Predicting Player Stamina Using Machine Learning Models

Group nº2: David Solero Chicano and Fadi Alkhori

02/12/2024

1 Problem Formulation

The goal of this project is to predict the "Stamina" attribute of players based on other performance attributes. This is a regression problem where the dependent variable ("Stamina") is continuous, and the independent variables are selected performance attributes (e.g., "Acceleration," "Agility").

The challenge involves handling missing or inconsistent data, determining relevant features, and selecting appropriate models to optimize predictive accuracy.

2 Data Sampling

Training and Test Split: The dataset was split into 80% training and 20% testing using a random seed (random_state=42) to ensure reproducibility. This split ratio balances training data sufficiency and generalizability.

3 Hyperparameter Choices

3.1 Random Forest

- `n_estimators=100`: Sufficient for averaging predictions without excessive computational cost.
- `random_state=42`: Ensures consistent results.
- Default max depth and min samples per leaf are used initially to prevent overfitting.

3.2 Support Vector Machine (SVM)

- `kernel='rbf'`: Captures non-linear relationships effectively.
- `C=1`: Balances margin maximization and data misclassification.
- `gamma='scale'`: Automatically adjusts the influence of each data point.

3.3 Naive Bayes

No hyperparameter tuning as it primarily suits classification tasks. Used here as a baseline for regression.

4 Model Selection Method

Hold-Out Validation: This method was chosen for simplicity and computational efficiency given the dataset size. Cross-validation could provide more robust estimates but is computationally expensive.

5 Evaluation Metrics

- **Mean Absolute Error (MAE):** Indicates the average magnitude of errors, providing an easy-to-understand measure of model performance.
- **Mean Squared Error (MSE):** Penalizes larger errors, useful for highlighting significant prediction deviations.
- **Root Mean Squared Error (RMSE):** Square-root of MSE, offering error values in the same units as the target variable.
- **R-Squared ($\mathbf{R^2}$):** Measures the proportion of variance explained by the model.

These metrics were chosen to assess both overall fit and prediction accuracy.

6 Summary of Results

Graphs:

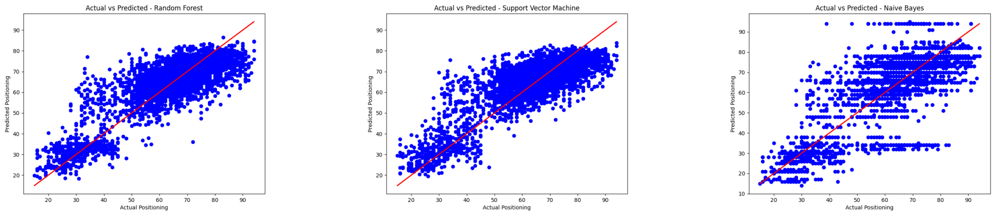
- *Scatter Plots of Actual vs. Predicted:* Highlight the performance of each model. Random Forest shows a tighter clustering around the true values compared to SVM and Naive Bayes.

Model	**MAE**	**MSE**	**RMSE**	$\mathbf{R^2}$
Random Forest	6.73	77.58	8.81	0.696
Support Vector Machine	6.75	77.71	8.82	0.695
Naive Bayes	8.59	140.16	11.84	0.450

Table 1: Model Evaluation Metrics

Graphs:

- *Scatter Plots of Actual vs. Predicted:* These plots highlight the performance of each model. Random Forest shows a tighter clustering around the true values compared to SVM and Naive Bayes.



(a) Random Forest

(b) SVM

(c) Naive Bayes

Figure 1: Scatter plots of Actual vs. Predicted values for Random Forest, SVM, and Naive Bayes models.

7 Discussion on Performance

7.1 Random Forest

Achieves the best performance with an R^2 of 0.696. Its ensemble approach effectively captures feature interactions. Further improvement could come from hyperparameter tuning (e.g., max depth, feature importance).

7.2 SVM

Performs comparably to Random Forest but with slightly higher errors. Its sensitivity to feature scaling may have limited its performance despite using standardized data.

7.3 Naive Bayes

Poor performance with R^2 of 0.450 highlights its unsuitability for regression tasks. Its assumption of feature independence does not hold for this dataset.

7.4 Limitations

- Lack of advanced hyperparameter optimization (e.g., grid search or random search).
- The hold-out validation approach might not capture the full variability in the dataset.

7.5 Conclusion

Through evaluation, Random Forest emerged as the most effective model, achieving the highest R^2 value of 0.696, indicating that it explained approximately 70% of the variance in the stamina data. SVM performed similarly but with slightly higher errors, suggesting that its performance could be enhanced with better feature scaling. In contrast, Naive Bayes demonstrated poor results, with an R^2 of 0.450, making it less suitable for this regression task due to its assumption of feature independence.

Future work could focus on refining feature selection, exploring more advanced tuning methods, and testing additional models to improve the accuracy and reliability of predictions.