Formula One Qualifying Lap Time Prediction Using Weather Predictors

Cole L. Wagner

Southern Methodist University

Table of Contents

[Abstract 3](#_Toc184995804)

[Formula One Qualifying Lap Time Prediction Using Weather Predictors 4](#_Toc184995805)

[Methods 5](#_Toc184995806)

[Data Description 5](#_Toc184995807)

[Data Preprocessing 6](#_Toc184995808)

[Modeling Procedure 7](#_Toc184995809)

[Shiny App 8](#_Toc184995810)

[Results 10](#_Toc184995811)

[Model Comparison 10](#_Toc184995812)

[Shiny App 11](#_Toc184995813)

[Discussion 12](#_Toc184995814)

[References 14](#_Toc184995815)

[Tables 15](#_Toc184995816)

[Figures 17](#_Toc184995817)

Abstract

The overall purpose of my project was to investigate the effect of different weather attributes on lap times in Formula One qualifying sessions. I sought to model the effect of weather conditions on lap times when accounting for the differences in the tracks themselves. This model could be used by teams to generate an expected lap time based on the weather and assess whether their driver outperforms or underperforms this expectation.

To achieve this, I pulled data on qualifying laps from the 2023 and 2024 seasons (via the OpenF1 API). After wrangling, I was left with a final dataset of 3,967 rows and 11 columns. Each row represented one lap, and columns consisted of track attributes (track name, length, and number of turns), weather attributes (air temperature, track temperature, pressure, humidity, etc), and the outcome (lap duration).

I then applied four modeling paradigms to this data (linear regression, Bayesian linear regression with a random intercept for track, random forest, and XGBoost), and found that the random forest model was best, achieving an RMSE of 0.5 seconds and explaining 99.8% of the variance in the outcome.

While track length and number of turns were the most important variables, I found that humidity, air and track temperature, and air pressure had predictive value. Faster lap times were associated with nonextreme humidity, low pressure, high air temperature, and moderately low track temperature. The most important weather factor was humidity, as there was a huge jump in lap times when the humidity was extremely high (which could be because high humidity often comes with rain). Because of this model’s extremely high predictive accuracy, it is a useful tool for teams to evaluate how their drivers handle different weather conditions.

Formula One Qualifying Lap Time Prediction Using Weather Predictors

If any other sport can be described as a “game of inches”, then Formula One should certainly be called a “game of millimeters” (and not because of its European heritage). In qualifying sessions to determine the grid order for the race, the difference between first and second is often only tenths or even hundredths of a second. As a result, many teams have employed advanced statistical techniques to squeeze every last drop of performance out of their cars and drivers. However, very little effort has been devoted to employing these same techniques to model factors outside of the teams’ control, namely the weather. There are many vague notions of how weather conditions impact a driver’s performance. For example, it is well known that extreme temperatures and precipitation causes the track to be slicker, forcing drivers to be conservative to prevent crashes (Burgess II, 2023). However, no comprehensive data analysis has been done to model the impact of weather conditions on qualifying lap times.

Prior research has primarily been focused on race outcomes or lap times within a race. For example, a published report in the MathSport International 2017 Conference sought to predict Formula One race lap times using generalized additive mixed models (GAMMs) (Casella and Vidoni, 2017). The researchers considered predictors such as lap number, tire compound and wear, team, and driver. The primary use case for this model presented by the researchers was the simulation of the remainder of a race to determine the optimal lap for a pit stop. Another article published in the Journal of Business and Economics Research explored the role of driver experience on finishing place in NASCAR races (Allender, 2008). The researcher created several linear regression models using predictors such as years of driver experience and several different interaction terms with this main effect. Neither of these articles explored the effects of weather conditions whatsoever.

Given this research gap, I performed an analysis of the effect that weather plays on qualifying lap times in Formula One. I acquired lap data for the 2023 and 2024 seasons from the OpenF1 API and Wikipedia and built ordinary least squares (OLS) linear regression, Bayesian linear regression, random forest, and XGBoost models to predict the duration of a lap using weather predictors. I then created a Shiny app to allow users to enter weather data and get real-time predictions. The app also includes other exploratory summary tables to give users a quick analysis of driver performance, weather conditions at each of the grand prix in the data, and laps that outperformed the model by the greatest amount of time. This final product could be used by Formula One teams to assess how well their drivers perform in various weather conditions.

# Methods

## Data Description

The initial data was split into five separate datasets for session, lap, weather, track, and driver data. The session, lap, weather, and driver datasets were pulled from the OpenF1 API. Each of these datasets had columns for meeting and session ID. Meeting IDs were unique to each grand prix, while session IDs were unique to each session within a meeting (practice, qualifying, race, etc.). As a result, it was possible to pull data exclusively for qualifying sessions using these ID variables. The track data, however, had to be web-scraped from a Wikipedia list of Formula One circuits. Because this Wikipedia page was not built for easy scraping, a web-based tool to extract the table information from the raw HTML code had to be used. The rest of the analysis was done in R.

The session dataset contained 49 rows (sessions) and 14 columns describing the country, session type, time, date, and unique IDs. The lap dataset contained 4,410 rows (laps) and 16 columns. These columns included the unique IDs, driver number, speed at intermittent points throughout the track, lap start time and date, total lap duration and section durations, whether the lap was an out lap from the pit, and the lap number. There were many more qualifying laps in the API, because lap data was recorded for all laps regardless of whether they were a “hot” lap (a lap where the driver was attempting to set a fast time). To filter out this noise, only the 90 fastest laps from each session were pulled. This number was settled on because only 45 laps are “counted” in a qualifying session, and drivers typically take no more than two attempts to set their fastest lap during a given round (Thorns, 2023). Thus, this was a conservative approach to retain as many “hot” laps as possible.

The weather dataset contained 4,051 rows and 10 columns. Each row represented the weather conditions in each minute during a qualifying session. Columns included air and track temperature, relative humidity, pressure, whether it was raining, the speed and direction of the wind, and the unique IDs. The track dataset contained 77 rows (tracks) and 11 columns describing the length of the track, number of turns, years used, type (street vs. race circuit), direction, location, and grand prix that used each track. Finally, the driver dataset contained 1,059 rows and 12 columns. Each row represented one driver participating in one session. The columns describe each driver’s name, number, team, country of origin, and unique IDs.

## Data Preprocessing

To begin preprocessing, the session and track datasets were prepared for joining, with the goal being to match each session with a track. The track dataset was filtered to only include tracks used in the 2023 and 2024 seasons and some track naming inconsistencies were corrected. The two datasets were then joined, duplicate and unnecessary columns were removed, and a numeric track length column was created.

The lap and weather dataset were then prepared for joining by rounding all date variables to the nearest minute. These two datasets were joined by date (meaning date and time to the minute) to ensure that each lap was merged with the proper weather conditions. Duplicate columns were then removed. The driver dataset was joined to the lap and weather data by session and meeting ID to complete this joined dataset. The result of these joins were two datasets, one with lap, weather, and driver data, and another with session and track data. These two datasets were then joined so that each lap was matched with a session and track and superfluous columns were removed. All missing data was then inspected, and it was found that 12 laps (rows) had no weather data, so these were removed. There were only 19 other missing values, so these were imputed using KNN imputation.

To further filter out noise in the data, the “107% rule” was applied. The 107% rule states that if a lap is not within 107% of the fastest lap in the session, it is too slow to be considered for race standing. 431 laps were removed when this rule was applied. To capture the intensity of rain during a given lap, the cumulative number of raining laps in the session was calculated and added to the data. A priori variable selection was then performed, to leave a final modeling dataset of 3,967 rows and 11 columns: lap duration, air temperature, track temperature, relative humidity, whether it was raining on the given lap, how many laps it had been raining during the session, wind speed, circuit name, number of turns, and the length of the track in kilometers. The final preprocessing steps before modeling were normalizing all numeric predictors and dummy encoding all categorical predictors.

## Modeling Procedure

Due to the relatively small data size, no train test split was performed. Instead, 10-fold cross-validation repeated twice and stratified by the outcome was used to estimate model performance metrics and tune the random forest and XGBoost models. Root mean square error (RMSE), mean absolute error (MAE), and R squared were the performance metrics used to compare the models. The first model fit was an OLS linear regression, and plots were created to check whether assumptions were met. This was followed by a Bayesian linear regression with random intercepts for each track. The shinystan package was used to check assumptions for this model. Neither of these models required tuning, so resampled performance metrics were easily calculated. The random forest and XGBoost models were tuned using five tuning parameter combinations generated from Latin hypercube sampling. The optimal tuning parameters were chosen to be those with the lowest RMSE value.

After comparing model performance, the random forest model was chosen for further analysis. A variable importance plot and ALE plots for the five most important predictors were created to analyze relationships between predictors and the outcome. To explore how much the weather predictors contributed to the model’s performance (as opposed to the track name, turns, and length), the random forest modeling procedure was run again with all track-related predictors removed. The resulting dataset contained only 8 columns (lap duration and all weather predictors). The final perturbation of the random forest model was to run the modeling procedure again using only weather predictors and excluding all laps from the Mexico Grand Prix. This was done due to the extreme pressure values in Mexico City.

## Shiny App

After modeling was complete, a Shiny app was created to give users the ability to input weather conditions and get a predicted lap time. The model chosen to generate predictions was the random forest model with only weather predictors and the full complement of laps. The app was divided into four main tabs. The first tab contained a sidebar with reactive inputs for each of the seven model predictors, and a predicted lap time was generated at the top of the tab. These functions were included to allow teams to input weather conditions and get real time expected lap times that they could use to assess their drivers. A variable importance plot and ALE plots for all predictors were also included on this tab to help users understand how the model makes predictions and how each variable is related to the predicted lap time.

The second tab contained a driver performance summary table. This table was created by generating predictions on the model data and attaching these predictions to the dataset. A time difference column was then engineered to describe the difference between actual time and predicted time. The data was then grouped by driver, and the total number of laps, average time difference, and percentage of laps that outperformed expectation were calculated. This table was integrated into the app via the reactable package, which turned the summary table into an interactive HTML object. This table gave users an efficient way to understand how drivers tended to perform against the model, and a way to compare drivers by quantifying how often and by how much they tended to underperform or outperform expectation.

The third tab contained a “best laps” summary table. The “best laps” were deemed those that outperformed the model by the greatest amount of time. This table was created by sorting the data by time difference and keeping only the top 20 laps. The reactable package was then used to make this table interactive and user friendly. This table added value by allowing users to examine different attributes of the best laps in the data to try to understand why these laps outperformed expectation by so much. The final tab contained a weather summary table. For this table, the data was grouped by track and year, and the average temperature, wind speed, relative humidity, pressure, and whether there had been any rain was calculated. Once again, this table was implemented into the app using the reactable package. This table gave users the ability to quickly understand a summary of the weather conditions in each of the grand prix in the data.

# Results

## Model Comparison

Table 1 shows the resampled performance metrics for each of the four modeling frameworks using track and weather predictors. There was a clear gap in performance between the linear and tree-based models. The random forest model had the lowest MAE (0.378), lowest RMSE (0.522), and highest R squared (0.998). Meanwhile, both linear models had an RMSE of 1.337 and an R squared of 0.987. Because lap duration is measured in seconds, the RMSE is the average error of predictions in seconds. Therefore, the random forest model missed the actual lap time by 0.522 seconds on average, while the linear models missed by an average of 1.337 seconds. The XGBoost model performed similarly to the random forest with an MAE of 0.402, an RMSE of 0.547, and an R Squared of 0.998. Ultimately, the random forest model was chosen for further analysis because it performed the best out of all modeling frameworks considered, and it was also less computationally expensive than the XGBoost and Bayesian linear regression models.

Table 2 shows a comparison of the three perturbations of the random forest model (all predictors, weather predictors only, weather predictors only and no Mexico City laps). The random forest with all predictors had the lowest MAE (0.378), lowest RMSE (0.522), and tied for the highest R squared (0.998). However, the random forest with only weather predictors performed extremely similarly with an MAE of 0.392, an RMSE of 0.567, and an R squared of 0.998. Additionally, the weather predictors model only included 7 predictors, while the full model contained 32. Because the weather predictors model performed so similarly to the full model, and was much more parsimonious, it was selected as the model to be fit for the Shiny app.

## Shiny App

Figure 1 depicts a screenshot of the first tab of the Shiny app. The user can input weather conditions via the boxes in the sidebar and can see the output lap time in the top card entitled “Model Prediction”. The “Feature Importance” card shows the importance of each predictor in the final model and is static. The “ALE Plot” card has a dropdown menu to select a feature, and the corresponding ALE plot is displayed. Figure 2 shows the “Driver Summary” tab. The only object in this tab is the driver performance summary table. The user can hover their cursor over a row in the table to highlight it and can use the vertical scroll bar to explore the performance of all drivers in the dataset. Figure 3 displays the “Best Laps” tab in the Shiny app, which is formatted in the same manner as the previous tab. The best laps summary table is also fitted with the same hover and scroll features as the driver performance table.

Finally, Figure 4 depicts the “Weather Summary” tab of the app. This tab shares a general format with the other two summary table tabs, but the table itself has a few key features. First, each column is searchable, meaning you can filter the table for a certain circuit, year, or weather condition value. The table also features a red to white color gradient for temperature, a horizontal bar chart for relative humidity, and a dot marker for the rain factor. These visuals add to the user experience by improving the app’s appearance and making the table easy to navigate.

# Discussion

After examining each of the modeling frameworks, I was led to believe that linear models are not appropriate for this data problem. Many of the relationships in the data are nonlinear, and the assumptions of OLS and Bayesian linear regression do not appear to be met. While I could account for nonlinearity by including splines in the model, tree-based algorithms have the advantage of automatically considering interactions in the data. As a result, I recommend future research on this topic focus on machine learning modeling approaches.

Based on the weather only random forest model (which was chosen because of its balance of performance and simplicity), Pressure, humidity, and temperature are the most important weather predictors in determining lap time. This is shown in Figure 5, where feature importance values are plotted on a horizontal bar chart. According to the ALE plot in Figure 6, higher temperatures tend to result in longer lap times, which corroborates the anecdotal notions that hotter tracks tend to be slicker and thus more difficult to maneuver.

Additionally, the final model’s performance proves that it is useful in giving teams a realistic expected lap time given different weather conditions. An average error of only 0.567 seconds without considering any driver, car, or track related attributes gives teams the confidence that the predicted lap time is an accurate depiction of how an average driver and car combination might perform in a given situation. This allows teams to assess cars and drivers using a realistic baseline in a variety of weather conditions.

The biggest limitation in this project is the lack of historical data. OpenF1 currently only has data from the 2023 and 2024 seasons, which limits the models’ abilities to examine trends over time. Less data also results in the need to perform resampling techniques instead of a train-test split, which would be preferable. I would anticipate model performance improving given more data, and nuances in the data that may be undetectable right now could be discovered. Despite this limitation, though, the models performed well and still provide useful insights. An area for future research could be an analysis of how weather impacts race results. This analysis proves that the weather has a significant association with lap time in qualifying, so it would be interesting to see how this extrapolates to a race.

References

Burgess, M. (2023, November 14). How does weather impact F1?. Red Bull. <https://www.redbull.com/us-en/how-does-weather-impact-f1>

Casella, C., & Vidoni, P. (2017). Formula 1 lap time modeling using generalized additive models. In Proceedings of MathSport International 2017 Conference (p. 87).

Allender, M. (2008). Predicting the outcome of NASCAR races: The role of driver experience. Journal of Business & Economics Research (JBER), 6(3).

Introduction. OpenF1 API | Real-time and historical Formula 1 data. (n.d.). <https://openf1.org/>

Wikimedia Foundation. (2024, December 9). List of formula One Circuits. Wikipedia. <https://en.wikipedia.org/wiki/List_of_Formula_One_circuits>

Thorns, T. (2023, November 17). What is a hot lap in F1? discover the secrets of a Hot lap. F1mix.com. https://f1mix.com/what-is-hot-lap-in-f1#:~:text=During%20each%20session%20a%20driver,can%20retake%20the%20Hot%20Lap.

Tables

Table 1

Comparison of Modeling Frameworks with Weather and Track Predictors

|  |  |  |  |
| --- | --- | --- | --- |
| Model Type | MAE | RMSE | R Squared |
| OLS Linear Regression | 0.939 | 1.337 | 0.987 |
| Bayesian Linear Regression | 0.936 | 1.337 | 0.987 |
| Random Forest | 0.378 | 0.522 | 0.998 |
| XGBoost | 0.402 | 0.547 | 0.998 |

Note: All metrics are resampled means from 10-fold cross-validation repeated twice.

Table 2

Comparison of Random Forest Models

|  |  |  |  |
| --- | --- | --- | --- |
| Model Type | MAE | RMSE | R Squared |
| RF with All Predictors | 0.378 | 0.522 | 0.998 |
| RF with Only Weather Predictors | 0.392 | 0.567 | 0.998 |
| RF with Only Weather Predictors and No Mexico City | 0.394 | 0.600 | 0.997 |

Note: All metrics are resampled means from 10-fold cross-validation repeated twice.

Figures

Figure 1

*Shiny App Predictions Tab*

A screenshot of a graph

Description automatically generated

Figure 2

Shiny App Driver Summary Tab

A screenshot of a graph

Description automatically generated

Figure 3

Shiny App Best Laps Tab

A screenshot of a graph

Description automatically generated

Figure 4

Shiny App Weather Summary Tab

A screenshot of a graph

Description automatically generated

Figure 5

Weather Only Random Forest Feature Importance Plot

A graph with a bar graph

Description automatically generated

Figure 6

Weather Only Random Forest Air Temperature ALE Plot

A graph showing the temperature

Description automatically generated

Note: Temperature is normalized, with zero being the mean, and each unit being a standard deviation.