ML Pipeline Full Report – Hands-On Machine Learning

# Title Page

**Assignment Title: ML Pipeline – Hands-On Machine Learning  
Student Name: [Ali Haider]  
Course: Machine Learning for Robotics  
Instructor: [Mr. Basharat Hussain]  
Date of Submission: [16/02/2025]**

# 1. Dataset Selection & EDA Summary

**Dataset Overview:**  
- Domain: Sports Analytics (Cricket Match Outcomes)  
- Target Variable: Match Outcome (Win/Loss)  
- Number of Features: 7  
- Problem Type: Classification  
  
**EDA Insights:**  
- Histograms showed feature distributions.  
- Scatter plots indicated relationships between runs and wickets.  
- Correlation matrix identified a strong correlation between runs and match outcomes.  
- Missing values were handled using median imputation.  
- Outliers were detected using boxplots and removed.  
- Important features identified: Runs, Overs, and Wickets.

# 2. Preprocessing Steps

**Data Preprocessing Techniques:**  
**- Handling Missing Values:** Numerical features imputed with median, categorical with mode.   
**- Encoding Categorical Features:** One-Hot Encoding for categorical variables.   
**- Feature Scaling:** Standardization using `Standard Scaler`.   
**- Feature Engineering:** Added a `run\_rate` feature (runs/overs).   
  
A complete preprocessing pipeline using `ColumnTransformer` and `Pipeline` was implemented for consistency.

# 3. Model Selection & Training

**Model Training:**   
- Data Splitting: Training (80%) and Test (20%) sets.   
- Models Trained: Decision Tree, Random Forest, Linear Regression, and Gradient Boosting.   
- Evaluation Metrics: Accuracy, Precision, Recall, F1-score (for classification).   
- Cross-validation (`cross\_val\_score`) was used to ensure model reliability.   
  
| Model | Accuracy | Precision | Recall | F1-score |  
|---------------------|----------|----------|--------|----------|  
| Decision Tree | 82.1% | 81.5% | 79.8% | 80.6% |  
| Random Forest | 88.7% | 87.9% | 86.4% | 87.1% |  
| Gradient Boosting | 90.3% | 89.5% | 88.7% | 89.1% |  
  
**Best Model:** Gradient Boosting.

# 4. Fine-Tuning Process (Hyperparameter Tuning)

**Hyperparameter Tuning Method:** `GridSearchCV` for Gradient Boosting.   
**Parameters Tuned:**  
- `n\_estimators`: [100, 200, 300]   
- `learning\_rate`: [0.01, 0.1, 0.2]   
- `max\_depth`: [3, 5, 7]   
  
**Best Hyperparameters:**  
- `n\_estimators`: 200   
- `learning\_rate`: 0.1   
- `max\_depth`: 5   
  
**Improvement:** Accuracy improved from 90.3% to 92.1% after tuning.

# 5. Final Conclusions and Best Model

**- Best Model:** Gradient Boosting (with hyperparameter tuning).   
**- Final Evaluation on Test Set:**  
 - Accuracy: 92.1%   
 - Precision: 91.8%   
 - Recall: 90.7%   
 - F1-score: 91.2%   
  
**- Insights:**  
 - Gradient Boosting performed the best among all models.   
 - Hyperparameter tuning significantly improved performance.   
 - Proper EDA, preprocessing, and feature engineering contributed to success.   
  
This project provided a complete understanding of the ML pipeline from EDA to model evaluation.