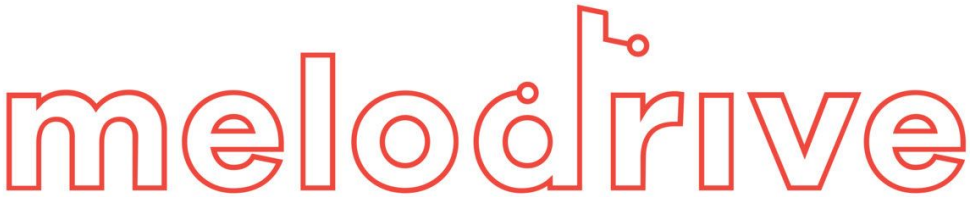
2. Gen Music Project Pipeline

Generative Algorithms for Sound and Music

2016


melodrive


boostVC
melodrive
Infinite AI Music
for the Digital Universe

“This is never gonna happen”

Executive at one of the 3 major labels.

2025

IMAGE CREDITS: BARRY CHIN/THE BOSTON GLOBE / GETTY IMAGES

STARTUPS

Legally embattled AI music startup Suno raises at $2.45B valuation on $200M revenue

Julie Bort — 11:21 AM PST · November 19, 2025

UNIVERSAL MUSIC GROUP AND UDIO ANNOUNCE UDIO'S FIRST STRATEGIC AGREEMENTS FOR NEW LICENSED AI MUSIC CREATION PLATFORM

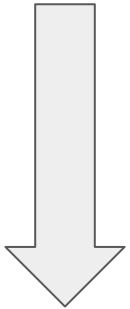
PRESS RELEASE

WARNER MUSIC GROUP AND SUNO FORGE GROUNDBREAKING PARTNERSHIP

November 25, 2025

Gen AI Mus is centre mind in the music industry

Gen AI Mus is centre mind in the music industry



AI Music Revolution

What does this mean for you?

What does this mean for you?

But…

1. You need to know what you’re doing
2. Most gen mus projects fail

Why do gen music projects fail?

Typical failure points

- Too much focus on model

Typical failure points

- Too much focus on model
- R&D / user needs mismatch

Typical failure points

- Too much focus on model
- R&D / user needs mismatch
- Data

Typical failure points

- Too much focus on model
- R&D / user needs mismatch
- Data
- No control

Typical failure points

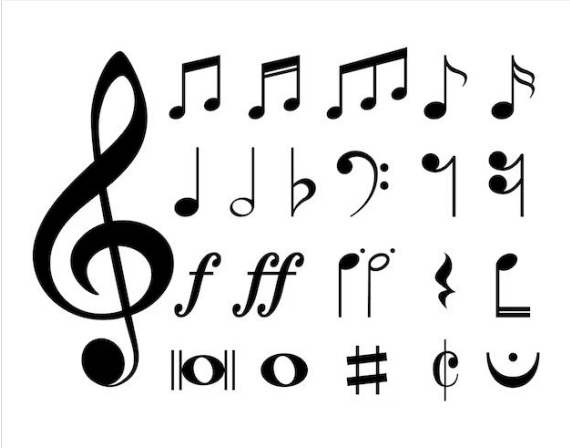
- Too much focus on model
- R&D / user needs mismatch
- Data
- No control
- No domain knowledge

Typical failure points

- Too much focus on model
- R&D / user needs mismatch
- Data
- No control
- No domain knowledge
- Weak evaluation

Lack of end-to-end planning

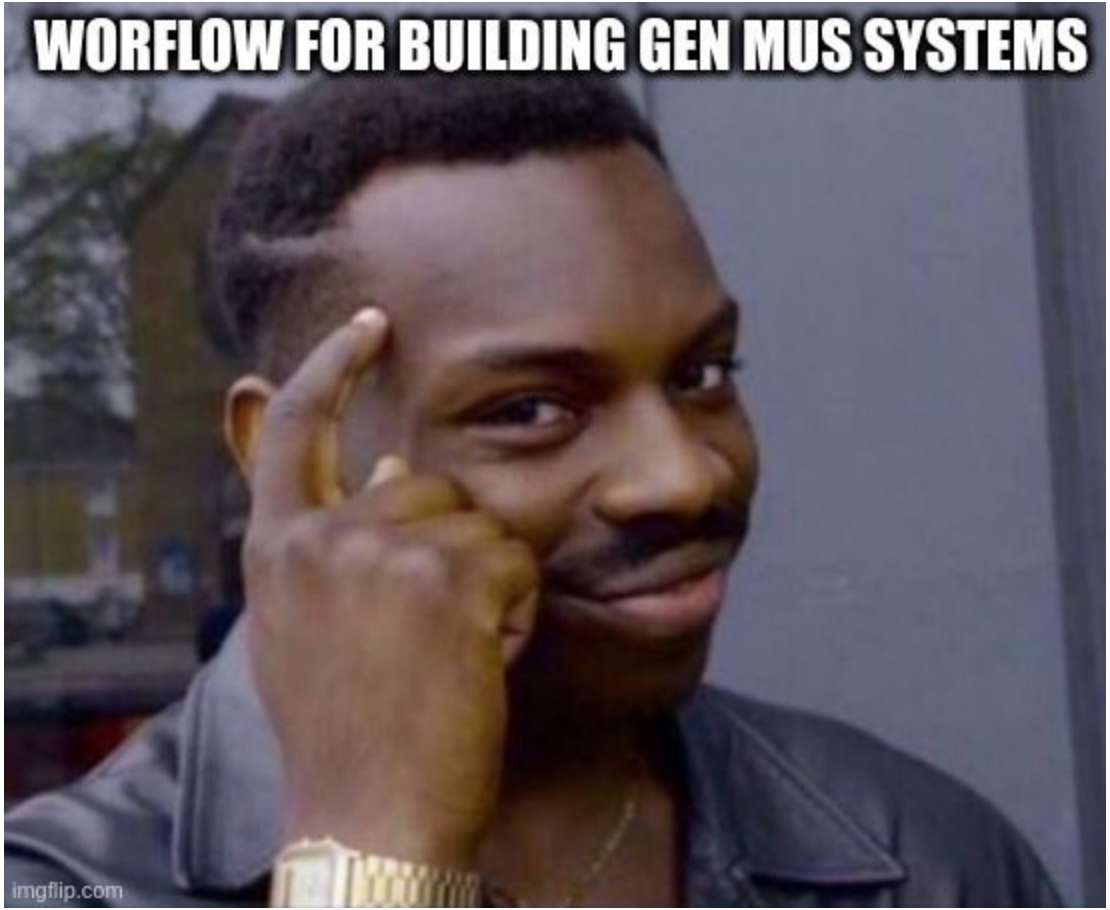
Symbolic / audio divide



Industry trajectory

1. Symbolic
2. Audio
3. Symbolic come-back (limited)

Lack of end-to-end planning


WORFLOW FOR BUILDING GEN MUS SYSTEMS
imgflip.com

End-to-end pipeline

1. Problem definition
2. Data strategy
3. Representation & pre-processing
4. Model choice
5. Training / implementation
6. Evaluation
7. Deployment

End-to-end pipeline

1. Problem definition
2. Data strategy
3. Representation & pre-processing
4. Model choice
5. Training / implementation
6. Evaluation
7. Deployment

Business

Data

Model

Operations

End-to-end pipeline

1. Problem definition

Researcher

2. Data strategy
3. Representation & pre-processing
4. Model choice
5. Training / implementation
6. Evaluation

7. Deployment

Business

Data

Model

Operations

End-to-end pipeline

1. Problem definition

Research engineer

2. Data strategy
3. Representation & pre-processing
4. Model choice
5. Training / implementation
6. Evaluation
7. Deployment

Business

Data

Model

Operations

Problem definition

- Who is the user?
- What musical decision is being automated or augmented? How?
- Degree of control of the user?
- Offline generation vs interactive?
- Web / desktop app / hardware?
- Audio / symbolic?
- …

The greatest sin

Going straight to code without deep understanding of the problem to attack

End-to-end pipeline

1. Problem definition
2. Data strategy
3. Representation & pre-processing
4. Model choice
5. Training / implementation
6. Evaluation
7. Deployment

Data strategy

- Do we need data?
- What data?
- How much?
- How do we get it?
- Legal, licensing, ethical constraints

End-to-end pipeline

1. Problem definition
2. Data strategy
3. Representation & pre-processing
4. Model choice
5. Training / implementation
6. Evaluation
7. Deployment

Representation and pre-processing

- What type of data?
- Pitch / rhythm / arrangement / harmony encoding
- Tokenization choices
- Temporal resolution
- Explicit vs implicit musical structure
- Cleaning / normalising data
- …

Symbolic music data

- MIDI
- Piano roll
- ABC
- Lead sheet
- MusicXML

MIDI

- Low-level event-based format
- Note on/off, velocity, timing, and control changes
- Focus on performance, not musical intention

MIDI

- Widespread, easy to obtain
- DAW-compatible and industry-standard
- Works well for rhythm and timing-based generation
- Easy to audition outputs

MIDI

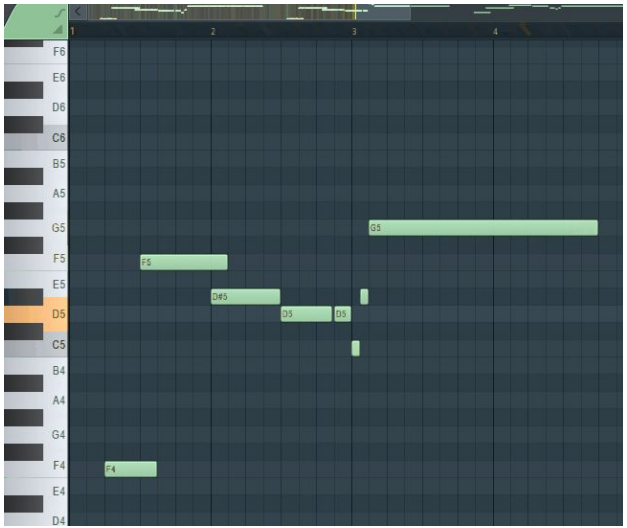
- Widespread, easy to obtain
- DAW-compatible and industry-standard
- Works well for rhythm and timing-based generation
- Easy to audition outputs
- No explicit musical structure (key, meter, harmony)
- Timing can be ambiguous (tempo changes)
- Velocity ≠ expression

MIDI

- Cleaning datasets is paramount and a nightmare
- Tokenize directly (e.g. MidiTok)
- Or convert to:
  - Piano-roll
  - Note-based rep -> *(time, pitch, duration, velocity)*

Piano roll

- Grid representation: pitch × time
- Binary (note on / off) or continuous (velocity)



Piano roll

- Simple and intuitive
- Easy to visualize
- Works well with CNNs and diffusion models
- Good for short-form, texture-based music

Piano roll

- Simple and intuitive
- Easy to visualize
- Works well with CNNs and diffusion models
- Good for short-form, texture-based music
- Poor representation of long-term structure
- Hard to encode harmony, voice leading, or form
- Time quantization is a strong modeling assumption

ABC

- Text-based
- Monophonic or simple polyphonic music
- Used for folk and dance music

```
X:1
T:Jingle Bells
C:James Lord Pierpont, 1857
M:4/4
L:1/4
K:C
EEE2 | EEE2 | EGCD | E4 |
FFFF | FEEE | EDDE | D2G2 |
EEE2 | EEE2 | EGCD | E4 |
FFFF | FEEE | GGFD | C4 |]
```

ABC

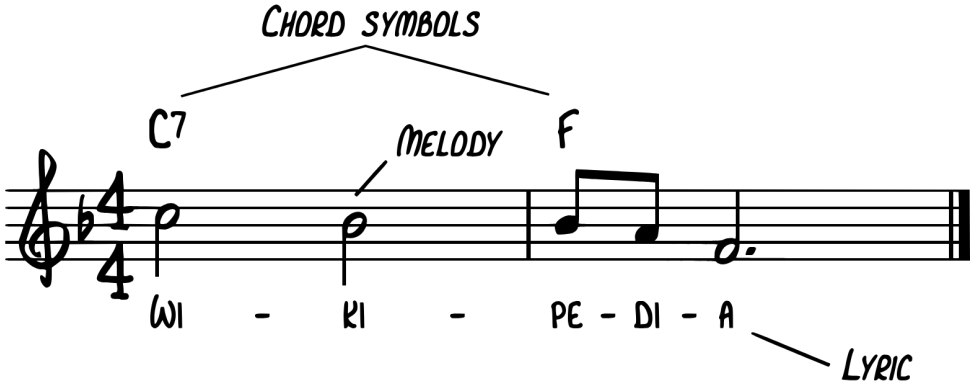
- Extremely compact
- Human-readable and editable
- Perfect for text-based models (Markov, LLMs, Transformers)
- Strongly constrained → fewer degenerate outputs

ABC

- Extremely compact
- Human-readable and editable
- Perfect for text-based models (Markov, LLMs, Transformers)
- Strongly constrained → fewer degenerate outputs
- Limited expressive power
- Poor support for complex polyphony and orchestration
- Niche repertoire
- Timing and articulation are coarse

Lead sheet

- Melody line + chord symbols
- Focuses on harmonic function



Lead sheet

- Musically meaningful
- Ideal for jazz, pop, songwriting systems
- Enables modular generation (melody ↔ harmony ↔ arrangement)

Lead sheet

- Musically meaningful
- Ideal for jazz, pop, songwriting systems
- Enables modular generation (melody ↔ harmony ↔ arrangement)
- Sparse representation
- Requires reconstruction into full arrangements
- Rhythm and voicing under-specified
- Few datasets (mainly jazz)

MusicXML

- Score-level representation of Western notation
- Encodes structure: measures, voices, articulations, dynamics
- Uses XML-based encoding

```
<note>
  <pitch>
    <step>E</step>
    <alter>-1</alter>
    <octave>4</octave>
  </pitch>
  <duration>2</duration>
  <type>half</type>
</note>
```

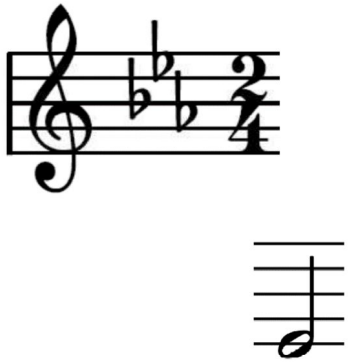


Figure 1.15 from [Müller, FMP, Springer 2015]

MusicXML

- Explicit musical intent
- Ideal for theory-aware systems
- Clear hierarchy (notes → measures → phrases)
- Good for classical and educational contexts

MusicXML

- Explicit musical intent
- Ideal for theory-aware systems
- Clear hierarchy (notes → measures → phrases)
- Good for classical and educational contexts
- Verbose and heavy
- Scarce, high-quality datasets
- Often biased toward Western classical music
- Overkill for many generative tasks


THERE ISN'T A "PERFECT" REPRESENTATION
imgflip.com

Music representation tips

- Context dependent

Music representation tips

- Context dependent
- My workflow
  - Check existing representations
  - If none works for my use case, create a custom one

Music representation tips

- Context dependent
- My workflow
  - Check existing representations
  - If none works for my use case, create a custom one
- Encode the minimal amount of musical aspects I need


THE RICHER THE
REPRESENTATION, THE RICHER THE CONTROL OPTIONS FOR THE USER
imgflip.com

Activity: Music representation

- Work in groups of 3 or 4 (7 minutes)
- Pick a use case
  - Adaptive game music generator
  - Sax jazz solo generator
  - Composer assistant for folk melody continuation
- Task
  - Choose representation (MIDI / Piano roll / ABC / Lead sheet / MusicXML), or create your own
  - Why it fits
  - One tradeoff you accept

Python tools for music representation

- music21
- muspy
- miditoolkit
- pretty_midi
- abcmidi

End-to-end pipeline

1. Problem definition
2. Data strategy
3. Representation & pre-processing
4. Model choice
5. Training / implementation
6. Evaluation
7. Deployment

Model choice

- Rule-based or deep learning?
- Why *this* model for *this* problem?

Rule-based vs deep-learning

Factor	Rule-Based	Deep Learning
Data needed	Low	High
Control	High	Medium–Low
Interpretability	High	Low
Expressiveness	Medium	High
Compute	Low	High
Real-time	Strong	Fragile
Debugging	Easier	Harder

When I use rule-based

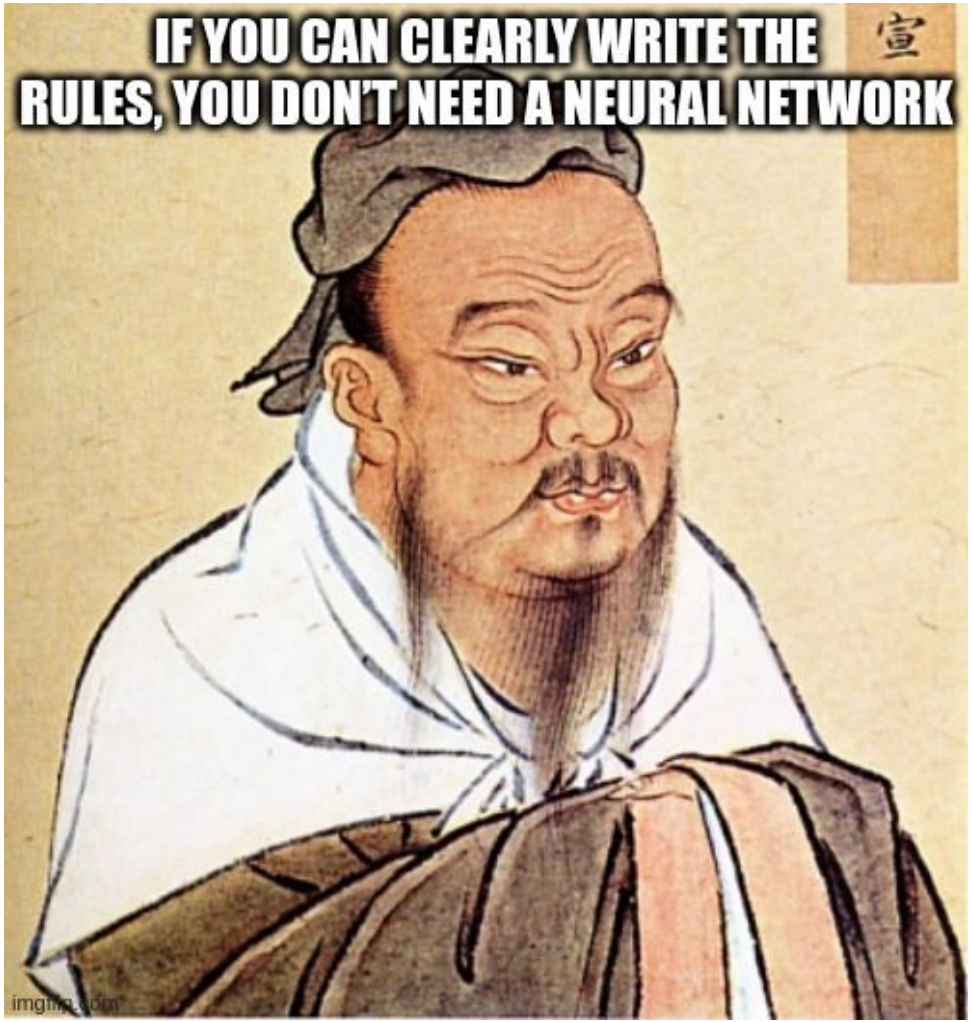
- Clear, constrained goals
- Well-defined musical rules
- Optimization problems (e.g. “best harmonization”)
- Deterministic or semi-deterministic outputs

When I use rule-based

- Clear, constrained goals
- Well-defined musical rules
- Optimization problems (e.g. “best harmonization”)
- Deterministic or semi-deterministic outputs
- Typical examples:
  - Harmonization
  - Counterpoint exercises
  - Style-constrained generation
  - Constraint satisfaction

When I use Deep Learning

- Open-ended goal
- Complex styles
- Production-based styles
- A lot of music data available
- Typical examples:
  - Full-score generative system
  - Adaptive music system


IF YOU CAN CLEARLY WRITE THE
RULES, YOU DON'T NEED A NEURAL NETWORK
宣

Often the real answer: Hybrid

- In practice, most successful systems I worked on are hybrid:
  - Rule-based preprocessing
  - DL generation
  - Rule-based post-processing
- Neural nets are great idea generators. Rules are great editors

End-to-end pipeline

1. Problem definition
2. Data strategy
3. Representation & pre-processing
4. Model choice
5. Training / implementation
6. Evaluation
7. Deployment

Training / implementation: Rule-based

1. Examine reference music
2. Extract stylistic rules
3. Encode rules mathematically / logically
4. Develop model
5. Reiterate

Training / implementation: Rule-based

1. Examine reference music
2. Extract stylistic rules
3. Encode rules mathematically / logically
4. Develop model
5. Reiterate

Engineer + music expert
MUST work together

Training / implementation: DL

1. Examine reference music
2. Train model injecting as much domain knowledge as possible
3. Reiterate

End-to-end pipeline

1. Problem definition
2. Data strategy
3. Representation & pre-processing
4. Model choice
5. Training / implementation
6. Evaluation
7. Deployment

Evaluation

- Check that system
  - Works as intended
  - Hits quality standards
- Who decides what’s good?
- Objective, subjective, expert, market-driven metrics


COMING UP TOMORROW....
EVALUATION
imgflip.com

End-to-end pipeline

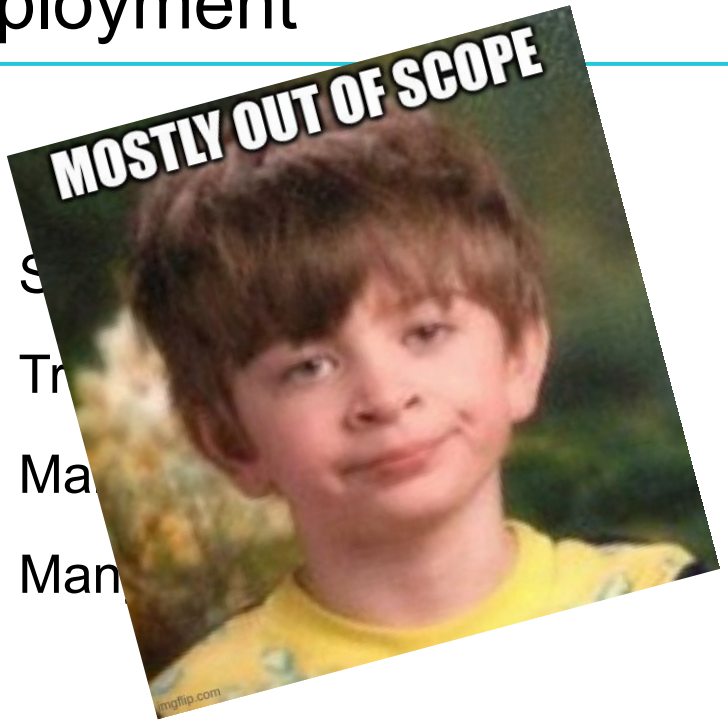
1. Problem definition
2. Data strategy
3. Representation & pre-processing
4. Model choice
5. Training / implementation
6. Evaluation
7. Deployment

Deployment

- Batch vs real-time
- Serving + scale
- Tracking
- Maintenance
- Many projects break here

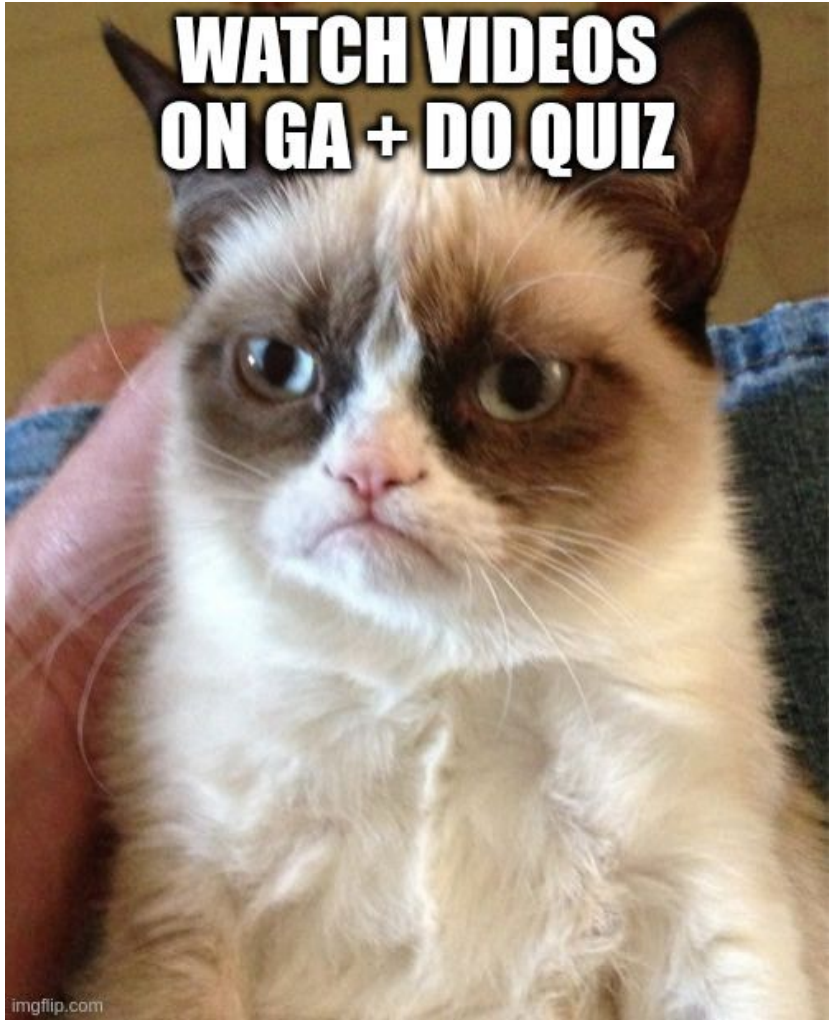
Deployment

-
- S
- Tr
- Ma
- Man



End-to-end pipeline

1. Problem definition
2. Data strategy
3. Representation & pre-processing
4. Model choice
5. Training / implementation
6. Evaluation
7. Deployment


WATCH VIDEOS
ON GA + DO QUIZ
imgflip.com