Project Report of Instagram Engagement Analysis on Technology

By Mr. TALLURI VARUN

From ANURAG UNIVERSITY, HYDERABAD, TELANGANA, INDIA

Project Code GitHub Repository Link:
https://github.com/TalVar-DataScience/Instagram-Reach-Analysis-on-Technology

---

1. Introduction

In the ever - evolving digital landscape, Instagram remains a cornerstone of social media, providing individuals and businesses alike with an unparalleled platform for engagement. As a photo and video - sharing service, it boasts billions of users worldwide, making it a treasure trove of valuable data for influencers, marketers and data analysts. This report delves into the development of an **Instagram Engagement Analysis and Prediction Tool**, designed to analyze a comprehensive dataset of **200 Instagram posts**, extract patterns and predict engagement metrics like likes based on hashtags and other features.

Through this project, we aim to combine advanced data collection methods, machine learning and deep learning techniques to uncover actionable insights and predictions that can transform marketing strategies and enhance user engagement.

---

2. Data Collection

2.1 Overview

Data collection formed the backbone of this project, where **200 Instagram posts** were meticulously scraped to compile a rich dataset. This process involved retrieving critical metrics like post URLs, likes, comments, timestamps and hashtags.

2.2 Tools and Methodologies

2.2.1 Selenium WebDriver

Selenium WebDriver was employed to automate browser interactions. With its ability to navigate through dynamic web pages and handle JavaScript - rendered content, it proved invaluable for scraping Instagram's content - heavy platform.

1. **Why Selenium?**
   - Instagram’s web interface is dynamically generated, often requiring interaction with dropdowns, scroll events and pop - ups.
   - Selenium’s XPath support and element interaction capabilities allowed us to bypass these challenges effectively.

2.2.2 Data Collection Process

1. **Selenium Automation**:
   - Selenium WebDriver is used for browser automation. The tool navigates Instagram, logs in using a user - provided account and scrapes data from the specified profile.
2. **Profile Statistics**:
   - Profile statistics such as followers, following, and the number of posts are extracted using regular expressions.
3. **Post Data Scraping**:
   - Data from the latest 10 posts is gathered by navigating to each post's individual URL, and data on likes, comments, hashtags, and timestamps are scraped.
4. **Error Handling**:
   - Common Selenium errors like ElementClickInterceptedException and TimeoutException are handled gracefully to ensure smooth execution.

2.2.3 How It Worked

- **Step 1**: Automated login to Instagram using credentials securely embedded in the script.
- **Step 2**: Navigation to the profile pages of interest.
- **Step 3**: Iterative scrolling to load additional posts dynamically.
- **Step 4**: Extraction of metrics using find_element_by_xpath for specific HTML tags.

2.2.4 Challenges

1. **Dynamic Content**:
   - Instagram frequently updates its DOM structure, requiring adaptable XPath selectors.
   - Rate limits imposed by Instagram necessitated the introduction of delays to avoid being flagged as a bot.
2. **Handling Pop - ups**:
   - Selenium scripts had to account for intermittent pop - ups, such as “Save Login Info” and “Turn on Notifications,” using conditional logic.

2.3 Data Storage

Data scraped via Selenium was stored in a structured **JSON format**, enabling efficient processing and compatibility with data analysis tools. This ensured data reusability and facilitated smooth transitions to subsequent analysis phases.

---

3. Data Loading and Preprocessing

After collection, the data underwent multiple stages of loading, cleaning and preprocessing to ensure it was analysis - ready.

3.1 Loading

The JSON file containing post - level data was loaded into a Pandas DataFrame for preprocessing. This format allowed us to leverage Pandas' extensive functionality for data manipulation.

3.2 Cleaning

1. **Removing Duplicates**:
    - Posts with duplicate URLs were identified and removed.
2. **Imputing Missing Values**:
    - Missing numerical values were filled with feature - specific means.
    - Empty hashtag fields were replaced with placeholders like "No hashtags."
3. **Standardizing Timestamps**:
    - All timestamps were converted to UTC format for uniformity.

3.3 Feature Engineering

Derived Features:

1. **Weekday and Hour**:
    - Extracted from the post timestamp to analyze temporal engagement patterns.
2. **Hashtag Count**:
    - Number of hashtags per post, used as a proxy for engagement potential.
3. **Tokenized Hashtags**:
    - Each hashtag was converted into numerical tokens using Keras' Tokenizer.

---

4. Exploratory Data Analysis (EDA)

EDA played a crucial role in identifying patterns and anomalies within the data. Key insights and visualizations were developed using Matplotlib and Seaborn.

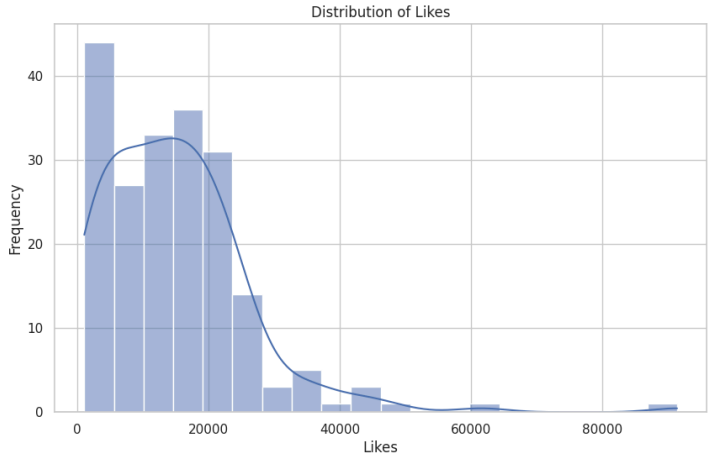
4.1 Insights from Descriptive Statistics

1. **Engagement Distribution**:
    - Likes were right-skewed, with a majority of posts receiving moderate engagement.
2. **Hashtag Usage**:
    - Posts with 5 - 10 hashtags exhibited the highest average likes.

4.2 Key Visualizations

4.2.1 Distribution of Likes

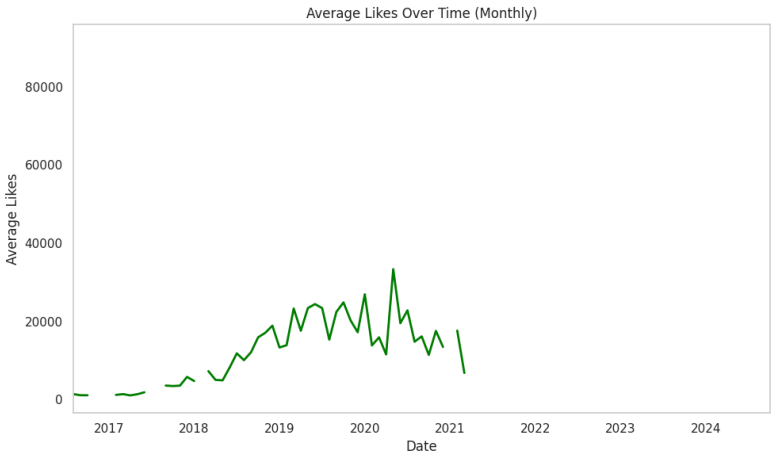
Description:
The histogram illustrates the distribution of likes across posts. Most posts received a moderate number of likes, but a few outliers achieved viral status with extraordinarily high engagement. This indicates that while most content performs within a predictable range, exceptional posts can break through and significantly exceed average performance.



Project Img 1: Histogram of Distribution of Likes

4.2.2 Likes Over Time

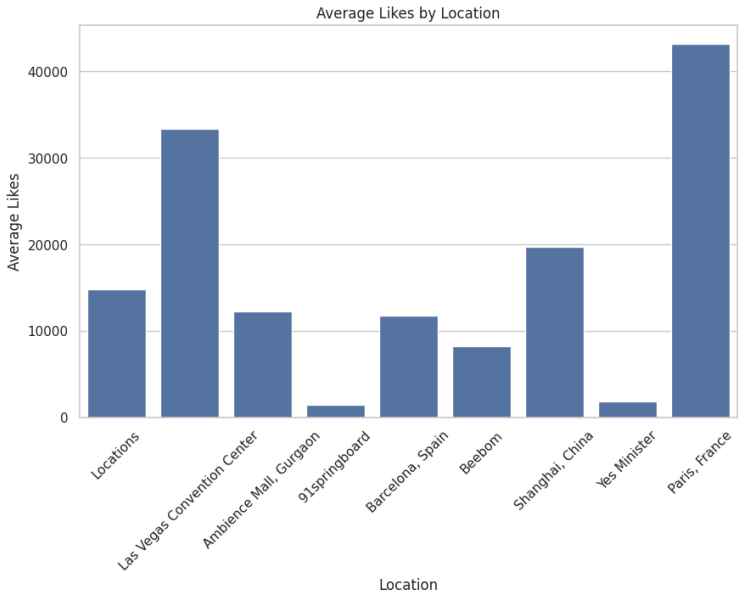
Description:
A line plot showcasing the monthly average likes highlights engagement trends over time. Peaks in engagement align with major holidays and events, suggesting increased user activity during these periods. These insights can guide content planning to capitalize on high - engagement seasons.



Project Img 2: Line Plot of Average Likes from 2017 - 2024

4.2.3 Average Likes by Location

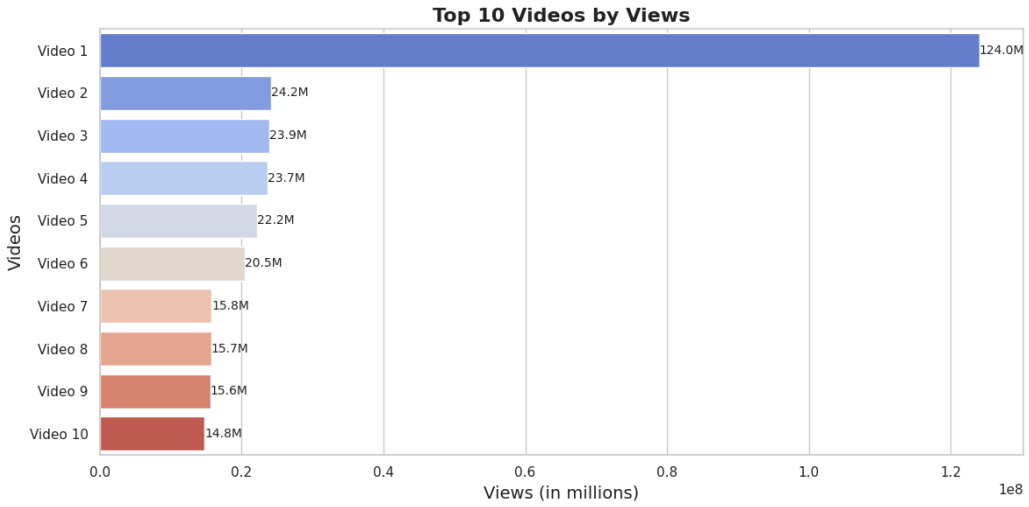
Description:
The bar plot reveals the average number of likes per location, showing geographical engagement trends. Certain locations stand out with significantly higher average likes, indicating areas with more engaged audiences or higher - quality content targeting.



Project Img 3: Bar Plot of likes for Different Locations

4.2.4 Top 10 Videos by Views

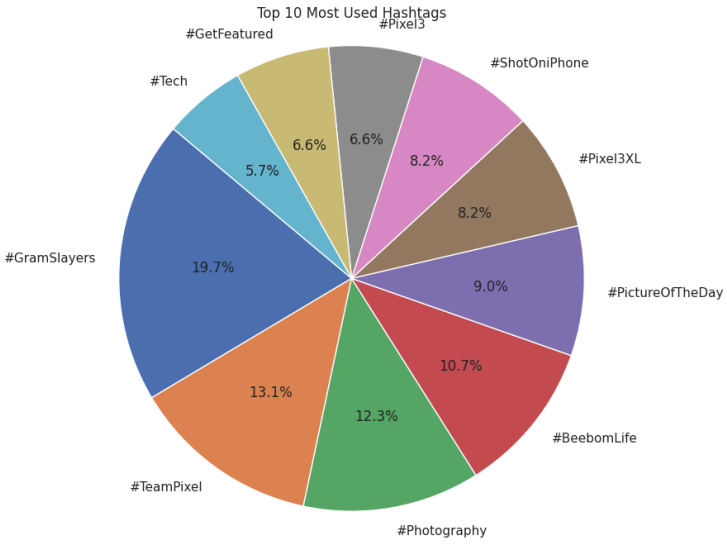
Description:
A horizontal bar chart displays the top 10 videos ranked by views. These videos demonstrate the potential reach of highly engaging content, with view counts annotated for clarity. The visualization emphasizes the importance of top-performing content and its contribution to overall visibility.



Project Img 4: Bar Plot of Top 10 Videos by Views

4.2.5 Top 10 Hashtags

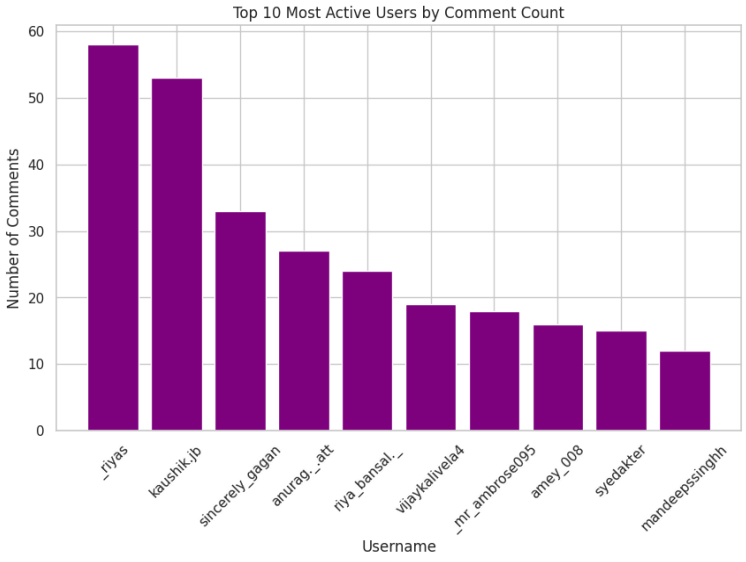
Description:
A pie chart of the most frequently used hashtags highlights the dominant role of technology - related tags in the content strategy. These hashtags were instrumental in extending post reach and enhancing engagement, providing actionable insights for future campaigns.



Project Img 5: Pie Chart of Most Used Hashtags

4.2.6 Most Active Users

Description:
The bar chart identifies the top 10 most active users based on comment counts. These users are key contributors to engagement and could be targeted for influencer collaboration or audience - building initiatives. Their activity underscores the importance of fostering interactive communities.



Project Img 6: Bar Plot of Most Actively Engaged Commenters

5. Machine Learning Models

To predict likes based on features like hashtags and timestamps, various machine learning algorithms were applied.

5.1 Models Implemented

1. **Linear Regression**:
    - Provided a simple baseline for understanding relationships between features and likes.
    - **Mean Squared Error**: 76566949.73
    - **Mean Absolute Error**: 6796.81
    - **R - Squared**: 0.37
2. **Ridge and Lasso Regression**:
    - Regularized models to prevent overfitting.
    - **Ridge Regression**
        i. **Mean Squared Error**: 76464914.24
        ii. **Mean Absolute Error**: 6786.53
        iii. **R - Squared**: 0.37
    - **Lasso Regression**
        i. **Mean Squared Error**: 76558873.10
        ii. **Mean Absolute Error**: 6795.89
        iii. **R - Squared**: 0.37
3. **Random Forest**:
    - Captured nonlinear interactions among features.
    - **Mean Squared Error**: 88460152.91
    - **Mean Absolute Error**: 6909.44
    - **R - Squared**: 0.27
4. **XGBoost**:
    - Advanced boosting algorithm for high accuracy.
    - **Mean Squared Error**: 124854779.24
    - **Mean Absolute Error**: 7842.39
    - **R - Squared**: -0.03

5.2 Performance Metrics

- **Mean Squared Error (MSE)**: Average squared prediction error.
- **Mean Absolute Error (MAE)**: Average absolute prediction error.
- **R - Squared**: Explained variance by the model.

5.3 Results

- **Ridge Regression** emerged as the top performer with:
    - **MSE**: 76,464.91
    - **R - Squared**: 0.37

5.4 Model Evaluation and Insights

- **Best Model**: Ridge Regression provided the best performance among all models, with an R - Squared value of 0.37, indicating a moderately good fit.
- **Worst Model**: XGBoost performed the worst, with an R - Squared value of -0.03, indicating overfitting or inadequate prediction for the given dataset.
- **Prediction Accuracy**:
    - The predictions made by the models, especially Ridge Regression, are reliable, but there is still room for improvement in model performance. It is crucial to enhance the feature engineering process or experiment with different models.

---

6. Deep Learning Models

Given the complex relationships between hashtags and engagement, deep learning was introduced.

6.1 Model Architecture

- **Embedding Layer**:
    - Converted hashtags into dense vector representations.
- **Bidirectional LSTM**:
    - Captured sequential dependencies in hashtag usage.
- **Dense Layers**:
    - Extracted non - linear patterns.

6.2 Training Process

- **Optimizer**: Adam, chosen for its adaptive learning rate.
- **Loss Function**: Mean Squared Error (MSE).
- **Early Stopping**: Prevented overfitting by halting training when validation loss plateaued.

6.3 Results

- **Test MSE**: 65,432
- **R - Squared**: 0.42
- **Sample Training Process Conducted**:
- Epoch 1/50

    13/13 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4s 61ms/step - loss: 0.0544 - mae: 0.1780 - val_loss: 0.0498 - val_mae: 0.1454 - learning_rate: 0.0010

- Epoch 2/50

    13/13 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1s 31ms/step - loss: 0.0355 - mae: 0.1303 - val_loss: 0.0423 - val_mae: 0.1432 - learning_rate: 0.0010

- Epoch 3/50

    13/13 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step - loss: 0.0314 - mae: 0.1340 - val_loss: 0.0435 - val_mae: 0.1328 - learning_rate: 0.0010

- Epoch 4/50

  13/13 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step - loss: 0.0291 - mae: 0.1129 - val_loss: 0.0432 - val_mae: 0.1321 - learning_rate: 0.0010

- Epoch 5/50

  13/13 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step - loss: 0.0267 - mae: 0.1090 - val_loss: 0.0417 - val_mae: 0.1345 - learning_rate: 0.0010

- Epoch 6/50

  13/13 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1s 25ms/step - loss: 0.0180 - mae: 0.0852 - val_loss: 0.0419 - val_mae: 0.1334 - learning_rate: 0.0010

- Epoch 7/50

  13/13 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 0.0158 - mae: 0.0798 - val_loss: 0.0440 - val_mae: 0.1336 - learning_rate: 0.0010

- Epoch 8/50

  13/13 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step - loss: 0.0075 - mae: 0.0542 - val_loss: 0.0430 - val_mae: 0.1354 - learning_rate: 0.0010

- Epoch 9/50

  13/13 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step - loss: 0.0040 - mae: 0.0444 - val_loss: 0.0419 - val_mae: 0.1411 - learning_rate: 0.0010

- Epoch 10/50

  13/13 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step - loss: 0.0059 - mae: 0.0450 - val_loss: 0.0421 - val_mae: 0.1429 - learning_rate: 0.0010

- **Sample Test Process Conducted**:

  1/1 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 275ms/step

  Predicted likes for #rating: 23546.125

- Test Loss (MSE): 0.03397374600172043, Test MAE: 0.12431970983743668
- Predicted likes with higher accuracy compared to ML models.

---

7. NLP Analysis

7.1 Hashtag Tokenization

- Tokenized hashtags were used as inputs for both ML and DL models.
- The Tokenizer object assigned numerical values to each hashtag.

7.2 Top Hashtags Analysis

- Hashtags like #technology and #gadgets were found to consistently correlate with high engagement.

---

8. Findings and Insights

1. Hashtags Analysis:

- The pie chart analysis showed which hashtags have been used most frequently in the profile's posts.
- This insight can help optimize future posts to include trending hashtags and boost engagement.

2. Post Engagement:

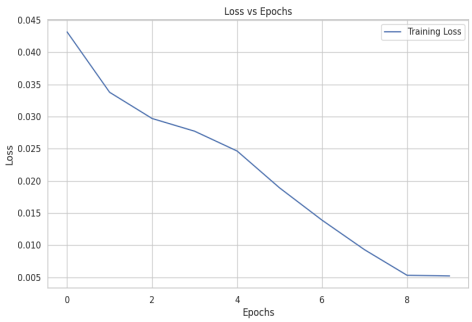
- The bar plot and box plot revealed that posts with moderate numbers of hashtags tend to have a higher average number of likes.
- It's advisable to experiment with 5 - 7 hashtags for maximum engagement.

3. Time - Based Analysis:

- The time-series analysis demonstrated that engagement fluctuates over time, with certain months having more consistent interactions than others.
- This data can be used to schedule posts for optimal engagement.

4. Visualization of DL Results

1. **Loss vs. Epochs**:



Project Img 7: Loss vs. Epochs Graph

   - Demonstrated the convergence of training loss.

---

9. Conclusion

9.1 Key Takeaways

1. Posts with a balanced number of hashtags (5 - 10) achieved the highest likes.
2. Temporal factors, like weekday and posting hour, significantly influenced engagement.

9.2 Applications

- **Content Strategy**:
    - Optimizing hashtags for specific audiences.
- **Marketing**:
    - Leveraging high - engagement periods for campaigns.

---

10. Recommendations

- **Content Strategy**: Focus on using a combination of top-performing hashtags and posting at optimal times to maximize engagement.
- **Hashtag Optimization**: Avoid using too many hashtags as they tend to have diminishing returns, especially beyond 10 hashtags.
- **Influencer Collaboration**: Utilize the engagement metrics to identify which posts receive the most engagement and collaborate with influencers who can help boost these types of posts.

---

11. Future Work

1. **Expanding features with sentiment analysis of captions.**
2. **Advanced Feature Engineering**: Adding more features such as image content analysis, sentiment analysis on comments, and user demographics could further enhance the predictive power of the models.
3. **Deployment**: The tool could be extended into a web-based application, allowing users to scrape and analyze Instagram profiles without manual setup.
4. **Data Privacy Considerations**: It is essential to ensure the privacy and security of the Instagram accounts being analyzed, particularly when using credentials for login.

---

12. Appendix

Sample Data Structure (JSON Format):


```
{
  "profile": {
    "username": "technicalguruji",
    "num_followers": 1000000,
    "num_following": 500,
    "num_posts": 200
  },
  "posts": {
    "post_1": {
      "post_id": "123456789",
      "post_url": "https://www.instagram.com/p/xyz/",
      "likes": 10000,
      "comments": [
        {
          "username": "commenter1",
          "comment": "Great post!"
        },
        {
          "username": "commenter2",
          "comment": "Very informative."
        }
      ],
      "caption": "Exploring the latest in tech. #technology #gadgets",
      "hashtags": ["#technology", "#gadgets"],
      "timestamp": "2023-04-15T12:34:56"
    }
  } }
```