Day 1(30-07-2025): Dataset Preparation

Goal: Prepare high-quality balanced and imbalanced datasets from the raw dataset.

Tasks:

Load the original dataset (`607448` reviews)

- Preprocess the text:
  - Lowercase, remove punctuation
  - Remove null values,duplicates etc

    Create 2 datasets for intial training and prototyping.

- **Balanced Dataset**:
  - 100,000 reviews
  - 20,000 samples for each star rating (1–5)
  - Used for training **Model A**
- **Imbalanced Dataset** (remaining 100,000):
  - Distributed according to a realistic rating distribution
  - Used for training **Model B**
  - Rating distribution:
    - ☆ 1 → 10% → 10,000
    - ☆ 2 → 15% → 15,000
    - ☆ 3 → 25% → 25,000
    - ☆ 4 → 30% → 30,000
    - ☆ 5 → 20% → 20,000
- Save: deep_balanced_dataset.csv, deep_imbalanced_dataset.csv

Steps:

1. **Balanced dataset creation:**
   - Use `.groupby('Rating').sample(n=16000)`
   - Save this 100k dataset for Model A
2. **Imbalanced dataset creation:**
   - Drop balanced dataset rows (`.drop(balanced_df.index)`)
   - Use `.groupby('Rating').sample(n=X)` with values per class from the table
3. **Ensure no overlap** between both sets

Output:

- Cleaned datasets ready for modeling
- Notebook: `deep_dataset_preparation.ipynb`

Day 2(31-07-2025): Model Training (Model A & Model B)

Goal: Train and save two separate deep learning models and document the training process of balanced model.

Tasks:

- Split both datasets into **Train (80%) / Test (20%)**
- Tokenize and pad sequences using `Tokenizer`
- Train:
  - **Model A** → Trained on balanced data
  - **Model B** → Trained on imbalanced data
- Model architecture: choose one
  - Simple LSTM / Bi-LSTM
  - RNN
  - GRU
- Save models:
  - `deep_model_A.h5`, deep_`model_B.h5`
  - `deep_tokenizer.pkl`
- Document the balanced model training process.

Output:

- Two trained models (A & B)
- Notebook:

`deep_balanced_model_training.ipynb,deep_imbalanced_model_training.ipynb`

- `Pdf: deep_balanced_model_training.pdf`

Day 3(01-07-2025): Evaluation & Cross-Testing

Goal: Test both models across both test sets to compare fairness and generalization and document the imbalanced training process

Tasks:

- Evaluate Model A on:
  - Balanced test set
  - Imbalanced test set
- Evaluate Model B on:
  - Imbalanced test set
  - Balanced test set
- Report:
  - Accuracy, precision, recall, F1-score (macro/weighted)
  - Confusion matrices (save visuals)
  - Class-wise performance comparison
- Document model behaviors, strengths, and weaknesses
- Document imbalanced training process.

Output:

- Notebook: `deep_evaluation_and_cross_test.ipynb`
- Plots: Confusion matrices, score tables
- Summary: `evaluation_report.pdf`, `deep_balanced_model_training.pdf`

Day 4(02-08-2025): Flask UI Development

Goal: Build a simple Flask web app that accepts review input and shows predictions from both models.

Tasks:

- Create folder: `ui/`
- File: `app.py`
  - Load both `.h5` models and tokenizer
  - Preprocess user input
  - Predict with both models
- Template: `templates/index.html`
  - Text input
  - Two prediction outputs (Model A & B)
- Test locally (`flask run`)
- Add screenshot for documentation

Output:

- Flask app with functional UI
- Folder: `ui/`
- Screenshot: `flask_ui.png`

Day 5(03-08-2025): Documentation & GitHub Push

Goal: Finalize all documentation and push code + reports to GitHub.

Tasks:

- Write `README.md`:
  - Project intro
  - Dataset creation
  - Preprocessing
  - Model architectures
  - Evaluation results
  - UI instructions
  - Screenshot of UI
- Create PDFs:
  - Model A report
  - Model B report
  - Balanced training report
  - Imbalanced training report

    - Cross test summary
    - UI design flow + deep learning summary
- Folder structure:

Output:

- Fully documented project
- Live local Flask UI
- Repo ready for submission