Horse Racing Prediction Model Report

Wentao Zou

May 2025

1 Data Summary

The horse racing prediction model uses a dataset of 1,757 entries with 122 features, reduced to 1,481 after dropping rows with missing target prediction values, which are `official_finish`, `speed_rating`, and `win_time`. The most related and significant features for prediction include `horse(name)`, `jockey`, `trainer` details, race conditions, and past performance information such as recent finishes. The external test dataset, named `CDX0515_filtered.csv`, contains 86 entries with 200 columns. I processed it to align with training features by referring to the sample column mapping provided by Professor Adam. Numerical features like `distance_f` and `dollar_odds` are normalized using `MinMaxScaler`. Missing values are handled by imputing -1 for numerical and `unknown` for categorical columns.

2 Model

The model is a custom `BartForRegression` based on the BART transformer (`facebook/bart-base`), designed for regression to predict three target outputs: `official_finish`, `speed_rating`, and `win_time`. It combines BART's encoder-decoder architecture with a linear regression head (768 to 3 dimensions). Input text sequences are created by concatenating feature values, tokenized to a maximum length of 512. The model is trained using PyTorch, with the `AdamW` optimizer and Mean Squared Error loss.

3 Training Process

The dataset is split into training (1,035 entries), validation (223), and test (223) sets randomly. Training runs for up to 100 epochs (pre-set) with a batch size of 64 when running under a GPU environment. The model triggers early stopping with a patience of 2 epochs to prevent overfitting. Early stopping monitors validation loss, saving the best model weights (`best_bart_model.pt`) when validation loss improves. The extended epoch count allows the model to converge thoroughly, as evidenced by the training log showing consistent loss reduction (e.g., train loss from 3,519.83 in epoch 1 to 91.86 in epoch 81). Training stopped early at epoch 82 when validation loss (208.28) did not improve in 2 steps, indicating optimal convergence.

4 Results

Internal test evaluation indicates moderate prediction accuracy, especially for `official_finish`, which is the most critical prediction. I also made predictions on the latest data for May 15th's races. However, the absence of the training scaler (`training_scaler.pkl`) when running predictions on another day under a different environment (CPU) may cause scaling inconsistencies.

5 Key Points I Learn

1. **Patience in Early Stopping**: A patience of 2 balances model convergence and overfitting prevention, allowing sufficient epochs for loss reduction while stopping when validation loss plateaus (e.g., epoch 82).

2. **Extended Epochs**: Pre-setting 100 epochs ensures the model explores the loss landscape relatively fully, critical for the complex BART architecture, as seen in the steady loss decline over 81 epochs.

3. **Data Challenges**: Column mismatches (e.g., `weight` containing sire names) and missing scaler files highlight the need for robust preprocessing and artifact management.

4. **Scalability**: The model handles diverse datasets via flexible column mapping and imputation, though target absence in test data limits evaluation.

This model demonstrates effective prediction for horse racing outcomes, especially on `official_finish`. Future improvements could include saving the training scaler and incorporating target data for external tests to enable metric computation.

6 Suggestions for this Course and Final Project

1. I hope we can have more homework questions or tasks on dealing with different formats of files from the very beginning of the course, so that we will be well prepared for the final project. This is my first time reformatting an XML file, and it took quite a lot of time. Thanks for your guiding code and other students' help.

2. Probably next time you can introduce horse racing at an early stage, so we can know more details about this sport. When processing raw data, we got confused about many variables and had to choose which variables besides `finish_position` could be better targets for predictions.