Data Collection

AUTHOR
Nandini Kodali

Introduction

In this section, we focus on the methods and sources used to collect data that are relevant to the research questions of this project. The quality of the data collected determines the accuracy of insights and predictions. Poor data collection practices can lead to biased analysis, inaccurate results, and ineffective models. Therefore, careful planning and execution of data collection is essential for the success of any data-driven project.

Challenges in Data Collection Some important points to be considered when collecting data are:

- **Data quality**: Ensuring the data is accurate, complete, and relevant to the research questions is crucial.
- **Inconsistencies**: Data form different sources may have different formats, structures, naming conventions, and units of measurement, requiring a thorough understanding of the data for pre-processing.
- **Ethics and Privacy**: Data collection methods must adhere to ethical guidelines, ensuring that no sensitive or private information is collected or misused.
- **Data Bias**: Collecting data from sources that introduces bias can lead to inaccurate results and models.
- **Technical Constraints**: Issues such as API rate limits, website restrictions, or incomplete data can hinder data collection.

By addressing these challenges, it is ensured that the data collected was relevant, accurate, and reliable.

Methods

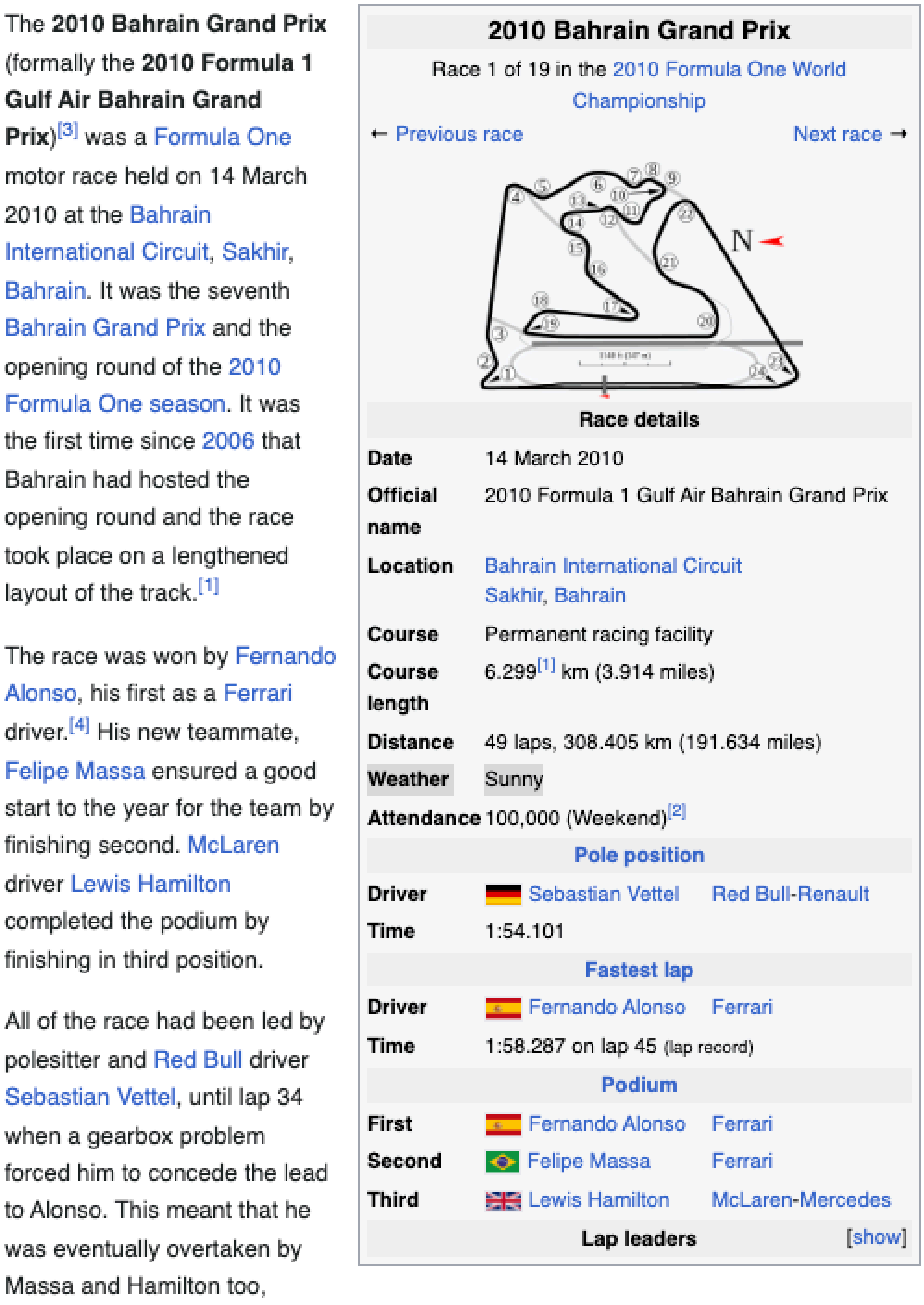
While there are various methods to collect data, `web scraping` and `APIs` are the two methods used in this project.

Web Scraping

Web Scraping is an automatic way to collect data from websites. It involves the use of automated scripts or tools to interact with the website’s structure to retrieve information. The data is extracted using selectors like tags, classes, or IDs. While web scraping can be tailored to collect a variety od data types from multiple web pages, one should be aware of the website's terms of service and ethical scraping practices, and manage rate limits to avoid being blocked.

In this Project: Using Python libraries like `requests` and `BeautifulSoup`, weather information was extracted from infoboxes of each race's Wikipedia pages.

Process:
- The URLs for each race were obtained from the race data collected, these links were used to locate the Wikipedia pages of each race.
- HTTP requests were sent to the Wikipedia pages to retrieve the HTML content.
- Parsing the HTML using `BeautifulSoup` locate the infobox containing the race metadata.
- Locating and extracting the **Weather** field from the table.



The **2010 Bahrain Grand Prix** (formally the **2010 Formula 1 Gulf Air Bahrain Grand Prix**)[3] was a Formula One motor race held on 14 March 2010 at the Bahrain International Circuit, Sakhir, Bahrain. It was the seventh Bahrain Grand Prix and the opening round of the 2010 Formula One season. It was the first time since 2006 that Bahrain had hosted the opening round and the race took place on a lengthened layout of the track.[1]

The race was won by Fernando Alonso, his first as a Ferrari driver.[4] His new teammate, Felipe Massa ensured a good start to the year for the team by finishing second. McLaren driver Lewis Hamilton completed the podium by finishing in third position.

All of the race had been led by polesitter and Red Bull driver Sebastian Vettel, until lap 34 when a gearbox problem forced him to concede the lead to Alonso. This meant that he was eventually overtaken by Massa and Hamilton too,

2010 Bahrain Grand Prix

Race 1 of 19 in the 2010 Formula One World Championship

← Previous race Next race →

Race details

Date	14 March 2010	
Official name	2010 Formula 1 Gulf Air Bahrain Grand Prix	
Location	Bahrain International Circuit Sakhir, Bahrain	
Course	Permanent racing facility	
Course length	6.299[1] km (3.914 miles)	
Distance	49 laps, 308.405 km (191.634 miles)	
Weather	Sunny	
Attendance	100,000 (Weekend)[2]	
Pole position		
Driver	Sebastian Vettel	Red Bull-Renault
Time	1:54.101	
Fastest lap		
Driver	Fernando Alonso	Ferrari
Time	1:58.287 on lap 45 (lap record)	
Podium		
First	Fernando Alonso	Ferrari
Second	Felipe Massa	Ferrari
Third	Lewis Hamilton	McLaren-Mercedes
Lap leaders		[show]

APIs

APIs (Application Programming Interfaces) are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols. APIs enable developers to access data or functionality from a system without having to know the underlying implementation details, making it easier to integrate data from multiple sources. They act as intermediaries, providing a structured way for programs to request and retrieve information or services.[2]

How do APIs Work?[3] APIs facilitate communication between applications, systems, or devices through a structured **request-response cycle**

1. **API Client**: The process begins with an API client, which sends a request to the API server, which can be triggered by user interaction or external events.
2. **API Request**: An API request typically contains the following components:
   1. Endpoint: URL that provides access to a specific resource.
   2. Method: Indicates that action to be performed on the resource.
   3. Parameters: Variables that are passed along with the request to customize the response.
   4. Headers: Key-value pairs that provide additional details about the request, such as authentication tokens or the content format.
   5. Request Body: Includes actual data required for operations like creating, updating, or deleting resources.
3. **API Server**: Receives the request and performs actions such as authenticating the client, validating the input, and processing the request by retrieving or updating the requested data.
4. **API Response**: The API server returns a response to the client, this typically includes:
   1. Status Code: A numerical code indicating the result of the request (e.g., 200 for success, 201 for resource creation, or 404 for resource not found).
   2. Headers: Additional metadata about the response.
   3. Response Body: The data requested by the client, or an error message.

API Architectural Styles:[1]

1. **REST (Representational State of Resource)**: A widely used style for data exchange over the internet. In RESTful APIs, resources are accessed through endpoints, and standard HTTP methods such as GET, POST, PUT, and DELETE are used to perform operations on these resources.
2. **SOAP (Simple Object Access Protocol)**: Protocol that uses XML to facilitate the transfer of highly structured messages between client and sever. While it provides features for security and reliability, it can be slower compared to other architectural styles.
3. **GraphQL**: An open-source query language designed to allow clients fetch only the data they need via a single API endpoint. This eliminates the

need for multiple requests, making it valuable for applications that operate over slower or less reliable network connections.

In this Project:

The Ergast Developer API was used to collect race, driver standings, circuit information, and pitstop data. “The Ergast Developer API is an experimental web service which provides a historical record of motor racing data for non-commercial purposes”[4].

Data Collection

Required Libraries

▶ Code

Race Information

Importance: Contains information about the season, round, date, grand prix name, location, results,wikipedia url of the race, driver and constructor details.

- Season: The year of the race.
- Round: The number of the race in the season.
- Grand Prix Name: The official name of the race.
- Results: Includes the finishing poistion and points earned.
- Driver Details: Provides information about the driver’s name, ID, nationality, and date of birth.
- Constructor Details: Constructor refers to the team that builds and maintains the cars. It contains information about the constructor’s name, ID, and nationality.

Source: ERGAST API

▶ Code

▶ Code

▶ Code

▶ Code

Driver Standings

After each round, drivers earn points based on their final position. These points are added to their overall tally, and the driver with the most points at the end of the season wins the World Driver’s Championship (WDC).

Importance: Contains total points earned by each driver in every season from 2000 to 2023. It is crucial for identifying trends in driver performance over the years.

Source: ERGAST API

▶ Code

Circuit Information

The race tracks are referred to as circuits.

Importance: This data includes the circuit name, locality, country as well as its longitude and latitude. These values can be used for collecting weather information on the race day.

Source: ERGAST API

▶ Code

▶ Code

News of Top 10 Drivers

Standings in 2024 season as on 11-24-2024

Silly Season in F1 refers to the period of speculation, rumors, and announcements surrounding driver lineups for the next season. This period typically begins during the latter half of the season, as drivers, teams, and sponsors negotiate deals for the future. Headlines during the silly season often speculate on whether drivers will extend their contract, switch teams, or retire from the sport, creating a buzz that fuels media interest.

Importance: Analyzing news coverage about drivers can provide insights into their career trajectories, and potential moves in the upcoming season.

Source: NEWS API

Resources: NEWS-API DSAN 5000 Lecture Content

▶ Code

▶ Code

Top 10 Drivers as of Round 22 (Las Vegas Grand Prix)

1. Max Verstappen
2. Lando Norris
3. Charles Leclerc
4. Oscar Piastri
5. Carlos Sainz
6. George Russell
7. Lewis Hamilton
8. Sergio Perez

9. Fernando Alonso
10. Nico Hulkenberg

▶ Code

▶ Code

Weather Data

Importance: Weather conditions play a crucial role in race strategy, tire choices, and driver performance.

Source: Wikipedia

▶ Code

▶ Code

▶ Code

▶ Code

	season	raceName	url
0	2010	Bahrain Grand Prix	http://en.wikipedia.org/wiki/2010_Bahrain_Gran...
24	2010	Australian Grand Prix	http://en.wikipedia.org/wiki/2010_Australian_G...
48	2010	Malaysian Grand Prix	http://en.wikipedia.org/wiki/2010_Malaysian_Gr...
72	2010	Chinese Grand Prix	http://en.wikipedia.org/wiki/2010_Chinese_Gran...
96	2010	Spanish Grand Prix	http://en.wikipedia.org/wiki/2010_Spanish_Gran...

▶ Code

Circuit Features

Around the F1 season, circuits vary widely in their features—some are known for tight, technical corners, others for long, high-speed straights, and a few for their narrow and challenging layouts. These unique characteristics influence car and driver performance significantly, with certain drivers or car designs excelling on specific track types.

Importance: Analyzing racetrack features is crucial for understanding how different teams and drivers perform under varying conditions. This information can be used to classify tracks, which can further be studied to identify patterns and trends in race results.

Source: fastf1

▶ Code

▶ Code

▶ Code

Pit stop data

Pit stop is when the car pulls in the pit lane, a designated area, for a quick maintenance, change of tires, mechanical repairs or any other actions necessary during the race. The teams have to strategically decide when to make a pit stop in order to gain a competitive advantage.

Importance: The speed and precision of pit crews play a crucial role in minimizing the time drivers lose during a pit stop.

- Lap: The specific lap during which the pit stop was made.
- Stop: Whether it is the first, second, or subsequent stop for the driver.
- Duration: Time spent in the pit lane.

Source: ERGAST API

▶ Code

▶ Code

All the datasets can be found here

References

1. DSAN 5000 lecture content, https://jfh.georgetown.domains/centralized-lecture-content/course-timelines/dsan-5000.html.

Footnotes

1. 2010 Bahrain Grand Prix ↩
2. What are APIs ↩
3. How do APIs work ↩
4. Ergast Developer API ↩