# TrackTempo AI: Transformer-Based Modeling Plan

## Race Intelligence Modeling System

**Pseudonym:** J. D. Farrell

**Tagline:** Structured Prediction in the Racing Domain

### Project Direction: Transformer-Based Modeling

- Model races as **sets of interacting runners** using Transformers.
- Each race becomes a matrix: `[num_runners, feature_dim]`
- Apply **softmax per race** to generate win probability distribution.
- Use embeddings for: **jockey_id, trainer_id, course**
- Include race context in features: **going, field_size, class, type**
- Train using **CrossEntropyLoss** on softmax output per race.
- Split using **GroupKFold/GroupShuffleSplit** by `race_id`
- Handle variable number of runners via **padding/masking**

### Feature Strategy

- Core features: `or, rpr, draw, age, ts, lbs`
- Trainer/Jockey form: last 14 runs, win %, profit, career stats
- Race metadata: distance, going, GoingStick, field size
- Use NLP embeddings for: `comment`, `spotlight` (MiniLM/BERT)
- Engineer race-relative features: `rpr_rank, or_percentile, trainer_form_rank`

### Training & Evaluation

- Loss: Softmax + log loss (per race)
- Evaluation: Accuracy, log loss, simulated ROI, win-rate
- Training batches = one race per sample
- Incorporate **trainer/jockey embeddings** for bias detection

### Next Steps (Post-Data Accumulation)

- Begin training once ~1000+ pre-race-clean horses are collected
- Build batcher for padded race matrices
- Develop trainer/course-aware embeddings
- Add leaderboard simulation mode
- Track model evolution and test online inference loops

*Generated by TrackTempo AI | Race Intelligence Modeling System | J. D. Farrell*