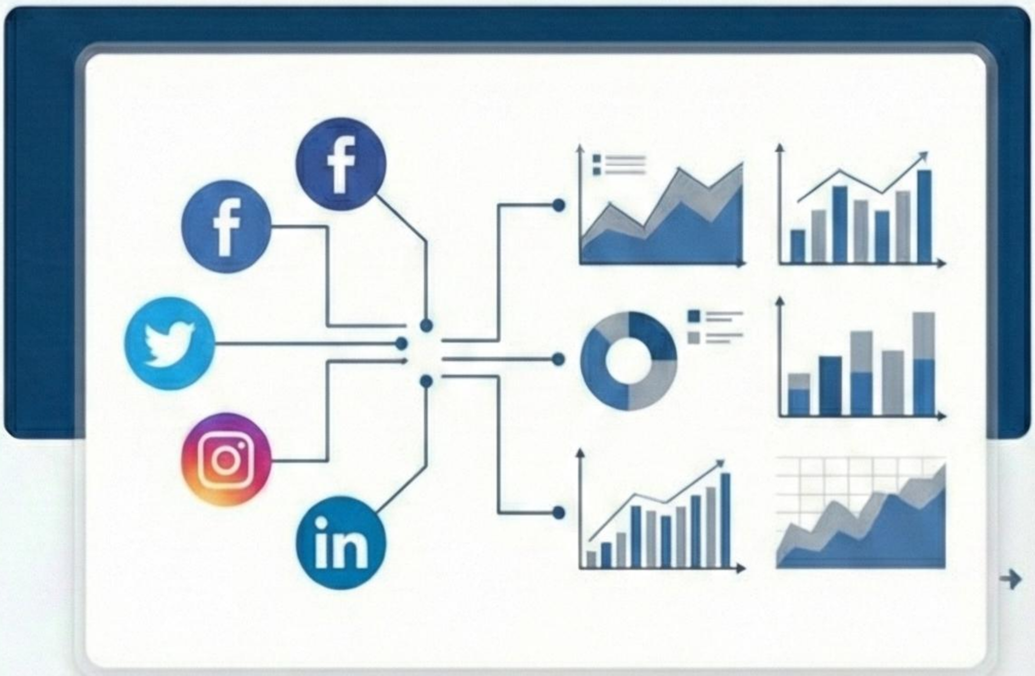
AMITY UNIVERSITY ONLINE

AMITY UNIVERSITY ONLINE

SOCIAL MEDIA ANALYTICS PROJECT

Reflection Document



Assignment: Reflection Document

Submitted by:
Hemant Borana
Semester: 5
Course: Data Modeling and Visualization (DMV)
Module: 5

Date:
December, 2025

SOCIAL MEDIA ANALYTICS PROJECT

Reflection Document

Student Name: Hemant Borana
Semester: 5
Course: Data Modeling and Visualization (DMV)
Module: 5
Date: December , 2025

Contents

PROJECT APPROACH 3

TECHNICAL CHALLENGES & SOLUTIONS 4

KEY LEARNINGS..... 6

DATA INSIGHTS & FINDINGS..... 8

PERSONAL REFLECTION 10

CONCLUSION 12

PROJECT APPROACH

1.1 Overall Strategy

When I first looked at this assignment, I realized the main challenge wasn't just loading different file formats, but creating a coherent story from fragmented data sources. My approach was to:

Phase 1: Understand the Data

- I started by thoroughly exploring each data source independently
- Identified the relationships between datasets (post_id, platform as join keys)
- Documented the structure and quirks of each format

Phase 2: Clean Systematically

- Rather than merging first and cleaning later, I cleaned each source independently
- This made it easier to track what transformations were applied where
- Added calculated fields that would be useful for later analysis

Phase 3: Strategic Integration

- Chose left joins to preserve all posts (our primary dataset)
- Created multiple views of the integrated data for different use cases
- Validated at each merge step to catch issues early

Phase 4: Visualize Insights

- Designed dashboard with audience in mind (marketing teams, data analysts)
- Focused on actionable metrics rather than just pretty charts
- Ensured each visualization answered a specific business question

1.2 Why This Approach Worked

This phased approach worked well because:

- **It was systematic:** Each phase had clear deliverables
- **It was reversible:** Could go back and fix issues without redoing everything
- **It was validated:** Checkpoints at each phase ensured quality
- **It was documented:** Easy to explain what was done and why

TECHNICAL CHALLENGES & SOLUTIONS

Challenge 1: Nested JSON Structure

Problem:
The user engagement JSON had a complex nested structure with engagement_history as an array within each user object. Standard pandas JSON reader couldn't flatten this properly.

My Solution:
I wrote a manual flattening loop that:

- Iterated through each user
- Extracted user-level attributes (user_id, join_date, favorite_platform)
- Looped through each engagement in the history array
- Created individual records preserving the user context
- Used .get() method for optional fields like sentiment

Learning:
Sometimes manual parsing gives you more control than automated tools. The extra code was worth it for clean, predictable results.

Challenge 2: XML Hierarchical Data

Problem:
XML data had multiple levels (platform → statistics/performance/demographics → individual metrics) and also had a separate weekly_metrics section. Needed to extract both flat statistics and time-series data.

My Solution:
Created two separate extraction loops:

- First loop: Platform-level statistics (one record per platform)
- Second loop: Weekly metrics (multiple records per platform)
- Used find() and findall() methods appropriately for navigation

Learning:
XML requires more thoughtful parsing than CSV/JSON. Understanding the structure before coding saves a lot of debugging time.

Challenge 3: Missing Data Strategy

Problem:
Different columns had different patterns of missing data:

- Categories: 5% missing (seemed random)
- Shares: 3% missing (could be actual zero or missing)
- Sentiment: 47% missing (but only expected for certain engagement types)

My Solution:
Used context-appropriate strategies:

- Categories: Filled with 'Uncategorized' (neutral, preserves record)
- Shares: Filled with 0 (reasonable assumption if not recorded)
- Sentiment: Left as None for non-comment engagements, created has_sentiment flag

Learning:
There's no one-size-fits-all approach to missing data. Understanding why data is missing matters more than just picking a fill method.

Challenge 4: Dashboard Readability

Problem:
Initial daily engagement trend chart had 164 data points and was completely unreadable with overlapping lines and cluttered date labels.

My Solution:

- Aggregated daily data to weekly data (reduced points from 164 to ~32)
- Increased line thickness and marker size
- Used platform-specific colors for brand recognition
- Set date formatter to show "Nov 01" style instead of "11-01"

Learning:
More data doesn't always mean better visualization. Sometimes aggregation improves clarity without losing the message.

Challenge 5: Bubble Chart Overlapping Labels

Problem:
Platform labels on the scatter plot were overlapping with bubbles and each other, making it hard to read.

My Solution:

- Positioned labels above bubbles with offset
- Added white background boxes with semi-transparent edges
- Increased bubble sizes for better visibility
- Added padding to axes so bubbles weren't cut off

Learning:
Good visualization is 50% data and 50% presentation. Small tweaks to positioning and colors make a huge difference.

KEY LEARNINGS

3.1 Technical Skills Developed

Multi-Format Data Handling:

- Gained hands-on experience with CSV (structured), JSON (semi-structured), and XML (hierarchical)
- Learned that each format requires different parsing strategies
- Understood when to use pandas built-in methods vs manual parsing

Data Integration Best Practices:

- Different join types serve different purposes (left, right, inner, outer)
- Always validate after merging (check row counts, look for nulls)
- Create intermediate datasets rather than one giant merge
- Document why you chose specific join strategies

Feature Engineering:

- Simple calculated fields can add significant analytical value
- Time-based features (hour, day, week) enable temporal analysis
- Categorizing continuous variables (time_spent → quick/medium/long) aids interpretation

3.2 Process Insights

Exploratory Data Analysis is Critical: Before writing any merging code, I spent time understanding:

- What each dataset represents
- What the grain/granularity is (post-level vs user-level vs platform-level)
- What the natural keys are
- What business questions the data can answer

This upfront time saved hours of confusion later.

Incremental Development Works: Rather than building everything at once, I:

- Loaded one source at a time
- Cleaned one dataset at a time
- Merged in steps with validation between each
- Built visualizations one by one

This made debugging much easier.

Documentation Pays Off: I documented as I went (in code comments and markdown cells) rather than at the end. This helped me:

- Remember why I made certain decisions
- Explain my work to others
- Catch inconsistencies early

DATA INSIGHTS & FINDINGS

4.1 Platform Performance Analysis

Instagram leads in reach but not engagement rate:

- Instagram: Highest total reach (480k+) but only 28.97% engagement rate
- Twitter: Lowest reach (231k) but highest engagement rate (51.68%)
- Finding: Reach ≠ Engagement. Twitter's smaller but more engaged audience might be more valuable.

Implication: For campaigns prioritizing conversions over impressions, Twitter might be the better platform despite lower reach.

4.2 Content Insights

Entertainment content dominates:

- Entertainment: 30% of all posts
- Business and Lifestyle: ~20-23% each
- Technology and Education: Under-represented at 11% each

Post type performance:

- Image posts get highest likes (avg ~1,600)
- Video posts have good balance across likes/shares/comments
- Text posts underperform on likes but get decent comments

Implication: Visual content (Images/Videos) drives engagement. Text posts might work better for thought leadership than viral reach.

4.3 User Behavior Patterns

Engagement timing:

- Most engagement happens between 9 AM and 6 PM (business hours)
- Spikes around 12 PM (lunch) and 6 PM (post-work)
- Different platforms have different optimal posting times

Time spent analysis:

- Negative sentiment content gets longest engagement time (148.5s)
- Positive sentiment: 158.6s
- Neutral sentiment: 163.4s

Finding: Users spend more time on neutral/positive content, suggesting they're more thoughtfully engaging rather than hate-scrolling.

Engagement types distribution:

- Like_comment (158) and Like (132) are most common
- Comments (127) and Shares (122) are less frequent but more valuable
- Finding: Most users engage passively; converting them to active engagement (comments/shares) is key

4.4 Weekly Trends

Consistency varies by platform:

- Facebook: Relatively stable week-to-week
- Instagram: More volatile, likely influenced by algorithm changes
- Twitter: High variability, possibly event-driven
- LinkedIn: Most consistent, professional audience with routine behavior

Implication: Facebook and LinkedIn are reliable for consistent reach, while Instagram and Twitter offer viral potential but are less predictable.

Sentiment Analysis Expansion:

- Currently only 47% of engagements have sentiment
- Apply NLP to comment text for automated sentiment detection
- Analyze sentiment trends over time

User Segmentation:

- Cluster users based on behavior patterns
- Identify high-value users (frequent engagers, positive sentiment)
- Create targeted content strategies per segment

Content Optimization:

- A/B testing framework for post variations
- Optimal posting time recommendations per platform
- Content calendar suggestions based on historical performance

PERSONAL REFLECTION

6.1 What Went Well

Problem-Solving Approach: I'm proud of how I tackled the nested JSON challenge. Instead of getting frustrated, I broke it down step-by-step and wrote clean, understandable code. The flattening logic works and is easy to maintain.

Visualization Design: The dashboard evolved through several iterations. I didn't settle for the first version – I kept improving readability based on what story the data was telling. The final version effectively communicates insights without being overwhelming.

Documentation Discipline: I documented as I worked rather than retroactively. This made writing these reflection and documentation reports much easier and ensured I captured my reasoning in the moment.

6.2 What I Struggled With

Initial Data Model Planning: I jumped into coding before fully mapping out how the datasets would connect. This led to some backtracking when I realized certain fields needed to be standardized before merging. Next time, I'll sketch out the data model first.

Time Management: I underestimated how long XML parsing would take. The hierarchical structure was more complex than expected. I should have allocated more time for the semi-structured data sources.

Feature Selection for Dashboard: I created many calculated fields during cleaning, but didn't use all of them in visualizations. I could have been more strategic about what features would actually drive insights.

6.3 Skills I Developed

Technical Skills:

- Multi-format data parsing (CSV, JSON, XML)
- Complex pandas operations (groupby, merge, pivot)
- Data validation and quality checking
- Visualization design principles

Analytical Skills:

- Translating business questions into data queries
- Identifying patterns and anomalies in data
- Drawing actionable insights from visualizations

Soft Skills:

- Breaking complex problems into manageable steps
- Documenting technical work for non-technical audiences
- Iterative improvement based on results

6.4 How This Applies to Real Work

This project mimics real-world data integration scenarios I'll face:

Marketing Analytics:

- Social media managers need exactly this type of cross-platform analysis
- Real-time monitoring is critical for campaign management
- Understanding engagement patterns informs content strategy

Data Engineering:

- Combining multiple data sources is a daily task
- Handling messy, real-world data with missing values
- Building reliable, validated data pipelines

Business Intelligence:

- Creating dashboards that tell a story
- Balancing detail with clarity
- Providing actionable recommendations, not just data

6.5 What I Would Do Differently

Start with Questions: Next time, I'd begin by defining specific business questions the dashboard should answer. This would guide what data to collect, what features to engineer, and what visualizations to create.

Automate Validation: I did manual validation checks, but I should write automated test functions that run after each transformation. This would catch issues faster and make the pipeline more production-ready.

Consider the Audience: I built the dashboard for a generic "data analyst" audience. If I knew specifically who would use it (executives vs. content creators vs. data scientists), I could tailor the complexity and focus accordingly.

Plan for Scale: The current solution works for 150 posts and 539 engagements. But what if it's 150,000 posts? I should consider:

- Database storage instead of in-memory dataframes
- Incremental updates instead of full reprocessing
- Sampling or aggregation for visualization performance

CONCLUSION

This project was an excellent exercise in end-to-end data integration. I learned that successful data integration is less about technical complexity and more about:

1. **Understanding your data** before you start coding
2. **Choosing the right tool** for each format (not forcing everything into the same approach)
3. **Validating continuously** rather than at the end
4. **Communicating insights** effectively through visualization
5. **Documenting thoroughly** for maintainability

The combination of structured (CSV), semi-structured (JSON, XML), and conceptual (streaming) data sources gave me confidence to handle diverse real-world scenarios. The visualization component reminded me that data without interpretation is just numbers – the goal is always to inform decisions.

I'm particularly proud of the systematic approach I developed. While the assignment had specific requirements, the methodology I used – explore, clean, integrate, visualize, validate – is transferable to any data integration project.

Most importantly: I now understand that data integration isn't a one-time task. In production, this would be an ongoing process with new data arriving constantly, quality issues emerging, and requirements evolving. Building maintainable, well-documented, validated pipelines is key to long-term success.

This project has prepared me to tackle complex data integration challenges in professional settings with confidence

AMITY UNIVERSITY ONLINE

THANK YOU

FOR YOUR TIME AND ATTENTION

CONTACT INFORMATION

Phone: +91 9284494154
Email: hemantpb123@gmail.com
LinkedIn: www.linkedin.com/in/hemant-parasmal-borana

Submitted by:
Hemant Borana

Date:
December 2025