**📊 YouTube Trending Video Analytics**

**Objectives**

1. Analyze YouTube trending videos dataset to identify popularity trends.
2. Understand which categories (Music, Gaming, News, etc.) gain the most traction.
3. Explore regional differences in trending content.
4. Detect factors that contribute to high engagement (views, likes, comments).
5. Spot anomalies such as videos with unusually high or low interaction.

**Dataset**

* **Source:** YouTube Trending Video Dataset (Kaggle)
* **Description:** Contains daily trending YouTube videos data across multiple regions.
* **Fields:** video\_id, title, publish\_time, views, likes, dislikes, comment\_count, category\_id, tags, region.

**Tools & Technologies**

* **Python** (pandas, matplotlib, seaborn, plotly for visualization)
* **SQL** (optional, for structured querying)
* **Jupyter Notebook / Google Colab** for development
* **(Optional)** Apache Spark (local mode) if you want to show “big data style” processing

**Step-by-Step Flow**

**1. Data Collection**

* Download dataset from Kaggle (CSV files per region).

**2. Data Cleaning**

* Handle missing values.
* Convert dates (publish\_time) into usable formats.
* Map category\_id to actual category names (provided in dataset).

**3. Data Exploration**

* Calculate basic stats: avg views, likes, dislikes per category.
* Identify top trending categories.
* Explore correlations (e.g., views vs likes).

**4. Trend Analysis**

* Plot category popularity trends over time.
* Identify hottest months/days for certain categories.
* Detect anomalies (e.g., very high like-to-view ratio).

**5. Regional Comparison**

* Compare trends between regions (e.g., US vs GB vs IN).
* See if some categories dominate in certain countries.

**6. Visualization & Reporting**

* Graphs: bar charts, line charts, heatmaps.
* Dashboards (optional, with Plotly or PowerBI/Tableau).
* Final report with key insights.

**Deliverables**

* Cleaned dataset ready for analysis.
* Trend graphs of top categories over time.
* Regional comparison charts.
* Anomaly detection (videos with unusual engagement).
* Final report with insights + recommendations.

Below will help you:

* Load the dataset
* Clean it
* Do some **first analysis & graphs**

# 🐍 Python Code Template: YouTube Trending Analytics

# -------------------------------

# Step 1: Import libraries

# -------------------------------

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Set style for graphs

sns.set(style="whitegrid")

# -------------------------------

# Step 2: Load dataset

# -------------------------------

# Example: US trending dataset (from Kaggle zip)

df = pd.read\_csv("USvideos.csv")

print("Dataset shape:", df.shape)

print("Columns:", df.columns)

# -------------------------------

# Step 3: Basic cleaning

# -------------------------------

# Convert publish\_time to datetime

df['publish\_time'] = pd.to\_datetime(df['publish\_time'], errors='coerce')

# Drop duplicates if any

df = df.drop\_duplicates()

# Check missing values

print("\nMissing values:\n", df.isnull().sum())

# -------------------------------

# Step 4: Map category\_id → category name

# -------------------------------

# Kaggle provides a JSON file "US\_category\_id.json"

import json

with open("US\_category\_id.json", "r") as f:

categories = json.load(f)

category\_dict = {}

for cat in categories['items']:

category\_dict[int(cat['id'])] = cat['snippet']['title']

df['category'] = df['category\_id'].map(category\_dict)

# -------------------------------

# Step 5: Quick stats

# -------------------------------

print("\nTop categories by average views:")

print(df.groupby('category')['views'].mean().sort\_values(ascending=False).head(10))

# -------------------------------

# Step 6: Visualization examples

# -------------------------------

# 1. Top categories by views

top\_categories = df.groupby('category')['views'].mean().sort\_values(ascending=False).head(10)

plt.figure(figsize=(10,6))

sns.barplot(x=top\_categories.values, y=top\_categories.index, palette="viridis")

plt.title("Top 10 YouTube Categories by Average Views")

plt.xlabel("Average Views")

plt.ylabel("Category")

plt.show()

# 2. Views vs Likes scatterplot

plt.figure(figsize=(8,6))

sns.scatterplot(data=df, x='views', y='likes', alpha=0.5)

plt.xscale('log')

plt.yscale('log')

plt.title("Views vs Likes (log scale)")

plt.show()

# 3. Publishing trend over months

df['month'] = df['publish\_time'].dt.month

monthly\_views = df.groupby('month')['views'].mean()

plt.figure(figsize=(10,6))

sns.lineplot(x=monthly\_views.index, y=monthly\_views.values, marker="o")

plt.title("Average Views by Month")

plt.xlabel("Month")

plt.ylabel("Avg Views")

plt.show()

## 🔹 What this code does

1. Loads the **YouTube dataset (USvideos.csv)**.
2. Cleans & parses publish\_time.
3. Maps **category\_id → category name** using the JSON file provided in Kaggle dataset.
4. Shows quick stats (avg views per category).
5. Creates **3 visualizations**:
   * Top 10 categories by views
   * Scatterplot of views vs likes
   * Trends of views by month

👉 This is enough to kick-start your project.  
You can later expand with:

* Regional comparisons (load GBvideos.csv, INvideos.csv, etc.)
* Engagement ratios (likes/views, comments/views)
* Anomaly detection