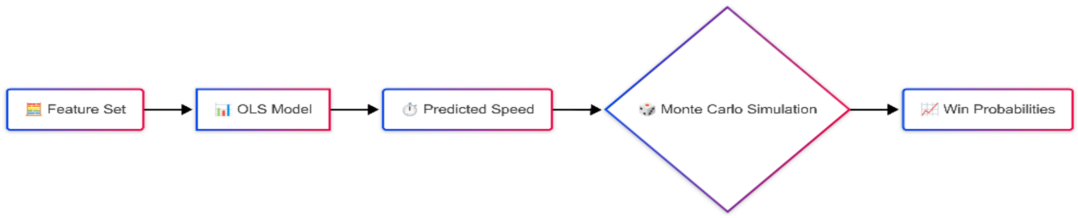
Horse Racing Probabilistic Modelling Report

1. Introduction

This project aims to generate valid, interpretable **win probabilities** by modelling **horse speed as a continuous target** rather than a binary outcome. During model selection, we evaluated various machine learning algorithms—including LightGBM, Random Forest, Support Vector Regression (SVR), and Multi-Layer Perceptron (MLP)—and found their performance in predicting horse speed and subsequently deriving win probabilities either marginally better or even inferior to **Ordinary Least Squares (OLS)**. Given OLS's simplicity, transparency, and adherence to statistical assumptions, we ultimately selected it as our primary model. Predictions from the OLS model were then converted into valid race-level probabilities summing to one through **Monte Carlo simulations**.

2. Methodology



2.1 Target Variable (Y):

We use a horse's **race speed as a continuous proxy for performance** because ultimately, races are won by the fastest runner. Unlike a simple win/lose outcome, speed provides a nuanced measure of performance. However, since race-day conditions and random fluctuations can affect speed, we treat the predicted speed as a **stochastic variable** and use **Monte Carlo simulation** to convert these speed estimates into win probabilities. This method captures the inherent uncertainty in determining which horse will win.

2.2 Features (X):

Our model incorporates a blend of **original features**—such as race distance, track condition, and trainer/jockey ratings—and **13 engineered race-relative features** designed to capture each horse's competitive standing within a race.

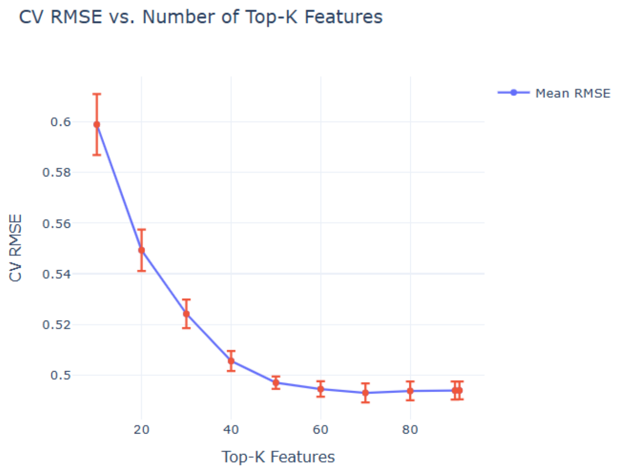
Since approximately 20% of the horses in the test data are new and were not present in the training set, horse-specific imputation isn't feasible. Instead, we **impute missing values** by grouping the training data by **track condition ("Going")** and using the **conditional median**, ensuring consistent and robust treatment of missing data.

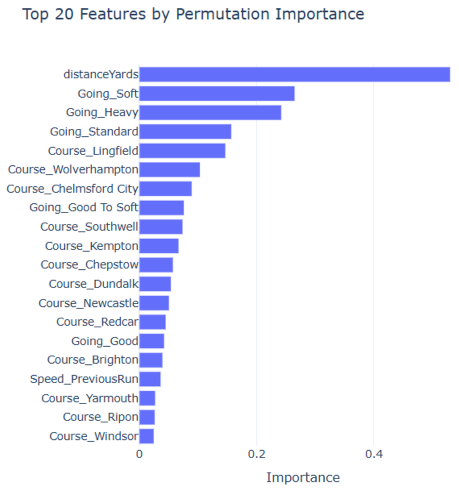
2.3 Feature Transformation:

To satisfy the **OLS normality assumption**, we applied the **Yeo-Johnson transformation**, which can normalise both positive and negative values while preserving data continuity.

- Applied **Yeo-Johnson transformation** to reduce skewness in numeric features and target (Speed).
- Used **One-Hot Encoding** for categorical features.
- To prevent **multicollinearity**, removed features with high correlation (>0.95)

2.4 Feature Selection:





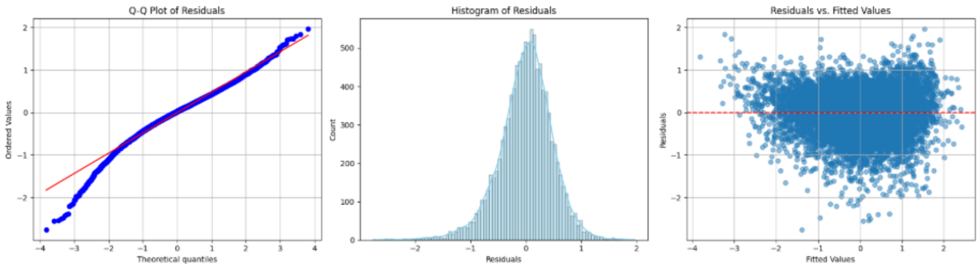
We selected the top **70 features** using **permutation importance** and **RMSE validation**. **distanceYards** emerged as the **strongest predictor**, confirming the impact of race length. Track condition (**Going**) and location (**Course**) also ranked highly, reflecting surface and environmental effects. While historical features like **Speed_PreviousRun** added value, they were less influential than current race conditions. Notably, **9 race-relative features** were retained, with **Rel_SpeedPrev**, **Age_Rank**, and **Rel_NMFPLTO** standing out—highlighting the importance of **relative speed**, **age**, and **recent form**.

3. Model Design

We model the horse's actual race speed as a continuous target variable **y**, using 70 selected features $\mathbf{x}_1$, $\mathbf{x}_2$, …, and $\mathbf{x}_{70}$ as explanatory variables to build the following linear regression model:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{70} x_{70} + \varepsilon$$

3.1 Model Diagnostics:

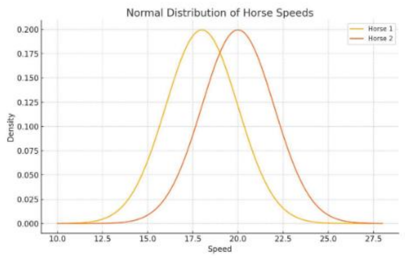


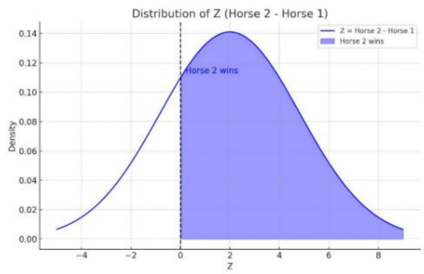
The OLS model met key assumptions: **linearity, homoscedasticity, residual independence** (Durbin-Watson statistic: ≈ 1.996), and acceptable **normality**.

3.2 Model Evaluation on Test Set:

Model	RMSE	MAE	R^2
OLS Model	0.4194	0.3005	0.6926

4. From Speed Predictions to Win Probabilities





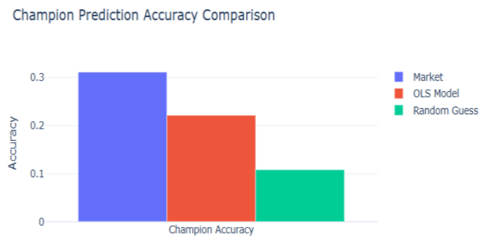
We assumed each horse's speed follows a **normal distribution $\mathbf{S_i} \sim \boldsymbol{N(\mu_i, \sigma)}$**, where $\boldsymbol{\mu_i}$ is the **predicted speed**, and $\boldsymbol{\sigma}$ is the **global standard deviation from training data**.

The first figure illustrates the individual speed distributions for two horses with predicted speeds of 18 and 20, respectively. To determine which horse is likely faster, we consider their **difference in speeds** as another normal distribution $\boldsymbol{Z} = \boldsymbol{S}_{\text{Horse 2}} - \boldsymbol{S}_{\text{Horse 1}}$. The second figure visualises this difference, clearly highlighting the region (shaded area) where Horse 2 outperforms Horse 1 (when $\boldsymbol{Z} > 0$).

Due to analytical complexity, we employed **Monte Carlo simulations** (50,000 iterations per race) to estimate each horse's win probability. This ensures probabilities are valid (0–1) and **sum to exactly 1 per race.**

5. Probabilistic Evaluation

Model	Log Loss	Brier Score
OLS + Monte Carlo	0.3281	0.0936



5.1 Market Benchmark:

- **Spearman correlation with market odds: 0.6455**
- **Horse-level accuracy**: Market **18.06%**, Model **14.97%**
- **Top-1 accuracy**: Market **31.09%**, Model **22.12%**, Random **10.81%**

While the market is more accurate, our model shows strong alignment (Spearman 0.6455) without relying on market data like **Betfair SP**. This makes it a viable, interpretable alternative—especially useful when market odds are unavailable or incomplete.

6. Assumptions, Limitations & Challenges

- Used an assumption of **normal distribution** to model horse speeds, which may oversimplify real-world variability.
- Used **global variance (σ^2)** due to data sparsity, rather than estimating for each horse individually.
- **Unseen horses/trainers/jockeys** in test data lacked historical context.
- **Non-finishers** (timeSecs = 0) are unpredictable and treated as **unobserved noise**.
- **Missing external factors** like weather and horse health were not available