Digital Business Analytics (DS-464)

Semester Project: Complete Data Analysis Report

Project Title:

Formula 1 Engagement & Predictive Modeling

YouTube Analytics: A Comprehensive Digital Footprint Analysis

Team BW Members:

Name	Student ID
Ahmed Musharaf	2022067
Muhammad Arsal	2022350
Saaim Ali Khan	2022519
Manal Ahsan	2022279
Amaan Zahir	2022902

Instructor:
Sir Hafiz Syed Muhammad Muslim

Submission Date:
December 29, 2025

Repository Link:
https://github.com/ahmedmusharaf31/dba-youtube-project

Abstract

This report documents Team BW's end-to-end Digital Business Analytics semester project centered on the Formula 1 (F1) ecosystem using YouTube as the primary source of fan interaction and engagement signals. Since F1 is highly visual and narrative-driven, YouTube serves as a rich environment to observe fan attention, emotional reaction, and discussion intensity through video performance metrics and comment-level sentiment.

Our solution is structured as a modular analytics system with two major components: (i) **Descriptive Analytics**, which extracts and transforms YouTube video metadata and comment threads to quantify engagement, Share of Voice (SoV), sentiment, and topic trends; and (ii) **Predictive Analytics**, which leverages structured race-performance data (via OpenF1 API) to engineer performance Key Performance Indicators (KPIs) and forecast season outcomes using a composite scoring model.

Key findings indicate that controversy-driven narratives (incidents, penalties, rivalries) generate disproportionately higher engagement than purely celebratory moments. Moreover, short-form content maximizes reach (views), while long-form analysis yields higher engagement quality (comments and likes per view), supporting community retention. This report also provides the system architecture, methodology, KPI definitions, evaluation strategy, reproducibility instructions, limitations, and future scope.

Keywords: YouTube Data API, OpenF1 API, ETL, NLP, Sentiment Analysis, Share of Voice, Predictive Modeling, Composite Score, Engagement Analytics

Contents

List of Figures

List of Tables

1 Executive Summary

This project delivers a complete analytics workflow that converts raw digital signals into actionable insights and forecasts. We selected **YouTube** as the central platform for descriptive analysis due to its:

- Rich engagement signals (views, likes, comments, publish time, duration),
- Comment threads as a high-volume sentiment dataset,
- Strong alignment with F1 media consumption (highlights, analysis, incidents).

To complement fan discourse with objective performance data, we integrated the **OpenF1 API** into a predictive modeling pipeline. The final deliverable includes:

- A reproducible ETL pipeline for YouTube data extraction and cleaning,
- Descriptive analytics on drivers, narratives, sentiment, and content formats,
- Predictive season outlook using engineered KPIs and a composite scoring model,
- A clear repository structure enabling separate scaling of descriptive and predictive modules.

2 Introduction

The exponential growth of digital media platforms has transformed how sports organizations, athletes, and fans interact. Among these platforms, **YouTube** has emerged as a dominant medium for long-form analysis, highlight dissemination, and community-driven discussion. In contrast to text-centric platforms, YouTube uniquely combines visual storytelling with measurable engagement metrics, making it ideal for advanced Digital Business Analytics (DBA).

Formula 1 (F1), as a globally followed and visually intensive sport, generates significant digital footprints through race highlights, technical breakdowns, and fan-driven commentary. These traces provide an opportunity to quantify attention, emotional response, and discussion intensity at scale. Understanding such patterns is valuable for teams, broadcasters, sponsors, and content creators seeking to optimize reach, monetization, and brand positioning.

This project leverages YouTube engagement data and official race telemetry to build a unified analytical framework addressing both *descriptive* (what happened and why) and *predictive* (what is likely to happen next) questions.

3 Problem Statement and Project Objectives

3.1 Business Motivation

F1 is an attention economy: teams, sponsors, broadcasters, and creators compete for fan attention. Understanding what drives engagement and how fans react enables better content strategy, better sponsor alignment, and stronger narrative framing.

3.2 Core Questions

The project addresses the following questions:

1. Which drivers dominate digital attention on YouTube (Share of Voice)?
2. What content types (short highlights versus long analysis) maximize reach versus retention?
3. How does fan sentiment vary across incidents (penalties and crashes) versus celebrations?
4. Can we forecast season outcomes using structured performance KPIs beyond points?

3.3 Project Objectives

- Build a scalable ETL pipeline to ingest YouTube data (videos and comments).
- Transform raw JSON into analysis-ready structured datasets.
- Compute descriptive metrics: engagement rate, SoV, sentiment distribution, topic trends, and content performance.
- Engineer predictive KPIs from OpenF1 race data and forecast standings.
- Ensure reproducibility through clean project structure, caching, and run instructions.

4 Project Architecture and Repository Structure

4.1 GitHub Repository

URL: https://github.com/ahmedmusharaf31/dba-youtube-project

4.2 High-Level Architecture

Figure 1 illustrates the system-level flow.

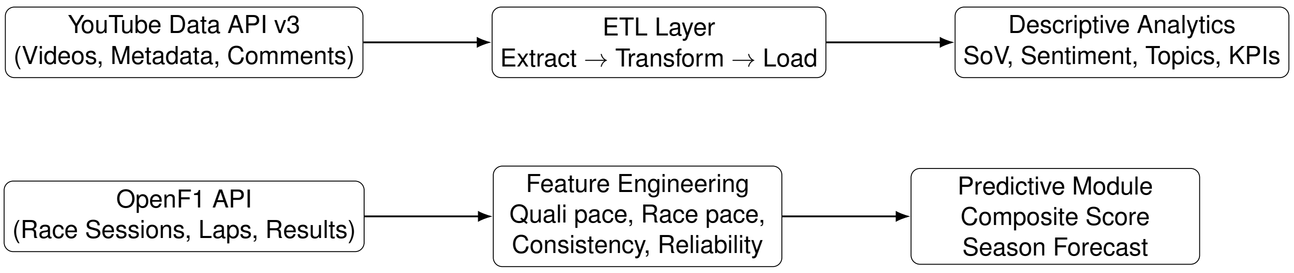
YouTube Data API v3
(Videos, Metadata, Comments)
ETL Layer
Extract → Transform → Load
Descriptive Analytics
SoV, Sentiment, Topics, KPIs
OpenF1 API
(Race Sessions, Laps, Results)
Feature Engineering
Quali pace, Race pace, Consistency, Reliability
Predictive Module
Composite Score
Season Forecast

Figure 1: System architecture: descriptive pipeline (YouTube) and predictive pipeline (OpenF1).

4.3 Directory Structure

The project is organized as follows:

```
dba-youtube-project
|---Descriptive
|   |---data
|   |   |---processed  (Cleaned, analysis-ready CSVs)
|   |   |---raw        (Raw JSON/CSV API responses)
|   |---notebooks      (EDA & Visualization environments)
|   |---src            (ETL & utility modules)
|   |---run_analytics.py (Primary execution entry point)
|---Predictive
    |---f1_data_cache  (Cached historical API data)
    |---main_script.py (Predictive modeling engine)
```

Listing 1: Project Directory Tree

4.4 Separation of Concerns

We intentionally split the codebase into two modules:

- **Descriptive module** focuses on fan discourse and media performance signals.

- **Predictive module** focuses on performance forecasting from structured telemetry and race data.

This separation improves maintainability, reduces coupling, and allows independent improvement of NLP methods versus forecasting logic.

5 Data Description

5.1 YouTube Data API (Descriptive Data)

The descriptive analytics component uses the **YouTube Data API v3**, which provides structured access to publicly available video and comment metadata. The dataset includes:

- Video-level metadata: titles, descriptions, publish timestamps, durations.
- Performance metrics: view count, like count, comment count.
- Comment text: fan sentiment and discussion signals.

5.2 OpenF1 API (Predictive Data)

The predictive component uses the **OpenF1 API** to obtain structured race-performance data, including:

- Session and event metadata.
- Lap-by-lap timing data (race and qualifying).
- Driver positions and race classifications.
- Reliability indicators such as DNFs.

5.3 Why Two Data Sources?

YouTube captures *fan attention and narrative*, while OpenF1 captures *performance reality*. Combining both enables:

- Reliable performance forecasting rooted in objective data,
- Actionable engagement insights rooted in audience behavior,
- A business-oriented interpretation connecting narrative and competition.

6 Data Pipeline and Methodology

6.1 ETL Design Principles

Our ETL design prioritizes:

- **Repeatability:** runs produce consistent outputs in `Descriptive/data/processed`.
- **Traceability:** raw API responses stored in `Descriptive/data/raw` for audit and debugging.
- **Robustness:** pagination handling, quota-aware extraction, and caching.
- **Analysis readiness:** flat tables (CSV) with standardized types and timezones.

6.2 Descriptive Pipeline (YouTube)

6.2.1 Extract

The extraction stage retrieves:

- Video metadata: title, description, publish time, duration, view count, like count.
- Comment threads: top-level comments and replies (where available).

Extraction must handle pagination and partial failures (for example, comments disabled or limited).

6.2.2 Transform

Key transformations include:

- **Flatten JSON** into tabular schema (video table and comment table).
- **Timestamp normalization** to a consistent timezone (UTC recommended).
- **Text cleaning:** remove URLs, repeated whitespace, and non-alphanumeric noise for NLP.
- **Feature creation:** engagement rate, comments per view, likes per view, upload hour and weekday.

A standard engagement rate proxy used in this report is:

$$\text{Engagement Rate} = \frac{\text{likes} + \text{comments}}{\text{views}}$$

6.2.3 Load

Final outputs are saved as analysis-ready files in `Descriptive/data/processed`, suitable for notebooks and dashboards (Looker Studio and similar tools).

6.3 Predictive Pipeline (OpenF1)

6.3.1 Extract and Cache

The predictive module collects historical race sessions and lap-level data. Caching under `Predictive/f1_data_ca`
reduces repeated calls and speeds retraining.

6.3.2 Feature Engineering

Raw laps and results are converted into driver-level KPIs:

- Pace indicators (qualifying and race),
- Consistency (variance or standard deviation across stable lap windows),
- Racecraft (positions gained or lost),
- Reliability (DNF patterns and finish rate).

7 Data Model and Analytical Schema

7.1 Logical Dataset Design

For descriptive analytics, we define two primary tables.

Table 1: Descriptive Tables (Logical Schema)

Table	Key Fields
`videos`	`video_id`, `title`, `published_at`, `duration_sec`, `view_count`, `like_count`, `comment_count`, `channel_id`
`comments`	`comment_id`, `video_id`, `author`, `published_at`, `text`, `like_count`, `parent_id (optional)`

7.2 Derived Metrics

We compute derived fields used throughout the report:

- Engagement Rate proxy: $(\text{likes} + \text{comments})/\text{views}$.
- Comments per view: $\text{comments}/\text{views}$.
- Likes per view: $\text{likes}/\text{views}$.
- Upload hour and weekday: derived from publish timestamp.

8 Descriptive Analytics Outcomes

8.1 Driver Engagement and Share of Voice (SoV)

Question: Which drivers drive the most traffic and discussion?

We estimate SoV by aggregating engagement over videos containing driver keywords in title, description, or tags. A general form of SoV:

$$\text{SoV(driver)} = \frac{\sum_{v \in V_d} \text{Engagement}(v)}{\sum_{v \in V} \text{Engagement}(v)} \tag{1}$$

Here, Engagement(v) may be a weighted score that combines views, likes, and comments with adjustable coefficients.

8.1.1 Narrative Impact

A consistent media insight is that **incidents and rivalries** create stronger reaction intensity than neutral updates. In the context of F1 content, controversies (penalties, crashes, steward decisions, team orders) tend to increase:

- Comment volume (discussion density),
- Polarized sentiment (positive and negative extremes),
- Return engagement (continuation of debate).

8.2 Audience Sentiment and Reaction (Comment NLP)

Question: How do fans feel about the content?
We apply polarity-based sentiment scoring using a three-class mapping:

Table 2: Sentiment Analysis Classification Matrix

Sentiment Class	Polarity Range	Representative Keywords
Positive	> 0.1	Amazing, Goat, Masterclass, Legend, Dominant
Neutral	$-0.1 \leq x \leq 0.1$	Race, Update, Gap, Tyres, Strategy
Negative	< -0.1	Robbed, Boring, Penalty, Biased, Rigged

8.2.1 Why Sentiment Matters

From a business analytics perspective:

- Sentiment indicates brand risk (toxicity and dissatisfaction),
- It helps identify content opportunities (topics driving positive momentum),
- It helps understand engagement drivers (outrage versus celebration).

8.2.2 Limitations of Basic Sentiment

We explicitly acknowledge:

- Sarcasm and memes are frequent in F1 communities.
- Polarity models can misread sarcastic phrases as positive sentiment.
- Mixed-language comments reduce accuracy without language detection.

8.3 Video Performance Metrics

Question: Which types of videos perform best?
We examine relationships between:

- Duration and views (reach),
- Duration and engagement rate (depth),
- Publish time and performance (timing advantage).

8.3.1 Reach versus Retention Interpretation

A strong practical insight is the tradeoff:

- **Short-form highlights** maximize reach (high views).
- **Long-form analysis** increases discussion density (higher comments per view).

8.4 Topic and Content Analysis

Question: What are the dominant themes of the season?
We recommend combining:

- Keyword frequency from titles and descriptions,
- Topic modeling (LDA) on cleaned comment corpus,
- Event-window segmentation (race weeks versus off-weeks).

8.4.1 Interpreting Topics as Narratives

Topics can be interpreted as narratives:

- Team dominance arcs,
- Strategy controversies,
- Driver market rumors,
- Regulations and stewarding debates.

9 Predictive Analytics Outcomes

9.1 Motivation: Beyond Points

Pure championship points can hide:

- Underlying pace (car performance),
- Consistency (variance in lap times),
- Reliability (DNFs),
- Racecraft (positions gained or lost).

Hence we designed a composite score that blends multiple KPIs.

9.2 Composite Scoring Methodology

Table 3 defines engineered KPIs.

Table 3: Definition of Predictive Performance Metrics (KPIs)

Metric	Data Source	Computational Logic (Conceptual)
Qualifying Pace	OpenF1 (Quali sessions)	Normalized best-lap and sector performance with a grid-position proxy.
Race Pace	OpenF1 (Race laps)	Median lap time after filtering pit-in/out and safety-car affected laps.
Consistency	OpenF1 (Race laps)	Standard deviation of rolling lap windows; penalize high variance.
Racecraft	OpenF1 (Results and position)	Grid-to-finish delta adjusted for typical overtaking difficulty.
Reliability	OpenF1 (Status)	Penalize technical DNFs; reward high finish rate.

9.3 Composite Weights

Weights can be tuned based on what the stakeholder values most (pace, stability, or reliability).

Table 4: Composite Model Weights

Metric	**Weight**
Race Pace Score	30%
Qualifying Performance	25%
Consistency Score	20%
Racecraft (Positions Gained)	15%
Reliability (Finish Rate)	10%

9.4 Forecasting Approach

A practical forecasting approach includes:

- Track trailing performance over the last N races,
- Compute a performance slope (improvement or decline),
- Project season totals using regression or weighted moving averages,
- Stress test uncertainty using reliability scenarios (for example, DNF shocks).

9.4.1 Interpretation of Predictions

The predictive module output should be interpreted as decision support:

- It ranks driver strength using multi-dimensional performance.
- It highlights which KPI (pace versus reliability) most influences outcomes.
- It supports narrative forecasting (who is rising, stable, or fragile).

10 Evaluation and Validation Strategy

10.1 Descriptive Validation

We validate descriptive outputs by:

- Spot-checking extracted counts against the YouTube interface for a small sample of videos,
- Ensuring counts are non-negative and consistent after transformation,
- Sampling cleaned text to confirm removal of URLs and noisy tokens.

10.2 Sentiment Validation

Sentiment is validated through:

- Manual labeling of a small comment sample (gold set),
- Comparing model predictions versus manual labels,
- Reporting misclassification patterns (sarcasm, slang, mixed language).

10.3 Predictive Validation

We recommend:

- Backtesting: train on earlier races and predict later races,
- Reporting error metrics (MAE and RMSE) for predicted points or ranks,
- Sensitivity analysis on the weight vector in Table 4.

11 Challenges and Limitations

1. **API Rate Limiting and Quotas:** YouTube API quotas are restrictive; caching and careful pagination are required.
2. **Comment Availability:** Some videos have comments disabled or heavily moderated, creating sample bias.
3. **Sentiment Nuance:** Sarcasm and slang reduce polarity-model accuracy.
4. **Language Mixing:** F1 audience is global; multilingual comments require language-aware processing.
5. **Missing Watch-time Metrics:** Public API does not provide retention curves; we rely on proxies such as comments per view.

12 Reproducibility and How to Run

12.1 Environment Setup

Recommended steps (adapt paths to your system):

1. Clone repository:

```
git clone https://github.com/ahmedmusharaf31/dba-youtube-project.git
cd dba-youtube-project
```

2. Create environment and install dependencies:

```
python -m venv .venv
# Windows:
# .venv\Scripts\activate
# Linux/Mac:
# source .venv/bin/activate
pip install -r requirements.txt
```

12.2 Running Descriptive Analytics

Execute the descriptive module via its entry point:

```
cd Descriptive
python run_analytics.py
```

12.3 Running Predictive Analytics

Execute the predictive module via:

```
cd Predictive
python main_script.py
```

12.4 Expected Outputs

- Cleaned CSV files in `Descriptive/data/processed`
- Raw API responses in `Descriptive/data/raw`
- Cached OpenF1 data in `Predictive/f1_data_cache`
- Console summaries and exported results depending on script configuration

13 Conclusion and Future Scope

13.1 Conclusion

This project demonstrates that YouTube is a powerful lens into the F1 attention economy. Through structured ETL and analytics design, Team BW produced both:

- A **diagnostic view** of engagement and sentiment drivers, and
- A **forecasting view** of future outcomes based on engineered performance KPIs.

13.2 Future Scope

High-impact improvements include:

- **Cross-platform fusion:** correlate YouTube spikes with X (Twitter) or Reddit during race windows.
- **Better sentiment models:** transformer-based sentiment or domain-tuned classifiers for sports sarcasm.
- **Event detection:** automatically detect incident narratives from comment bursts and keywords.
- **Thumbnail analysis:** computer vision to correlate imagery (crashes, flags, driver faces) with engagement.
- **Time-series models:** recurrent or probabilistic models for points-trajectory forecasting.

A Appendix A: Recommended File-to-Report Mapping

This section is designed so you can paste exact filenames and class names once you confirm them from the repo.

Component	Where to Reference in Repo (fill exact path)
YouTube extraction logic	`Descriptive/src/...`
Comment cleaning utilities	`Descriptive/src/...`
Analytics runner	`Descriptive/run_analytics.py`
Notebook EDA	`Descriptive/notebooks/...`
OpenF1 pipeline	`Predictive/...`
Forecast engine	`Predictive/main_script.py`
Cache folder	`Predictive/f1_data_cache`

B Appendix B: Code Snippets (Templates)

Paste key snippets here (ETL steps, sentiment function, KPI calculations).

B.1 Example: Sentiment Scoring Function (Template)

```python
def classify_sentiment(polarity: float) -> str:
    if polarity > 0.1:
        return "positive"
    elif polarity < -0.1:
        return "negative"
    return "neutral"
```

Listing 2: Sentiment scoring template (replace with your actual function)

B.2 Example: Engagement Rate Metric (Template)

```python
def engagement_rate(views: int, likes: int, comments: int) -> float:
    if views <= 0:
        return 0.0
    return (likes + comments) / views
```

Listing 3: Engagement rate template

C Appendix C: Figures to Add (Checklist)

To make the report visually complete for grading, add screenshots or plots and reference them:

- Top drivers SoV bar chart
- Sentiment distribution bar chart
- Comments per view versus duration scatter plot
- Topic modeling top words per topic
- Composite score ranking table
- Forecast plot: cumulative points projection

D Appendix D: Repository Link

Repository: https://github.com/ahmedmusharaf31/dba-youtube-project