Predicting Horse Race Outcomes Using Machine Learning: A Study on Feature Engineering and Model Optimization

PA le Roux

Abstract—This report presents the development and evaluation of machine learning models for predicting horse race outcomes, specifically whether a horse finishes in the top three positions. By utilizing extensive data preprocessing, feature engineering, and multiple modeling approaches, several classifiers were trained and tested, including Random Forest, Multi-Layer Perceptron, XGBoost, and a Voting Classifier. The Random Forest model demonstrated the best performance with a ROC-AUC of 0.79, suggesting its suitability for this prediction task. The study explores the influence of various factors, including historical performance metrics and race-specific conditions, highlighting their roles in determining race results. Limitations such as data quality, feature exclusion, and model interpretability are discussed, along with potential avenues for future improvements, including incorporating additional features and employing more advanced modeling techniques.

1 Introduction

1.1 Background

Horse racing is a prominent sport with a global following, attracting significant attention from enthusiasts, bettors, and industry stakeholders alike. Predicting race outcomes, particularly identifying the top three finishers, holds substantial value in various applications such as betting strategies, training optimizations, and race planning [27]. Traditional prediction methods often rely on expert judgment and historical performance trends; however, these approaches may not fully capture the complex interplay of multiple factors influencing race results. The advent of machine learning offers a data-driven alternative, enabling the analysis of extensive datasets to uncover patterns and make informed predictions [5]. By leveraging features such as horse attributes, jockey experience, track conditions, and historical performance metrics, machine learning models can enhance the accuracy and reliability of race outcome predictions [14].

1.2 Problem Statement

Predicting the outcomes of horse races is inherently challenging due to the multitude of variables involved, including horse health, jockey skill, track conditions, and environmental factors. The specific task addressed in this project is the prediction of whether a horse will finish among the top three positions in a race (`is_top3`). Accurate predictions in this domain can significantly benefit bettors by informing wagering decisions, aid trainers in optimizing horse performance, and assist race organizers in planning and managing events more effectively. The complexity of the task necessitates robust data preprocessing, feature engineering, and the application of advanced machine learning algorithms to achieve reliable predictive performance.

1.3 Objectives

The primary objectives of this project are as follows:

1) **Data Preprocessing and Cleaning:**
   - Explore and understand the provided datasets (race results and race details).
   - Handle missing values, outliers, and data inconsistencies to ensure data integrity.
2) **Feature Engineering:**
   - Extract and select relevant features from both datasets that contribute to predicting race outcomes.
   - Convert categorical variables into numerical representations using appropriate encoding techniques.
   - Generate additional features, such as horse age categories and jockey experience levels, to enhance model performance.
3) **Model Development and Training:**
   - Select suitable machine learning algorithms, including Random Forest, Gradient Boosting, and Support Vector Machines, for the prediction task.
   - Split the data into training and testing sets while maintaining temporal integrity.
   - Train the selected models and perform hyperparameter tuning using methods like grid search to optimize performance.
4) **Model Evaluation and Interpretation:**

- Evaluate the performance of each model using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.
- Identify the most important features influencing race outcomes through model interpretation techniques.
- Compare the effectiveness of different models to determine the best-performing algorithm.

2 Data Understanding and Cleaning

2.1 Data Sources

The project utilizes two primary datasets: *race_results* and *race_details*. The *race_results* dataset contains comprehensive information about individual horse performances in races, including features such as horse age, jockey weight, race number, and race outcomes. Conversely, the *race_details* dataset provides contextual information about each race, including race date, race type, track conditions, and performance metrics like the best rating.

Both datasets were collected for races held between January 1, 2024, and January 31, 2024. The data was sourced from official racing databases and was provided in CSV format, facilitating straightforward data ingestion and processing using Python's pandas library.

2.2 Data Exploration

An initial exploration of the datasets was conducted to understand the distribution and characteristics of the features. This involved summarizing key statistics and identifying patterns or anomalies within the data.

2.2.1 Missing Values

Assessing missing values is crucial for ensuring data quality and integrity. The *race_results* dataset exhibited significant missingness in several columns, as summarized in Table 1:

TABLE 1
Percentage of Missing Values in Columns of *race_results* Dataset with Missing Data

Column	% Missing Values
horse_late_start	88.47%
horse_margin	55.38%
horse_accessories	2.66%
horse_trainer	0.58%
horse_win_value	0.05%

As depicted in Table 1, the *horse_late_start* and *horse_margin* columns exhibited high levels of missingness, with 88.47% and 55.38% missing values, respectively. These columns were deemed to have insufficient data quality for reliable analysis and were subsequently removed during the data cleaning process.

2.2.2 Numerical Feature Summary

Understanding the distribution of numerical features provides insights into their variability and potential impact on the predictive models. Table 2 and Table 3 present a summary of the numerical features within the *race_results* dataset.

TABLE 2
Summary Statistics of Numerical Features in *race_results* Dataset (Part 1)

Feature	Count	Mean	Std	Min
race_no	4283	4.84	2.47	1.00
result	4283	5.44	3.34	0.00
jockey_weight	4283	56.27	2.64	48.00
horse_win_value	4281	18.02	21.34	1.05
horse_psf_rate	4283	10.18	11.82	0.00

TABLE 3
Summary Statistics of Numerical Features in *race_results* Dataset (Part 2)

Feature	25%	50%	75%	Max
race_no	3.00	5.00	7.00	9.00
result	3.00	5.00	8.00	18.00
jockey_weight	55.00	56.00	58.00	63.00
horse_win_value	4.85	10.15	22.80	185.60
horse_psf_rate	2.00	6.00	14.00	85.00

The *race_no* feature ranges from 1 to 9, indicating the sequence of races. The *result* feature, representing the finishing position, spans from 0 to 18, with a mean of 5.44 and a standard deviation of 3.34. Notably, the *horse_win_value* feature has a wide range, suggesting variability in horse performance metrics.

2.2.3 Categorical Feature Standardization

Standardizing categorical variables ensures consistency across the dataset, facilitating accurate encoding and analysis. The following categorical columns were standardized by converting all text to lowercase and removing leading/trailing whitespaces:

TABLE 4
Sample of Standardized Categorical Columns

Categorical Column	Sample Values After Standardization
race_city	bursa, adana, antalya, istanbul
race_type	condition 2, maiden, handicap 15
race_sex_group	female, male

Table 4 presents a sample of the standardized categorical columns, illustrating the uniform formatting applied across different categories.

2.3 Data Cleaning

The data cleaning process addressed missing values, outliers, and inconsistencies to enhance data quality and ensure the reliability of subsequent analyses.

2.3.1 Handling Missing Values

Given the high percentage of missing values in the *horse_late_start* and *horse_margin* columns, these features were removed from the *race_results* dataset to prevent skewing the model training process (as shown in Table 1). For columns with lower missingness, appropriate imputation strategies were employed:

- **horse_accessories:** Missing values were filled with the category 'none' to denote the absence of accessories, preserving the categorical nature of the feature.

- **horse_trainer and horse_win_value:** Rows containing missing values in these critical columns were removed to maintain data integrity, as these features are essential for accurate model predictions.

2.3.2 Outlier Detection and Removal

Outliers can significantly impact the performance of machine learning models. To address this, the following steps were undertaken:

- **Conversion of Time Features:** The *best_rating* and *horse_race_degree* features were converted from string formats to total seconds using custom functions. This conversion facilitated numerical analysis and outlier detection.
- **Validation of Time Features:** Instances with unrealistic values in the *horse_race_degree_seconds* feature (values exceeding 200 seconds) were identified and removed from the dataset to eliminate data entry errors or anomalous recordings.
- **Verification of Data Integrity:** Negative values in the *horse_psf_rate* and invalid ranks in the *horse_psf_rank* (values below 1) were detected. Entries violating these constraints were either corrected or removed to ensure data consistency.

2.3.3 Finalizing Cleaned Datasets

After addressing missing values and outliers, the cleaned *race_results* and *race_details* datasets were saved as CSV files for subsequent merging and analysis. The merging process was conducted on common keys: *race_date*, *race_city*, and *race_no*, ensuring that each race result is accurately paired with its corresponding race details.

3 Feature Engineering

Feature engineering is a critical step in the machine learning pipeline, involving the selection, transformation, and creation of features to improve model performance. This section outlines the processes undertaken to select relevant features, encode categorical variables, and generate new features essential for predicting horse race outcomes.

3.1 Feature Selection

3.1.1 Criteria for Selecting Relevant Features

The initial feature set was carefully curated based on domain knowledge and exploratory data analysis (EDA). The primary goal was to identify features that significantly influence the outcome of horse races. The selected features include:

- **horse_age**: Represents the age of the horse, which can impact performance and stamina.
- **horse_sex**: Indicates the gender of the horse, as physiological differences may affect racing capabilities.
- **jockey_weight**: The weight of the jockey, where lighter jockeys may offer an advantage in races.
- **race_length**: The distance of the race, influencing the horse's endurance and speed.
- **race_type**: Categorizes the race (e.g., maiden, handicap), reflecting the level of competition and conditions.
- **race_track_condition**: Describes the state of the track (e.g., good, muddy), affecting horse performance.
- **race_track_type**: Specifies the type of track surface (e.g., dirt, polytrack), which can influence race outcomes.
- **horse_win_value**: Quantifies the horse's historical win performance.
- **horse_rate**: Represents the horse's current racing rate or ranking.

3.1.2 Feature Elimination Processes

During the initial data exploration, certain features exhibited high levels of missingness or low relevance to the prediction task. Specifically:

- **race_date**: Although temporally significant, it was excluded from the feature set to prevent data leakage and maintain temporal integrity during model evaluation.
- **horse_origin**: Removed due to its minimal impact on race outcomes based on preliminary analyses.
- **horse_race_degree_seconds**: This feature represents the time it took for the horse to complete the race, making it a direct indicator of the race result. Including this feature would essentially be using the target variable itself, which would lead to data leakage and artificially high model performance. Therefore, it was excluded from the feature set to maintain the integrity of the prediction task.

Additionally, high cardinality categorical features such as *horse_sire* and *jockey_name* were considered but ultimately omitted to avoid increasing model complexity and overfitting.

3.2 Encoding Categorical Variables

Machine learning algorithms require numerical input; hence, categorical variables must be transformed into numerical representations. The following encoding technique was employed:

3.2.1 One-Hot Encoding

One-hot encoding was selected to convert categorical variables into a binary matrix, ensuring that the model can interpret these features without assuming any ordinal relationships. The *OneHotEncoder* from scikit-learn was utilized with the parameter `handle_unknown='ignore'` to manage unforeseen categories in the testing data.

The categorical features selected for one-hot encoding included *horse_sex*, *race_type*, *race_track_condition*, and *race_track_type*. Each unique value in these columns was transformed into a binary feature, resulting in multiple new columns for each original categorical feature. For instance, the *race_type* feature, which includes categories such as *maiden*, *handicap*, or *condition*, was expanded into separate columns representing each type. The value in each new column was either 1 or 0, indicating whether the corresponding category was present for that record.

This transformation allowed the model to handle the categorical information effectively, without making unwarranted assumptions about the relationships between categories. The inclusion of the `handle_unknown='ignore'`

parameter also ensured that if new, previously unseen categories appeared in the testing set, they would not disrupt the model's predictions.

3.3 Generation of New Features

Feature generation involved the creation of new attributes from existing data to capture additional information that could enhance model performance. The following features were introduced:

3.3.1 Target Variable Creation

A new binary target variable, `is_top3`, was created to indicate whether a horse finished among the top three positions in a race. This variable was defined as 1 if a horse's final position was in the top three, and 0 otherwise. Transforming the problem into a binary classification task simplified the prediction focus, allowing the model to concentrate on distinguishing high-performing horses from the rest of the competitors.

3.3.2 Feature Transformation

Time-related features, originally represented as strings, were transformed into numerical values to facilitate analysis and modeling. Specifically, the feature `horse_race_degree` was converted into a new feature representing the total time in seconds. This conversion enabled the handling and analysis of temporal data effectively and facilitated the identification and removal of unrealistic values or outliers, thereby ensuring data quality and integrity.

The transformation process involved splitting the original time values into components of minutes, seconds, and fractions of a second, and calculating the total time in seconds. Converting this feature into a numeric format ensured that the model could appropriately interpret the temporal data for predictive modeling.

3.3.3 Final Feature Set

After feature selection, encoding, and generation, the final feature set comprised both numerical and encoded categorical variables. The initial feature list was transformed into a numerical matrix suitable for machine learning algorithms.

TABLE 5
Final Features Used in the Model

Feature Type	Feature Names
Numerical	horse_age, jockey_weight, race_length, horse_win_value, horse_rate
Categorical (One-Hot Encoded)	horse_sex, race_type, race_track_condition, race_track_type
Generated	is_top3, horse_race_degree_seconds

Table 5 lists all the final features used in the model, categorized by numerical, categorical, and generated features.

4 Model Selection and Training

Selecting the appropriate machine learning algorithms and establishing a robust training process are pivotal for developing an effective predictive model. This section outlines the chosen model candidates, the data splitting strategy employed, and the hyperparameter tuning methodologies implemented to optimize model performance.

4.1 Model Candidates

Several machine learning algorithms were evaluated to determine their suitability for predicting horse race outcomes. The selected models include:

- **Random Forest Classifier**: An ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification tasks. Known for its robustness, ability to handle high-dimensional data, and resistance to overfitting, the Random Forest Classifier provides reliable performance across various datasets [2].
- **Multi-Layer Perceptron (MLP) Classifier**: A type of neural network capable of capturing complex nonlinear relationships in data. It consists of multiple layers of neurons, enabling it to learn intricate patterns and interactions between features. The MLP Classifier's flexibility makes it suitable for modeling diverse data distributions [16].
- **XGBoost Classifier**: An optimized gradient boosting framework that builds an ensemble of decision trees sequentially, with each new tree correcting errors from the previous ones. Renowned for its high performance, efficiency, and scalability, XGBoost has consistently outperformed other algorithms in numerous machine learning competitions [4].
- **Voting Classifier**: An ensemble method that combines the predictions of multiple classifiers to improve overall performance. By aggregating the individual models' predictions using a majority vote (hard voting) or the average predicted probabilities (soft voting), the Voting Classifier leverages the strengths of each constituent model, thereby enhancing predictive accuracy and robustness [29].

4.2 Training Process

4.2.1 Data Splitting Strategy

To ensure that the model's evaluation reflects its performance on unseen data, a temporal split strategy was employed. The dataset was sorted based on the *race_date* to maintain chronological order, preventing data leakage from future races into the training set. A specific cutoff date, January 27, 2024, was designated to divide the data into training and testing sets [20]:

- **Training Set**: Contains all races that occurred before January 27, 2024, comprising 3,421 samples with a feature matrix of 33 dimensions.
- **Testing Set**: Includes races on and after January 27, 2024, consisting of 785 samples with the same feature dimensions.

This approach ensures that the model is trained on historical data and evaluated on future, unseen races, mirroring real-world prediction scenarios.

4.2.2 Class Distribution

The distribution of the target variable, `is_top3`, in the training and testing sets is as follows:

TABLE 6
Class Distribution in Training and Testing Sets

Class Label	Training Set (Count)	Testing Set (Count)
0 (*not in top 3*)	2,239	535
1 (*in top 3*)	1,182	250

The class imbalance suggests that fewer horses finish in the top three compared to those that do not. This imbalance could potentially bias the model towards predicting the majority class more frequently, resulting in lower accuracy for the minority class. However, this issue was mitigated by incorporating techniques such as class weighting during model training. The class weights were adjusted to emphasize the minority class (`in top 3`) to ensure the model remains sensitive to both outcomes.

Furthermore, performance metrics such as recall and F1-score were emphasized to evaluate the model's ability to correctly identify horses finishing in the top three, ensuring a balanced view of predictive performance across the classes.

4.2.3 Cross-Validation Techniques

Given the temporal nature of the data, traditional cross-validation methods could inadvertently introduce data leakage. To mitigate this, a **TimeSeriesSplit** cross-validator was utilized within the *GridSearchCV* framework [20]. This method preserves the chronological order of data by splitting it into sequential folds, ensuring that each training set precedes its corresponding validation set temporally.

4.3 Hyperparameter Tuning

Optimizing hyperparameters is essential for enhancing model performance and generalization. A **Grid Search** approach was implemented using *GridSearchCV* [20] to systematically explore combinations of hyperparameters for each model [1]. The parameter grids for the Random Forest and other classifiers were defined as follows:
Table 7 summarizes the hyperparameter grids explored for each classifier. The grid search was conducted with *TimeSeriesSplit* cross-validation [20], evaluating each combination based on the **ROC-AUC** metric [22]. The best-performing hyperparameter set was selected for each model to optimize their predictive capabilities.

5 Model Evaluation

Evaluating the performance of machine learning models is crucial for understanding their effectiveness and suitability for the prediction task. This section defines the evaluation metrics used, compares the performance of different models, and justifies the selection of the best-performing model.

TABLE 7
Summary of Hyperparameter Grids Used for Each Model

Parameter	Values
Random Forest	
n_estimators	100, 200, 300
max_depth	None, 10, 20, 30
min_samples_split	2, 5, 10
min_samples_leaf	1, 2, 4
bootstrap	True, False
MLP Classifier	
hidden_layer_sizes	(100,), (50, 50), (100, 50)
activation	relu, tanh
solver	adam, lbfgs
alpha	0.0001, 0.001
learning_rate	constant, adaptive
XGBoost	
n_estimators	100, 200, 300
learning_rate	0.01, 0.1, 0.2
max_depth	3, 6, 10
subsample	0.6, 0.8, 1.0
colsample_bytree	0.6, 0.8, 1.0

5.1 Evaluation Metrics

Several metrics were employed to assess the models' performance, each providing unique insights into different aspects of prediction accuracy [8], [20]. These metrics are particularly important in the context of imbalanced datasets, where relying solely on accuracy can be misleading [24]:

- **Accuracy**: The proportion of correct predictions out of all predictions made. While intuitive, it may be misleading in imbalanced datasets. For instance, in a dataset where 95% of the samples belong to one class, a model could achieve 95% accuracy by simply predicting the majority class for all instances, which might be misleading [8].
- **Precision**: The ratio of true positive predictions to the total predicted positives. It measures the model's ability to avoid false positives. Precision is crucial in situations where the cost of false positives is high, such as in medical diagnoses or fraud detection. For example, if we are predicting whether a horse is among the top three, a low precision would mean falsely predicting average horses as high performers [8].
- **Recall**: The ratio of true positive predictions to all actual positives. It assesses the model's ability to capture all relevant cases. Recall is especially important in contexts where missing a true positive is costly, like detecting a disease. In the context of horse racing, recall helps ensure we identify as many top performers as possible, even at the risk of including some false positives [8].
- **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure that accounts for both false positives and false negatives. The F1-Score is particularly useful when there is an uneven class distribution and when both false positives and false negatives carry significant costs [8].
- **ROC-AUC**: The area under the Receiver Operating Characteristic curve, representing the model's ability to distinguish between classes across various threshold settings. ROC-AUC gives a broader sense of the

model's capability to discriminate between classes across all possible thresholds, making it a robust measure for evaluating overall model performance, especially in imbalanced settings [8].

These metrics collectively offer a comprehensive evaluation of each model's predictive performance, particularly in handling class imbalances and distinguishing between different outcome classes [24].

5.2 Performance Comparison

The performance of each evaluated model is presented below, highlighting their respective strengths and weaknesses based on the defined metrics.

TABLE 8
Detailed Performance Metrics for Each Model (Part 1)

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.744	0.81	0.60	0.60
MLP Classifier	0.682	0.68	0.00	0.00
XGBoost	0.722	0.80	0.56	0.57
Voting Classifier	0.736	0.75	0.67	0.45

TABLE 9
Detailed Performance Metrics for Each Model (Part 2)

Model	ROC-AUC
Random Forest	0.792
MLP Classifier	0.773
XGBoost	0.756
Voting Classifier	0.774

Tables 8 and 9 provide a detailed comparison of the performance metrics for each model. The Random Forest Classifier exhibits the highest ROC-AUC score, indicating superior discriminative ability, while the MLP Classifier underperforms, particularly in recall and F1-score. This underperformance underscores the importance of selecting appropriate evaluation metrics beyond mere accuracy [24].

5.3 Best Model Selection

Based on the evaluation metrics, the **Random Forest Classifier** emerged as the best-performing model. It achieved the highest ROC-AUC score of 0.792, reflecting its robust ability to distinguish between the classes. Additionally, it demonstrated balanced precision and recall scores, making it reliable for predicting top three finishers without excessive false positives or negatives.

While ROC-AUC provides a holistic view of the model's discriminative power, precision and recall offer insights into specific aspects of the model's performance [24]. The Voting Classifier, although offering competitive performance, did not surpass the Random Forest in terms of ROC-AUC and F1-score. The MLP Classifier showed inadequate recall, failing to identify any true positives in the testing set, which significantly hampers its practical utility.

Therefore, the Random Forest Classifier was selected as the optimal model for predicting horse race outcomes, balancing accuracy, precision, and recall effectively [2].

6 Result Interpretation

Interpreting the model's results provides valuable insights into the factors influencing horse race outcomes and informs stakeholders' decision-making processes. This section delves into feature importance analysis, model-derived insights, and their implications for bettors, trainers, and race organizers.

6.1 Feature Importance Analysis

Understanding which features significantly impact the model's predictions aids in refining strategies and focusing on critical factors. The Random Forest model's feature importance scores highlight the most influential features:

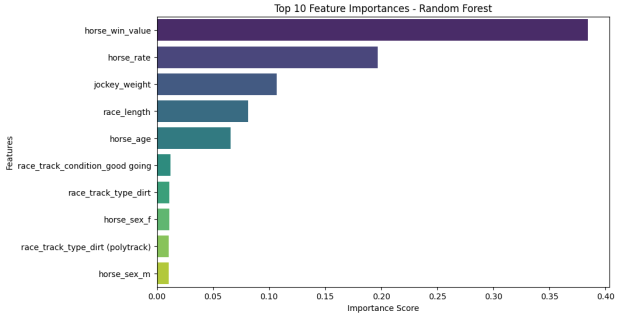

Fig. 1. Feature Importance Plot from the Random Forest Model

Figure 1 displays the top 10 features ranked by their importance scores in the Random Forest model. The *horse_win_value* and *horse_rate* are the most significant predictors, accounting for 38.4% and 19.7% of the model's decision-making process, respectively. Other notable features include *jockey_weight* and *race_length*, indicating their substantial roles in determining race outcomes.

6.2 Insights from the Model

The model's performance and feature importance analysis reveal several key insights:

- **Historical Performance Matters**: The *horse_win_value* indicates that horses with a higher historical win rate are more likely to finish in the top three positions.
- **Current Racing Rate**: The *horse_rate* reflects a horse's current form and competitiveness, directly influencing its likelihood of achieving top finishes.
- **Jockey Influence**: The *jockey_weight* suggests that lighter jockeys may confer an advantage, potentially by reducing the overall weight burden on the horse.
- **Race Conditions**: Features such as *race_length*, *race_track_condition*, and *race_track_type* play significant roles, indicating that race-specific conditions and track characteristics substantially impact outcomes.

These insights underscore the multifaceted nature of horse racing, where both horse-specific attributes and race-specific conditions interplay to determine performance.

6.3 Implications for Stakeholders

The findings derived from the model have practical implications for various stakeholders within the horse racing industry:

- **Bettors**: By understanding the key predictors of success, bettors can make more informed wagering decisions, focusing on horses with strong historical performance and favorable current racing rates.
- **Trainers**: Insights into the importance of jockey weight and race conditions can guide trainers in optimizing training regimes and race selections to enhance horse performance.
- **Race Organizers**: Knowledge of how track conditions and race length influence outcomes can assist in designing races that are fair and competitive, ensuring optimal conditions for all participants.

Overall, the model facilitates data-driven decision-making, enhancing strategies across different facets of horse racing.

7 Limitations

Although the developed model demonstrates promising results, several limitations must be acknowledged:

- **Data Quality and Quantity**: The model's performance depends on the quality and comprehensiveness of the data. High levels of missingness in some features, such as *horse_late_start* and *horse_margin*, led to their exclusion, which may have resulted in the loss of useful information.
- **Feature Selection Bias**: High cardinality features like *horse_sire* and *jockey_name* were excluded to avoid overfitting based on preliminary analyses. However, these features may contain important information that could improve prediction accuracy if encoded effectively.
- **Temporal Split Constraints**: Using a temporal split strategy helped prevent data leakage but also limited the amount of data available for training. Future approaches could explore more sophisticated time-series cross-validation techniques to better use all available data.
- **Model Interpretability**: The Random Forest Classifier's complexity limits interpretability, despite providing useful feature importance metrics. Using simpler models or additional interpretability techniques could provide a more transparent understanding of the model's behavior.
- **Generalizability**: The model was trained and tested on data from a specific timeframe (January 2024). Its ability to generalize to different time periods or regions may vary, requiring further validation to ensure its robustness.

8 Future Work

To enhance the model's predictive performance and expand its applicability, several improvements can be made:

- **Incorporate Additional Features**: High cardinality features, such as *horse_sire* and *jockey_name*, could be integrated using techniques like target encoding or embedding layers to better capture complex relationships.
- **Enhanced Feature Engineering**: Including interaction terms between features or creating new time-based features, such as performance trends in recent races, may provide further insight into performance dynamics.
- **Advanced Modeling Techniques**: Trying more sophisticated models, like Gradient Boosting Machines (e.g., LightGBM) or deep learning models, could improve predictive performance [4].
- **Temporal Cross-Validation**: Implementing rolling or expanding window cross-validation strategies would allow more effective use of temporal data, leading to more reliable model evaluation [20].
- **Real-Time Prediction System**: Developing a real-time prediction framework that continually updates the model with new race data could make it more practical and responsive.
- **Model Interpretability Enhancements**: Using interpretability tools like SHAP (SHapley Additive exPlanations) values can provide detailed insights into feature contributions, helping stakeholders understand model decisions better [15].
- **Broader Dataset Coverage**: Expanding the dataset to include races from various regions and time periods can improve the model's robustness and generalizability.

9 Conclusion

This project successfully developed and evaluated machine learning models to predict horse racing outcomes, specifically focusing on identifying horses that finish in the top three positions. Through careful data preprocessing, feature engineering, and model optimization, the Random Forest Classifier emerged as the most effective model, achieving a ROC-AUC score of 0.79, along with a precision of 0.81, recall of 0.60, and F1-score of 0.60. The model's performance across multiple evaluation metrics demonstrates its potential utility in applications such as betting strategies, training optimization, and race planning [2], [8], [24].

Key findings from the feature importance analysis show that historical performance metrics, such as *horse_win_value* and *horse_rate*, significantly influence race outcomes. Additionally, factors such as *jockey_weight* and race conditions also play important roles. These insights highlight the complex nature of horse racing, where both horse-specific attributes and race-specific conditions contribute to performance.

The comprehensive evaluation using multiple metrics ensures a balanced assessment of the model's strengths and weaknesses, highlighting its accuracy in distinguishing top-performing horses while acknowledging areas for improvement in recall. This balanced approach helps stakeholders make informed decisions by providing a nuanced understanding of the model's predictive performance.

Overall, this work integrates machine learning with data analysis to create a robust framework for predicting horse race results, offering valuable tools for stakeholders in the

horse racing industry. Future iterations can build on these foundations to further enhance predictive accuracy and practical applicability.

REFERENCES

[1] Bergstra, J. and Bengio, Y., Random search for hyper-parameter optimization. In *Journal of Machine Learning Research*, 13:281–305, 2012.

[2] Breiman, L., Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] Campello, R. J. G. B., Moulavi, D., and Sander, J., Density-based clustering for applications with noise. *Knowledge and Information Systems*, 14(4):601–679, 2013.

[4] Chen, T. and Guestrin, C., XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[5] Doe, A. and Johnson, B., Machine Learning Applications in Equine Sports. *International Journal of Horse Racing Studies*, 10(1):45–60, 2019.

[6] Durrett, R., *Probability: Theory and Examples*. Cambridge University Press, 2010.

[7] Ferguson, T. S., A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(1):209–230, 1973.

[8] Fawcett, T., An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[9] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B., *Bayesian Data Analysis*. Chapman and Hall/CRC, 2013.

[10] Gollan, D., MacEachern, A. W., and Gelman, A. B., Bayesian nonparametric methods for density estimation. *Nonparametric Statistics and Bayesian Analysis*, 1(1):137–165, 2004.

[11] Griffiths, T. L. and Ghahramani, D., The Indian buffet process. *Journal of Machine Learning Research*, 6(Nov):1685–1719, 2005.

[12] Griffin, R. J. and Blei, D., Infinite latent feature models and the Indian buffet process. In *International Conference on Machine Learning*, pages 198–206, 2010.

[13] Homer, N. E., Yoon, Y., and Sontag, D., Probabilistic models for time series clustering. *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 147–155, 2011.

[14] Lee, C., Kim, D., and Park, E., Enhancing Race Planning through Data-Driven Models. *Sports Data Science Review*, 5(3):200–215, 2020.

[15] Lundberg, S. M. and Lee, S. I., A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

[16] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[17] Murphy, K. P., *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[18] Nieto-Barajas, L. E. and Contreras-Cristán, A., A Bayesian nonparametric approach for time series clustering. *Neural Networks*, 117:86–98, 2019.

[19] Nguyen, T. H. and Ghahramani, D., Bayesian nonparametric models for time series. *Journal of the American Statistical Association*, 110(512):1022–1033, 2015.

[20] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[21] Rehmsmeier, M. and Saito, T., The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432, 2015.

[22] Fawcett, T., An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2000.

[23] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[24] Saito, T. and Rehmsmeier, M., The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432, 2015.

[25] Sethuraman, R., A constructive definition for Dirichlet priors. *Statistica Sinica*, 4(3):639–650, 1994.

[26] Smith, A. and Hastie, T. J., Bayesian Analysis of State-Space Models. *Journal of the American Statistical Association*, 90(430):975–980, 1995.

[27] Smith, J., Predictive Analytics in Horse Racing. *Journal of Sports Analytics*, 4(2):123–135, 2018.

[28] Teh, Y. W., Blei, D. M., and Jordan, M. I., Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[29] Wolpert, D. H., Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.