Internship Report – NULLCLASS (June–July 2025)

Name: Agneya S Nambiar
Internship Period: 01 June 2025 – 01 July 2025

Introduction

This internship at **NULLCLASS** focused on hands-on application of machine learning techniques, model training, and data analysis using real-world datasets. Over the course of one month, I completed three tasks involving natural language processing, public dataset analysis, and fine-tuning a text-to-image model. These tasks helped me connect academic learning to practical implementation, preparing me for future roles in AI and data science.

Background

Before starting this internship, I had a basic understanding of Python, data visualization, and machine learning concepts. I had also worked with tools like Pandas, NumPy, Matplotlib, and Jupyter Notebooks. However, I had minimal hands-on experience with Hugging Face Transformers, TensorFlow Datasets, or advanced deep learning model training. This internship helped me bridge those gaps.

Learning Objectives

- Learn how tokenization and encoding work using pretrained language models.
- Understand the structure and composition of a public image dataset (Oxford Flowers 102).
- Explore and visualize dataset statistics and sample images.
- Fine-tune a text-to-image model like Stable Diffusion with a custom dataset.
- Overcome computational limitations with lightweight training strategies.
- Save models, use evaluation metrics, and present findings effectively.

Activities & Tasks

Task 1: Tokenization and Encoding using Pretrained Language Model

In this task, I used **Hugging Face Transformers** to load a pretrained model (BERT) and tokenize various input sentences. Tokenization breaks a sentence into words or subwords, and encoding converts them into numerical input suitable for the model. I printed out:

- Tokens
- Input IDs
- Attention masks
- Embeddings (model outputs)

This gave me a clear idea of how language models interpret and represent text internally. I also learned how to extract the last hidden states as features for further downstream tasks.

Task 2: Public Dataset Exploration – Oxford Flowers 102

In this task, I used tensorflow_datasets to explore the Oxford Flowers 102 dataset, which contains images of 102 flower categories.

Steps Taken:

1. Loaded the dataset using tfds.load().
2. Counted the number of classes and the number of images in train/test/validation sets.
3. Sampled 200 images and measured their dimensions (height, width).
4. Used matplotlib to create a 3x3 grid of flower images with class labels.
5. Created a boxplot to visualize image resolution distribution.
6. Saved plots and a CSV of image stats in a folder called images_and_statistics.

This task helped me understand the importance of dataset preprocessing, class balance, and input normalization before applying deep learning models.

Task 3: Fine-Tuning a Text-to-Image Model (Stable Diffusion)

This task was the most challenging yet exciting. I attempted to fine-tune **Stable Diffusion** using a small custom dataset of images and captions (e.g., "a red flower on a white background").

At first, I used the standard training loop with diffusers, transformers, and accelerate, but my **laptop started lagging** heavily and became unusable. I resolved this by implementing **lightweight training techniques**:

- Froze VAE and text encoder.
- Reduced batch size and image resolution.
- Trained for fewer steps (200–300 instead of thousands).

Steps Performed:

- Set up the training pipeline using UNet, VAE, scheduler, tokenizer.
- Prepared dataset with prompts.json and images/.
- Encoded inputs, added noise, predicted with UNet, and calculated MSE loss.
- Used torch.save() to store trained weights.
- Generated sample images after training.

The final model was able to generate basic images from simple prompts. I saved the model weights and samples and uploaded larger files to Google Drive (linked in README.md).

Skills & Competencies Developed

- Hugging Face Transformers (tokenization, encoding)
- TensorFlow Datasets (data exploration and visualization)
- Model evaluation with metrics (accuracy, loss)
- Fine-tuning generative models (Stable Diffusion)
- Lightweight training optimization
- Data preprocessing and augmentation

- GitHub version control, Google Drive model linking
- Python programming and debugging

Feedback & Evidence

I documented every step with:

- Screenshots of training logs and graphs
- Sample outputs from the tokenizer and image generator
- Boxplots and CSV from dataset analysis
- All .ipynb files with markdown explanations and code outputs
- GitHub repository with project folders, saved models, and Drive links

These have been pushed to GitHub and shared as instructed.

Challenges & Solutions

Challenge	Solution
Laptop was lagging during text-to-image model training	Switched to **lightweight training** by freezing VAE/text encoder and reducing batch size
Difficulty in dataset formatting for Stable Diffusion	Studied Hugging Face documentation and GitHub examples
Time constraints due to training steps	Limited training to fewer epochs and verified results manually
Model evaluation confusion	Used confusion_matrix, accuracy_score, and precision_score from sklearn

Outcomes & Impact

- Completed all 3 tasks independently and submitted within deadline.
- Gained deep understanding of NLP and computer vision pipelines.
- Learned to work with limited hardware and find practical solutions.
- Developed portfolio-worthy projects for GitHub.
- Improved confidence in working with real-world ML applications.

Conclusion

This internship at **NULLCLASS** was a transformative experience. Each task was hands-on, research-intensive, and pushed me to solve problems independently. I particularly enjoyed Task 3 as it combined creativity with machine learning.

The structured approach taught me the importance of proper data handling, training optimization, and model evaluation. I'm grateful for this opportunity and am confident that these skills will support my academic and professional journey in AI and Data Science.