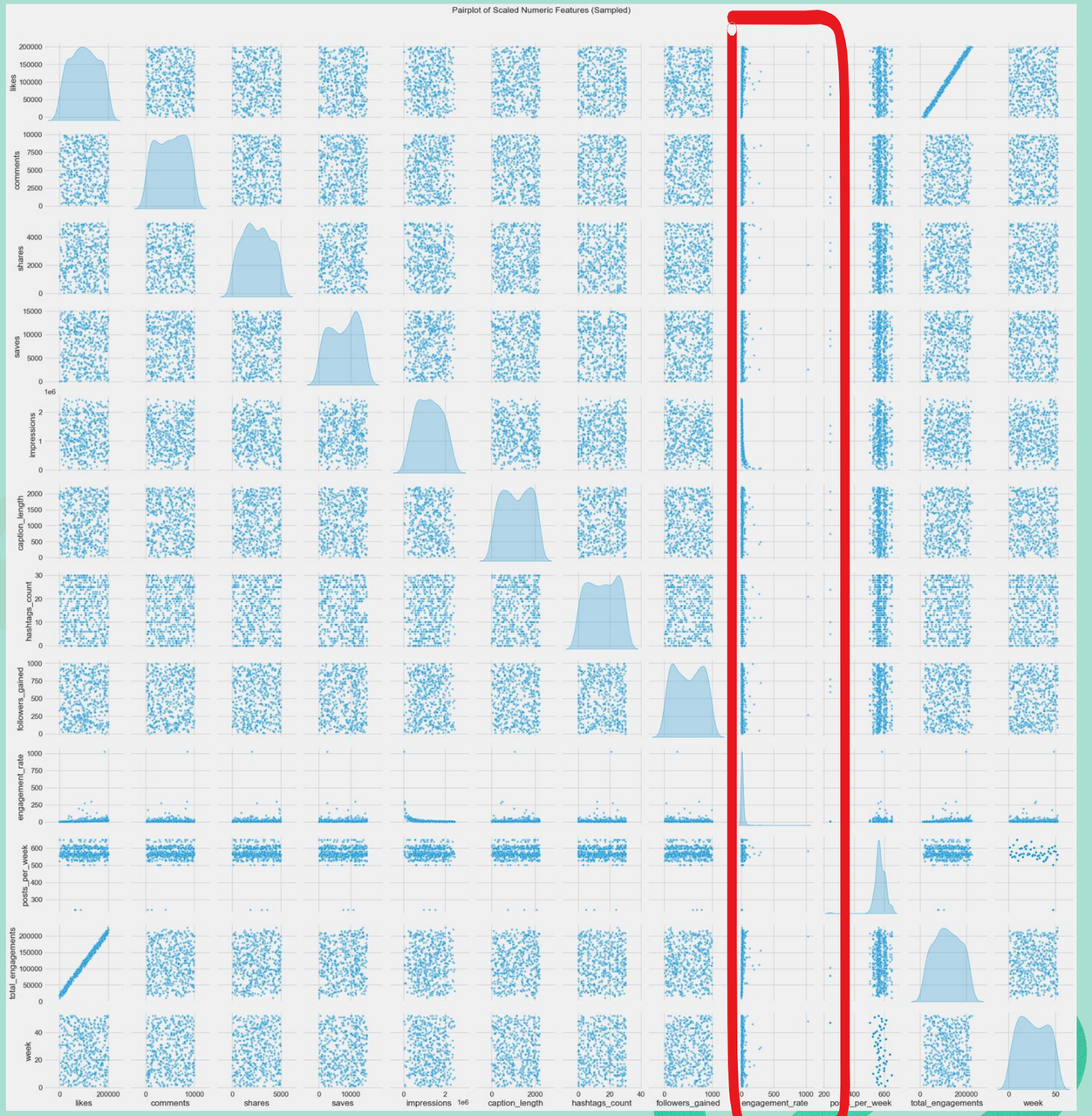
```
<Axes: >
```



Exploratory Data Analysis (EDA)

- It can be seen that most of features do not affect *engagement_rate* significantly except a few.

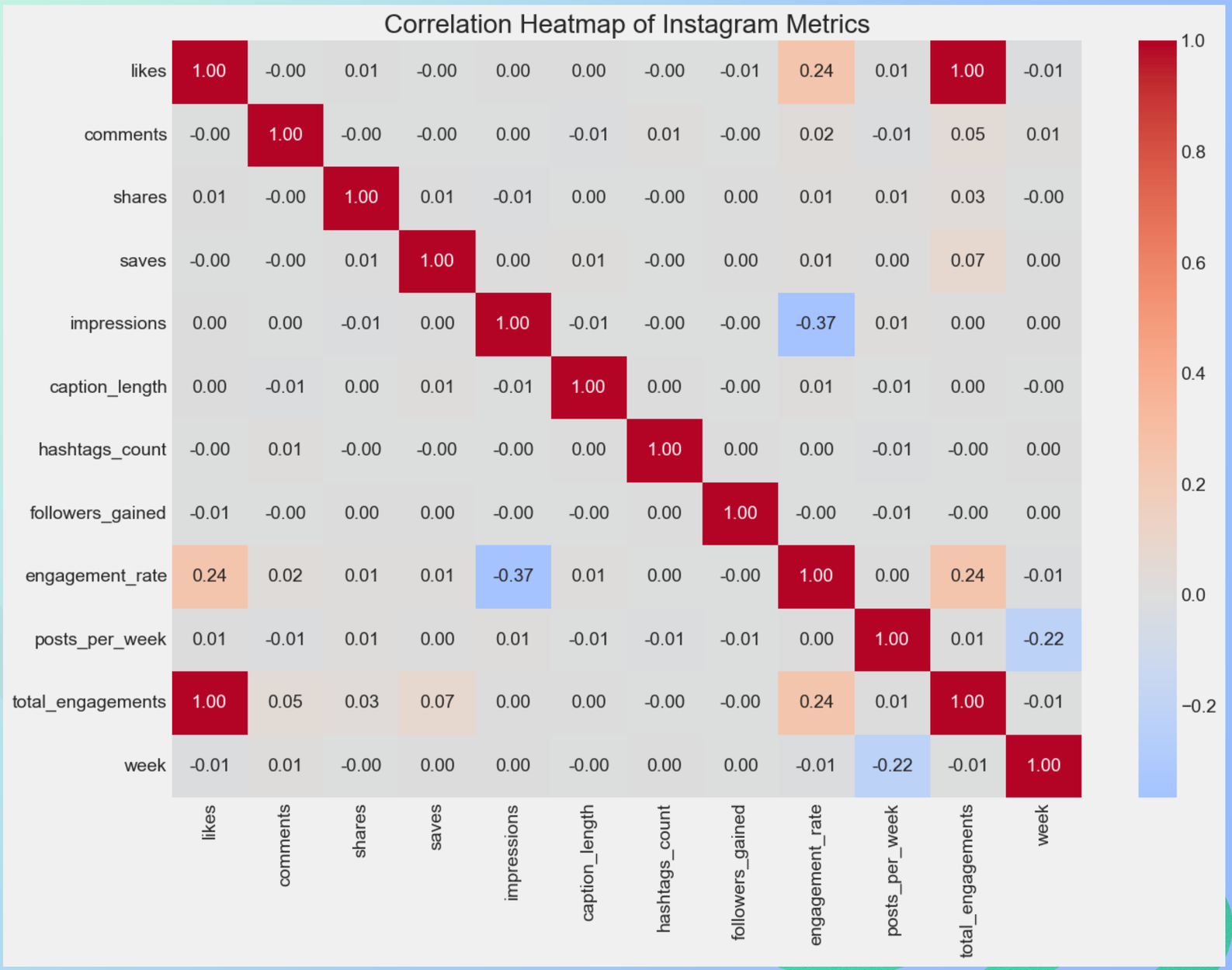
- The correlation between the **number of likes** and **total engagements** is highly linear meaning that *likes feature* has the most effect on *engagement_rate*.
- The correlation between **engagement_rate** and **impressions**, is logarithmically negative.



Exploratory Data Analysis (EDA)

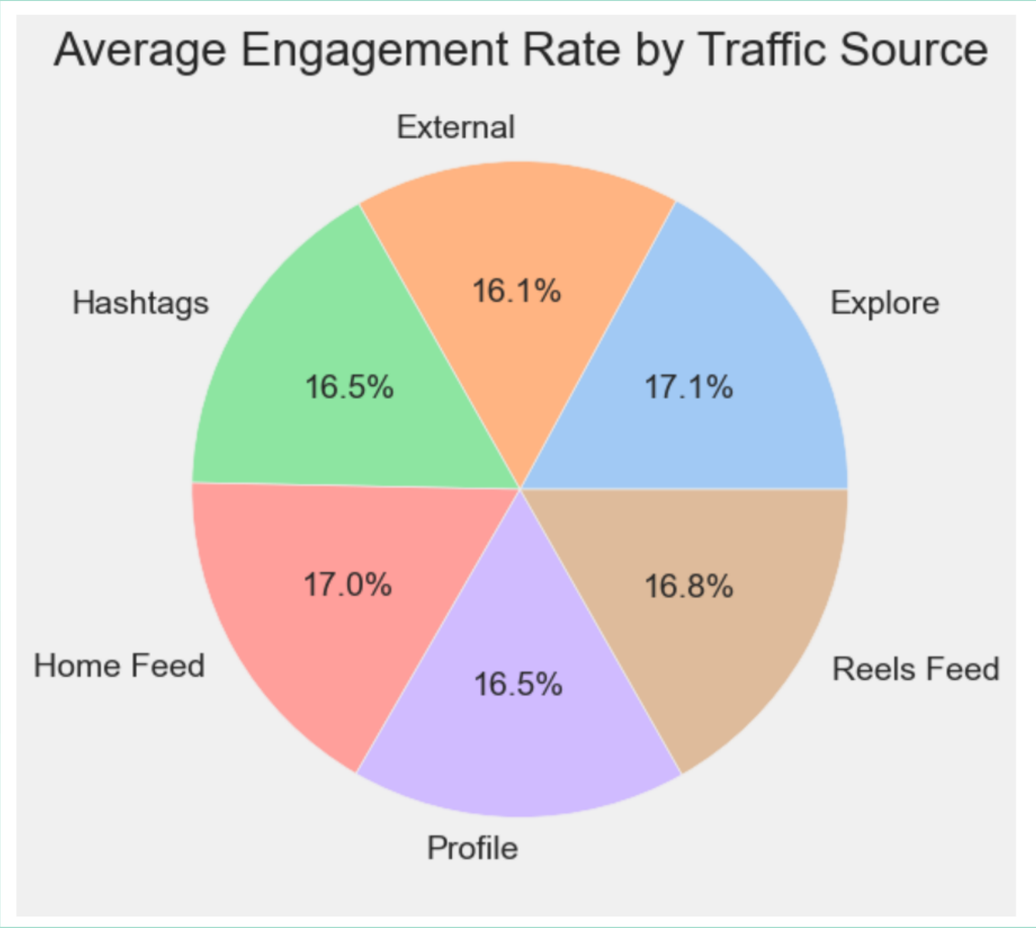
- Since the correlation between features is very weak, to improve accuracy we can:

1. Perform **tree-based regression**
2. Convert the *engagement_rate* numeric value to **categorical** and make it a **classification problem.**



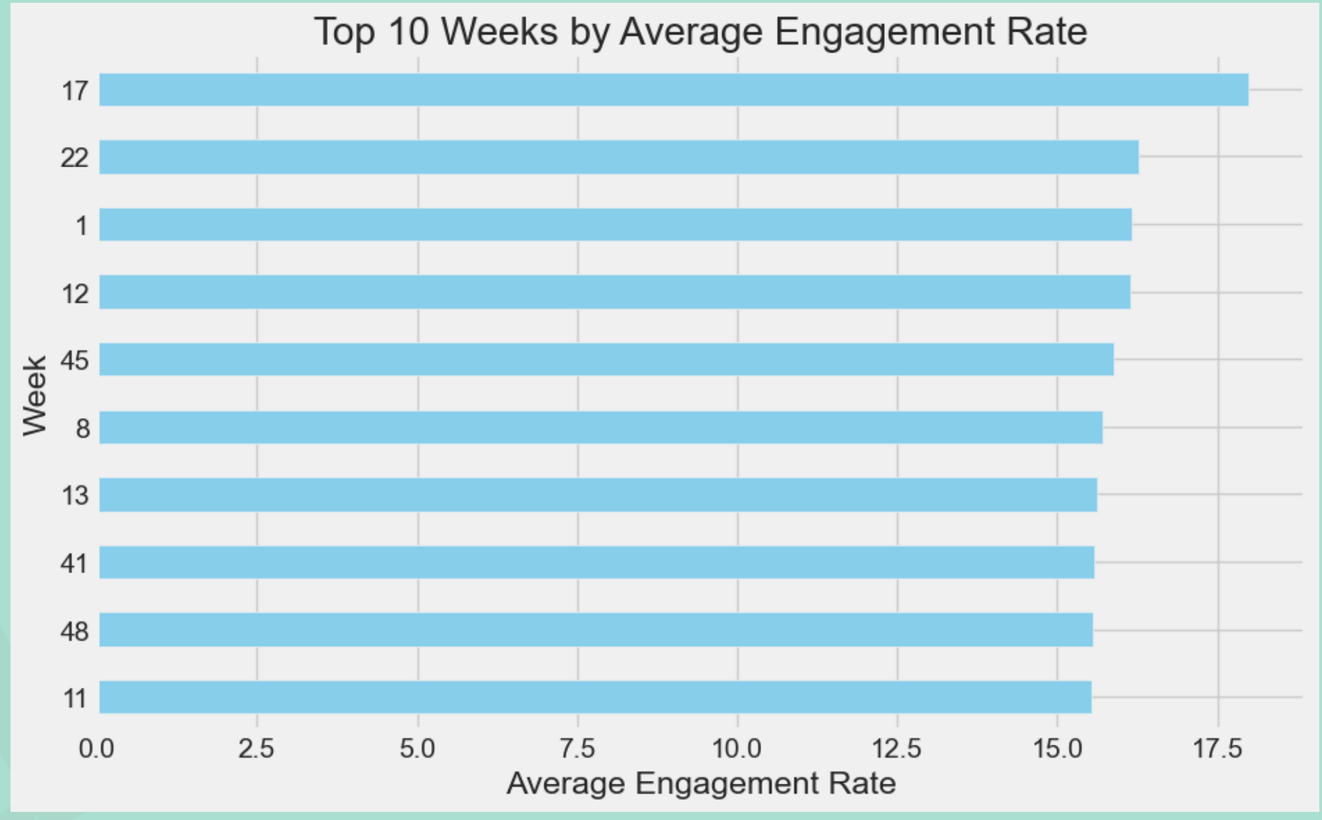
Exploratory Data Analysis (EDA)

- The difference among traffic destinations is insignificant.
- We can say engagement rate is not heavily influenced by the type of traffic source.




```
# Aggregate numerical values
agg = df_new.groupby('traffic_source')['engagement_rate'].mean()

# Pie chart
plt.figure(figsize=(6,6))
plt.pie(agg, labels=agg.index, autopct='%1.1f%%', colors=sns.color_palette('pastel'))
plt.title('Average Engagement Rate by Traffic Source')
plt.show()
```

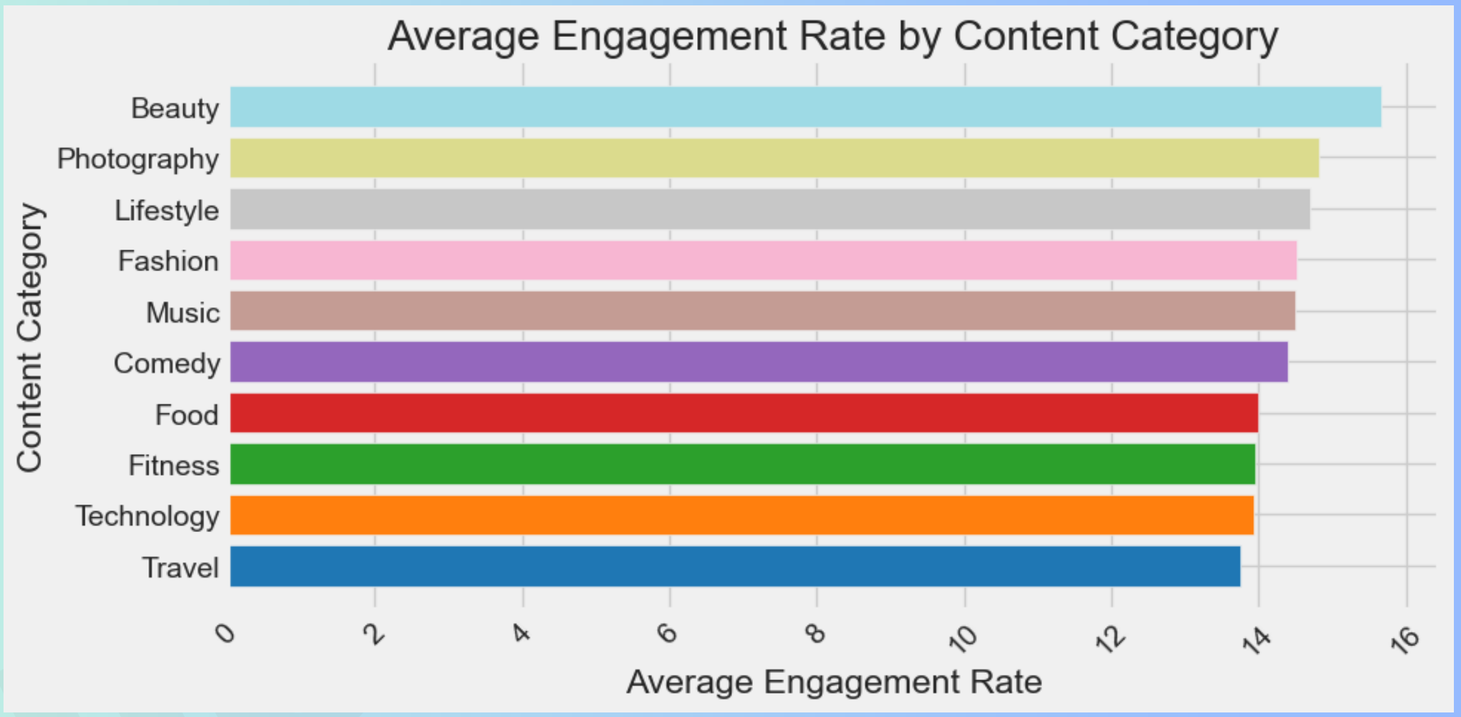
Exploratory Data Analysis (EDA)



Week **number 17** has noticeable higher engagement rate compared to other weeks. It might be related to **special events or occurrences**. We should check the calendar.

```
# We can see that weekdays and trrafic source have insignificant impact on impressions and engagement.
# For the week number, we will plot the top 10 weeks with highest engagement_rate to see if we can find valuable insights.

# Group by week and calculate mean engagement rate
week_summary = df_new.groupby('week')['engagement_rate'].mean()

# Get top 10 weeks by engagement rate
top10_weeks = week_summary.sort_values(ascending=False).head(10)

# Plot horizontal bar chart
top10_weeks.plot(kind='barh', figsize=(10,6), color='skyblue')
plt.xlabel('Average Engagement Rate')
plt.ylabel('Week')
plt.title('Top 10 Weeks by Average Engagement Rate')
plt.gca().invert_yaxis()  # Optional: highest at the top
plt.show()
```

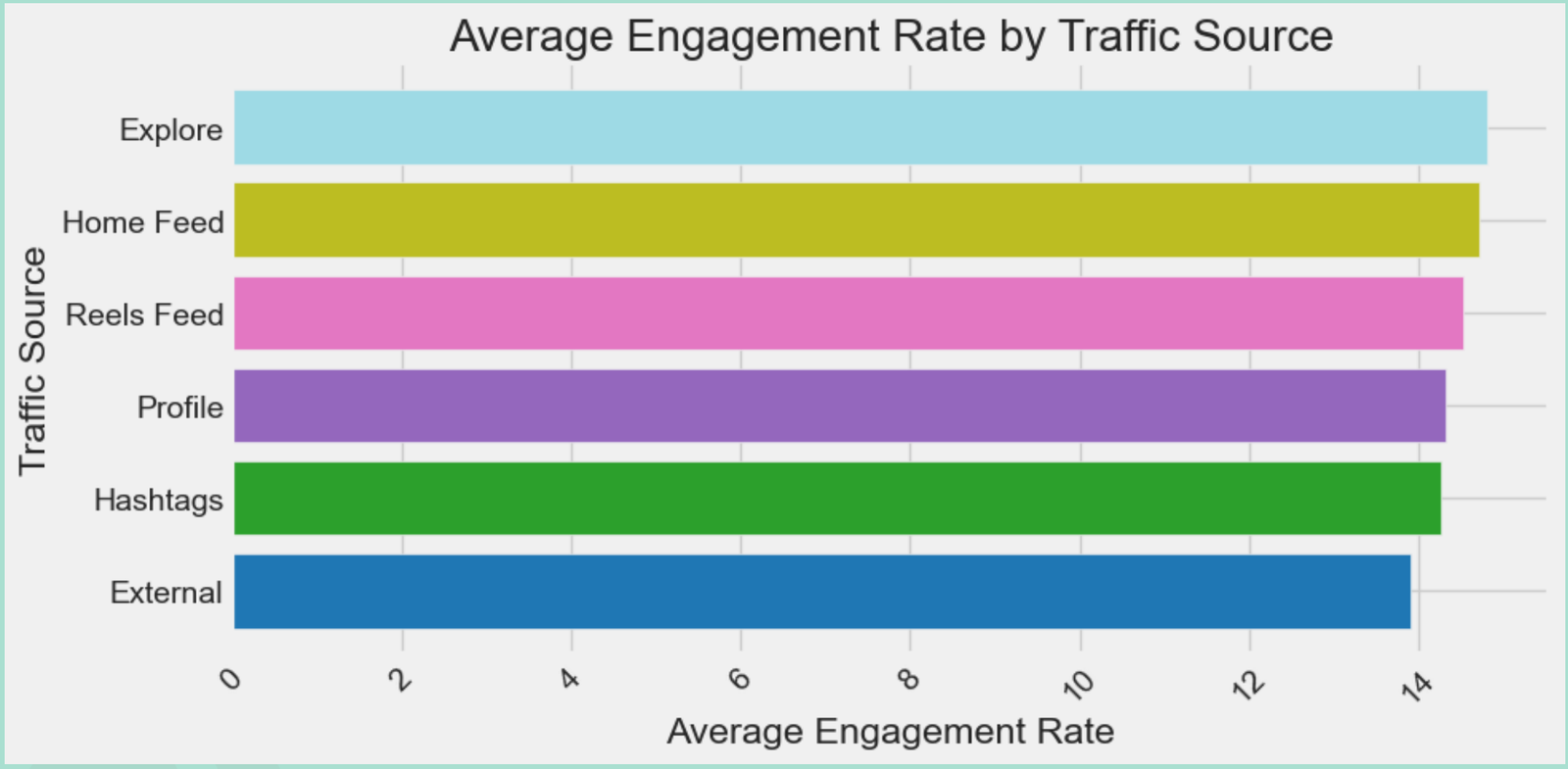
Exploratory Data Analysis (EDA)



Categories **Beauty, Photography** and **Lifestyle,** rank the top 3 highest categories with the highest engagement rate.

```
# Sum + sort categories
top_categories = (
    df_new.groupby('content_category')['engagement_rate']
      .mean()
      .sort_values(ascending=True)
)

# Generate a distinct color for each bar
colors = plt.cm.tab20(np.linspace(0, 1, len(top_categories)))

plt.figure(figsize=(10,5))
plt.barh(top_categories.index, top_categories.values, color=colors)
plt.title("Average Engagement Rate by Content Category")
plt.xlabel("Average Engagement Rate")
plt.ylabel("Content Category")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

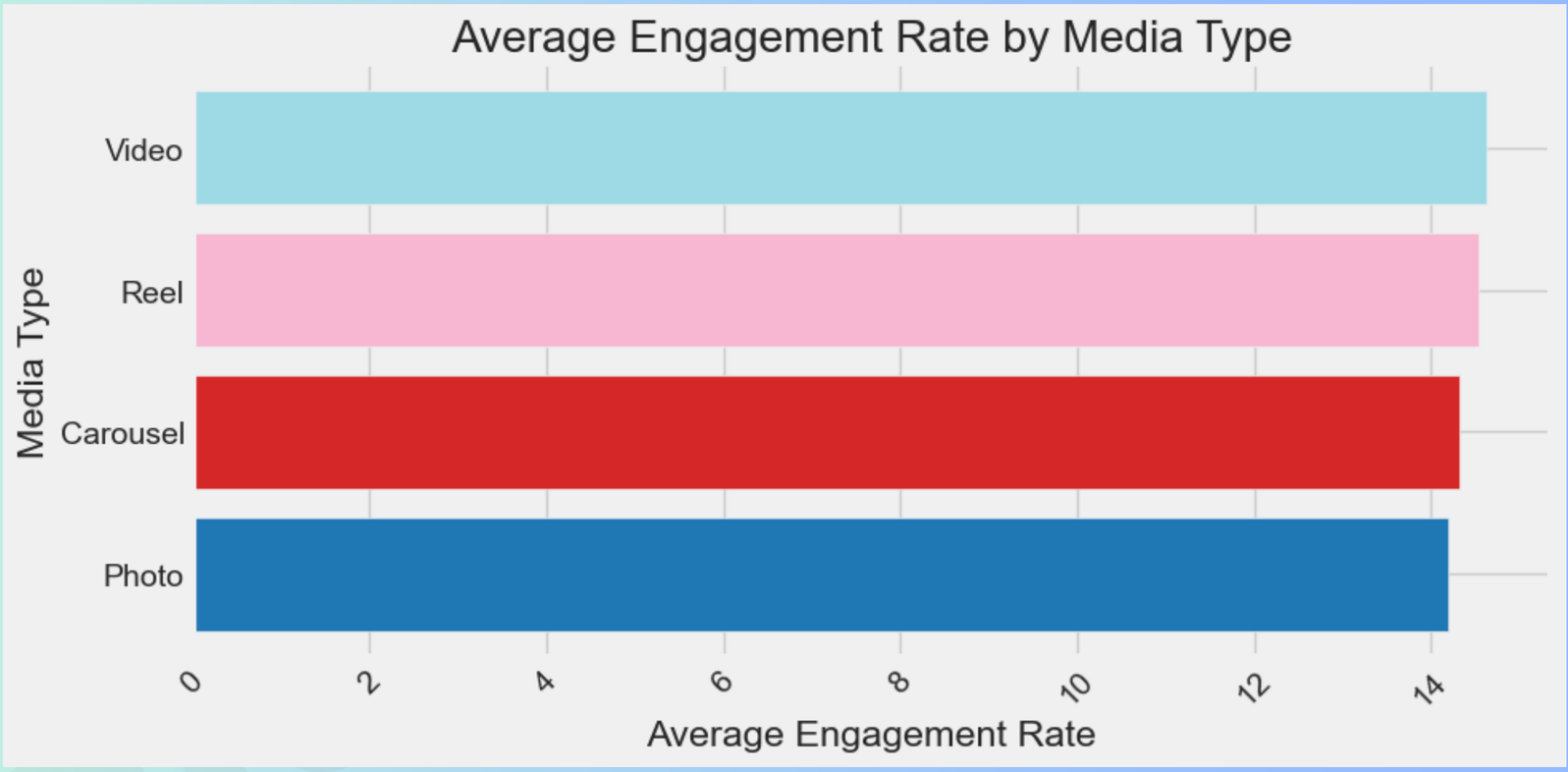
Exploratory Data Analysis (EDA)



Most of the engagements were gained through **Explore** channel.

```
# Sum + sort categories
top_categories = (
    df_new.groupby('traffic_source')['engagement_rate']
      .mean()
      .sort_values(ascending=True)
)

# Generate a distinct color for each bar
colors = plt.cm.tab20(np.linspace(0, 1, len(top_categories)))

plt.figure(figsize=(10,5))
plt.barh(top_categories.index, top_categories.values, color=colors)
plt.title("Average Engagement Rate by Traffic Source")
plt.xlabel("Average Engagement Rate")
plt.ylabel("Traffic Source")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Exploratory Data Analysis (EDA)



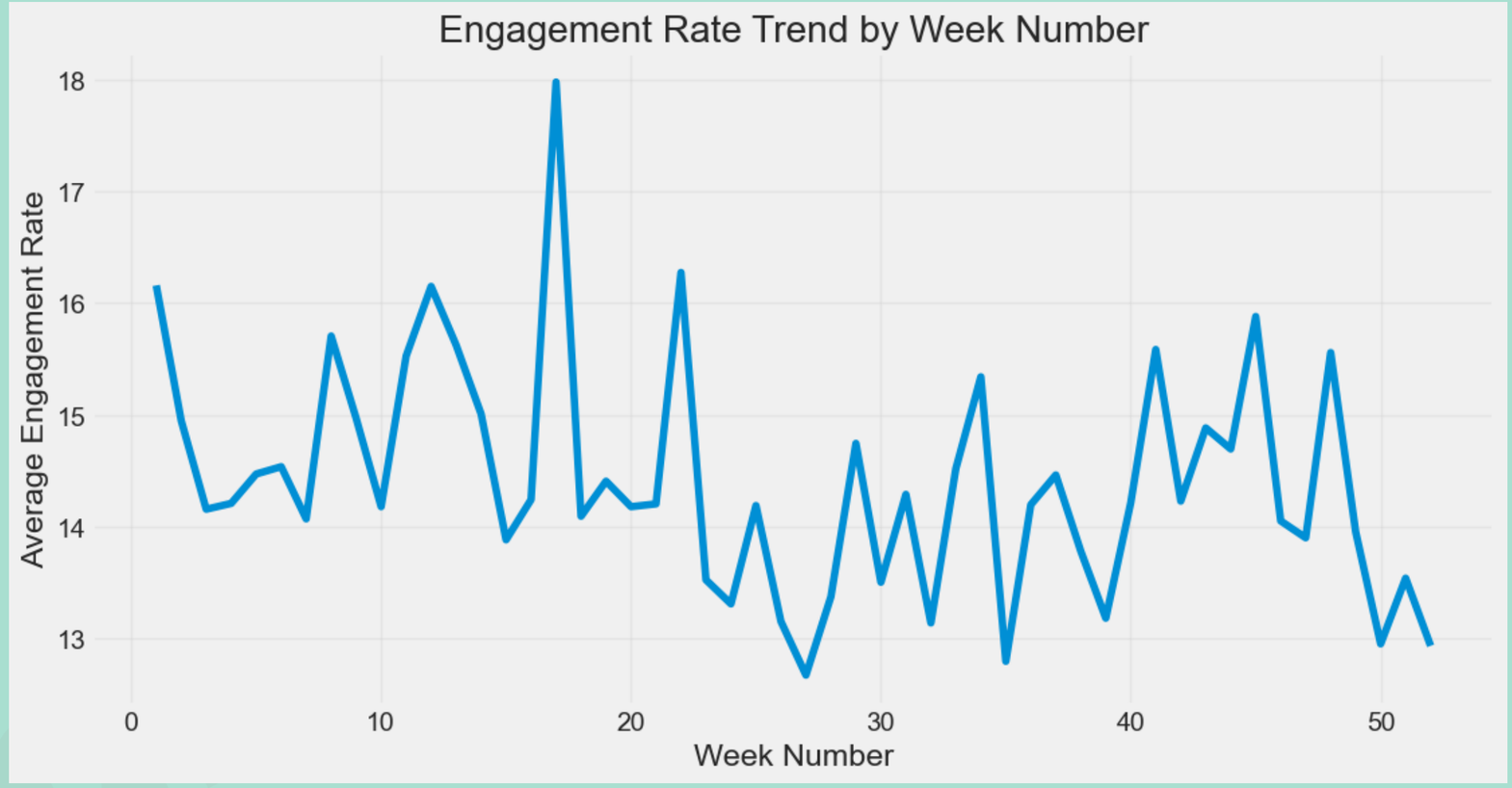
Videos are slightly more likely to gain engagements than **Photos.**

```
# Sum + sort categories
top_categories = (
    df_new.groupby('media_type')['engagement_rate']
      .mean()
      .sort_values(ascending=True)
)

# Generate a distinct color for each bar
colors = plt.cm.tab20(np.linspace(0, 1, len(top_categories)))

plt.figure(figsize=(10,5))
plt.barh(top_categories.index, top_categories.values, color=colors)
plt.title("Average Engagement Rate by Media Type")
plt.xlabel("Average Engagement Rate")
plt.ylabel("Media Type")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Exploratory Data Analysis (EDA)

There isn't any continuous trend in engagement rate throughout the one year period.

But we have to look for peaks and lows as they may refer to special events in a specific time of the year.




```python
# Calculate average engagement rate per week
week_trend = df_new.groupby('week')['engagement_rate'].mean().sort_index()

plt.figure(figsize=(12,6))
sns.lineplot(data=week_trend)

plt.title('Engagement Rate Trend by Week Number')
plt.xlabel('Week Number')
plt.ylabel('Average Engagement Rate')
plt.grid(True, alpha=0.3)

plt.show()
```

Exploratory Data Analysis (EDA)

Plotly Interactive Dashboard

http://127.0.0.1:8050

Key Insights

1

Based on the weekly trend of post engagement, strategists should avoid concentrating on content creation during periods of low engagement.

2

Among the numerical features affecting engagement rates, social media strategists should focus on increasing number of likes on posts, since it is the most directly linked to the level of audience engagement.

3

Posts in beauty category, usually get higher engagement rates.

Key Insights

4

Explore drives the highest engagement, but all traffic sources perform almost equally, signaling that the content (not the channel) is the main driver of engagement.

5

Engagement rate is not affected much by the type of media.

6

Based on the weekly trend of post engagement, strategists should avoid concentrating on content creation during periods of low engagement.

Future Recommendations

1. In real world, there are many other factors contributing to engagement rates which should be considered. Duration, catching phrases, hashtag content, background sounds and captions are a few examples

2. A Comment Sentiment Analysis could be implemented to measure the overall sentiment of the posts and their impact on engagement.

3. To improve accuracy, given the weak feature correlation, we can use tree-based regression for capturing non-linear relationships and transform the engagement rate into a categorical variable for classification to see if it makes more sense.

4. To validate the results statistically, we can conduct hypothesis testing before proceeding to develop machine learning models.

Presented by Mehdi Bohloul

Thank you very much!

https://github.com/eddie-bhl

https://linkedin.com/in/mehdi-bohloul