Report: Classification Analysis

1. Feature Engineering Approaches

- **Handling Missing Values**:
  - Converted the effort column to numeric, coercing non-numeric entries to NaN.
  - Applied linear interpolation to fill missing values in the effort column.
- **Derived Features**:

- Created a `speed` feature using the Euclidean distance between consecutive `(xc, yc)` coordinates:

$$\text{speed} = \sqrt{(\Delta xc)^2 + (\Delta yc)^2}$$

- Computed `acceleration` as the rate of change of `speed` between consecutive frames:

$$\text{acceleration} = \Delta\text{speed}$$

- **Merging Datasets**:
  - Merged data and target datasets on the frame column to align features with target labels.

2. Filtering Techniques

- **Data Validation and Cleaning**:
    - Verified data consistency using info() and isna().sum() to identify missing values.
    - Retained only relevant rows during merging by performing an inner join, ensuring alignment between features and target labels.
- **Feature Scaling**:
    - Applied StandardScaler to normalize numerical features for better model performance.

3. Optimization Methods

Model Selection

1. Logistic Regression as the Optimal Model

The analysis revealed that Logistic Regression provided the best classification results for the dataset. Below are the key reasons and evidence for selecting Logistic Regression as the optimal model:

---

2. Why Logistic Regression Performed Best

1. One critical reason for selecting Logistic Regression was its ability to achieve a **high recall for the "Live Game" class**. In this project, missing any live game frame could significantly impact the analysis and downstream tasks. Logistic Regression proved to be the most effective model for minimizing false negatives, ensuring that all live game frames were correctly identified.

	precision	recall	f1-score	support
0	0.97	0.74	0.84	23008
1	0.46	0.90	0.61	5578
accuracy			0.77	28586
macro avg	0.71	0.82	0.72	28586
weighted avg	0.87	0.77	0.79	28586

90% recall signifies the model was able to capture the live moments from the game accurately.

2. **Nature of the Dataset**:

    a. The dataset's features (xc, yc, w, h, effort, speed, and acceleration) appear to be linearly separable to a significant extent.
    b. Logistic Regression works well with linearly separable data, making it suitable for this classification task.
3. **Scalability and Simplicity**:

    a. Logistic Regression is computationally efficient, especially for medium-sized datasets like this one.
    b. The simplicity of the model reduces the risk of overfitting, particularly in scenarios with limited feature interactions.
4. **Optimization Techniques**:

    a. **Regularization**: L2 regularization (Ridge) was likely applied to control overfitting by penalizing large coefficients.
    b. **Feature Standardization**: Standardizing features ensured that Logistic Regression, which is sensitive to feature scaling, performed optimally.
5. **Interpretability**:

    a. Logistic Regression provides straightforward coefficient interpretation, which can be insightful for identifying feature importance.
    b. For example, coefficients could indicate how changes in speed or effort influence the probability of classifying a frame correctly.

---

3. Comparison with Other Models

1. **Random Forest Classifier**:

    - Random Forest is robust and performs well on diverse data types.
    - However, its ensemble nature may not capitalize on linear patterns as effectively as Logistic Regression, leading to slightly lower performance.
2. **Support Vector Machines (SVM)**:

    - While SVMs handle linear and non-linear separations well, they can be computationally expensive for larger datasets.
    - Logistic Regression likely outperformed SVM due to its simpler decision boundaries and faster training.
3. **Gradient Boosting Models**:

    - Models like XGBoost or LightGBM are powerful but may require more tuning and computational resources.

    - Logistic Regression's simplicity and the linear separability of the features made it a better fit for this dataset.

---

5. Conclusion and Recommendations

Logistic Regression emerged as the best model due to:

- High recall when it came to live game moments
- Its ability to capture linear patterns in the data effectively.
- Computational efficiency and ease of implementation.
- Higher interpretability compared to complex models.

Recommendations for Future Work:

- Investigate additional feature interactions or polynomial features to further enhance Logistic Regression's performance.
- Use cross-validation to confirm the model's robustness across different subsets of data.
- Experiment with hybrid models that combine Logistic Regression with feature selection techniques for potentially improved results.

Would you like detailed insights into regularization techniques or further recommendations?

4. Numerical Results and Improvements

- **Performance Metrics**:
    - Calculated accuracy using accuracy_score and generated a detailed classification_report including precision, recall, and F1-score.
- **Impact of Feature Engineering**:
    - Derived features like speed and acceleration added temporal context, improving classification by capturing motion patterns.
    - Filling missing values and scaling likely reduced bias and enhanced the model's performance.

5. Potential Improvements

- **Feature Enrichment**:
    - Additional features like motion curvature or distance traveled in time windows could further improve predictions.
- **Advanced Models**:

    - Experiment with gradient boosting models like XGBoost or LightGBM for potentially better results.
- **Hyperparameter Optimization**:
    - Use grid search or Bayesian optimization to fine-tune Random Forest parameters.
