## Data Exploration “data\_exploration.ipynb”

To begin the horse racing project, I worked with a dataset stored in XML format. The dataset contains a vast number of fields—potentially thousands—covering various aspects of each race. My primary objective is to predict the finishing position of a horse, so I focused on exploring variables that may influence this outcome.

Guided by both domain knowledge and data descriptions found online, I categorized the data into two main areas:

* Race-Level Information: Variables that describe the overall race context, such as "Surface", "RaceNumber", "Track", "TrackCondition", and "Distance".
* Horse-Level Information: Attributes specific to each horse and its performance history, including "HorseName", "Jockey", "Trainer", "HorseWins", and other performance indicators.

Given the volume of available fields, it was impractical to examine each one individually. Therefore, I implemented a search mechanism to check whether specific variables I hypothesized to be relevant—such as "Wind", "Humidity", and other environmental factors—were present in the data files. This allowed me to efficiently filter and shortlist features for further analysis and modeling.

## Data Preparation for Model Training “data\_construction\_final.py”

Despite the extensive variable exploration in the previous stage, several concerns arose regarding the usability and consistency of the data. The dataset includes a mix of numerical and categorical variables, and many fields were found to be either redundant, ambiguous in meaning, or lacking documentation, which complicated their interpretation and preprocessing.

Due to these challenges, I ultimately decided to utilize the preprocessed training dataset provided by the professor. This dataset was curated to include relevant features with standardized formats, allowing for more efficient modeling and analysis without the need to resolve every inconsistency in the raw XML files.

## Model Training “model\_construction\_bart\_and\_prediction.ipynb”

As the first step, I performed data exploration and preprocessing on the training dataset. This involved several transformations to prepare the data for modeling:

* Categorical Encoding: Categorical variables such as "Horse", "Surface", and "Jockey" were encoded using random numerical identifiers. I chose this approach over traditional one-hot encoding (dummy variables) because the combinatorial explosion of interactions between high-cardinality categorical features would have made the feature space excessively large and sparse.
* Unit Conversion: Race distances originally recorded in furlongs were converted to meters for consistency and interpretability.
* Field Extraction: Fields such as "RaceNumber" were parsed and explicitly extracted to facilitate merging and grouping operations.

After preprocessing, I trained a predictive model using the Bayesian Additive Regression Trees (BART) algorithm. BART is a non-parametric, ensemble-based method that captures nonlinear relationships and interactions between variables effectively, making it well-suited for modeling complex outcomes like horse finishing positions.

## Prepare the Prediction file “prediction\_file\_perparation.ipynb”

To ensure consistency between the training and test datasets, I applied the same set of predictors used during model training to the test data. For categorical variables such as "Horse", "Trainer", and "Jockey", I reused the same encoding mappings established in the training phase.

For any new, unseen categories in the test set that did not appear in the training set, I assigned a random encoding from the existing pool. This approach preserved the structure of the feature space while preventing issues during model inference caused by unknown categorical values.

## Prediction “model\_construction\_bart\_and\_prediction.ipynb”

Using the trained BART model, I predicted the finishing position for each horse in the test set. In this context, a lower predicted finishing position indicates a better performance, with the lowest value corresponding to the winning horse in each race.

To translate these predicted rankings into probabilistic outcomes, I applied a softmax transformation to the inverse of the predicted finishing positions. This yielded a probability distribution over horses in each race, representing the estimated probability of each horse winning. This probabilistic output can be particularly useful for applications such as betting strategy, ranking evaluation, or ensemble modeling.

## Things Could Be Done in the Future

Several aspects of the dataset and domain knowledge present opportunities to further improve prediction performance:

* Historical Performance: The dataset includes previous race records for each horse, which can provide valuable temporal features such as recent form, consistency, and momentum. Incorporating these sequential patterns could enhance predictive accuracy.
* Horse Identity and Lifecycle: The current training set is based on races from 2023, and many of the horses in the dataset may have since retired. Instead of assigning random numerical encodings, we could leverage biological and lineage information—such as breed, sire, or dam—to build more generalizable representations of horses, capturing performance potential across related individuals.
* Strategic Behavior: In real-world race planning, horses often participate in multiple races on the same day or across short intervals. It is plausible that jockeys or trainers may intentionally underperform in certain races to conserve energy or optimize outcomes in higher-stakes events. Modeling such strategic behavior—for example, by identifying races with higher expected payoff or using conditional performance history—could lead to more realistic predictions.

These insights highlight the potential to enrich the model using domain-aware features and temporal dynamics, paving the way for future iterations of this work.