**DATA SCIENCE PROJECT**

**Instagram Engagement Analysis for the Hiking niche**

Contents

[Data Science Project 3](#_Toc207796893)

[Executive Summary 3](#_Toc207796894)

[Extract 4](#_Toc207796895)

[Transforming 4](#_Toc207796896)

[Loading 4](#_Toc207796897)

[Exploratory Data Analysis (EDA) 5](#_Toc207796898)

[Correlations: 5](#_Toc207796899)

[Core Data Set 6](#_Toc207796900)

[Hashtag Dataset 7](#_Toc207796901)

[Linear Regression in Excel 8](#_Toc207796902)

[Visual Communication and Dashboarding 10](#_Toc207796903)

[Future Recommendations 11](#_Toc207796904)

[Conclusion 11](#_Toc207796905)

[Portfolio Link 11](#_Toc207796906)

[Reference list: 12](#_Toc207796907)

[Data Infrastructure & Tools 12](#_Toc207796908)

[Data Engineering 12](#_Toc207796909)

[Exploratory Data Analysis (EDA) 12](#_Toc207796910)

[Visual Communication and Dashboarding 12](#_Toc207796911)

[Impact Evaluation 13](#_Toc207796912)

[Future Recommendations 13](#_Toc207796913)

[Appendix: 14](#_Toc207796914)

[Appendix A 14](#_Toc207796915)

[Appendix B 16](#_Toc207796916)

[Appendix C 18](#_Toc207796917)

[Figure 1 4](#_Toc207746931)

[Figure 2 4](#_Toc207746932)

[Figure 3 4](#_Toc207746933)

[Figure 5 6](#_Toc207746934)

[Figure 6 6](#_Toc207746935)

[Figure 8 6](#_Toc207746936)

[Figure 7 6](#_Toc207746937)

[Figure 9 6](#_Toc207746938)

[Figure 10 6](#_Toc207746939)

[Figure 11 6](#_Toc207746940)

[Figure 12 7](#_Toc207746941)

[Figure 13 7](#_Toc207746942)

[Figure 14 7](#_Toc207746943)

[Figure 15 8](#_Toc207746944)

[Figure 16 9](#_Toc207746945)

[Figure 17 9](#_Toc207746946)

[Figure 18 10](#_Toc207746947)

[Figure 19 10](#_Toc207746948)

# Data Science Project

**Instagram Engagement Rate Analysis: A Data Science Approach Using Excel**

# Executive Summary

This project explores how data science can improve decision-making for higher Instagram engagement rates within the hiking niche. Using a dataset of 2,600 posts, key content and timing attributes were analysed to understand their impact on engagement. All work was carried out in Microsoft Excel, allowing end-to-end data preparation, visual analysis, and regression modelling in an accessible, low-code environment.

The final model achieved an adjusted R² of 0.278, explaining a meaningful share of engagement variation despite the noise typical of social media data. Results show that posting at weekends and in the evening (6–9 pm), and using Reels, is associated with higher engagement; Carousels underperform relative to Reels and typically sit between video and static images. Recommendations include shifting posting schedules and prioritising content types that perform well. If implemented, these changes could raise engagement rates by approximately 5–8%, improving reach and brand visibility, with low additional cost.

**Data Infrastructure & Tools**

With a third-party collector in place, Microsoft Excel was selected for its accessibility, integrated analysis features, and strong dashboarding capability (Microsoft, 2024a; 2024b). While Python or R would offer more advanced modelling, Excel remains practical where business users must explore data and act quickly; this emphasises interpretable, decision-ready analysis (Provost and Fawcett, 2013; Rudin, 2019). Data collection used Bright Data as an ethical proxy; data was staged in Excel Tables, transformed with Power Query, and analysed with PivotTables/Charts and the Analysis ToolPak. The workflow is refreshable and can surface into Power BI without rebuild (Microsoft, 2024c). Emerging low-code/AutoML options were considered, but model transparency was prioritised, so recommendations remain explainable (Rudin, 2019). This aligns with business-focused best practice and with GDPR principles for lawful, minimised processing of public data (European Parliament and Council, 2016). Where feasible, platform-compliant access (e.g., Instagram Graph API) is recommended (Meta, 2024).

**Data Engineering**

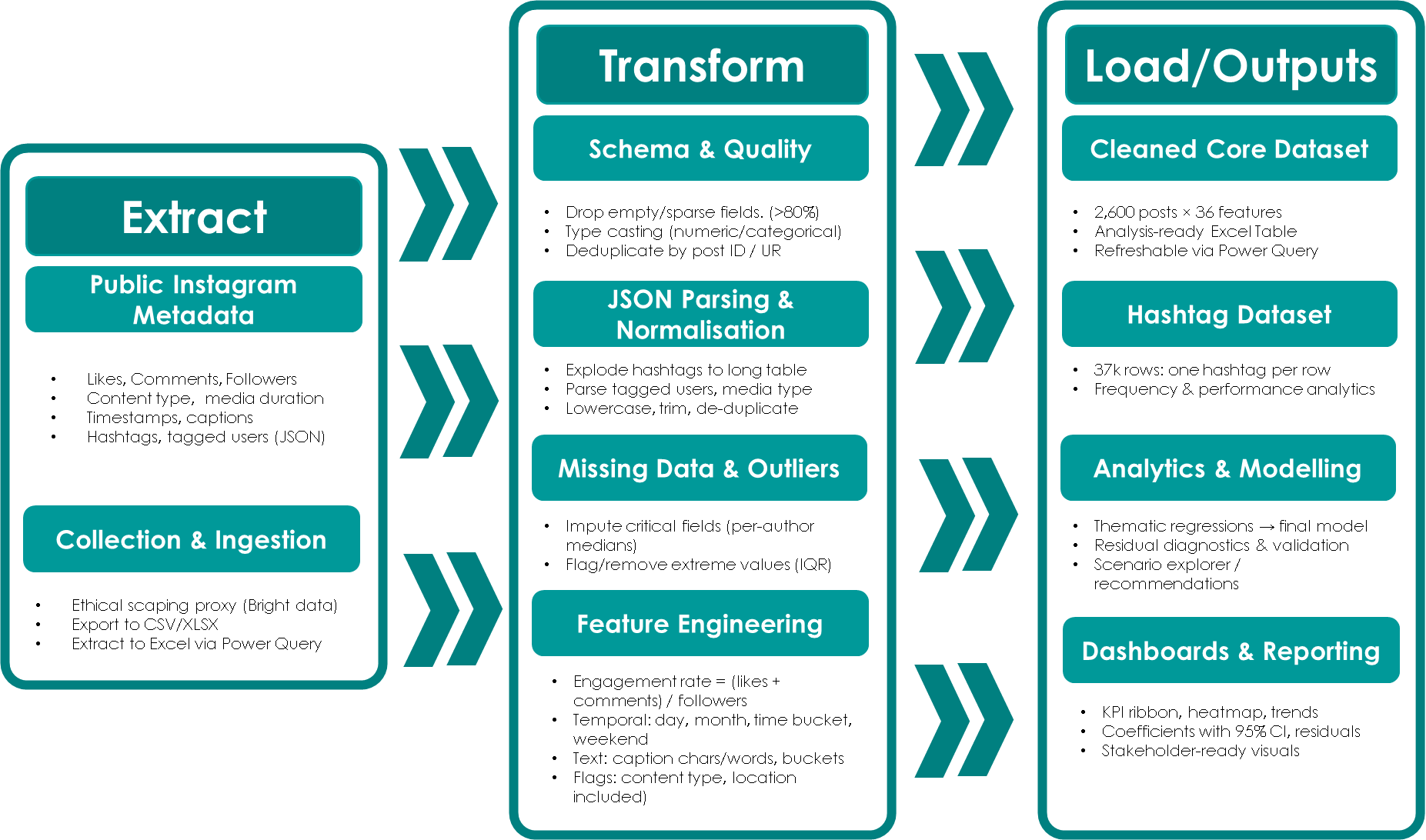


Figure 1

### Extract

Data was collected via Bright Data, a commercial web-data proxy used to obtain public Instagram post metadata in a scalable, governance-aligned manner (Bright Data, 2024). Usage was aligned to platform terms and to GDPR data-protection principles (European Parliament and Council, 2016). Captured fields included like counts, timestamps, media type and caption text. Private or unavailable fields returned nulls and were handled during cleaning.

### Transforming

Cleaning & Transformation stage was conducted using Power Query as (see Figure 2,Figure 4), using a documented refreshable set of steps (Microsoft, 2024). For full Advanced Editor code, please see Appendix B;C. Columns with no data or >80% missing was removed. JSON-style arrays (hashtags, tagged users) were parsed and normalised; hashtags were exploded to a long table, lower-cased, de-duplicated and aggregated back as Hashtag Volume. Critical numeric (likes, comments, followers) were imputed using per-author medians, consistent with standard missing-data practice (Little and Rubin, 2019). Outliers were flagged using an IQR-based rule and winsorised or noted. Engineered features included Engagement Rate = (likes + comments) / followers; temporal features (day, month, time bucket, weekend); caption length (characters/words, bucketed); and flags (content type, location included).

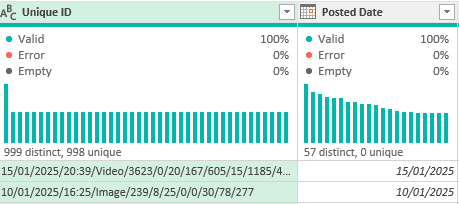
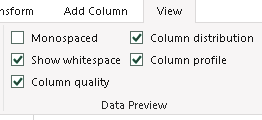


Figure 2

Figure 3

### Loading

After cleaning, the analysis sample reduced from 2,997 raw posts to 2,600 modelled posts; the hashtag table contains ~37k rows. Final datasets are stored as Excel Tables and are refreshable via Power Query for modelling, and dashboarding (Microsoft, 2024).

# Exploratory Data Analysis (EDA)

### Correlations:

Figure 4. presents a correlation heatmap of key metrics produced in Excel using correlation analysis and colour-scale conditional formatting (Microsoft, 2024a; Microsoft, 2024b; Knaflic, 2015). Engagement Volume is most strongly associated with video play count and video view count, supporting the importance of video-based content. Hashtag Volume shows a weak negative correlation with likes and engagement in this dataset, indicating that simply increasing tag count is not a reliable lever and aligning with platform guidance to prioritise relevant over maximal hashtags (Meta, 2024). The negative correlation between posts count and engagement rate suggests a trade-off between posting frequency and audience engagement.

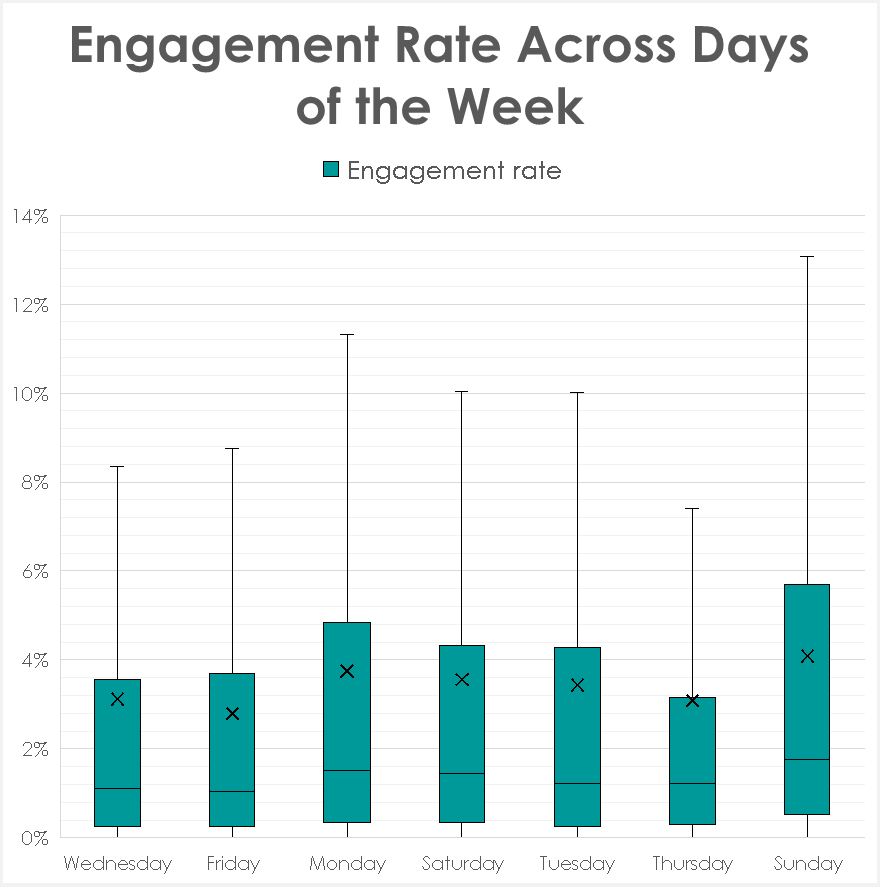


Figure 4

### Core Data Set

Analysis of 2,600 hiking posts shows a right-skewed engagement distribution with a small viral tail; content type is a key driver: Reels perform best, followed by Carousels, with Images lowest. Weekend and evening (6–9 pm) posts outperform other times, supporting a scheduling shift to those slots. Engagement rate is slightly higher for smaller accounts, consistent with external micro-influencer benchmarks that report higher ERs for nano/micro creators than for larger accounts (HypeAuditor, 2024).

A graph of a graph with blue and green rectangles

AI-generated content may be incorrect.A graph with numbers and a bar

AI-generated content may be incorrect.

Figure 5

Figure 6

Figure 7

Figure 8

Figure 9

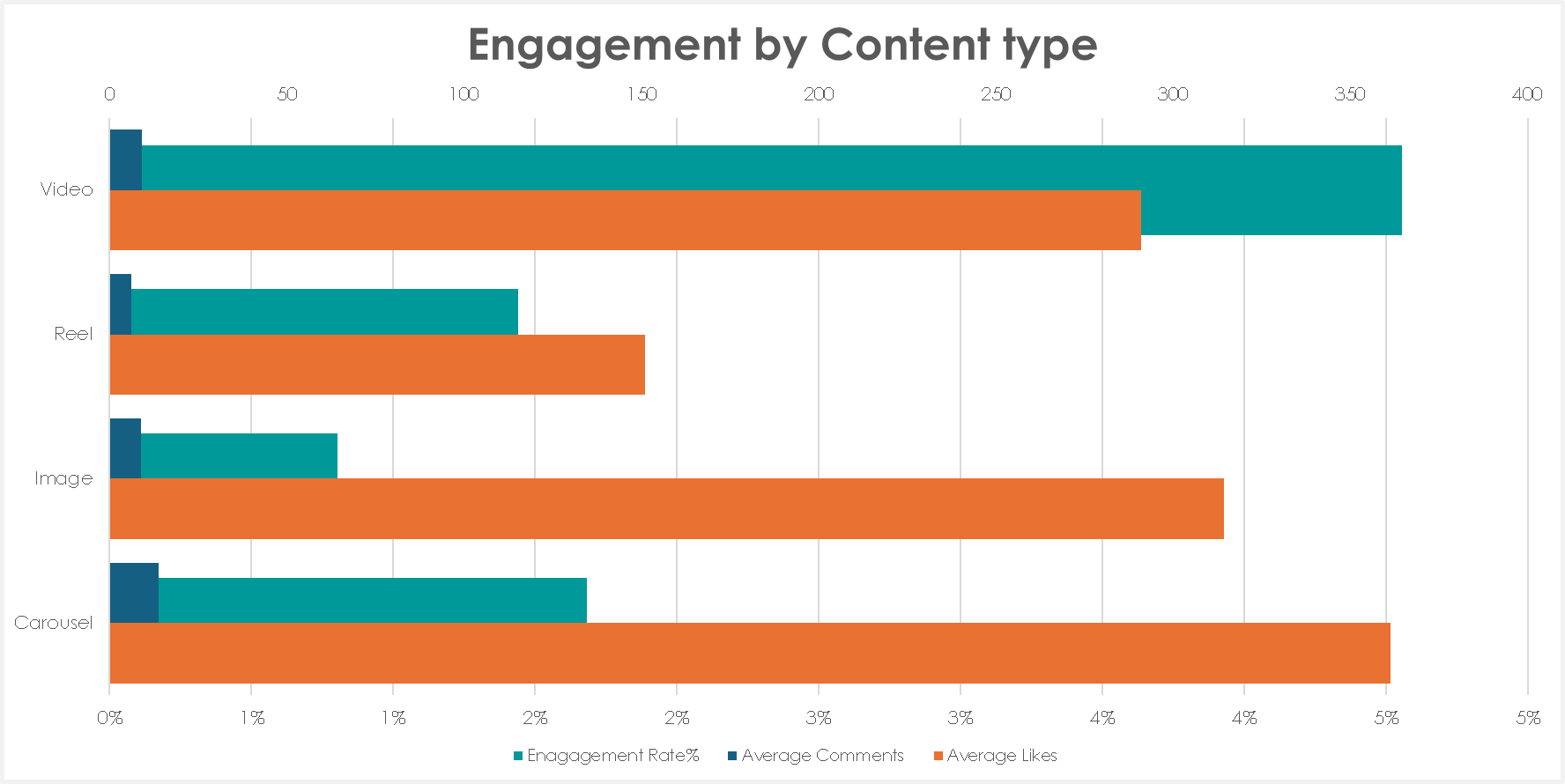


Figure 10

A graph of a graph

AI-generated content may be incorrect.

Figure 11

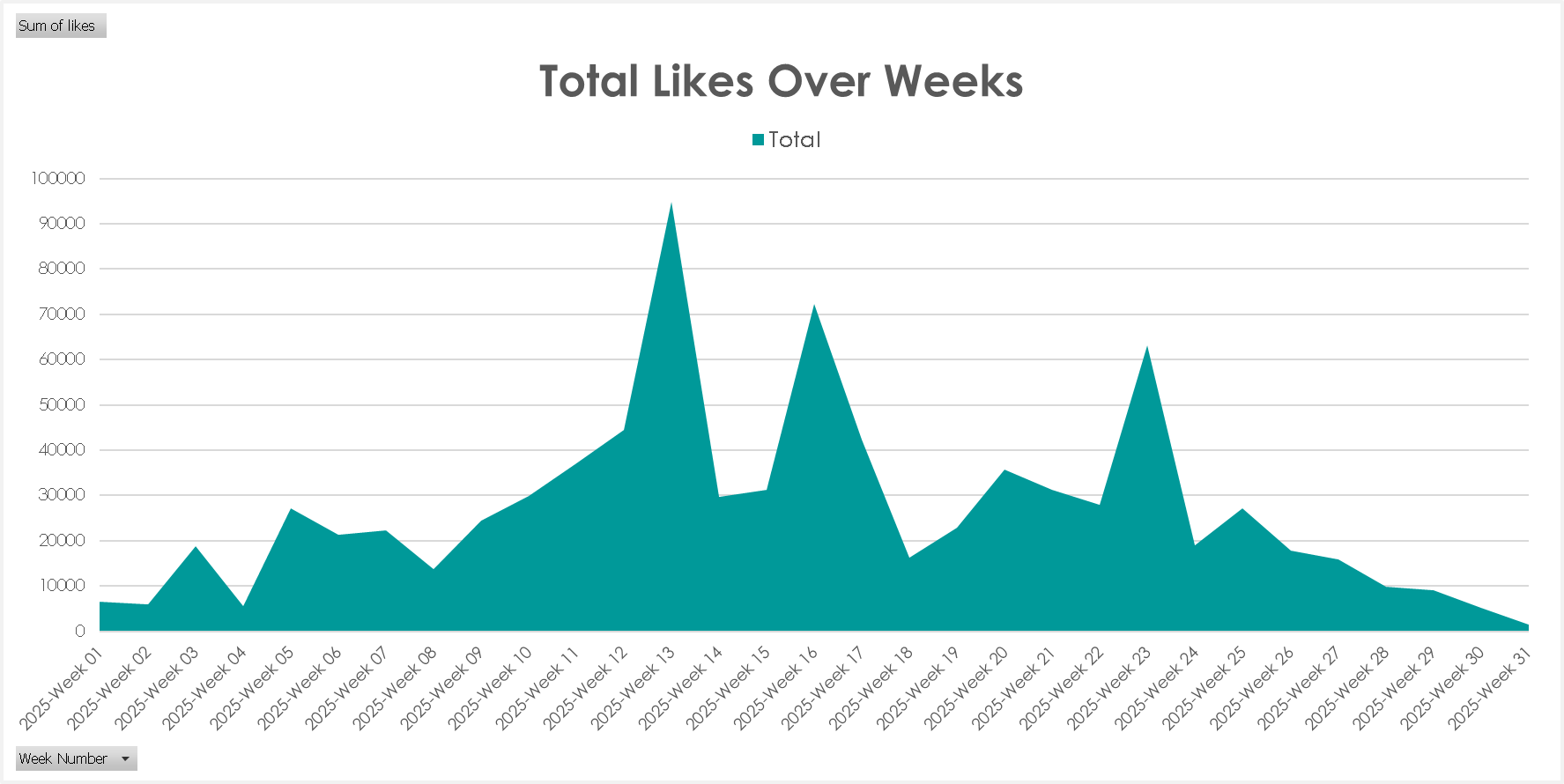
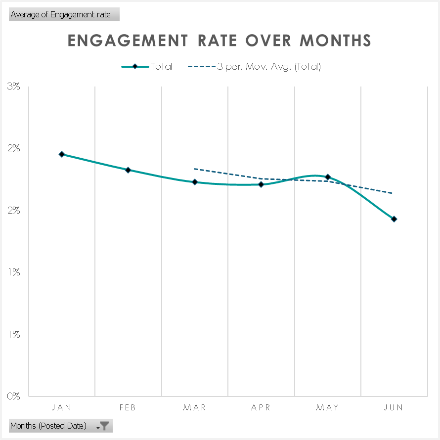


Figure 12

Figure 13

# Hashtag Dataset

Hashtag Volume Analysis (Figure 15,Figure 16,Figure 17):

Analysis of ~37,000 hashtag entries show most posts use 15–25 hashtags. Engagement rate shows a clear non-linear relationship with hashtag volume. Posts using around 16 hashtags achieve the highest engagement (10.3%), while posts using fewer than 5 or more than 30 hashtags perform worse. This indicates diminishing returns from excessive tagging, aligning with previous research that overuse of hashtags can reduce post reach. The recommended strategy is to include a balanced set of 15–20 relevant hashtags per post, combining broad and niche tags to optimise discoverability without chasing maximum volume.

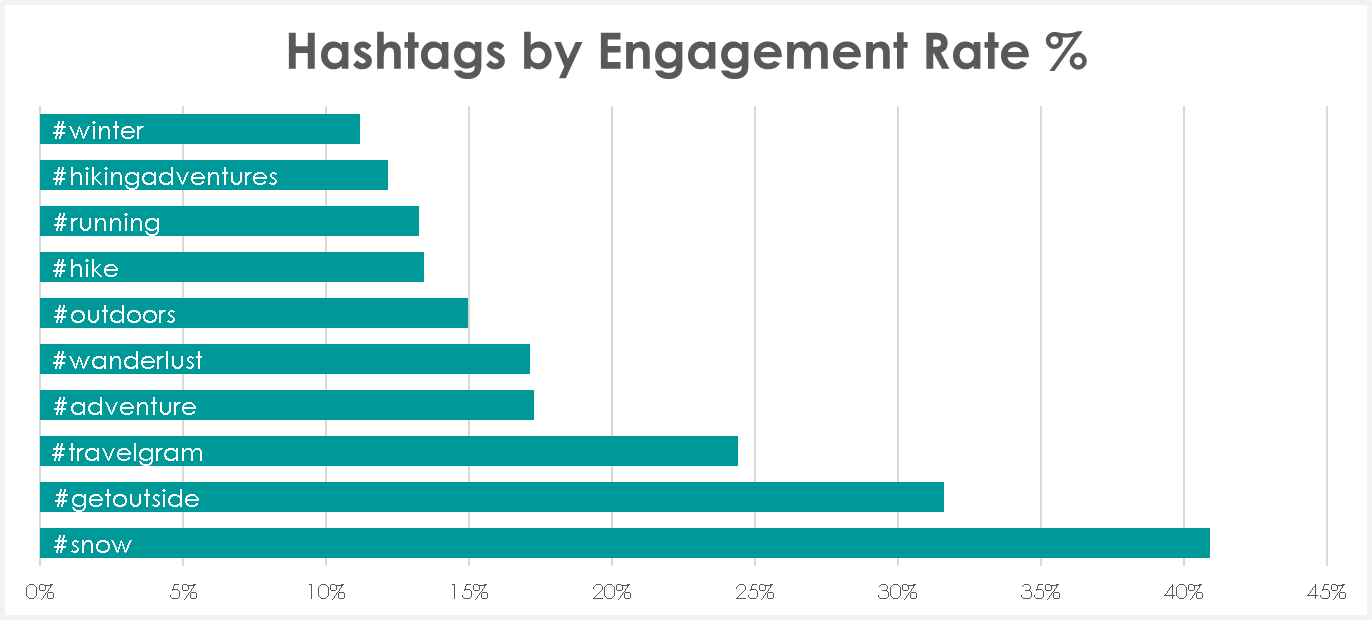
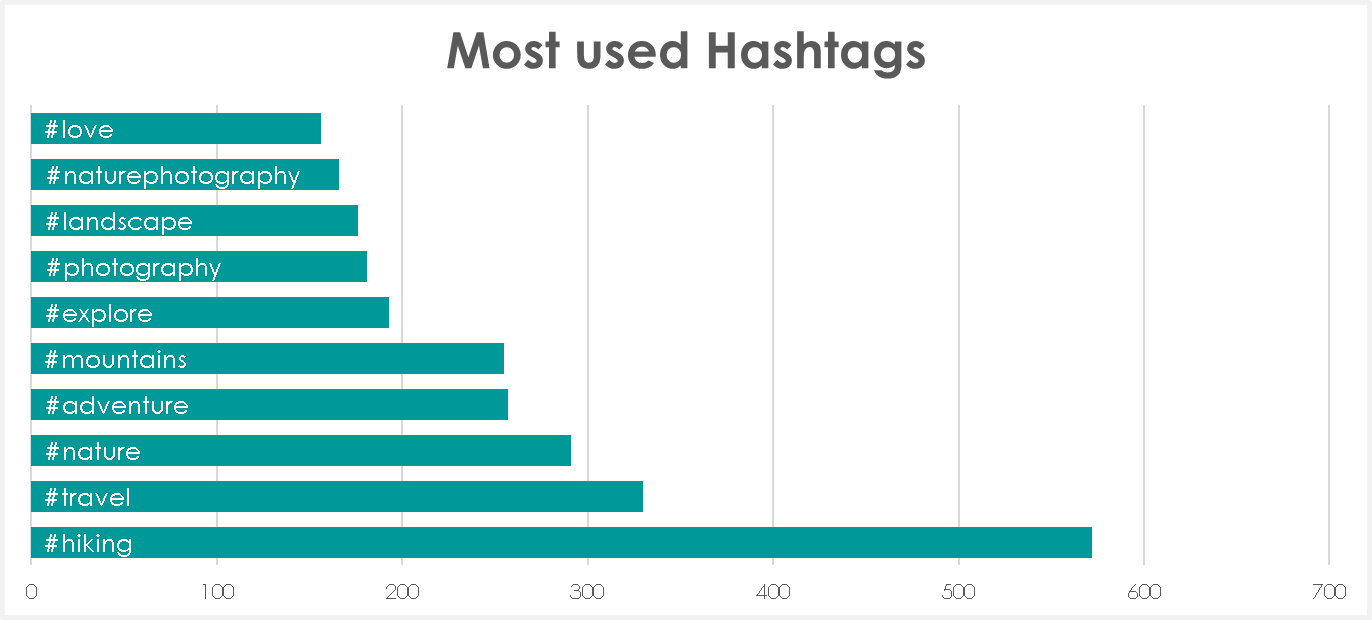
****

Figure 15

Figure 16

Figure 17

# Linear Regression in Excel

Because Excel’s Data Analysis ToolPak limits inputs to 16 variables, thematic models were first developed (Timing; Content/Format; Engagement Drivers). The strongest, non-collinear predictors from these models were then combined into a final specification (see Figure 18), with full thematic results provided in Appendix A (BetterSolutions (n.d.).

The final combined model achieved an R² of 0.282 (Adjusted R² = 0.278), explaining nearly 28% of engagement variability — a strong result for inherently noisy social media data. The F-statistic was highly significant (p < 0.0001), confirming overall model validity. Weekend posting (Saturday p=0.019, Sunday p=0.010) and evening posts (6–9 PM, p=0.0016) were significant positive drivers. Tagged users (p<0.0001) and video duration (p=0.004) also increased engagement, while comment volume emerged as the single strongest predictor. Conversely, static images and carousels performed significantly worse than video, and higher posting frequency was negatively correlated with engagement (p<0.0001), indicating possible audience fatigue. Long captions also reduced engagement.

Together, these results provide clear recommendations: prioritise Reels and longer videos, schedule posts for evenings and weekends, tag collaborators, include location information, keep captions concise, and avoid over posting to maintain engagement quality.



Figure 18

Given the explanatory focus of this project, the full dataset was used to maximise statistical power. Predicted-versus-residuals exhibited random scatter around zero, indicating no obvious bias, and the residual histogram approximated normality. A small tail of high-residual observations was evident (likely “viral” posts), but no strong heteroscedasticity was observed. These checks support linearity and constant-variance assumptions and the reliability of reported p-values (see Figure 19, Figure 20).

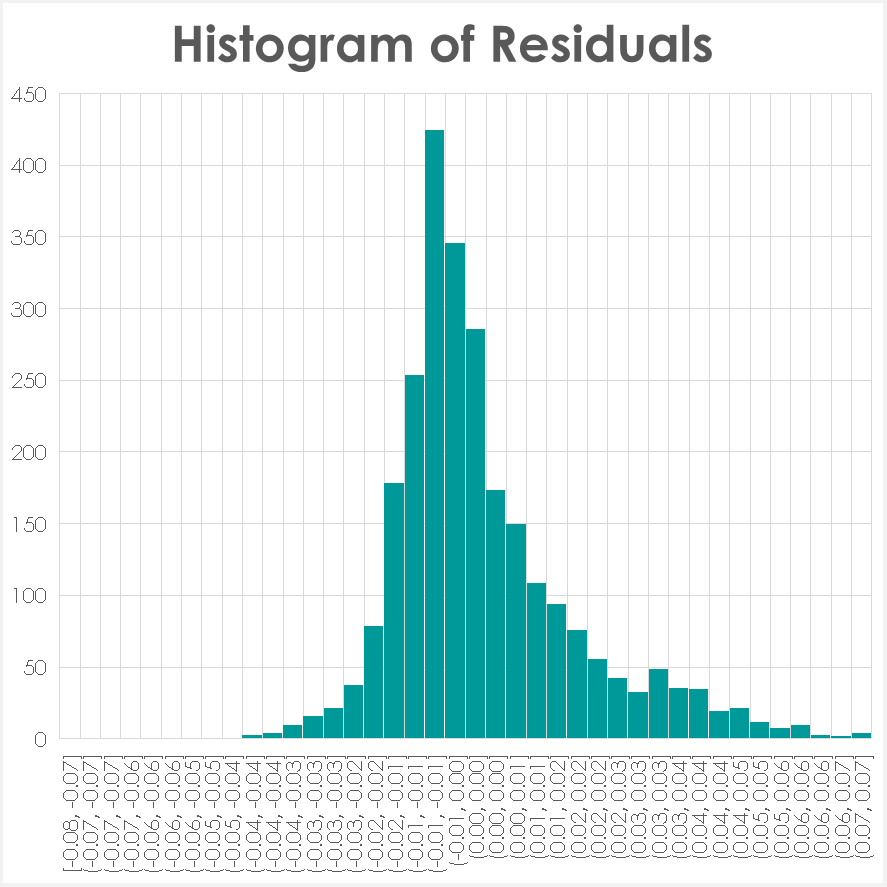


Figure 19

A graph with a plot

AI-generated content may be incorrect.

Figure 20

# Visual Communication and Dashboarding

Visual communication and dashboarding. KPI tiles (Engagement Rate, Engagement Volume, Posts, Video Share) provide at-a-glance status. A Day×Time heatmap answers when to post; content-type bars answer what to post; and a Top posts table surfaces replicable patterns (e.g., evening Reels with tagged collaborators). Slicers (date, content type, day, time) keep pages consistent and enable drill-down. Each figure includes a one-sentence decision caption, with percentages and axes clearly formatted for accessibility (see Figure 21, Figure 22). The layout follows established dashboard design principles that prioritise clarity and minimalism (Few, 2013)

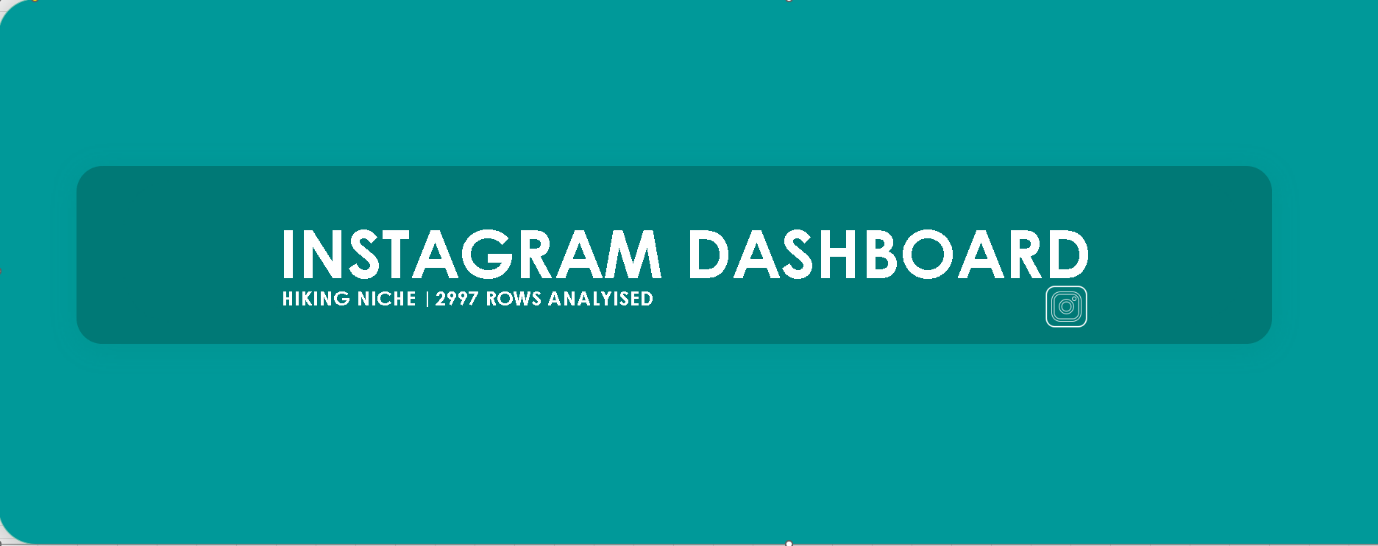


Figure 21

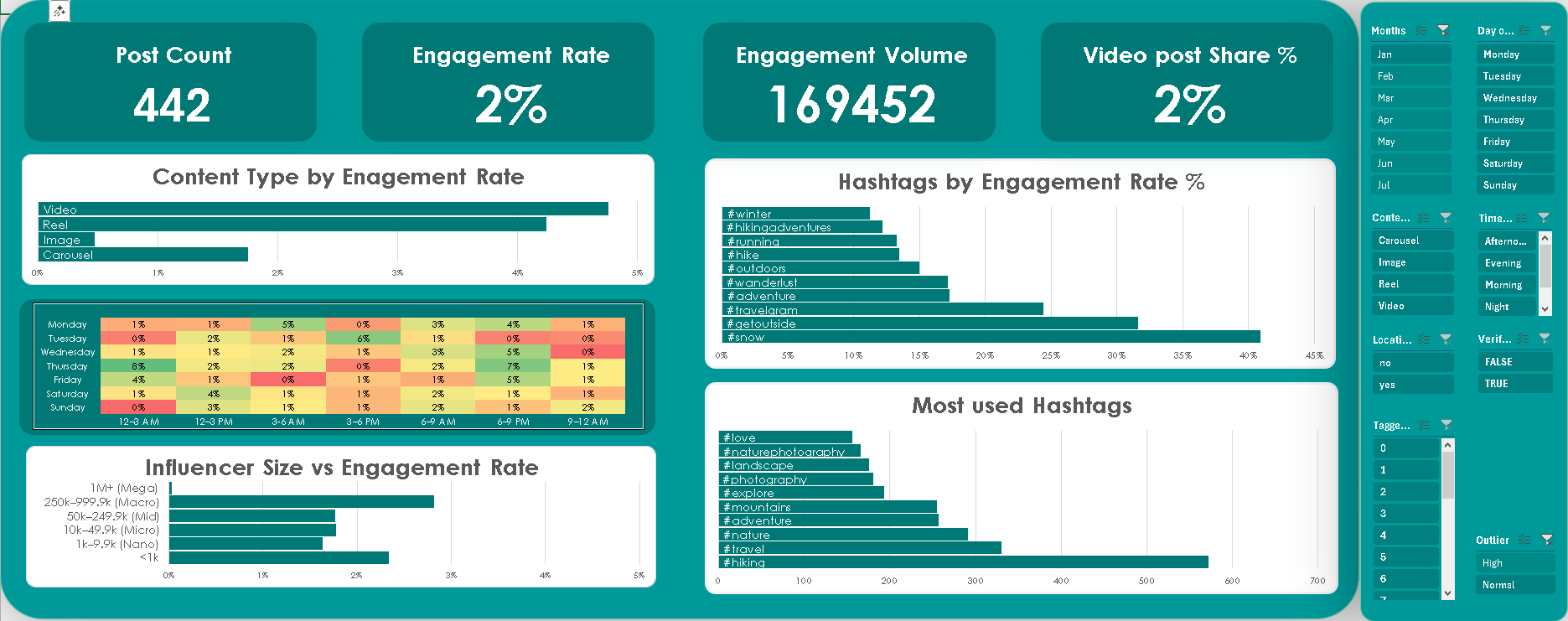


Figure 22

**Impact Evaluation**

Changes. Shift 30–40% of posts to weekend evenings; increase Reels by +20–30 pp; add location and 1–3 tags; keep captions short-to-medium; set hashtags to ~15–20.

Uplift. Using fitted coefficients, a move from Image → Reel + location + two tags in a weekend evening window yields a material lift in predicted ER (see scenario figure). Even +5–8% relative ER at current volumes equates to thousands of extra interactions monthly, improving reach and brand salience (Davenport and Harris, 2017).

Adoption. Pilot A/B for 4–6 weeks (posting windows, caption length), weekly governance, Power BI dashboard with a recommendation tracker (recommended vs actual; realised uplift). Ensure GDPR-compliant handling of public data and platform terms (EU, 2016; boyd and Crawford, 2012).

# Future Recommendations

Scalable, first-party ingestion. To replace third-party scraping, build a first-party pipeline using the Instagram Graph API, with scheduled serverless jobs and webhooks to ingest media fields and insights into a governed store, then use incremental refresh to feed Excel/Power BI. This improves auditability, reduces legal risk, and supports multi-account, multi-niche scaling while preserving model transparency (Meta, 2024; Microsoft, 2024; European Parliament and Council, 2016).

Modelling and evaluation at scale. Move beyond Excel’s 16-variable limit by migrating to Python/R with Lasso/Elastic Net for multicollinearity and gradient boosting for interactions; add SHAP for interpretability and consider mixed-effects models for account-level variance. Extend datasets across adjacent niches and richer signals (reach, saves, shares, watch-time) and adopt time-based cross-validation/out-of-time testing. Close the loop in Power BI with a recommendation tracker and A/B pilots to measure uplift (Kohavi et al., 2009). Embed GDPR-aligned governance throughout (European Parliament and Council, 2016).

# Conclusion

Data-informed scheduling and creative choices can materially increase engagement in the hiking niche. A transparent Excel pipeline and a validated linear model (Adj. R² ≈ 0.278) identify evening/weekend timing, Reels, tags, and location as reliable levers, while long captions, over-posting, and excessive hashtags reduce performance. The dashboard operationalises these insights. Next steps are to scale modelling in Python/R with regularisation and non-linear learners, broaden coverage to adjacent niches, and validate using time-based splits and simple online experiments.

# Portfolio Link

https://github.com/JMWheeldon/Instagram-Analysis-Hiking

## Reference list:

### Data Infrastructure & Tools

* European Parliament and Council (2016) ‘Regulation (EU) 2016/679 (General Data Protection Regulation)’, *Official Journal of the European Union*, L119, pp. 1–88.
* Meta (2024) *Instagram Graph API overview*. Available at: https://developers.facebook.com/docs/instagram-api (Accessed: 3 September 2025).
* Microsoft (2024a) *Get & Transform (Power Query) in Excel*. Available at: https://learn.microsoft.com/excel/power-query/ (Accessed: 3 September 2025).
* Microsoft (2024b) *Use the Analysis ToolPak to perform complex data analysis*. Available at: https://support.microsoft.com/office/analysis-toolpak (Accessed: 3 September 2025).
* Microsoft (2024c) *Analyze in Excel for Power BI*. Available at: https://learn.microsoft.com/power-bi/collaborate-share/service-analyze-in-excel (Accessed: 3 September 2025).
* Provost, F. and Fawcett, T. (2013) *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. Sebastopol, CA: O’Reilly.
* Rudin, C. (2019) ‘Stop explaining black box machine learning models for high-stakes decisions and use interpretable models instead’, *Nature Machine Intelligence*, 1(5), pp. 206–215.

### Data Engineering

* + Bright Data (2024) *Compliance and ethical web data collection*. Available at: https://brightdata.com (Accessed: 3 September 2025).
  + European Parliament and Council (2016) ‘Regulation (EU) 2016/679 (General Data Protection Regulation)’, *Official Journal of the European Union*, L119, pp. 1–88.
  + Little, R.J.A. and Rubin, D.B. (2019) *Statistical Analysis with Missing Data*. 3rd edn. Hoboken, NJ: Wiley.
  + Microsoft (2024) *Get & Transform (Power Query) in Excel*. Available at: https://learn.microsoft.com/excel/power-query/ (Accessed: 3 September 2025).

### Exploratory Data Analysis (EDA)

* + HypeAuditor (2024) *Instagram Engagement Rate Benchmarks*. Available at: https://hypeauditor.com/blog/ (Accessed: 3 September 2025).
  + Meta (2024) *Instagram Creators: Best practices for hashtags*. Available at: [https://www.instagram.com/creators/](https://www.instagram.com/creators/?utm_source=chatgpt.com) (Accessed: 3 September 2025).
  + Microsoft (2024a) *Use the Analysis ToolPak to perform complex data analysis*. Available at: https://support.microsoft.com/office/analysis-toolpak (Accessed: 3 September 2025).
  + Microsoft (2024b) *Apply conditional formatting to highlight information*. Available at: https://support.microsoft.com/office/conditional-formatting (Accessed: 3 September 2025).
  + BetterSolutions (n.d.) *Analysis ToolPak – Regression*. Available at: [https://bettersolutions.com/excel/add-ins/analysis-toolpak-regression.htm](https://bettersolutions.com/excel/add-ins/analysis-toolpak-regression.htm?utm_source=chatgpt.com) (Accessed: 3 September 2025).

### Visual Communication and Dashboarding

* Few, S. (2013) *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring*. 2nd edn. Burlingame, CA: Analytics Press.

### Impact Evaluation

* boyd, d. and Crawford, K. (2012) ‘Critical questions for big data’, Information, Communication & Society, 15(5), pp. 662–679.
* European Parliament and Council (2016) ‘Regulation (EU) 2016/679 (General Data Protection Regulation)’, Official Journal of the European Union, L119, pp. 1–88.
* Kohavi, R., Longbotham, R., Sommerfield, D. and Henne, R.M. (2009) ‘Controlled experiments on the web: survey and practical guide’, Data Mining and Knowledge Discovery, 18(1), pp. 140–181.

### Future Recommendations

* + European Parliament and Council (2016) ‘Regulation (EU) 2016/679 (General Data Protection Regulation)’, *Official Journal of the European Union*, L119, pp. 1–88.
  + Meta (2024) *Instagram Graph API overview*. Available at: https://developers.facebook.com/docs/instagram-api (Accessed: 3 September 2025).
  + Microsoft (2024) *Get & Transform (Power Query) in Excel*. Available at: https://learn.microsoft.com/excel/power-query/ (Accessed: 3 September 2025).
  + Kohavi, R., Longbotham, R., Sommerfield, D. and Henne, R.M. (2009) ‘Controlled experiments on the web: survey and practical guide’, *Data Mining and Knowledge Discovery*, 18(1), pp. 140–181.

## Appendices:

### Appendix A

**Thematic Linear Regression Models**











### Appendix B

**Instagram core data set Power Query Editor Code**

let

Source = Csv.Document(File.Contents("C:\Users\jorda\Downloads\Raw\_Hiking\_Data\_Set.csv"),[Delimiter=",", Columns=39, Encoding=65001, QuoteStyle=QuoteStyle.Csv]),

#"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),

#"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"url", type text}, {"user\_posted", type text}, {"description", type text}, {"hashtags", type text}, {"num\_comments", Int64.Type}, {"date\_posted", type datetime}, {"likes", Int64.Type}, {"photos", type text}, {"videos", type text}, {"location", type text}, {"latest\_comments", type text}, {"post\_id", Int64.Type}, {"discovery\_input", type text}, {"has\_handshake", type text}, {"shortcode", type text}, {"content\_type", type text}, {"pk", Int64.Type}, {"content\_id", type text}, {"engagement\_score\_view", Int64.Type}, {"thumbnail", type text}, {"video\_view\_count", Int64.Type}, {"product\_type", type text}, {"coauthor\_producers", type text}, {"tagged\_users", type text}, {"video\_play\_count", Int64.Type}, {"followers", Int64.Type}, {"posts\_count", Int64.Type}, {"profile\_image\_link", type text}, {"is\_verified", type logical}, {"is\_paid\_partnership", type logical}, {"partnership\_details", type text}, {"user\_posted\_id", Int64.Type}, {"post\_content", type text}, {"audio", type text}, {"profile\_url", type text}, {"videos\_duration", type text}, {"images", type text}, {"alt\_text", type text}, {"photos\_number", Int64.Type}}),

#"Removed Columns" = Table.RemoveColumns(#"Changed Type",{"has\_handshake", "discovery\_input", "shortcode", "pk", "content\_id", "profile\_image\_link", "user\_posted\_id", "post\_id", "user\_posted", "photos", "videos", "thumbnail", "images", "alt\_text", "profile\_url", "partnership\_details", "latest\_comments"}),

#"Added Custom2" = Table.AddColumn(#"Removed Columns", "Video Duration", each let

raw = [videos\_duration],

parsed = try Json.Document(raw),

duration = if parsed[HasError] then null else parsed[Value]{0}[video\_duration]

in

duration),

#"Added Custom" = Table.AddColumn(#"Added Custom2", "Like Type", each if [likes] = null or [likes] = "" then "Private" else "Public"),

#"Duplicated Column" = Table.DuplicateColumn(#"Added Custom", "date\_posted", "date\_posted - Copy"),

#"Renamed Columns" = Table.RenameColumns(#"Duplicated Column",{{"date\_posted - Copy", "Posted Time"}}),

#"Changed Type1" = Table.TransformColumnTypes(#"Renamed Columns",{{"Posted Time", type time}}),

#"Changed Type2" = Table.TransformColumnTypes(#"Changed Type1",{{"date\_posted", type date}}),

#"Filtered Rows" = Table.SelectRows(#"Changed Type2", each ([Like Type] = "Public")),

#"Added Custom1" = Table.AddColumn(#"Filtered Rows", "Custom", each if [hashtags] = null or [hashtags] = "" then 0

else List.Count(List.Transform(Text.Split(Text.Middle([hashtags], 2, Text.Length([hashtags]) - 2), ","), each Text.Trim(Text.Replace(Text.Replace(\_, """", ""), "#", ""))))),

#"Changed Type3" = Table.TransformColumnTypes(#"Added Custom1",{{"Video Duration", Int64.Type}}),

#"Renamed Columns1" = Table.RenameColumns(#"Changed Type3",{{"Custom", "Hashtag Volume"}}),

#"Removed Errors" = Table.RemoveRowsWithErrors(#"Renamed Columns1", {"Video Duration"}),

#"Changed Type4" = Table.TransformColumnTypes(#"Removed Errors",{{"Video Duration", type text}}),

#"Removed Columns1" = Table.RemoveColumns(#"Changed Type4",{"videos\_duration"}),

#"Parsed JSON" = Table.TransformColumns(#"Removed Columns1",{{"audio", Json.Document}}),

#"Expanded audio" = Table.ExpandRecordColumn(#"Parsed JSON", "audio", {"ig\_artist\_id", "original\_audio\_title"}, {"audio.ig\_artist\_id", "audio.original\_audio\_title"}),

#"Renamed Columns2" = Table.RenameColumns(#"Expanded audio",{{"audio.original\_audio\_title", "Audio Title"}, {"audio.ig\_artist\_id", "Audio Artist"}}),

#"Added Custom3" = Table.AddColumn(#"Renamed Columns2", "Custom", each if [Audio Title] = null then "No" else "Yes"),

#"Renamed Columns3" = Table.RenameColumns(#"Added Custom3",{{"Custom", "Audio Used"}}),

#"Removed Columns2" = Table.RemoveColumns(#"Renamed Columns3",{"post\_content"}),

#"Added Custom4" = Table.AddColumn(#"Removed Columns2", "Location Included", each if Text.Trim(Text.From([location])) = "" then "no" else "yes"),

#"Removed Columns3" = Table.RemoveColumns(#"Added Custom4",{"location"}),

#"Added Custom5" = Table.AddColumn(#"Removed Columns3", "Custom", each Json.Document([tagged\_users])),

#"Added Custom6" = Table.AddColumn(#"Added Custom5", "Custom.1", each List.Count([Custom])),

#"Renamed Columns4" = Table.RenameColumns(#"Added Custom6",{{"Custom.1", "Tagged Users"}}),

#"Removed Columns5" = Table.RemoveColumns(#"Renamed Columns4",{"Custom"}),

#"Replaced Errors" = Table.ReplaceErrorValues(#"Removed Columns5", {{"Tagged Users", 0}}),

#"Removed Columns4" = Table.RemoveColumns(#"Replaced Errors",{"tagged\_users"}),

#"Parsed JSON1" = Table.TransformColumns(#"Removed Columns4",{{"coauthor\_producers", Json.Document}}),

#"Added Custom7" = Table.AddColumn(#"Parsed JSON1", "Author Volume", each List.Count([coauthor\_producers])),

#"Replaced Errors1" = Table.ReplaceErrorValues(#"Added Custom7", {{"Author Volume", 0}}),

#"Removed Columns6" = Table.RemoveColumns(#"Replaced Errors1",{"coauthor\_producers"}),

#"Added Custom8" = Table.AddColumn(#"Removed Columns6", "Post Description", each Text.Combine(

List.Select(

Text.Split(Text.From([description]), " "),

each not Text.StartsWith(\_, "#")

),

" "

)),

#"Parsed JSON2" = Table.TransformColumns(#"Added Custom8",{{"hashtags", Json.Document}}),

#"Removed Columns7" = Table.RemoveColumns(#"Parsed JSON2",{"hashtags", "engagement\_score\_view"}),

#"Added Custom9" = Table.AddColumn(#"Removed Columns7", "Custom", each Text.BeforeDelimiter([Post Description], "#")),

#"Removed Columns8" = Table.RemoveColumns(#"Added Custom9",{"Post Description"}),

#"Renamed Columns5" = Table.RenameColumns(#"Removed Columns8",{{"Custom", "Post Description"}}),

#"Added Custom10" = Table.AddColumn(#"Renamed Columns5", "Custom", each Text.Length([Post Description])),

#"Renamed Columns6" = Table.RenameColumns(#"Added Custom10",{{"Custom", "Post Description Characters"}}),

#"Added Custom11" = Table.AddColumn(#"Renamed Columns6", "Post Description Word Count", each List.Count(Text.Split(Text.Trim([Post Description]), " "))),

#"Added Custom12" = Table.AddColumn(#"Added Custom11", "Post Decription Bucketing", each if Text.Length([Post Description]) < 100 then "Short" else if Text.Length([Post Description]) < 250 then "Medium" else "Long"),

#"Added Custom13" = Table.AddColumn(#"Added Custom12", "Post Link", each if Text.Contains([Post Description], "http") then 1 else 0),

#"Added Custom14" = Table.AddColumn(#"Added Custom13", "Custom", each let

text = Text.Lower([Post Description]),

positiveWords = {"awesome", "cool", "great", "love", "fun", "exciting", "amazing", "fantastic"},

negativeWords = {"bad", "boring", "hate", "terrible", "sad", "annoying", "awful", "disappointing"},

posCount = List.Count(List.Select(positiveWords, each Text.Contains(text, \_))),

negCount = List.Count(List.Select(negativeWords, each Text.Contains(text, \_))),

sentiment = if posCount > negCount then "Positive"

else if negCount > posCount then "Negative"

else "Neutral"

in

sentiment),

#"Renamed Columns7" = Table.RenameColumns(#"Added Custom14",{{"Custom", "Post Sentiment"}}),

#"Removed Columns9" = Table.RemoveColumns(#"Renamed Columns7",{"Post Description", "description"}),

#"Reordered Columns" = Table.ReorderColumns(#"Removed Columns9",{"url", "date\_posted", "num\_comments", "likes", "content\_type", "video\_view\_count", "product\_type", "video\_play\_count", "followers", "posts\_count", "is\_verified", "is\_paid\_partnership", "Audio Artist", "Audio Title", "photos\_number", "Video Duration", "Like Type", "Posted Time", "Hashtag Volume", "Audio Used", "Location Included", "Tagged Users", "Author Volume", "Post Description Characters", "Post Description Word Count", "Post Decription Bucketing", "Post Link", "Post Sentiment"}),

#"Removed Columns10" = Table.RemoveColumns(#"Reordered Columns",{"url"}),

#"Reordered Columns1" = Table.ReorderColumns(#"Removed Columns10",{"date\_posted", "Posted Time", "Like Type", "content\_type", "product\_type", "followers", "is\_verified", "num\_comments", "likes", "Hashtag Volume", "photos\_number", "video\_view\_count", "video\_play\_count", "Video Duration", "posts\_count", "is\_paid\_partnership", "Audio Artist", "Audio Title", "Audio Used", "Location Included", "Tagged Users", "Author Volume", "Post Description Characters", "Post Description Word Count", "Post Decription Bucketing", "Post Link", "Post Sentiment"}),

#"Renamed Columns8" = Table.RenameColumns(#"Reordered Columns1",{{"date\_posted", "Posted Date"}, {"content\_type", "Content Type"}, {"is\_verified", "Verified"}, {"num\_comments", "Comment Volume"}}),

#"Replaced Value" = Table.ReplaceValue(#"Renamed Columns8",null,0,Replacer.ReplaceValue,{"video\_play\_count"}),

#"Changed Type5" = Table.TransformColumnTypes(#"Replaced Value",{{"video\_play\_count", Int64.Type}}),

#"Replaced Value1" = Table.ReplaceValue(#"Changed Type5",null,0,Replacer.ReplaceValue,{"video\_view\_count"}),

#"Added Custom15" = Table.AddColumn(#"Replaced Value1", "Custom", each [Comment Volume]+[likes]+[video\_view\_count]+[video\_play\_count]),

#"Renamed Columns9" = Table.RenameColumns(#"Added Custom15",{{"Custom", "Engagement Volume"}}),

#"Reordered Columns2" = Table.ReorderColumns(#"Renamed Columns9",{"Posted Date", "Posted Time", "Like Type", "Content Type", "product\_type", "followers", "Verified", "Comment Volume", "likes", "video\_view\_count", "video\_play\_count", "Engagement Volume", "Hashtag Volume", "photos\_number", "Video Duration", "posts\_count", "is\_paid\_partnership", "Audio Artist", "Audio Title", "Audio Used", "Location Included", "Tagged Users", "Author Volume", "Post Description Characters", "Post Description Word Count", "Post Decription Bucketing", "Post Link", "Post Sentiment"}),

#"Added Custom16" = Table.AddColumn(#"Reordered Columns2", "Engagement rate", each [Engagement Volume]/[followers]),

#"Changed Type6" = Table.TransformColumnTypes(#"Added Custom16",{{"Engagement rate", Percentage.Type}}),

#"Removed Errors1" = Table.RemoveRowsWithErrors(#"Changed Type6", {"Engagement rate"}),

#"Reordered Columns3" = Table.ReorderColumns(#"Removed Errors1",{"Posted Date", "Posted Time", "Like Type", "Content Type", "product\_type", "followers", "Verified", "Comment Volume", "likes", "video\_view\_count", "video\_play\_count", "Engagement Volume", "Engagement rate", "Hashtag Volume", "photos\_number", "Video Duration", "posts\_count", "is\_paid\_partnership", "Audio Artist", "Audio Title", "Audio Used", "Location Included", "Tagged Users", "Author Volume", "Post Description Characters", "Post Description Word Count", "Post Decription Bucketing", "Post Link", "Post Sentiment"}),

#"Added Custom17" = Table.AddColumn(#"Reordered Columns3", "Day of Week", each Date.DayOfWeekName([Posted Date])),

#"Added Custom18" = Table.AddColumn(#"Added Custom17", "Custom", each Time.Hour([Posted Time])),

#"Added Custom19" = Table.AddColumn(#"Added Custom18", "Time of Day", each if([Custom]) >= 5 and ([Custom]) < 12 then "Morning"

else if ([Custom]) >= 12 and ([Custom]) < 17 then "Afternoon"

else if ([Custom]) >= 17 and ([Custom]) < 21 then "Evening"

else "Night"),

#"Removed Columns11" = Table.RemoveColumns(#"Added Custom19",{"Custom"}),

#"Reordered Columns4" = Table.ReorderColumns(#"Removed Columns11",{"Posted Date", "Day of Week", "Posted Time", "Time of Day", "Like Type", "Content Type", "product\_type", "followers", "Verified", "Comment Volume", "likes", "video\_view\_count", "video\_play\_count", "Engagement Volume", "Engagement rate", "Hashtag Volume", "photos\_number", "Video Duration", "posts\_count", "is\_paid\_partnership", "Audio Artist", "Audio Title", "Audio Used", "Location Included", "Tagged Users", "Author Volume", "Post Description Characters", "Post Description Word Count", "Post Decription Bucketing", "Post Link", "Post Sentiment"}),

#"Inserted Merged Column" = Table.AddColumn(#"Reordered Columns4", "Merged", each Text.Combine({Text.From([Posted Date], "en-GB"), Text.From([Posted Time], "en-GB"), [Content Type], Text.From([followers], "en-GB"), Text.From([Comment Volume], "en-GB"), Text.From([likes], "en-GB"), Text.From([video\_view\_count], "en-GB"), Text.From([video\_play\_count], "en-GB"), Text.From([Hashtag Volume], "en-GB"), Text.From([posts\_count], "en-GB"), Text.From([Post Description Characters], "en-GB")}, "/"), type text),

#"Renamed Columns10" = Table.RenameColumns(#"Inserted Merged Column",{{"Merged", "Unique ID"}}),

#"Reordered Columns5" = Table.ReorderColumns(#"Renamed Columns10",{"Unique ID", "Posted Date", "Day of Week", "Posted Time", "Time of Day", "Like Type", "Content Type", "product\_type", "followers", "Verified", "Comment Volume", "likes", "video\_view\_count", "video\_play\_count", "Engagement Volume", "Engagement rate", "Hashtag Volume", "photos\_number", "Video Duration", "posts\_count", "is\_paid\_partnership", "Audio Artist", "Audio Title", "Audio Used", "Location Included", "Tagged Users", "Author Volume", "Post Description Characters", "Post Description Word Count", "Post Decription Bucketing", "Post Link", "Post Sentiment"}),

#"Replaced Value2" = Table.ReplaceValue(#"Reordered Columns5",null,"N/A",Replacer.ReplaceValue,{"Audio Artist"}),

#"Replaced Value3" = Table.ReplaceValue(#"Replaced Value2",null,"N/A",Replacer.ReplaceValue,{"Audio Title"}),

#"Replaced Value4" = Table.ReplaceValue(#"Replaced Value3",null,0,Replacer.ReplaceValue,{"photos\_number"}),

#"Replaced Value5" = Table.ReplaceValue(#"Replaced Value4",null,"0",Replacer.ReplaceValue,{"Video Duration"}),

#"Removed Columns12" = Table.RemoveColumns(#"Replaced Value5",{"product\_type"}),

#"Added Custom20" = Table.AddColumn(#"Removed Columns12", "Row Count", each 1),

#"Filtered Rows1" = Table.SelectRows(#"Added Custom20", each [Engagement rate] < 0.25)

in

#"Filtered Rows1"

### Appendix C

**Hashtag data set Power Query Editor Code**

let

Source = Csv.Document(File.Contents("C:\Users\jorda\Downloads\Raw\_Hiking\_Data\_Set.csv"),[Delimiter=",", Columns=39, Encoding=65001, QuoteStyle=QuoteStyle.Csv]),

#"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),

#"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"url", type text}, {"user\_posted", type text}, {"description", type text}, {"hashtags", type text}, {"num\_comments", Int64.Type}, {"date\_posted", type datetime}, {"likes", Int64.Type}, {"photos", type text}, {"videos", type text}, {"location", type text}, {"latest\_comments", type text}, {"post\_id", Int64.Type}, {"discovery\_input", type text}, {"has\_handshake", type text}, {"shortcode", type text}, {"content\_type", type text}, {"pk", Int64.Type}, {"content\_id", type text}, {"engagement\_score\_view", Int64.Type}, {"thumbnail", type text}, {"video\_view\_count", Int64.Type}, {"product\_type", type text}, {"coauthor\_producers", type text}, {"tagged\_users", type text}, {"video\_play\_count", Int64.Type}, {"followers", Int64.Type}, {"posts\_count", Int64.Type}, {"profile\_image\_link", type text}, {"is\_verified", type logical}, {"is\_paid\_partnership", type logical}, {"partnership\_details", type text}, {"user\_posted\_id", Int64.Type}, {"post\_content", type text}, {"audio", type text}, {"profile\_url", type text}, {"videos\_duration", type text}, {"images", type text}, {"alt\_text", type text}, {"photos\_number", Int64.Type}}),

#"Removed Columns" = Table.RemoveColumns(#"Changed Type",{"has\_handshake", "discovery\_input", "shortcode", "pk", "content\_id", "profile\_image\_link", "user\_posted\_id", "post\_id", "user\_posted", "photos", "videos", "thumbnail", "images", "alt\_text", "profile\_url", "partnership\_details", "latest\_comments"}),

#"Added Custom2" = Table.AddColumn(#"Removed Columns", "Video Duration", each let

raw = [videos\_duration],

parsed = try Json.Document(raw),

duration = if parsed[HasError] then null else parsed[Value]{0}[video\_duration]

in

duration),

#"Added Custom" = Table.AddColumn(#"Added Custom2", "Like Type", each if [likes] = null or [likes] = "" then "Private" else "Public"),

#"Duplicated Column" = Table.DuplicateColumn(#"Added Custom", "date\_posted", "date\_posted - Copy"),

#"Renamed Columns" = Table.RenameColumns(#"Duplicated Column",{{"date\_posted - Copy", "Posted Time"}}),

#"Changed Type1" = Table.TransformColumnTypes(#"Renamed Columns",{{"Posted Time", type time}}),

#"Changed Type2" = Table.TransformColumnTypes(#"Changed Type1",{{"date\_posted", type date}}),

#"Filtered Rows" = Table.SelectRows(#"Changed Type2", each ([Like Type] = "Public")),

#"Added Custom1" = Table.AddColumn(#"Filtered Rows", "Custom", each if [hashtags] = null or [hashtags] = "" then 0

else List.Count(List.Transform(Text.Split(Text.Middle([hashtags], 2, Text.Length([hashtags]) - 2), ","), each Text.Trim(Text.Replace(Text.Replace(\_, """", ""), "#", ""))))),

#"Changed Type3" = Table.TransformColumnTypes(#"Added Custom1",{{"Video Duration", Int64.Type}}),

#"Renamed Columns1" = Table.RenameColumns(#"Changed Type3",{{"Custom", "Hashtag Volume"}}),

#"Removed Errors" = Table.RemoveRowsWithErrors(#"Renamed Columns1", {"Video Duration"}),

#"Changed Type4" = Table.TransformColumnTypes(#"Removed Errors",{{"Video Duration", type text}}),

#"Removed Columns1" = Table.RemoveColumns(#"Changed Type4",{"videos\_duration"}),

#"Parsed JSON" = Table.TransformColumns(#"Removed Columns1",{{"audio", Json.Document}}),

#"Expanded audio" = Table.ExpandRecordColumn(#"Parsed JSON", "audio", {"ig\_artist\_id", "original\_audio\_title"}, {"audio.ig\_artist\_id", "audio.original\_audio\_title"}),

#"Renamed Columns2" = Table.RenameColumns(#"Expanded audio",{{"audio.original\_audio\_title", "Audio Title"}, {"audio.ig\_artist\_id", "Audio Artist"}}),

#"Added Custom3" = Table.AddColumn(#"Renamed Columns2", "Custom", each if [Audio Title] = null then "No" else "Yes"),

#"Renamed Columns3" = Table.RenameColumns(#"Added Custom3",{{"Custom", "Audio Used"}}),

#"Removed Columns2" = Table.RemoveColumns(#"Renamed Columns3",{"post\_content"}),

#"Added Custom4" = Table.AddColumn(#"Removed Columns2", "Location Included", each if Text.Trim(Text.From([location])) = "" then "no" else "yes"),

#"Removed Columns3" = Table.RemoveColumns(#"Added Custom4",{"location"}),

#"Added Custom5" = Table.AddColumn(#"Removed Columns3", "Custom", each Json.Document([tagged\_users])),

#"Added Custom6" = Table.AddColumn(#"Added Custom5", "Custom.1", each List.Count([Custom])),

#"Renamed Columns4" = Table.RenameColumns(#"Added Custom6",{{"Custom.1", "Tagged Users"}}),

#"Removed Columns5" = Table.RemoveColumns(#"Renamed Columns4",{"Custom"}),

#"Replaced Errors" = Table.ReplaceErrorValues(#"Removed Columns5", {{"Tagged Users", 0}}),

#"Removed Columns4" = Table.RemoveColumns(#"Replaced Errors",{"tagged\_users"}),

#"Parsed JSON1" = Table.TransformColumns(#"Removed Columns4",{{"coauthor\_producers", Json.Document}}),

#"Added Custom7" = Table.AddColumn(#"Parsed JSON1", "Author Volume", each List.Count([coauthor\_producers])),

#"Replaced Errors1" = Table.ReplaceErrorValues(#"Added Custom7", {{"Author Volume", 0}}),

#"Removed Columns6" = Table.RemoveColumns(#"Replaced Errors1",{"coauthor\_producers"}),

#"Added Custom8" = Table.AddColumn(#"Removed Columns6", "Post Description", each Text.Combine(

List.Select(

Text.Split(Text.From([description]), " "),

each not Text.StartsWith(\_, "#")

),

" "

)),

#"Parsed JSON2" = Table.TransformColumns(#"Added Custom8",{{"hashtags", Json.Document}}),

#"Removed Columns7" = Table.RemoveColumns(#"Parsed JSON2",{"url", "description", "Post Description"}),

#"Expanded hashtags" = Table.ExpandListColumn(#"Removed Columns7", "hashtags"),

#"Grouped Rows" = Table.Group(#"Expanded hashtags", {"hashtags", "content\_type", "product\_type", "is\_verified", "is\_paid\_partnership", "Audio Artist", "Audio Title", "Like Type", "Audio Used", "Location Included"}, {{"Comment Volume", each List.Average([num\_comments]), type nullable number}, {"Like Volume", each List.Average([likes]), type nullable number}, {"Engagement Score", each List.Average([engagement\_score\_view]), type nullable number}, {"Video View Count", each List.Average([video\_view\_count]), type nullable number}, {"Account Followers", each List.Average([followers]), type nullable number}, {"Account Post Count", each List.Average([posts\_count]), type nullable number}, {"Photo Volume", each List.Average([photos\_number]), type nullable number}, {"Video Duration", each List.Average([Video Duration]), type nullable text}, {"Hashtag Volume", each List.Average([Hashtag Volume]), type number}, {"Tagged Users", each List.Average([Tagged Users]), type number}, {"Author Volume", each List.Average([Author Volume]), type number}, {"Video Play Count", each List.Average([video\_play\_count]), type nullable number}}),

#"Changed Type5" = Table.TransformColumnTypes(#"Grouped Rows",{{"Comment Volume", Int64.Type}, {"Like Volume", Int64.Type}, {"Engagement Score", Int64.Type}, {"Video View Count", Int64.Type}, {"Account Followers", Int64.Type}, {"Account Post Count", Int64.Type}, {"Photo Volume", Int64.Type}, {"Video Duration", Int64.Type}, {"Hashtag Volume", Int64.Type}, {"Tagged Users", Int64.Type}, {"Author Volume", Int64.Type}}),

#"Replaced Errors2" = Table.ReplaceErrorValues(#"Changed Type5", {{"Video Duration", 0}}),

#"Reordered Columns" = Table.ReorderColumns(#"Replaced Errors2",{"hashtags", "content\_type", "product\_type", "Comment Volume", "Like Volume", "Video View Count", "Account Followers", "is\_verified", "is\_paid\_partnership", "Audio Artist", "Audio Title", "Like Type", "Audio Used", "Location Included", "Engagement Score", "Account Post Count", "Photo Volume", "Video Duration", "Hashtag Volume", "Tagged Users", "Author Volume", "Video Play Count"}),

#"Removed Columns8" = Table.RemoveColumns(#"Reordered Columns",{"Engagement Score"}),

#"Reordered Columns1" = Table.ReorderColumns(#"Removed Columns8",{"hashtags", "content\_type", "product\_type", "Comment Volume", "Like Volume", "Video View Count", "Video Play Count", "Account Followers", "is\_verified", "is\_paid\_partnership", "Audio Artist", "Audio Title", "Like Type", "Audio Used", "Location Included", "Account Post Count", "Photo Volume", "Video Duration", "Hashtag Volume", "Tagged Users", "Author Volume"}),

#"Replaced Value" = Table.ReplaceValue(#"Reordered Columns1",null,0,Replacer.ReplaceValue,{"Video View Count"}),

#"Added Custom10" = Table.AddColumn(#"Replaced Value", "volume", each 1),

#"Replaced Value1" = Table.ReplaceValue(#"Added Custom10",null,0,Replacer.ReplaceValue,{"Video Play Count"}),

#"Replaced Value3" = Table.ReplaceValue(#"Replaced Value1",null,0,Replacer.ReplaceValue,{"Account Followers"}),

#"Replaced Value2" = Table.ReplaceValue(#"Replaced Value3",null,0,Replacer.ReplaceValue,{"Comment Volume", "Like Volume"}),

#"Added Custom9" = Table.AddColumn(#"Replaced Value2", "Engagement Volume", each ([Comment Volume]+[Like Volume]+[Video View Count]+[Video Play Count])/[Account Followers]),

#"Removed Errors1" = Table.RemoveRowsWithErrors(#"Added Custom9", {"Engagement Volume"}),

#"Removed Blank Rows" = Table.SelectRows(#"Removed Errors1", each not List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(\_), {"", null}))),

#"Replaced Errors3" = Table.ReplaceErrorValues(#"Removed Blank Rows", {{"Engagement Volume", 0}}),

#"Filtered Rows1" = Table.SelectRows(#"Replaced Errors3", each (not Number.IsNaN([Engagement Volume]) and [Engagement Volume] <> 0 and [Engagement Volume] <> #infinity)),

#"Renamed Columns5" = Table.RenameColumns(#"Filtered Rows1",{{"Engagement Volume", "Engagement Rate"}}),

#"Added Custom11" = Table.AddColumn(#"Renamed Columns5", "Engagement Volume", each [Comment Volume]+[Like Volume]+[Video View Count]+[Video Play Count])

in

#"Added Custom11"