General Imputation Assignment Report

1. Introduction

The objective of this assignment was to develop a thought architecture for the general imputation problem using neural networks, specifically Multi-Layer Perceptrons (MLPs). The focus was to compare two approaches:

- Independent MLPs: One MLP per feature, trained independently to predict that feature using the other 15 features as input.
- Conjunct MLP: A single MLP with multiple outputs, trained to reconstruct all features (autoencoder-style).

I experimented with different training strategies:

- Independent MLPs: Trained on full data vs. subsets (but focused on full data for best results). Explored imputation strategies like single-pass, iterative, and optimized Mode (reverse order with temperature scaling and refinement).
- Conjunct MLP: Trained on full data with reconstruction loss, using iterative imputation.

The dataset (DATA.csv) contains 19,900 rows and 16 categorical features (f1 to f16), each with values A–P (16 categories). No missing values in original data. I split 80/20 for train/test (15,920 / 3,980 rows).

All experiments were conducted in a Jupyter notebook using PyTorch on a GPU for efficiency.

2. Methods: What I Did and Why

2.1 Data Preprocessing

- **Label Encoding**: Used LabelEncoder per feature to convert A–P to 0–15 for targets.
- **One-Hot Encoding**: Global OneHotEncoder for inputs (256 dims total: 16 features × 16 categories).
- **Why?** Categorical data requires numerical input for MLPs; one-hot captures nominal nature, label encoding for loss calculation.
- **Test Simulation**: Introduced missing values at rates 5%, 10%, 20% by zeroing one-hot blocks.

2.2 Independent MLPs Approach

- **Architecture**: 16 separate MLPs, each with:
- Input: 256 (full one-hot, target block zeroed)

- Hidden layers: 128 → 64 (ReLU, Dropout 0.2)
- Output: 16 (softmax)
- **Training Strategy**:
  - Trained on complete training data (vectorized one-hot tensor for speed).
  - CrossEntropyLoss, Adam optimizer, 40 epochs, batch 128.
  - Total time: ~45 seconds for 16 models on GPU.
- **Imputation**:

Single-pass prediction using observed and imputed values, with reverse-order prediction and temperature scaling (T=0.5) for improved confidence.

- **Rationale:** Feature-specific models allow specialization and parallel training.

2.3 Conjunct MLP Approach

- **Architecture:** Single MLP with:

  Input/Output: 256

  Hidden layers: 256 → 256 (ReLU, Dropout 0.3)

- **Training:** MSELoss on one-hot reconstruction, Adam, 80 epochs, batch size 256.
- **Imputation:** 7 iterative reconstruction passes with clamping and renormalization.
- **Rationale:** Joint model may capture inter-feature correlations but increases complexity.

2.4 Evaluation

- **Metric**: Accuracy on imputed missing positions (compared to ground truth).
- **Baseline**: Mode imputation per feature (~25% accuracy).
- **Rationale:** Mode baseline provides a non-learning reference; any valid model must outperform it.

3. Results

3.1 Training Insights

- **Independent MLPs**: Rapid convergence (loss ~0.1–0.2). Full data training optimal.
- **Conjunct MLP**: Slower convergence (loss ~0.05), but failed to learn meaningful imputation patterns.
- **Key Difference:** Independent models are faster to train, parallelizable, and more accurate.

3.2 Imputation Accuracy

Final comparison (from experiments; seed=99 for reproducibility):

Missing Rate	Independent MLPs	Conjunct MLP	Mode Baseline
5%	**97.65%**	22.41%	25.00%
10%	**96.40%**	22.39%	24.97%
20%	**92.38%**	22.38%	24.91%

3.3 Analysis of Results

- **Independent MLPs** achieve 92.38% accuracy at 20% missing, demonstrating exceptional robustness.
- **Conjunct MLP** performs at ~22.4%, equivalent to random guessing in a 16-class problem.
- **Mode baseline** achieves ~25%, as expected from the most frequent category.
- **Independent MLPs outperform mode by 3.7×** and conjunct by **4.1×,** confirming superior learning of inter-feature dependencies.
- **No performance gap** between imputation strategies — optimized single-pass matches multi-iteration, indicating stable predictions.

4. Discussion and Insights

- **Key Insight**: Independent MLPs are more convenient (parallel training, feature-specific) and accurate for categorical imputation. Conjunct better for continuous data (not this case).
- **Training Strategies**: Full data optimal; subsets useful for large datasets to reduce time, but hurt accuracy here. Iterative imputation essential for multiple missings.
- **Recommendations**: Use independent MLPs with Optimized Mode for best results. Future: Embeddings instead of one-hot for scalability.

Overall, this experiment showed independent MLPs as the superior architecture for general imputation on this dataset.