Ilya Shafir
UCSD AI/ML Bootcamp - March 2024 Cohort

33.4.1 Capstone Project Deployment Method

31st March 2025

Chosen Deployment Method: Streamlit (Community Edition)

I have selected Streamlit Community Edition for deploying my Twitter Sentiment Analysis models due to its simplicity, quick setup, interactive user interface, and cost-effectiveness since it has a free tier and offers cloud compute. It is particularly suited for small-scale applications, prototyping, and demos. I've considered Gradio and FastAPI as alternatives and the comparison is below.

Comparison of Deployment Tools:

Criteria	Streamlit	Gradio	FastAPI
Ease of Use	High (simple, intuitive)	High (simple, intuitive)	Medium (requires some setup)
Speed of Deployment	Fast	Very Fast (minimal steps)	Moderate (requires coding)
Cost	Free (Community tier)	Free	Free
Interactivity/UI	High (rich UI elements)	Medium-High (good UI elements)	Low (minimal built-in UI)

Performance	Medium (good for demos)	Medium (ideal for quick demos)	High (optimized for production)
Monitoring & Logging	Basic built-in	Basic built-in	Advanced (requires integration)
Flexibility	Medium	Medium	High (very customizable)

Rationale for Choosing Streamlit:

- **Cost-Effectiveness:** Ideal as it has a free-tier service that supports this small scale project.
- **Interactivity:** Robust UI options to provide for end-users.
- **Ease and Speed:** I can perform rapid prototyping without extensive setup or coding changes migrating from Google Colab.

Post-Deployment Plan:

- **Monitoring:** Leverage Streamlit's basic monitoring and logging for tracking usage patterns and model predictions.
- **Performance Tracking:** Regularly assess model accuracy by periodically sampling predictions. Some of this may require isolating certain tweets to perform human judgement on sentiment.
- **Model Options:** If newer sentiment models get released, they may offer faster and/or more accurate sentiment analysis. I could then swap out the ensemble for those newer models as needed.
- **Scaling:** Since Streamlit Community Edition has limited resources, the dataset sizes must be limited for analysis. If moving on to larger data sizes, I would consider a paid platform, either staying on Streamlit, or another similar platform that is cost effective and offers another good UI.
- **Improvements:** Add more result analysis features, such as additional charts, more user customization as to what and how they want to extract statistics. Also options for user to choose the models in their ensemble and make sentiment comparisons that way.

Integration in ML Pipeline:

Streamlit seamlessly integrates with existing Python ML workflows, providing a unified, interactive interface that is easy to deploy, update, and makes it user-friendly. Since I'm using existing

Sentiment Analysis models, part of the pipeline would be to update to more modern and effective models.

Raw Data ➜ Preprocessing ➜ Streamlit App (Inference/UI) ➜ User Interaction

↳ **Monitoring & Logging ➜ Ensemble Model Review ➜ Update Ensemble**