

Name: Sandeep Yedla
Course: 3FA2025 Machine Learning Fundamentals (AIML-500-01N1D)
Assignment: LLM, Custom GPTs, and STORM AI Practice
Date: 10/31/25

Task 1 – Practice with an LLM

Prompt Used :

You are a beginner-level music producer using FL Studio. Explain step-by-step how to make a beginner-friendly lo-fi chill beat from scratch. Keep the explanation under 250 words, use simple language, and mention at least one stock plugin that comes with FL Studio

Chat 1

Make a simple lo-fi chill beat in FL Studio (beginner guide)

1. **New project & tempo:** Open FL Studio → set BPM to **75–85**. Turn **swing** to ~10% (top toolbar) for groove.
2. **Drums (easy mode):** Add **FPC** (stock). Pick a mellow kit (lo-fi/boom-bap). In the **Channel Rack**, make a 4-bar loop: kicks on 1 & 3-ish, snares on 2 & 4, soft hi-hats every 1–2 steps. Shift some hats slightly right for “lazy” feel.
3. **Humanize:** Lower **velocity** of a few hats and ghost snares. Nudge timing a tiny bit.
4. **Chord loop:** Add **FL Keys** or **FLEX** (both stock). Record or draw **jazzy minor-7** chords (e.g., A-min7 → D-min7 → G-maj7). Keep it simple.
5. **Melody:** New **FLEX** patch (pad/keys) or **3xOSC** for a soft lead. Play a short 2–4 note motif that repeats.
6. **Bass:** Add **3xOSC**. Turn off two oscillators, low-pass with **Filter**, and follow root notes of your chords.
7. **Dust & vibe:** Import a quiet vinyl/noise sample (free pack or any noise). Turn it down so it’s barely there.
8. **Mix (basic):** Route tracks to the Mixer.
 - **Fruity Parametric EQ 2:** Cut low rumble on chords/pads (<100 Hz).
 - **Fruity Reverb 2** on snare/keys (small size, short decay).
 - **Fruity Delay 3** subtle on melody.
 - **Sidechain bass** to kick with **Fruity Limiter** (Comp mode) for a gentle dip.
9. **Master touch:** Add **Fruity Soft Clipper** on the Master, turn **post** until it’s a bit warm, not crushed.
10. **Arrange:** Intro (chords + noise) → bring hats → full drums → drop melody → breaks/mutes.
11. **Automation:** Fade the noise or filter the chords in the intro/outro.
12. **Export:** File → Export → WAV/MP3, 24-bit WAV for best quality.

Very detailed, structured like a tutorial. Covered every stage from setup to mastering, including plugins like *FPC*, *FLEX*, *Fruity Reeverb 2*, and *Fruity Limiter*.

It followed the word limit closely and offered clear step-by-step sequencing (intro → drums → chords → melody → mix → export).

Tone: Practical and precise.

Chat 2

Thought for 6s >

Here's a simple, comfy lo-fi recipe in FL Studio:

1. New project → set BPM to 75–85. Turn Swing (top toolbar) to ~10–20% for groove.
2. Drums: Add **FPC**. Pick a dusty kit or load soft kick, snare, hi-hat samples. Make a 2-bar pattern: kick on 1 & "and of 2," snare on 2 & 4, gentle hats every 1/8. Lower velocities randomly for human feel.
3. Keys: Add **FL Keys** or **FLEX** (e.g., Lo-Fi Piano). Lay a mellow 4-chord loop (try minor 7ths). Quantize lightly, then nudge a few notes off-grid.
4. Bass: Add 3xOSC or **FLEX** bass. Follow root notes of your chords. Keep it simple and low.
5. Melody bits: A soft bell, Rhodes, or guitar plucks—short, sparse phrases.
6. Texture: Drop a vinyl crackle sample on a new track at very low volume.
7. EQ: On keys/bass/drums, use **Fruity Parametric EQ 2** to remove mud (cut ~200–300 Hz on keys; high-pass textures).
8. Space: Add **Fruity Reeverb 2** (small room, short decay) and **Fruity Delay 3** (very low mix) on keys/melody.
9. Lo-fi tone: Gentle saturation with **Fruity Blood Overdrive** (very low preamp) or **Fruity Waveshaper**. Low-pass your master bus a little for warmth.
10. Glue: On the master, **Fruity Soft Clipper** (ceiling ~0 dB, small post gain). Arrange: 8 bars intro (drums+keys), 16 bars main, 8 bars outro. Export WAV/MP3.

That's it—chill, simple, and vibey.

Observation :

Creative and conversational tone ("That's it—chill, simple, and vibey").

Suggested plugins such as FPC, Fruity Blood Overdrive, Fruity Waveshaper, and Fruity Soft Clipper.

Had slightly more personality, and gave mixing/EQ advice with specific frequencies.

Tone: Friendly, slightly more artistic than instructional.

Chat 3

Here's a simple, beginner-friendly way to make a lo-fi chill beat in FL Studio:

1. Set the vibe

Open a new project. Set tempo to **70–85 BPM**. Add a little **swing** (10–20%) on the Channel Rack.

2. Drums

Use the **FPC** stock plugin (lo-fi kit or soft kicks/snares). Make a 2-bar loop: kick on 1 & 3, snare on 2 & 4, soft hats every 2 steps. Randomize hat **velocity** a bit.

3. Chords

Add **FLEX** (stock). Pick a mellow keys/pad preset. Draw simple 7th chords (e.g., Cmaj7–Am7–Dm7–G7). Nudge some notes slightly off-grid for human feel.

4. Bass

Try **BooBass** (stock) or a warm **FLEX** bass. Follow the chord roots. Keep notes short and simple.

5. Melody

Add a gentle lead (**FLEX** keys, bell, or guitar). Use 3–5 notes, leave space. Lo-fi loves minimal ideas.

6. Texture

Add vinyl crackle or rain sample at low volume. High-pass it at ~200 Hz.

7. Lo-fi tone

Put **Fruity Parametric EQ 2** on keys/melody: low-pass around 8–12 kHz. Add **Fruity Reeverb 2** (small size, short decay). Optional: a touch of **Fruity Chorus**.

8. Glue & Pump

On the master, add **Fruity Soft Clipper** (ceiling ~0 dB, small pre-gain). Gentle sidechain: use **Fruity Limiter** (Comp mode) to duck keys/melody slightly to the kick.

9. Arrange

Intro (chords + crackle) → main loop → small break (mute drums or melody) → return. Automate low-pass to open/close between sections.

10. Export

Keep peaks under 0 dB, export to WAV/MP3. You've got a cozy lo-fi loop!



Observation :

Balanced between technical clarity and creative guidance.

Included musical theory elements (chord progressions like *Cmaj7–Am7–Dm7–G7*), humanization techniques, and detailed mastering notes.

Offered multiple plugin options (*BooBass*, *FLEX*, *Fruity Chorus*).

Tone: Organized and encouraging.

Comparison of Outputs

Aspect	Chat 1 (GPT-5)	Chat 2 (Claude)	Chat 3 (Gemini)
Follows word limit	Yes	Yes	Yes
Mentions stock plugins	(FPC, FLEX, Fruity Reverb 2, Limiter)	(FPC, Blood Overdrive, EQ 2)	(FPC, FLEX, BooBass, Reverb 2)
Clarity for beginners	Excellent — concise and structured	Good — readable but looser	Excellent — clear and well-paced
Creativity / Tone	Moderate — focused on workflow	High — friendly, casual, artistic	High — musical detail and warmth
Technical depth	Strong — covered all production stages	Medium — EQ and tone focus	Strong — added chord theory and arrangement
Overall coherence	Very structured	Conversational	Balanced and natural

All three models understood the intent of the prompt and produced usable instructions for making a lo-fi beat.

However, their **styles differed**:

- **Chat 1 (GPT-5):** Most precise and technical; best for actual beginners following step-by-step.
- **Chat 2 (GPT-5):** More creative tone; emphasized vibe and emotion over structure.
- **Chat 3 (GPT-5):** Combined clarity with musical knowledge; most musically insightful.

Across all three, the core workflow—tempo (75–85 BPM), swing, FPC drums, jazzy 7th chords, vinyl crackle, soft clipper—was consistent, proving the prompt was clear and well-bounded.

Observation :

Testing this single prompt across three LLMs showed how prompt clarity and tone directly shape responses. All followed my constraints but emphasized different strengths—structure (Chat 1), creativity (Chat 2), and musical detail (Chat 3). I learned that when prompting for creative-technical domains like music production, specifying both audience level (beginner) and output style (step-by-step vs conversational) helps control the AI's focus. This exercise highlighted that while LLMs share general understanding, their unique writing and reasoning styles make each more suitable for different creative goals.

Criteria Used to Assess Quality

- Accuracy (follows constraints, uses stock plugins)
- Clarity (easy for a beginner to follow)
- Creativity (engaging tone and originality)
- Practicality (steps usable in real FL Studio workflow)
- Consistency (maintained lo-fi mood and structure)

Conclusion

Each LLM chat can act like a virtual instructor but with its own personality: one precise, one expressive, one balanced.

For technical-creative work such as **music making**, combining structured prompts with specific constraints produces the most coherent and usable results.

Task 2 – Practice with Custom GPTs

Exploring Open AI Data Analyst Custom Gpt

Prompt : **What's the difference between mean, median, and mode? Give an example**



Great question! The **mean**, **median**, and **mode** are all ways to describe the "**central tendency**" of a set of numbers — essentially, they give you an idea of what a "typical" value might be. But they do it in different ways.

📌 Definitions

1. Mean (Average):

- Add up all the numbers, then divide by how many numbers there are.
- Sensitive to **extreme values** (outliers).

2. Median:

- The **middle value** when the numbers are arranged in order.
- If there's an even number of values, take the average of the two middle ones.
- **Not affected** by extreme values.

3. Mode:

- The number that appears **most often**.
 - There can be **more than one mode**, or no mode at all.
-

Example

Let's use this set of numbers:

csharp

 Copy code

```
[2, 4, 4, 6, 9, 10, 100]
```

1. Mean:

$$\text{Mean} = \frac{2 + 4 + 4 + 6 + 9 + 10 + 100}{7} = \frac{135}{7} \approx 19.29$$

So the **mean** is 19.29 – notice how the 100 skews it higher.

2. Median:

- Ordered list: [2, 4, 4, 6, 9, 10, 100]
- Middle value = 4th number = 6
- So, **median** is 6

3. Mode:

- The number 4 appears **twice**, more than any other.
- So, **mode** is 4

Prompt: Now change your explanation to make it easier for a 10-year-old to understand.

Sure! Here's a simple way to explain **mean**, **median**, and **mode** to a 10-year-old:

What Are They?

Imagine you have some numbers, like scores or how many candies your friends have. The **mean**, **median**, and **mode** are just different ways to figure out what a "typical" number looks like.

1. Mean (Average)

👉 Add up all the numbers, then divide by how many there are.

Example:

Let's say your friends have this many candies:

```
2, 4, 4, 6, 9, 10, 100
```

- Add them up:

```
2 + 4 + 4 + 6 + 9 + 10 + 100 = 135
```

- Divide by 7 (because there are 7 numbers):

```
135 ÷ 7 ≈ 19.29
```

So the **mean** is about 19.

 But wait — most people don't have 19 candies! T ↴ s because **one person has 100**, and that makes the average too high.

Prompt : Why did you explain it that way instead of showing code first?

Great question — and a smart one! Here's why I explained it with **words and examples first** instead of showing **code right away**:

🧠 1. Understanding Comes Before Tools

Before using code, it's important to **understand the idea**. If you know **what** mean, median, and mode are — and **why they matter** — the code will make a lot more sense when you see it.

Think of it like learning to bake a cake:

- First, you learn **what a cake is** and **what it should taste like**.
 - Then you can follow a recipe (like code!) to make one.
-

Ἁ 2. Step-by-Step Learning Works Better

Jumping into code too fast can be confusing, especially if someone doesn't know:

- What the numbers represent,
- What the code is trying to do, or
- Why one answer might be better than another.

Starting with a real-life example (like candy or scores) helps the brain connect to the idea.

💻 3. Code is a Tool, Not the Goal

The goal is to **understand mean, median, and mode** — code is just a **tool** to calculate them faster. Once the concept is clear, you can learn:

That said — **if you already understand the ideas** and want to jump to code next time, just say so! I can switch it up based on how *you* learn best.

Prompt : **Can you make your explanation shorter?** or “**Show the same data as a pie chart instead of a bar chart**.

Absolutely! Here's a **shorter version** of the explanation:

📊 Mean, Median, Mode — Super Simple

Let's say these are the numbers:

2, 4, 4, 6, 9, 10, 100

Mean (Average):

Add them all up and divide by how many:

- $(2 + 4 + 4 + 6 + 9 + 10 + 100) \div 7 = 19.29$

Median (Middle):

Put them in order and pick the middle:

- Middle number is 6

Mode (Most Often):

The number that shows up the most:

- 4 appears twice → **Mode = 4**

Now, for your second request — you want to **see the same data as a pie chart** instead of a bar chart. Let's visualize how often each number appears in that list as a pie chart.

It looks like there was a hiccup generating the pie chart due to too many requests at the moment. No worries — here's a quick summary of how the **pie c!** would look:

Prompt : **Actually, I think the outlier threshold should be 2.5 instead of 3. Can you redo the analysis?**



Got it! You're asking to **redo the outlier analysis** using a threshold of **2.5 standard deviations** instead of **3**.

Let's go step-by-step using your original data:

📊 Data:

csharp

Copy code

```
[2, 4, 4, 6, 9, 10, 100]
```

⌚ Step 1: Calculate the Mean and Standard Deviation

We'll find:

- **Mean (average)**
- **Standard deviation** (how spread out the numbers are)

Then we'll define an **outlier** as any number more than **2.5 standard deviations** away from the mean.

Let me run the numbers for you now.

It looks like the system still isn't processing code due to high activity, but no worries — I'll walk you through the math manually!



Prompt : **can you summarize the conversation**

Sure! Here's a summary of our conversation:

🧠 Key Concepts Covered

1. Mean, Median, Mode:

- Explained what each one means and how they're different.
- Used a simple number set [2, 4, 4, 6, 9, 10, 100] to calculate:
 - Mean ≈ 19.29
 - Median = 6
 - Mode = 4

2. Child-Friendly Version:

- Gave a simple explanation for a 10-year-old using candy examples.
- Focused on making the idea easy and relatable.

3. Why No Code at First:

- Explained that it's better to understand the *concepts* before diving into *code*.
- Code is a tool — not the goal.

4. Pie Chart Request:

- You asked to show the number frequencies as a pie chart.
- System error prevented it, but I explained how the chart would look:
 - 4 would have the biggest slice (appears twice).
 - All others would be equal (appear once).

5. Outlier Analysis with 2.5 Standard Deviations:

- Redid the outlier check using a tighter threshold (2.5 instead of 3).
- Found that 100 is close to the cutoff, but **not quite an outlier** statistically.

Observation :

The Data Analyst GPT responded in a clear, step-by-step manner, adapting explanations based on the user's needs. It simplified complex concepts, answered follow-up questions thoughtfully, adjusted calculations when asked, and provided helpful summaries — all while maintaining a user-friendly tone. Even when technical limitations (like chart rendering) occurred, it offered descriptive alternatives to keep the conversation useful.

Conclusion

The Custom GPT — Data Analyst — demonstrated strong communication, flexibility, and analytical reasoning. It not only answered my questions but adjusted to new requests, simplified concepts, and explained its own logic clearly. Despite a technical hiccup with chart generation, it offered useful alternatives, showing reliability and user-centered design. Overall, it performed like a knowledgeable, responsive assistant well-suited for both learning and data exploration tasks.

Task 3 : Practice with a Research Assistant

Prompt : Write a short research article on *Machine Learning in Healthcare*

Writing Purpose *

Please share the motivation and what you hope to achieve with your topic

I am exploring options and ideas for machine learning use case in health care domain

Submit

Basic Fact Writer, Ethicist, Data Scientist, Healthcare Professional

Observation :

When I looked at the brainstorming stage in STORM AI, it felt like the tool was thinking out loud before writing the actual paper. It started by asking focused questions about the topic and then divided the information into different perspectives — for example, one tab showed general facts, another focused on ethics, one analyzed the data science side, and another reflected a healthcare professional's point of view. Each section added a different kind of knowledge or reasoning, almost like a team of experts sharing ideas before drafting.

The tool didn't just list random points; it organized them in a logical order — first explaining what machine learning does in healthcare, then moving to challenges, ethical issues, and ways to fix them. It even referenced example sources and repeated key patterns across tabs, showing that it was building a full picture of the topic.

Overall, the brainstorming process seemed to help STORM AI structure its thoughts, connect different aspects of the subject, and make sure the final article was balanced and detailed. It's like watching an AI map out its plan step by step before turning it into a polished research paper.

Experience using Strom AI for Research :

From what I observed while using STORM AI, it seems to work by breaking down the topic into smaller, connected parts before writing anything. It first identifies the main areas related to the topic — in my case, things like medical imaging, predictive analytics, ethics, and data privacy — and then gathers information or examples about each one. The brainstorming tabs (like "Fact Writer," "Ethicist," and "Data Scientist") make it look like the system uses different viewpoints to organize its thoughts before generating the final article.

I think it uses some kind of language model that's been trained on a wide range of academic and professional information. It probably compares patterns from what it already knows, then builds an outline and fills in each section logically, almost like how a person would plan an essay before writing. Once it has the structure, it combines all the ideas into paragraphs that read smoothly and make sense together.

Overall, it feels like STORM AI is simulating a collaborative writing process — almost like having a small research team made up of different experts (a scientist, an ethicist, a healthcare worker) working together to write one complete paper. It doesn't just give random facts; it tries to reason through the topic in a structured way and explain it clearly.

Conclusion :

Using STORM AI to generate a research article was actually pretty interesting. It felt like the tool was guiding me through the entire process — from brainstorming ideas to building a clear, structured paper. I liked how it showed different perspectives, like the ethical and technical sides, before writing the final draft. The article it produced was well-organized, easy to understand, and covered most of the key points I expected. Overall, it felt like working with a smart research partner that helps you think deeper and write faster.

4 - Chatbot

Website Used : <https://mizou.com/explore>

Creating an student account at Mizou

Clicking on build a chat bot with AI Generated option

The screenshot shows the 'ChatBot' interface. On the left, there's a form for setting learning objectives and grade level. The 'Learning Objectives' section contains a text input field with the placeholder: 'Students will analyze and evaluate the strategic decisions made by Napoleon Bonaparte during the Battle of Austerlitz. They will identify key military tactics that led to his victory and discuss how this battle solidified his reputation as a military genius.' Below it is a 'Grade Level' dropdown set to 'Choose a grade...'. A large blue button labeled 'Get ideas' is at the bottom. To the right is a 'Preview' area with a 'Clear' button.

This screenshot shows the same interface but with different input. The 'Learning Objectives' section now has the placeholder: 'Students would like to discuss how World War II could have been avoided.' The 'Grade Level' dropdown is set to 'University'. The 'Get ideas' button is present at the bottom.

Choose option and generate

Engage students in a structured debate on key events and decisions that could have changed history, enhancing critical thinking and argumentation skills.

← → ⌛ mizou.com/create-bot?type=guided&page=explode

≡

← ChatBot

Learning Objectives ⓘ

Students would like to discuss how World War II could have been avoided.

Grade Level ⓘ

University

Generating ChatBot ... (98%)

• •

Preview

This screenshot shows the 'ChatBot' creation interface on the mizou.com website. It includes fields for 'Learning Objectives' (a box containing 'Students would like to discuss how World War II could have been avoided.'), 'Grade Level' (set to 'University'), and a progress bar labeled 'Generating ChatBot ... (98%)' with three dots below it. To the right is a 'Preview' section which is currently empty.

← → ⌛ mizou.com/create-bot?type=guided&page=explode

≡

Oh no!

We're having some temporary technical difficulties right now. We're working on the problem as fast as we can!

In the meantime, try refreshing the page or hitting "back" on your browser just in case things are working again...

We're really sorry!

If this message does not go away within five minutes, please report the following code to the team: 1761946185617x176031785055613440

This screenshot shows an error message page with a red header. The main text reads: 'Oh no!' followed by 'We're having some temporary technical difficulties right now. We're working on the problem as fast as we can!'. It also includes a message about refreshing the page, an apology, and a specific reporting code: 'If this message does not go away within five minutes, please report the following code to the team: 1761946185617x176031785055613440'.

Unfortunately due to server issue, Mizou wasn't responding, but post this conversational chat bot would be prepared