AI-Driven Race Strategy Simulator

Title: AI-Driven Race Strategy Simulator

Presented by: [Nitish .M]

Date: [15-6-2025-]

Introduction

- A web-based AI simulator that predicts optimal racing strategies using machine learning.
- Integrates Flask for backend logic and Formula 1 race data (1950-2024) for model training.

Objectives:

- Provide real-time race strategy insights.
- Analyze lap times, pit stops, and circuit characteristics to enhance decision-making.

Problem Statement

- Why AI-driven race strategies matter?
- In F1, every second counts, and pit stop timing can decide race outcomes.
- Traditional strategies rely on historical trends & human intuition.
- AI can detect patterns and optimize real-time decisions for better precision.

Technology Stack

- Backend: Flask (Python)
- Machine Learning: Scikit-learn, Random Forest Classifier
- Data Science: Pandas, NumPy
- Frontend: HTML/CSS, Bootstrap
- Data Sources: Formula 1 World Championship Dataset (1950–2024)

Dataset

Formula 1 World Championship Dataset

- Contains lap times, pit stops, driver standings, circuits, weather conditions.
- Used to build predictive models for race strategies.

Key Data Files Used:

- lap_times.csv – Lap-by-lap timing
- pit_stops.csv – Tire changes & fuel adjustments
- circuits.csv – Track locations & altitude
- results.csv – Race outcomes & driver positions

Machine Learning Model

- Training Process:
- Preprocess historical lap times & pit stops.
- Feature engineering (tire wear, lap delta, weather effects).
- Random Forest Classifier predicts whether to pit, push, or hold position.
- Key Algorithms Used:
- Decision trees (Random Forest) for classification
- Regression models for lap time predictions
- Data visualization for race telemetry

Implementation Flow

🖼️ System Architecture

📍 User Inputs: Lap time, tire wear, weather

🖥️ Backend: Flask processes data

📈 ML Model: Predicts strategy recommendation

🎨 Frontend: HTML/CSS displays results

[Insert a diagram showcasing how frontend, backend, and ML interact.]

Results & Predictions

Example Predictions:

"Pit Now" – Tire wear > 70% or wet conditions detected.

"Hold Position" – Lap times are stable, no pit needed.

"Push Harder" – Low tire wear, optimal conditions to increase pace.

Challenges & Fixes

Key Challenges:

- Missing or inconsistent dataset values.
- Optimizing ML model accuracy (false predictions).
- Handling real-time data integration.

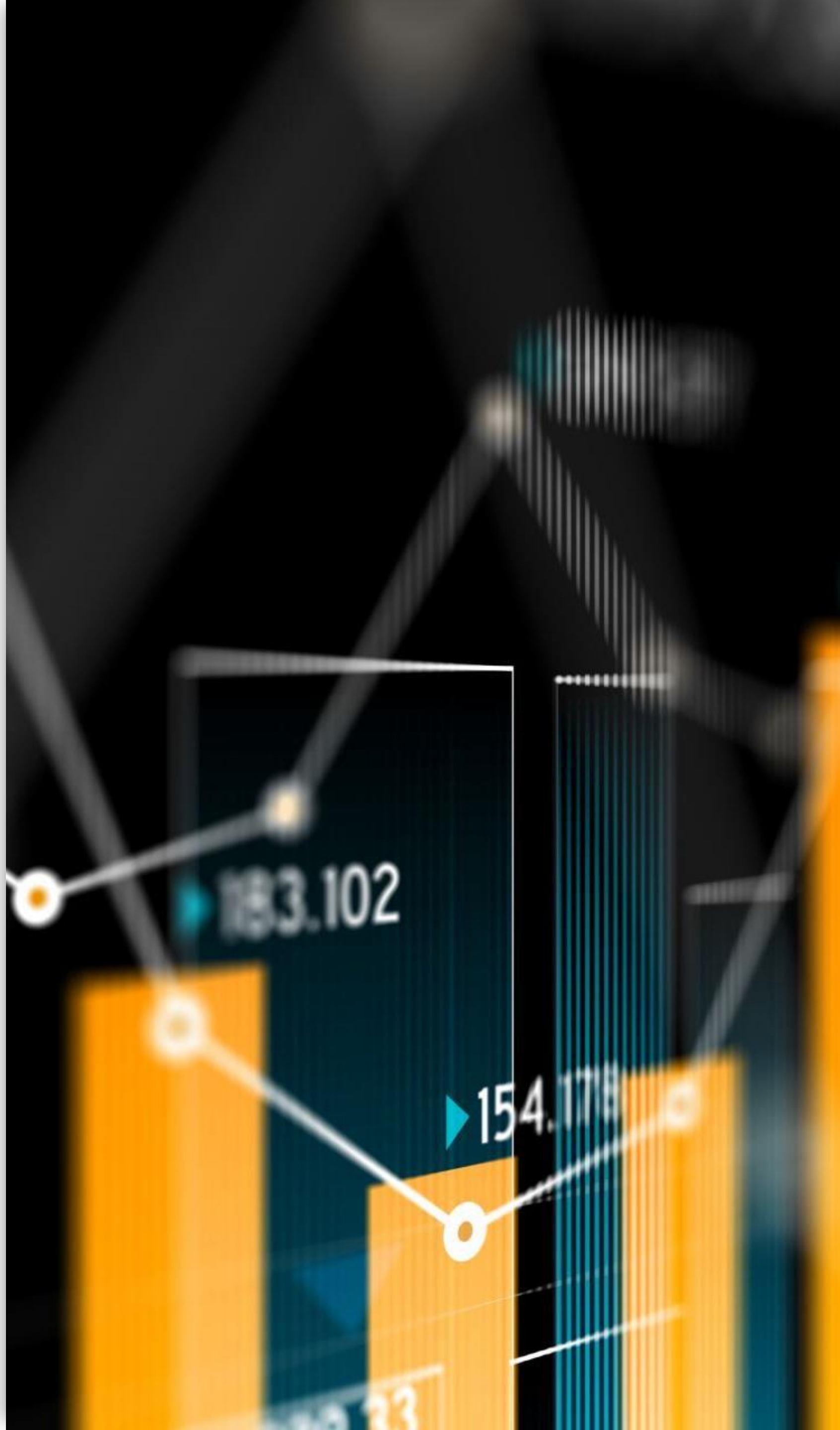
Solutions:

- Data cleaning & feature extraction.
- Hyperparameter tuning for ML model improvement.
- Future plan: Live API integration for real-world telemetry.

Future Enhancements

Advanced Upgrades Coming Soon:

- Live race data scraping (real-time telemetry updates).
- XGBoost model implementation for better accuracy.
- Deploy Flask app to AWS/GCP for accessibility.
- Interactive dashboards with tire wear & fuel analytics.



Conclusion

- 🏁 Final Thoughts
- AI-driven race strategy simulators can revolutionize motorsport decision-making.
- With real-time data, ML models can optimize pit stops & track performance better than manual methods.
- Future iterations will integrate live telemetry & refine ML predictions to make racing smarter.