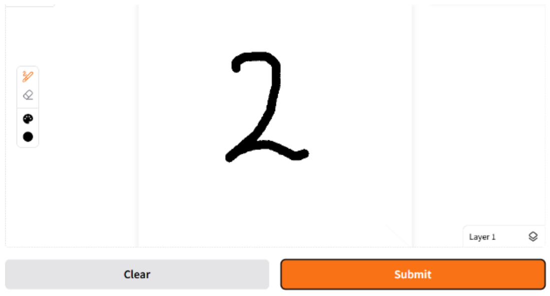
1. MNIST Digit Classification

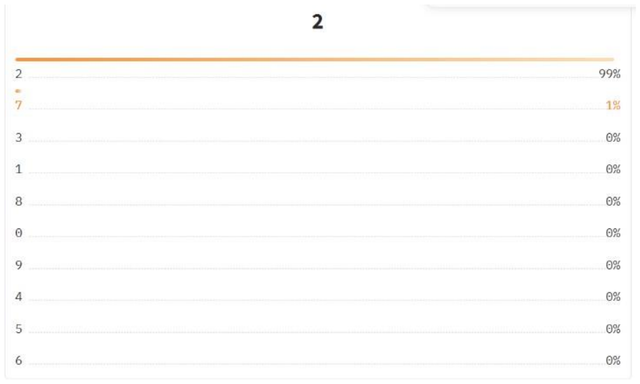
Pipeline:

- **Data**: Use load_digits() from scikit-learn or a small subset of MNIST.
- **Preprocessing**: Handle the data as needed — clean, scale, transform, or encode based on what improves your model.
- **Modeling**:
  - Use a Multi-Layer Perceptron (MLP) or tree-based models (e.g., RandomForest, XGBoost) or any suitable model.
  - Evaluate performance: accuracy, classification report.
- **UI**:
  - Gradio app: User draws a digit on a canvas → displays the predicted digit.

Demo



→



```
2
2   99%
7   1%
3   0%
1   0%
8   0%
0   0%
9   0%
4   0%
5   0%
6   0%
```

2. Heart Failure Prediction

Pipeline:

- **Data:** Heart Failure Dataset
- **Preprocessing:** Handle the data as needed — clean, scale, transform, or encode based on what improves your model.
- **Modeling:**
  - Use a Multi-Layer Perceptron (MLP) or tree-based models (e.g., RandomForest, XGBoost) or any suitable model.
  - Evaluate performance: accuracy, classification report.
- **UI:**
  - Gradio app: User inputs patient data → displays prediction: at-risk / not at-risk.

3. Credit Card Fraud Detection

Pipeline:

- **Data:** Credit Card Fraud Dataset
- **Preprocessing:** Handle the data as needed — clean, scale, transform, or encode based on what improves your model.
- **Modeling:**
  - Use a Multi-Layer Perceptron (MLP) or tree-based models (e.g., RandomForest, XGBoost) or any suitable model.
  - Evaluate performance: accuracy, classification report.
- **UI:**
  - Gradio app: User inputs transaction features → displays fraud/not fraud.

4. Garbage Image Classification

Pipeline:

- **Data:** Garbage Dataset
- **Preprocessing**: Handle the data as needed — clean, scale, transform, or encode based on what improves your model.
- **Modeling**:
  - Use a CNN or VGG, ResNet, or any suitable CNN model.
  - Evaluate performance: accuracy, classification report.
- **UI:**
  - Gradio app: User inputs a photo → displays the predicted class.

5. Face Mask Detection

Pipeline:

- **Data:** Face Mask Dataset
- **Preprocessing**: Handle the data as needed — clean, scale, transform, or encode based on what improves your model + use Data Augmentation.
- **Modeling**:
  - Use a CNN or VGG, ResNet, or any suitable CNN model.
  - Evaluate performance: accuracy, classification report.
- **UI:**
  - Gradio app: User inputs a photo → displays the predicted class.

Deadline: Thursday, August 7, 2025

Evaluation:

- **Project hosted publicly on GitHub, HuggingFace.**
- **All team members present a live presentation, covering:**
    - **Data Preprocessing: What transformations were made and why?**
    - **Model Performance: Show results (accuracy, classification report, R^2, etc.).**
    - **Obstacles s Challenges: What went wrong? How did you solve it?**
    - **Future Work: What can be improved or added?**

Bonus Points (Optional)

- Use **Streamlit**, **React.js**, or any modern **frontend framework.**
- Add a **Flask** or **FastAPI** backend, e.g., (/predict route).
- Build and publish a **Docker image** to **Docker Hub.**

Note: Each member should deeply understand the entire project and especially their contribution. You may be asked about any part.

Feel free to contact me if you need help or if any information is missing

Good Luck, and Have Fun <3