TrackTempo: Encoding Strategy and Dataset Flow 2025-04-08 06:06

1. Core Goals (Recap)

Your preprocessing pipeline outputs an inference dataset named like:
  inference_dataset_YYYY-MM-DDTHH-MM.pkl

This dataset must be upgraded to a training-ready dataset by appending:

- Final race positions
- Winner flag
- Any other ground-truth labels

This merge happens AFTER encoding categorical fields.

2. Whats Needed for Training

a) .idx Fields (Embedding Indices)

These are required by the model per runner:

- country_idx
- going_idx
- sex_idx
- type_idx
- ...

They are generated using LabelEncoders saved as:
  embedding_encoders_YYYY-MM-DDTHH-MM.pkl

These must be present in:

- Training dataset
- Inference dataset (for evaluation/prediction)

3. What Went Wrong (Root Cause)

We incorrectly tried moving encoding into the flattening stage.

Consequences:

- Encoding logic was applied too early
- LabelEncoders rely on a full clean dataset
- NaNs and inconsistent columns caused transform errors
- The original flow (with an explicit encoding step) was more robust

4. Recommended Flow (Restoration Plan)

Step 1: Preprocessing Pipeline

- Output: Clean inference file (no encodings)
- e.g. inference_dataset_YYYY-MM-DDTHH-MM.pkl

Step 2: Apply Encoders Script

- Input: Inference dataset + LabelEncoders
- Output: Encoded dataset with _idx fields
- Output: inference_dataset_with_idx.pkl

Step 3: Merge for Training

- Input: Encoded inference + results.csv
- Output: model_ready_train.pkl

5. Dataset Encoding Matrix

Dataset Stage	Encoded?	.idx Columns	Labels Present
Inference (raw)	No		
Inference + Encoded	Yes		
Training-Ready	Yes		
Evaluation	Yes		Optional

Conclusion

Do NOT encode during flattening.
Keep a clean 3-stage pipeline:
   preprocess encode merge (for training)

This gives you flexibility to evolve label handling, model types, and evaluation logic indepen