Formula 1 Tire Strategy Prediction Report

April 26, 2024

1 Introduction

In the high-stakes world of Formula One racing, strategic decision-making is the key to success on the track. With modern technology and a fiercely competitive environment, Formula One demands rigorous planning and quick flexibility to ever-changing race variables. Tyre selection is an important part of racing strategy because it has a significant impact on a driver's performance and, ultimately, the race outcome.

This machine learning project goes deep into Formula One, with the goal of forecasting optimal tyre strategy using extensive historical race data and driver tactics. The project's primary goal is to create a predictive model designed for Formula One races by analysing a large dataset that includes race details, circuit characteristics, weather variables, historical lap timings, stint durations, pit stop times, and past tyre selections. The ultimate goal is to provide teams and drivers with data-driven insights so that they can make well-informed decisions on tyre selection, stint durations, and overall race strategy by leveraging machine learning algorithms.

In Formula One, the importance of tyre selection cannot be highlighted, since it influences race outcomes and strategic actions. Tyre compound variations influence crucial variables such as grip levels, cornering speeds, lap timings, and pit stop frequencies, all of which have a direct impact on a driver's performance and race pace. Effective tyre strategies are critical in race planning, as teams must handle tyre degradation, optimise pit stop timings, and control stint lengths to achieve peak performance while adhering to race regulations. Mastering tyre selection tactics reveals strategic benefits such as streamlined pit stops, quick responses to track fluctuations, and fine-tuned race pacing, all of which contribute to a competitive advantage and victories on the Formula One stage.

This project uses statistics from 2018 to 2023, with Pirelli serving as the sole tyre supplier over this time. The ultimate goal is to develop a robust and precise prediction algorithm capable of providing critical insights into optimal tyre selections under a variety of race scenarios. This project aims to be an enabler in increasing race strategy optimisation inside the fast-paced and fiercely competitive world of Formula One racing by combining industry expertise with powerful machine learning approaches.

2 Data Collection and Preprocessing

Data Collection For this project, we've drawn on a wealth of data sources to support our analytical and modelling efforts. The primary data source is the fastf1 Python package, which gives us direct access to an extensive amount of Formula 1 telemetry and race-related data via the official Formula 1 API. This real-time data contains a wide range of information, such as lap timings, pit stop details, tyre strategies, weather conditions, driver performance indicators, and

much more. To supplement this real-time data stream, I used historical data from the large Kaggle dataset "Formula 1 World Championship 1950-2020." This Kaggle dataset offers a deep dive into the historical structure of Formula One, spanning numerous decades and including a wide range of characteristics such as race specifics, driver insights, team dynamics, circuit particulars, standings, and comprehensive race results. Furthermore, to enrich our dataset with nuanced circuit-specific attributes, I explored the Formula One Circuits Wikipedia page. This source provided us with vital information about Formula 1 circuits, such as their names, locations, unique track characteristics, and an abundance of historical race history. This thorough data collection has provided us with a strong dataset suitable for in-depth analysis, feature engineering, and the creation of complex machine learning models.

In the data repository, the raw data folder contains essential CSV files drawn from the Kaggle dataset, spanning the years 1950 to 2023. These files, which include `circuits.csv`, `drivers.csv`, `lap_times.csv`, `pit_stops.csv`, and `races.csv`, are the basis of the historical data analysis and modelling efforts.

The `in_progress_data` folder contains the results of the continuous data processing and refinement. Cache folders from the fastf1 Python package are included here, along with carefully selected data collected from Wikipedia and the fastf1 module. Information extracted from Wikipedia, such as `race_circuits.csv` detailing race-specific circuits, and data on weather conditions (`weather.csv`) and tyre specifics (`tire.csv`) are pivotal in enhancing the dataset's depth and granularity.

The `final_data` folder, contains the meticulously selected and refined dataset prepared for advanced feature engineering and the development of effective machine learning models. The data collection process is meticulously executed through the `data_collection.ipynb` file.

Data Preprocessing Thorough preprocessing techniques were used to guarantee the collected data's accuracy, consistency, and suitability for further analysis and modelling.

Certain datasets were selected with the intention of highlighting relevant periods and aspects of Formula 1 racing. `final_pitstops.csv` and `final_laptimes.csv`, for instance, only provide data from 2018 to 2023, guaranteeing that our studies appropriately capture current racing patterns.

In order to enhance our understanding of circuit dynamics, final_circuits.csv combined information from `circuits.csv` and `race_circuits.csv`, offering a thorough view of circuit-specific features that are essential for the simulation.

A single dataset was created by integrating and merging other datasets, such as `races.csv`, `tire.csv`, `weather.csv`, `final_circuits.csv`, `drivers.csv`, `final_laptimes.csv`, and `final_pitstops.csv`. This aggregated dataset is the basis for in-depth examination, feature engineering, and the development of robust machine learning models specifically designed to optimise Formula 1 tyre strategies.

The meticulous preprocessing tasks, encompassing data integration, cleaning, and transformation, were systematically executed within the `data_cleaning.ipynb` file. These preparatory measures ensure that the dataset is prepped for advanced analytical techniques and model development, laying the foundation for perceptive findings and useful forecasts about race strategy..

3 Machine Learning Evaluation and Evaluation Metrics

In this project we start with a model evaluation before proceeding with feature engineering which is done in the `model_selection_training.ipynb` file. Below are the reasons why I chose to do it this way:

1. **Efficiency and Focus :** Conducting model evaluation upfront allows to quickly assess the performance of various machine learning algorithms with minimal preprocessing. This approach helps us focus on identifying the most promising models early in the project, saving time and computational resources.
2. **Baseline Performance Benchmark :** By evaluating multiple models initially, we establish a baseline performance benchmark using metrics such as accuracy, F1 score, or other relevant metrics. This baseline serves as a reference point for evaluating the impact of subsequent feature engineering efforts and model enhancements.
3. **Model Selection Guidance :** The process of model evaluation guides us in selecting the model that demonstrate the best overall performance on our dataset. Choosing the model with the highest F1 score, for example, provides a starting point for further optimization and feature engineering.
4. **Iterative Approach :** Machine learning projects often follow an iterative approach where initial evaluations inform subsequent steps. Starting with model evaluation allows us to iterate and refine our modeling strategy based on insights gained during the evaluation phase.
5. **Resource Optimization :** It enables us to prioritize feature engineering efforts on models that show the most promise from the outset.
6. **Domain Adaptability :** Understanding the domain and data characteristics is crucial. Model evaluation helps to gain insights into how different algorithms perform on our specific dataset, guiding our feature engineering strategies to address data nuances effectively.

By prioritizing model evaluation before feature engineering, I established a strong foundation for this machine learning project, enabling me to make informed decisions and optimize our modeling approach for better performance and predictive accuracy.

I conducted a thorough evaluation of machine learning models to predict tire compounds in Formula 1 races based on historical race data obtained from the `final_data.csv` file. I employed four different classifiers: Random Forest Classifier, Gradient Boosting Classifier, Support Vector Machine Classifier (SVM) with a radial basis function (RBF) kernel, K-Nearest Neighbors Classifier (KNN), and Multi-Layer Perceptron Classifier (MLP). The evaluation focused on assessing each model's accuracy, precision, recall, and F1-score across various tire compound categories.

Random Forest Classifier Evaluation The Random Forest Classifier, trained with 100 estimators, achieved an accuracy of approximately 73.57% on the test data. This model exhibited a decent level of performance across multiple tire compound categories. Notably, compounds like HARD, INTERMEDIATE, and WET showed relatively high precision and recall scores, indicating that the model effectively distinguished these compounds. However, some classes, such as ULTRASOFT and SUPERSOFT, had lower precision and recall scores, suggesting potential challenges in accurately predicting these compounds. The model correctly predicted ULTRASOFT as the tire compound for the new data input, showcasing its predictive capabilities.

Classification Report for Random Forest Classifier:

- Accuracy: 73.57%
- Precision: The model demonstrated good precision for HARD (77%), INTERMEDIATE (91%), and SOFT (70%) compounds.
- Recall: The recall rates were particularly high for INTERMEDIATE (97%) and HARD and HYPERSOFT (80%) compounds.
- F1-Score: The F1-scores for most compounds were in the range of 0.58 to 0.94.

Gradient Boosting Classifier Evaluation The Gradient Boosting Classifier, also trained with 100 estimators, achieved an accuracy of approximately 73.65% on the test data. This model exhibited performance comparable to the Random Forest Classifier. It maintained consistent precision and recall scores across different tire compound categories, with a balanced performance overall. Notably, the model correctly predicted ULTRASOFT as the tire compound for the new data input, showcasing its predictive capabilities.

Classification Report for Gradient Boosting Classifier:

- Accuracy: 73.65%
- Precision: The model showed precision scores across various compounds, ranging from 66% to 93%.
- Recall: Most compounds had recall rates above 60%, with INTERMEDIATE compounds showing higher recall rates.
- F1-Score: The F1-scores were balanced across different compounds, indicating a stable performance.

Support Vector Machine Classifier (SVC) Evaluation In contrast, the Support Vector Machine Classifier (SVM) with an RBF kernel exhibited lower performance, with an accuracy of approximately 33.91% on the test data. This model struggled to accurately predict tire compounds, for most classes except the MEDIUM which also had a loe precision, recall and f-1 score. The classification report highlighted poor precision and recall scores, indicating challenges in distinguishing between different tire compounds.

Classification Report for Support Vector Machine Classifier (SVM):

- Accuracy: 33.91%
- Precision: The precision scores were all 0.0, except for MEDIUM compounds which had a precision score of 34%.
- Recall: The recall rates were also low, indicating difficulties in correctly identifying certain tire compounds.
- F1-Score: The F1-scores for most compounds were considerably lower than the other classifiers, reflecting the model's overall poor performance.

K-Nearest Neighbors Classifier (KNN) Evaluation The K-Nearest Neighbors Classifier with 5 neighbors achieved an accuracy of 32.43% on the test data. This model exhibited lower performance compared to other classifiers, with relatively low precision and recall scores across various tire compound categories.

Classification Report for K-Nearest Neighbors Classifier (KNN):

- Accuracy: 32.43%

- Precision: The precision scores were generally low for most compounds, indicating challenges in correctly predicting tire compounds.
- Recall: The recall rates were also low, reflecting difficulties in identifying certain tire compounds.
- F1-Score: The F1-scores were considerably lower than other classifiers, indicating the model's poor performance overall.

Multi-Layer Perceptron (MLP) Classifier Evaluation The Multi-Layer Perceptron Classifier, configured with hidden layer sizes of (100, 50) and 500 iterations, achieved an accuracy of 23.57% on the test data. This model exhibited poor performance compared to other classifiers, with low precision and recall scores across various tire compound categories.

Classification Report for Multi-Layer Perceptron (MLP) Classifier:

- Accuracy: 23.57%
- Precision: The precision scores were generally low for most compounds, indicating challenges in correctly predicting tire compounds.
- Recall: The recall rates were also low, reflecting difficulties in identifying certain tire compounds.
- F1-Score: The F1-scores were considerably lower than other classifiers, indicating the model's poor performance overall.

Decision Tree Classifier Evaluation The Decision Tree Classifier, trained with default parameters, achieved an accuracy of 32.43% on the test data. This model showed competitive performance in predicting tire compounds, with notable precision and recall scores for several compounds. However, it struggled with certain classes like ULTRASOFT, which had lower precision and recall scores. The model correctly predicted ULTRASOFT as the tire compound for the new data input, showcasing its predictive capabilities.

Classification Report for Decision Tree Classifier:

- Accuracy: 68.43%
- Precision: The model demonstrated good precision for compounds like HARD (74%), INTERMEDIATE (91%), and WET (91%).
- Recall: Most compounds had recall rates above 50%, with INTERMEDIATE and HYPERSOFT compounds showing higher recall rates.
- F1-Score: The F1-scores were relatively balanced across different compounds, indicating a decent overall performance.

3.0.1 Conclusion

Overall, the Random Forest Classifier and Gradient Boosting Classifier demonstrated competitive performance in predicting tire compounds based on historical race data. These models showed strengths in distinguishing between different tire compounds, with some classes requiring further fine-tuning for improved accuracy. The Decision Tree Classifier also showed promising performance, especially in distinguishing between compounds like HARD, INTERMEDIATE, and SOFT. In contrast, the Support Vector Machine Classifier (SVM) with an RBF kernel, the Multi-Layer Perceptron (MLP) Classifier and the K Nearest Neighbours struggled to achieve satisfactory performance, highlighting the importance of selecting appropriate algorithms for specific prediction tasks.

Further optimization and feature engineering may be necessary to enhance the models' predictive capabilities, particularly for classes with lower precision and recall scores.

4 Model Trainng Predictive Analysis

I conducted model training using the Gradient Boosting Classifier and utilized the trained model to predict tire compounds for specific drivers and circuits based on historical race data. The race data was loaded from the `final_data.csv` file into a pandas dataframe. I split the dataset into features (X) and target variables (y_compound and y_stint), where 'compound' represents the tire compounds to be predicted, and 'stint length' represents the stint lengths.

After performing one-hot encoding on categorical variables in X_encoded and standardizing the features using StandardScaler, I defined and trained the Gradient Boosting Classifier model (gb_model_compound) using the standardized training data (X_train_scaled_compound, y_train_compound).

The evaluation of the model's performance using an example test dataset revealed an accuracy of approximately 72.17%. This accuracy indicates the model's overall ability to correctly predict tire compounds. Precision, recall, and F1-score metrics were calculated for each tire compound category, showing varying performance across different categories. Compounds like INTERMEDIATE and WET exhibited high precision and recall scores, indicating good predictive performance. However, compounds like ULTRASOFT and SUPERSOFT showed lower precision and recall, suggesting challenges in accurately predicting these compounds.

Additionally, I implemented a prediction function (predict_tires_and_stints) that takes driver ID, circuit ID, and the number of stints as inputs. This function filters the dataset for the specific driver and circuit, sorts the data by stint order, and selects data for the specified number of stints. It then preprocesses the input data (including one-hot encoding and standardization) and makes predictions using the trained Gradient Boosting Classifier.

For example, when calling the predict_tires_and_stints function with driver ID 4, circuit ID 1, and 3 stints, it predicted the tire compounds ['ULTRASOFT', 'HARD', 'MEDIUM'] for the first 3 stints, with corresponding stint lengths [25, 39, 8] laps.

The prediction function enables real-time predictions of tire compounds for specific drivers and circuits, providing valuable insights for race strategy planning in Formula 1.

The choice of the Gradient Boosting Classifier aligns with the project's goal of accurately predicting tire compounds based on historical race data, and its accuracy was satisfactory during model evaluation.

`predictive_analysis.ipynb` demonstrates the process of training a Gradient Boosting Classifier model, evaluating its performance, and implementing a predictive function for tire compound predictions in Formula 1 race strategy optimization.

5 Feature Engineering

Feature engineering played a crucial role in enhancing the predictive power of machine learning models for predicting tire strategies in Formula 1 races. Various steps were undertaken to extract meaningful insights and create new features that capture important aspects of race dynamics.

1. **Polynomial Features:** The dataset was enriched with polynomial features derived from 'stint length' and 'avg_lap_time'. This transformation allowed the models to capture non-linear relationships, especially in how tire wear and lap times vary throughout a race. By incorporating polynomial terms, the model gained the ability to understand more complex patterns inherent in the data.

2. **One-Hot Encoding:** Categorical variables such as 'raceid', 'round', 'circuitid', 'driverid', and 'compound' were one-hot encoded to transform them into a suitable format for machine learning algorithms. This encoding strategy ensured that categorical distinctions were properly represented as binary features, enabling the models to utilize this information effectively during training and prediction.

3. **Splitting Data:** The dataset was split into training and testing sets using a 80-20 ratio. The training set was used to train the machine learning models, while the testing set was kept aside for evaluating the models' performance on unseen data.

4. **Standardization:** Numerical features underwent standardization using the StandardScaler to ensure all features were on a consistent scale. This step is critical for algorithms sensitive to feature magnitudes, preventing biases that may arise from features with larger scales dominating the model's learning process.

5. **Gradient Boosting Models:** Two types of Gradient Boosting models were trained on the engineered features. A Gradient Boosting Classifier was trained to predict tire compounds based on the engineered features and other relevant variables. Simultaneously, a Gradient Boosting Regressor was trained to predict stint lengths, providing insights into optimal tire usage strategies during races.

6. **Model Evaluation:** The trained models were evaluated using the testing data to assess their performance. Metrics such as accuracy for the classifier and appropriate regression metrics for the regressor were computed and analyzed to gauge how well the models generalized to new data.

7. **Predictive Functionality:** A predictive function was developed to demonstrate the models' utility in making predictions based on specific race scenarios. Given driver ID, circuit ID, and the number of stints, the function predicts the likely tire compounds and stint lengths, providing actionable insights for race strategy planning.

This feature engineering is done in the `feature_engineering.ipynb` file.

6 Results and Analysis

The project involved an extensive evaluation of machine learning models for predicting tire compounds in Formula 1 races, along with feature engineering to enhance predictive capabilities. The evaluation focused on several classifiers, including Random Forest, Gradient Boosting, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP), and Decision Tree classifiers.

1. **Random Forest and Gradient Boosting Classifiers:** Both classifiers demonstrated competitive performance, with accuracies around 73.57% and 73.65%, respectively. They showed strengths in distinguishing between different tire compounds, particularly for compounds like HARD, INTERMEDIATE, and WET. However, challenges were observed in accurately pre-

dicting compounds like ULTRASOFT and SUPERSOFT, indicating areas for improvement through further fine-tuning.

2. **Support Vector Machine (SVM):** The SVM with an RBF kernel exhibited lower performance, with an accuracy of approximately 33.91%. It struggled to accurately predict tire compounds for most classes, highlighting difficulties in distinguishing between different compounds.

3. **K-Nearest Neighbors (KNN) and Multi-Layer Perceptron (MLP) Classifiers:** Both models showed lower performance compared to Random Forest and Gradient Boosting classifiers, with accuracies around 32.43% and 23.57%, respectively. They faced challenges in accurately predicting tire compounds across various categories.

4. **Decision Tree Classifier:** The Decision Tree Classifier showed competitive performance with an accuracy of 68.43%. It exhibited good precision and recall scores for compounds like HARD, INTERMEDIATE, and WET, but struggled with certain classes like ULTRASOFT.

Overall, the Random Forest and Gradient Boosting classifiers emerged as top performers, showcasing strengths in distinguishing between tire compounds. The Decision Tree Classifier also showed promise but required refinement for certain compounds. The SVM, KNN, and MLP classifiers faced challenges and may benefit from further optimization and feature engineering.

The feature engineering process, including polynomial features, one-hot encoding, data splitting, standardization, and training Gradient Boosting models, significantly enhanced the models' predictive capabilities. The predictive function developed based on the Gradient Boosting Classifier successfully provided real-time predictions of tire compounds for specific drivers and circuits, offering valuable insights for race strategy planning in Formula 1.

7 Conclusion

In conclusion, this project delved into the intricate world of Formula 1 tire strategy prediction using machine learning techniques and extensive feature engineering. By leveraging historical race data encompassing key factors like circuit characteristics, weather conditions, and driver performance metrics, we aimed to develop a robust predictive model to assist teams and drivers in making informed tire selection decisions.

Through rigorous data collection and preprocessing, we ensured the quality and relevance of our dataset, integrating real-time telemetry data with historical insights. The evaluation of multiple machine learning classifiers revealed that Random Forest and Gradient Boosting classifiers exhibited competitive performance, showcasing strengths in distinguishing between different tire compounds. The Decision Tree Classifier also showed promise but required further refinement for certain compounds.

Feature engineering played a pivotal role in enhancing our models' predictive power. Techniques such as polynomial feature generation and one-hot encoding enabled us to capture complex relationships and categorical distinctions effectively. The developed predictive function demonstrated the utility of our models in real-time tire compound predictions for specific race scenarios, providing actionable insights for race strategy planning.

To further develop this machine learning model, several avenues can be explored:

1. **Incorporate Real-Time Data:** Integrate real-time data feeds during races to update the model with current track conditions, driver performance, and tire wear. This can improve the model's accuracy by accounting for dynamic race situations.
2. **Driver and Team Strategies:** Include historical data on driver and team strategies to understand how different approaches impact tire performance. Analyze trends and patterns to provide strategic recommendations tailored to specific drivers and teams.
3. **Weather Prediction:** Integrate weather forecasting data to predict how weather conditions will affect tire performance. Develop algorithms that adjust tire strategy predictions based on anticipated weather changes during races.
4. **Track-Specific Insights:** Analyze track characteristics such as surface grip, tire degradation rates, and cornering demands. Incorporate this information into the model to generate track-specific tire strategy recommendations for optimal performance.
5. **Deep Learning Techniques:** Explore deep learning algorithms such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs) to capture complex temporal and spatial relationships in race data. These models can enhance predictive capabilities and adapt to varying race scenarios.
6. **Interactive Visualization:** Develop interactive visualization tools to present model predictions, insights, and recommended tire strategies in an intuitive and user-friendly manner. Incorporate dashboards that allow teams and drivers to explore different scenarios and make informed decisions.
7. **Continuous Model Updating:** Establish a framework for continuous model updating and retraining based on new data and feedback loops from race outcomes. Implement mechanisms for model monitoring, validation, and version control to ensure ongoing reliability and relevance.

By continuing to iterate, refine, and innovate in these areas, we can further elevate the predictive capabilities of this machine learning model for Formula 1 tire strategy prediction, contributing to more strategic and competitive racing outcomes in the dynamic world of Formula 1.

8 References

Here are the references used for this project:

1. FastF1 Python Module: https://fastf1.io/
2. Kaggle Dataset - Formula 1 World Championship 1950-2020: https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020?resource=download&select=constructor_standings.csv
3. Wikipedia - List of Formula One Circuits: https://en.wikipedia.org/wiki/List_of_Formula_One_circuits
4. Official Formula 1 Website: https://www.formula1.com/
5. Pirelli Motorsport - Formula 1 Tyres: https://www.pirelli.com/tyres/en-ww/motorsport/f1/tyres

These references were instrumental in obtaining data, conducting analysis, and gaining insights for the project on Formula 1 race strategy prediction.