**40-Hour YouTube Analytics Project Workflow (CONDENSED)**

**2-Person Team: Max Giesinger & Wincent Weiss Analysis**

**Team Roles**

**Person A (Data Engineer):** API, database, dbt models, automation  
**Person B (Data Analyst):** Features, analysis, visualizations, insights

**WEEK 1: Setup & Data Collection (10-12 hours)**

**Day 1-2: Environment Setup (4 hours)**

**Person A (2h):**

* Create project structure (folders: data, scripts, models, notebooks, outputs)
* Initialize Git repo + .gitignore
* Create DuckDB database with empty schemas (raw, staging, intermediate, marts)
* Set up pre-commit hooks

**Person B (2h):**

* Get YouTube API key from Google Cloud Console
* Find channel IDs for both artists
* Browse videos manually to understand content patterns
* Create data dictionary template
* Install Python packages and test imports

**Day 3-5: Data Collection (6-8 hours)**

**Person A (4h):**

* Build API crawler script with quota tracking
* Fetch 50 videos per artist + 30 comments per video
* Save raw data to JSON
* Build loader script to insert JSON into DuckDB
* Validate data loaded correctly

**Person B (3-4h):**

* Design database schema (videos, comments, channels tables)
* Write database initialization script
* Create data validation script (check nulls, row counts, date ranges)
* Run validation and document issues

**Day 6-7: Automation (2 hours)**

**Person A (1.5h):**

* Modify fetch script for incremental updates
* Create GitHub Actions workflow for daily data collection
* Add API key to GitHub Secrets
* Test manual trigger

**Person B (0.5h):**

* Document monitoring process
* Create troubleshooting guide

**WEEK 2: Cleaning & Feature Engineering (12-14 hours)**

**Day 8-10: dbt Setup & Staging (6-7 hours)**

**Person A (3-4h):**

* Initialize dbt project with profiles.yml and dbt\_project.yml
* Create sources.yml with data quality tests
* Build stg\_videos.sql (clean, cast types, filter nulls)
* Build stg\_comments.sql (clean text, remove empty)
* Run dbt run and dbt test for staging

**Person B (3h):**

* Design feature engineering plan (time, text, content, engagement features)
* Create feature definitions document
* Write custom dbt tests (e.g., engagement\_rate between 0-100)

**Day 11-14: Feature Engineering (6-7 hours)**

**Person A (3h):**

* Build int\_video\_features.sql (Part 1):
  + Time features (day, hour, weekend, time\_of\_day)
  + Text features (title\_length, word\_count, tag\_count)
  + Content classification (music video, live, acoustic, etc.)
  + Collaboration detection

**Person B (3-4h):**

* Continue int\_video\_features.sql (Part 2):
  + Engagement metrics (like\_rate, comment\_rate, engagement\_rate)
  + Performance tiers (Viral, High, Medium, Low)
  + Percentile rankings
* Build int\_comment\_features.sql:
  + Comment length, emoji count, punctuation
  + Simple keyword-based sentiment
* Run and test all intermediate models

**WEEK 3: Analysis & Visualization (12-14 hours)**

**Day 15-17: Analytical Models (6-7 hours)**

**Person A (3h):**

* Build fct\_video\_performance.sql (rankings, artist comparisons)
* Build fct\_artist\_comparison.sql (aggregated metrics)
* Build fct\_comment\_analysis.sql (sentiment by video)
* Run dbt run --models marts and test

**Person B (3-4h):**

* Write 05\_answer\_dvs.py script
* Query DV1: Video characteristics vs engagement
* Query DV2: Upload timing patterns
* Query DV3: Artist benchmarking
* Query DV4: Sentiment analysis
* Save all results to CSV files (15-20 CSVs)

**Day 18-21: Visualization (6-7 hours)**

**Person A (3h):**

* Create visualizations.ipynb
* Chart 1: Engagement by content type (bar chart)
* Chart 2: Upload timing heatmap (2 artists)
* Chart 3: Artist comparison dashboard (interactive HTML)
* Chart 4: Performance tier distribution (bar chart)

**Person B (3-4h):**

* Chart 5: Sentiment breakdown (stacked bar)
* Chart 6: Weekend vs weekday (grouped bar)
* Chart 7: Views vs engagement scatter (interactive HTML)
* Write insights.md with:
  + Executive summary
  + Findings for each DV (1-4)
  + Recommendations per artist
  + Limitations and next steps

**WEEK 4: Documentation & Polish (8-10 hours)**

**Day 22-24: Documentation (5-6 hours)**

**Person A (3h):**

* Write comprehensive README.md:
  + Project overview, key findings, tech stack
  + Project structure diagram
  + Data pipeline architecture
  + Setup instructions (step-by-step)
  + Usage guide, sample queries
  + Future enhancements
* Add docstrings to all Python scripts
* Format code with black, fix flake8 issues
* Update requirements.txt

**Person B (2-3h):**

* Create 5-slide presentation:
  + Overview, key findings, recommendations, architecture, next steps
* Create final checklist document:
  + All deliverables checked off
  + Code quality verification
  + Portfolio readiness items
  + Skills demonstrated list

**Day 25-28: Optional Enhancement (3-4 hours)**

**Pick ONE:**

**Option 1: Streamlit Dashboard**

* Person A: Setup, layout, filters, KPIs
* Person B: Build 4 tabs with interactive charts

**Option 2: Advanced NLP Sentiment**

* Person A: Setup transformers, detect language
* Person B: Run ML model, compare results, visualize

**Option 3: Predictive Model**

* Person A: Train models (RF, XGBoost), evaluate
* Person B: Feature importance, predictions, documentation

**Option 4: Thumbnail Analysis**

* Person A: Extract colors, brightness from images
* Person B: Analyze correlation with engagement

**Option 5: Automated Reports**

* Person A: Build HTML template, generation script
* Person B: Design layout, add executive summary

**Option 6: Expand to 4-5 Artists**

* Person A: Add artists, re-run pipeline
* Person B: Update visualizations, competitive analysis

**Final Delivery Checklist (30 min - Both)**

**Code Quality:**

* Run black and flake8
* Remove debug prints
* Check for hardcoded values

**Testing:**

* Run full pipeline from scratch
* Verify all outputs generated
* Test dbt models pass

**Documentation:**

* Proofread README
* Verify all links work
* Add screenshots

**Repository:**

* Clean git history
* Add topics/tags
* Create LICENSE file
* Write meaningful commit messages

**Portfolio:**

* Take screenshots of best visuals
* Prepare 30-second pitch
* Add to portfolio site

**Time Summary**

| **Week** | **Person A** | **Person B** | **Total** |
| --- | --- | --- | --- |
| Week 1 | 6h | 4h | 10h |
| Week 2 | 6h | 7h | 13h |
| Week 3 | 6h | 7h | 13h |
| Week 4 | 5h | 5h | 10h |
| **Total** | **23h** | **23h** | **46h** |

*With optional enhancement: 50-54h total*

**Daily Standup (10 min)**

* What did you complete?
* What's next?
* Any blockers?

**Success Criteria**

✅ All 4 research questions answered with data  
✅ 15+ CSV outputs and 7 visualizations  
✅ Professional GitHub repo with clear README  
✅ Automated data pipeline working  
✅ 5-minute presentation ready  
✅ Portfolio-ready project showcasing skills