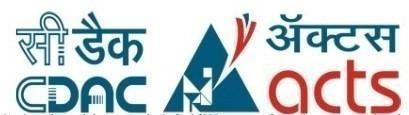
**Project Report**

**On**

**Fast Lane Forecasts:**

**Analysing and Predicting F1 Performance**

*Submitted*

*In partial fulfilment for the award of the Degree of*

# PG-Diploma in Big Data Analytics

**(C-DAC, ACTS (Pune))**

|  |  |
| --- | --- |
| **Guided By:** | **Submitted By:** |
| Mr. Sanjay Sane | Sanket Ugale (230340125042) |
|  | Shashwat Banger (230340125045) |
|  | Shubhashini Dubey (230340125048) |
|  | Tushar Patil (230340125052) |

**Centre for Development of Advanced Computing**

**(C-DAC), ACTS (Pune- 411008)**

**Acknowledgement**

We extend our heartfelt gratitude to Mr. Sanjay Sane, our esteemed Project Guide for "Fast Lane Forecasts: Analysing and Predicting F1 Performance," undertaken as a part of the PG-DBDA curriculum at CDAC ACTS Pune. His invaluable guidance and unwavering support have been instrumental throughout this project, from its inception to its completion. His expertise and insightful suggestions have significantly enriched our understanding and implementation of advanced concepts in the fields of Hadoop, PySpark, SparkSQL, Tableau, and Machine Learning.

We also wish to express our sincere appreciation to Ms. Gayatri Pandit, our dedicated Course Coordinator, for her continuous encouragement and assistance. Her commitment to fostering a conducive learning environment has been a driving force behind our academic growth.

Furthermore, we acknowledge CDAC ACTS Pune for providing us with the opportunity to undertake the project "Fast Lane Forecasts: Analysing and Predicting F1 Performance." This platform has enabled us to apply theoretical knowledge to real-world scenarios and expand our proficiency in diverse technical domains.

In this journey of knowledge and skill enhancement, we are grateful to have received support from various quarters. We would like to extend our thanks to all those who played a significant role, directly or indirectly, in our project's success. Your contributions have been invaluable.

|  |  |
| --- | --- |
|  | **Submitted By:** |
|  | Sanket Ugale (230340125042) |
|  | Shashwat Banger (230340125045) |
|  | Shubhashini Dubey (230340125048) |
|  | Tushar Patil (230340125052) |

**ABSTRACT**

"Fast Lane Forecasts: Analyzing and Predicting F1 Performance" is an academic project delving into Formula 1 (F1) racing data using advanced technologies. Employing Hadoop, PySpark, SparkSQL, Tableau, and Machine Learning, this project investigates historical F1 race data to unveil pivotal determinants of race outcomes. By efficiently processing extensive datasets with PySQL and PySpark, the project sets the groundwork for model training.

Central to the project is the deployment of machine learning models, complementing rigorous data analysis, to predict F1 race results. These models meticulously parse historical race data, spotlighting influential features that guide the predictive algorithms. To translate insights effectively, interactive visualizations are curated through Tableau, facilitating comprehension of intricate analyses and enhancing accessibility. "Fast Lane Forecasts" represents a comprehensive paradigm that reshapes data insights into a navigational tool for comprehending and anticipating F1 racing dynamics.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Title** | **Page No.** |
| **1**  **1.1**  **1.2** | **Introduction**  Introduction  Objective | **01-02**  **01**  **02** |
| **2** | **Literature Review** | **03-04** |
| **3** | **Methodology and Techniques** | **05-11** |
| **3.1**  3.1.1  3.1.2  3.1.3  3.1.4  3.1.5 | **Methodology**  Data Extraction and Storage  Data Cleaning and Transformation with PySpark:  Exploratory Data Analysis (EDA) with Python Libraries  Tableau Visualization and Dashboard Creation  Machine Learning Model Application | **05**  05  06  06  06  07 |
| **3.2** | **Dataset** | **07** |
| 3.3  3.3.1  3.3.2  3.3.3  3.3.4  3.3.5  3.3.6  3.3.7  3.3.8  3.3.9  3.3.10 | Model Description  Preprocessing  Feature Standardization and Encoding  Cross-Validation  Ensemble Methods - Random Forest Classifier  Hyperparameter Tuning  Support Vector Classifier (SVC)  K-Nearest Neighbors (KNN)  Gaussian Naive Bayes  Evaluation Metrics and Confusion Matrices  Logistic Regression | **08**  08  08  09  09  09  10  10  10  11  11 |
| **4** | **Implementation** | **12-14** |
| **5** | **Results** | **15-20** |
| **5.1**  **5.2**  **5.3**  **5.4** | Analysis  Tableau  Machine Learning  Deployment | 15  16  17  20 |
| **6** | **Future Enhancement** | **22** |
|  |  |  |
| **7** | **References** | **23** |
|  |  |  |

**Table of Figures**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Title** | **Page No.** |
| 1 | Proposed Work | 5 |
| 2 | Data Source Connection | 6 |
| 3 | CRISP-DM | 8 |
| 4 | Query and Output | 15 |
| 5 | Query and Output | 15 |
| 6 | Query and Output | 16 |
| 7 | Query and Output | 16 |
| 8 | Dashboard for Driver | 17 |
| 9 | Dashboard for Circuit | 17 |
| 10 | Code for training model | 18 |
| 11 | Code for Feature Importance | 19 |
| 12 | Feature Importance | 19 |
| 13 | Home page | 20 |
| 14 | Driver’s position prediction | 20 |
| 15 | Tableau Dashboard 1 | 21 |
| 16 | Tableau Dashboard 2 | 21 |

**Chapter 1 Introduction**

**1.1 Introduction**

Formula 1, commonly known as F1, is a premier international motorsport competition that showcases the pinnacle of automotive engineering and racing expertise. It involves high-performance single-seater cars racing on circuits across the globe. F1 races are characterized by their precision, speed, and technical innovation, making them a highly competitive and captivating sporting spectacle.

The world of F1 racing generates an immense volume of data encompassing variables such as race times, driver performance, team strategies, weather conditions, and circuit characteristics. Analyzing this data is crucial as it provides insights into the factors that influence race outcomes. By deciphering patterns and relationships, data analysis helps teams make informed decisions, optimize strategies, and enhance performance.

Machine learning, a subset of artificial intelligence, involves developing algorithms that enable computers to learn from data and make predictions or decisions without explicit programming. In the context of F1 racing, machine learning can be employed to build predictive models that forecast race outcomes based on historical data. These models learn from past races to recognize complex interactions between variables, enhancing the accuracy of predictions.

Visualization is a powerful tool that transforms complex data into visually appealing representations. In the realm of F1 racing, visualization aids in presenting insights in an easily understandable manner. Interactive graphs, charts, and dashboards created using tools like Tableau allow stakeholders to explore data trends, compare performance metrics, and gain a comprehensive view of the dynamics shaping F1 races. Visualization bridges the gap between raw data and actionable insights, facilitating more informed decision-making.

**1.2 Objective**

The objectives of the project work are as following-

* Utilizing PySQL and PySpark to efficiently extract, transform, and preprocess extensive datasets necessary for training machine learning models.
* Creating interactive visualizations using Tableau to effectively communicate intricate analysis findings in a user-friendly manner.
* Developing and deploying machine learning models to accurately predict race results by leveraging historical race data and identifying performance-influencing features.
* Performing comprehensive data analysis on F1 racing data to uncover key factors impacting race outcomes.

**Chapter 2 Literature Review**

**Ankur Patil et al. [1]** The study aims to identify key variables that contribute to a driver's success in F1 races by using data-driven approaches. The researchers collected data spanning five years (2015–2019) from web scraping and performed an in-depth analysis of various F1 race-related variables. The paper highlights the correlation and interdependency among these variables and employs Principal Component Analysis (PCA) to reduce the dimensionality of the dataset.

**Veronica Nigro [2]** In this article, the author presents a comprehensive project aimed at predicting Formula 1 Grand Prix winners using a machine learning approach. The project involves data collection from various sources, including race results, driver standings, qualifying times, weather conditions, and more. The author analyzes the collected data to explore factors such as circuit locations, pole positions, home country advantage, dangerous circuits, and driver ages. The article also delves into the machine learning modeling process, discussing the use of regression and classification models, with neural networks showing promising results. The author's predictive model achieves a 62% accuracy rate in predicting race winners for the 2019 season and compares these predictions against bookmakers' odds, demonstrating the potential practicality of the approach.

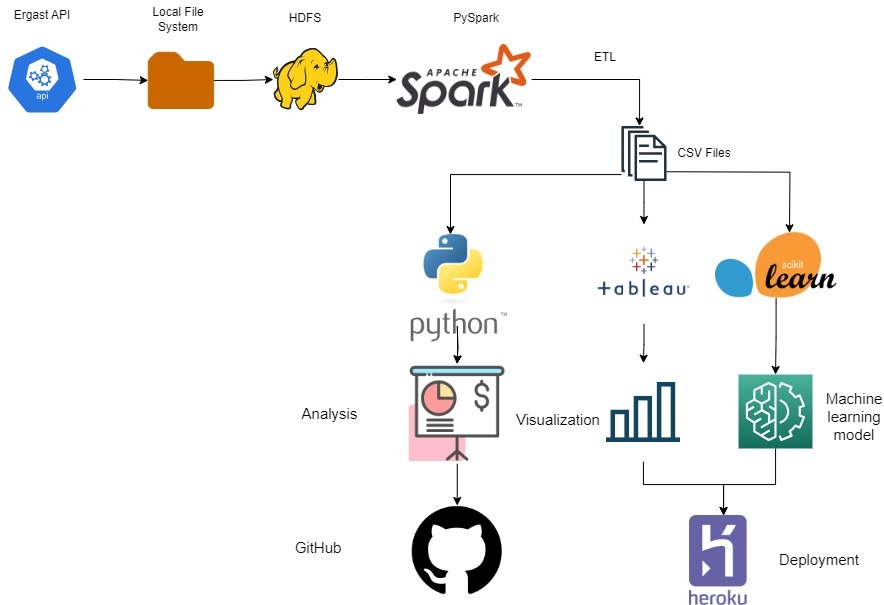
**Alexander Heilmeier et al. [3]** This project aims to enhance Formula 1 race strategies through the application of neural networks (NNs) for automated pit stop and tire compound decisions. Utilizing various NN architectures, including a hybrid model combining feed-forward and LSTM cells, the project develops predictive models for optimizing pit stop timings and tire compound choices. The integration of these models into a Virtual Strategy Engineer (VSE) framework allows for simulation-based analysis of race scenarios. The results highlight the potential benefits of the VSE in improving average result positions, particularly considering safety car phases. This research contributes to the literature by showcasing the efficacy of data-driven decision-making in optimizing race strategies and demonstrates the applicability of NNs in the dynamic context of Formula 1 racing.

**Horatiu Sicoie [4]** The project employs supervised Machine Learning algorithms to predict race winners and championship standings in Formula 1 racing. It involves data retrieval from an open-source API and web scraping for historical F1 data, utilizing regression and ensemble learning algorithms like Random Forest Regressor, Gradient Boosting Regressor, and Support Vector Regressor. Evaluation metrics including Spearman's rank correlation, Pearson correlation, R-squared, MSE, and RMSE assess model performance. Results indicate promising prediction of top 10 finishers, with features like Mercedes affiliation, driver age, and grid positions being significant. The study successfully predicts championship standings, while suggesting avenues for further improvement, collaboration, and data enrichment within the Formula 1 analytics field.

**Chapter 3 Methodology and Techniques**

**3.1 Methodology:**

The methodology employed for the Formula 1 race analysis project embodies a structured approach that encompasses data extraction, transformation, exploratory analysis, visualization, and machine learning. Leveraging a range of tools and libraries, the project aimed to unearth insights that facilitate informed decision-making within the dynamic realm of Formula 1 racing. In the subsequent sections, each phase of the methodology will be elaborated upon to provide a comprehensive understanding of how the project unfolded and successfully derived valuable insights from the intricate world of Formula 1 racing.



**Fig.1 Proposed Work**

**3.1.1 Data Extraction and Storage:**

The project began by extracting raw race data from the Ergast API, a comprehensive source of Formula 1 racing statistics. This data, rich with information about races, drivers and teams was then stored in the Hadoop Distributed File System (HDFS) for efficient data management and processing.

**3.1.2 Data Cleaning and Transformation with PySpark:**

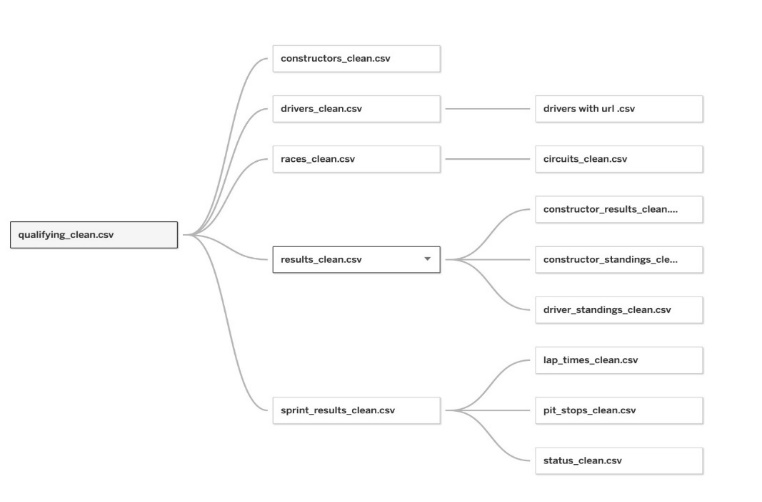
To ensure dataset quality, the team employed Apache Spark's PySpark library for Extract, Transform, Load (ETL) processes. Raw data in CSV and JSON formats underwent rigorous cleaning, handling various formats, removing missing values, and standardizing data representations using PySpark's versatile capabilities.

**3.1.3 Exploratory Data Analysis (EDA) with Python Libraries:**

The transformed and cleaned dataset was then subjected to Exploratory Data Analysis (EDA) using popular Python libraries such as NumPy, pandas, seaborn, and matplotlib. This stage involved thorough statistical analysis and visualizations to uncover patterns, trends, and insights within the data. Metrics related to race outcomes and driver performance were extracted, facilitating a comprehensive understanding of Formula 1 dynamics.

**3.1.4 Tableau Visualization and Dashboard Creation:**

With valuable insights derived from EDA, the next step involved employing Tableau for visualization. Using Tableau's intuitive interface, interactive visualizations were crafted to represent complex race data comprehensibly. Dashboards were designed to provide interactive insights of drivers and constructors, enabling one to monitor driver and team performance effectively.



**Fig.2 Data source connection**

**3.1.5 Machine Learning Model Application:**

Building on the enriched dataset, the project then turned towards applying machine learning algorithms. Logistic regression, decision trees, random forests, support vector machines (SVM), k-nearest neighbors (KNN), and Gaussian Naive Bayes were implemented to predict race outcomes, analyze the impact of different variables, and uncover key performance indicators. These models provided predictive capabilities that enhanced strategic decision-making for teams and drivers.

**3.2 Dataset**

The Ergast Developer API offers a unique and experimental web service tailored to provide historical motor racing data exclusively for non-commercial purposes. The API is a treasure trove of Formula One series data, spanning all the way back to the inception of world championships in 1950. This resource-rich API caters to developers and enthusiasts alike, offering a manual interface for non-programmers to query the database and also allowing the option to download the database tables in CSV format. These tables can be imported into spreadsheets or analysis software for further examination.

The API documentation emphasizes the availability of diverse response formats including XML, JSON, and JSONP, each enabling distinct methods of data retrieval. The API's data structure supports an array of queries, ranging from race schedules, results, standings, driver and constructor information, to lap times and more. One can also utilize response paging and implement caching mechanisms to enhance the efficiency of applications and reduce the load on the API server. The API's community-driven nature is underscored by the availability of source code and development tools on GitHub, with a commitment to user feedback and bug reporting for continuous improvement.

**3.3 Model Description**

**3.3.1 Preprocessing**

Preprocessing served as a critical phase in the project, involving a thorough refinement of the raw data to enhance its quality and relevance. Following the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework, the team iteratively refined the dataset by applying a series of filters to extract essential columns and features. This iterative process involved multiple iterations of changing dataframes, reconsidering approaches, and adjusting parameters to achieve optimal results.



**Fig.3 CRISP-DM**

The CRISP-DM framework guided the iterative approach, allowing the team to progressively fine-tune the dataset and ensure that only the most pertinent information was retained for analysis. Iterative preprocessing was vital due to the project's evolving nature, enhancing accuracy through adaptability. It optimized the dataset, enabling successful subsequent analysis and modelling.

**3.3.2 Feature Standardization and Encoding:**

Fair comparison across different algorithms requires features to be on a common scale. This is achieved through feature standardization, where features are transformed to have a mean of zero and a standard deviation of one. Categorical features, such as team names or driver nationalities, are encoded into numerical representations using techniques like one-hot encoding. This process readies the data for integration into machine learning models.

**3.3.3 Cross-Validation:**

Accurately estimating a model's performance necessitates robust evaluation techniques. The project employs Stratified K-Fold cross-validation, a method that partitions the dataset into K subsets while maintaining the class distribution. This approach ensures that each model is trained and validated on diverse data points, mitigating the risk of overfitting. Cross-validation provides a comprehensive assessment of a model's generalization capabilities, aiding in selecting the best algorithm for the task.

**3.3.4 Ensemble Methods - Random Forest Classifier:**

The algorithm adopts an ensemble learning approach, demonstrating superior performance compared to alternative algorithms in the selection. The rationale behind opting for this specific algorithm is rooted in its adeptness at handling noise inherent in the data, as well as effectively managing the abundance of categorical variables present in the dataset. This choice is further justified by the algorithm's remarkable speed in execution, which is crucial for real-time prediction scenarios, while also minimizing the risk of overfitting – a common challenge in predictive modelling. RF posses high classification accuracy, tolerate outliers and noise well and never got overfitting. (Liu, Wang, & Zhang, 2012) [5].

**3.3.5 Hyperparameter Tuning:**

In the quest for optimal performance, the code harnesses the potential of hyperparameter tuning. A randomized search is employed, efficiently navigating through the hyperparameter space of the Random Forest Classifier. By randomly selecting combinations of hyperparameters, the search strikes a balance between exploration and exploitation, identifying the configuration that maximizes predictive prowess. This fine-tuning process elevates the model's ability to capture underlying patterns in the data.

**3.3.6 Support Vector Classifier (SVC):**

Recognizing the complexity of Formula 1 race outcomes, the project incorporates the Support Vector Classifier (SVC). Operating in a high-dimensional space, SVC excels in segregating data into classes through optimal hyperplanes. Its non-linear kernel functions enable the detection of intricate relationships between features and outcomes. The inclusion of SVC offers a point of comparison against the Random Forest's performance, shedding light on each algorithm's strengths.

**3.3.7 K-Nearest Neighbors (KNN):**

KNN classifies a new data point by considering the class labels of its k-nearest neighbors in the training data. It measures distances between data points to determine proximity. KNN is simple and adaptable to various data distributions, but it can be sensitive to noisy data and requires careful selection of k. KNN was chosen to consider the influence of similar past races on current predictions. It can capture localized patterns that other algorithms might overlook, making it suitable for capturing track-specific dynamics.

**3.3.8 Gaussian Naive Bayes:**

Gaussian Naive Bayes is based on Bayes' theorem and assumes that features are normally distributed within each class. It calculates the likelihood of each feature's value given the class and combines this with prior probabilities to make predictions. Despite the "naive" assumption, it performs well for many real-world applications. Gaussian Naive Bayes was employed to capture probabilistic relationships between features and race outcomes. While its "naive" assumption might not always hold true, it offers a quick and interpretable way to estimate the likelihood of specific events occurring during races.

**3.3.9 Evaluation Metrics and Confusion Matrices:**

The project goes beyond accuracy as the sole evaluation metric. Precision, F1-score, and recall metrics provide nuanced insights into model performance across different categories. Confusion matrices visualize classification outcomes, delineating true positives, true negatives, false positives, and false negatives. These matrices paint a holistic picture of how well each algorithm predicts different race outcomes, enabling a comprehensive assessment.

**3.3.10 Logistic Regression:**

Logistic Regression is a linear classification algorithm that predicts the probability that an input belongs to a particular class. It uses the logistic function to transform the linear combination of features into a probability score. Logistic Regression is interpretable and works well when the relationship between features and outcome is roughly linear. Logistic Regression was chosen for its simplicity and interpretability, making it a good starting point for understanding the relationships between features and race outcomes.

**Chapter 4 Implementation**

1. Use of Python Platform for writing the code with PySpark

2. Hardware and Software Configuration:

**Hardware Configuration:**

● CPU: intel i5 or similar, 8 GB RAM

**Software Required:**

1. **Hadoop**: The core framework that enables the distributed storage and processing of large datasets across clusters of computers. HDFS is a fundamental component of Hadoop for storing data across multiple machines.

2. **HDFS**: Hadoop Distributed File System provides a scalable and reliable method for storing large volumes of data across multiple servers.

3. **PySpark**: The Python library for Apache Spark, used for big data processing and analytics. It provides APIs for ETL tasks, machine learning, and data analysis on distributed clusters.

4. **Python**: The programming language used to write PySpark scripts, interact with various components of the project, and create Streamlit apps.

5. **Spyder**: Spyder, the Scientific Python Development Environment, Spyder is a free integrated development environment (IDE) included with Anaconda. Tailored for scientific use, it offers features like code editing, interactive testing, debugging, and data visualization. It's designed for Python-based data analysis, engineering, and scientific research.

6. **Anaconda**: Anaconda is a powerful package management software that offers free distributions of Python and R programming languages. It's an essential tool for data science, machine learning, and scientific computing, simplifying the setup of complex environments and dependencies.

7. **Jupyter Notebook**: Jupyter Notebook is an interactive web-based environment that combines code, data, and visualizations. It's highly flexible, supporting various workflows in data science and machine learning. Its extensibility allows integration of new components through plugins. Jupyter is extensible and modular: write plugins that add new components and integrate with existing ones.

8. **Visualization** Tools: Visualizing ETL and analysis results with the help of tools like Matplotlib, and Seaborn.

9. **Docker**: A containerization platform used to package the application and its dependencies into a single container. Docker ensured consistent behaviour across different environments and simplifies deployment.

10. **Heroku:** Herokuwas chosen as the deployment platform for our project due to its user-friendly interface and streamlined deployment process. It enabled us to host our web application, showcasing our machine learning models and interactive visualizations without the complexities of managing server infrastructure. Heroku's integration with version control systems also allowed for seamless updates and changes, making our project accessible to a broader audience in a convenient and efficient manner.

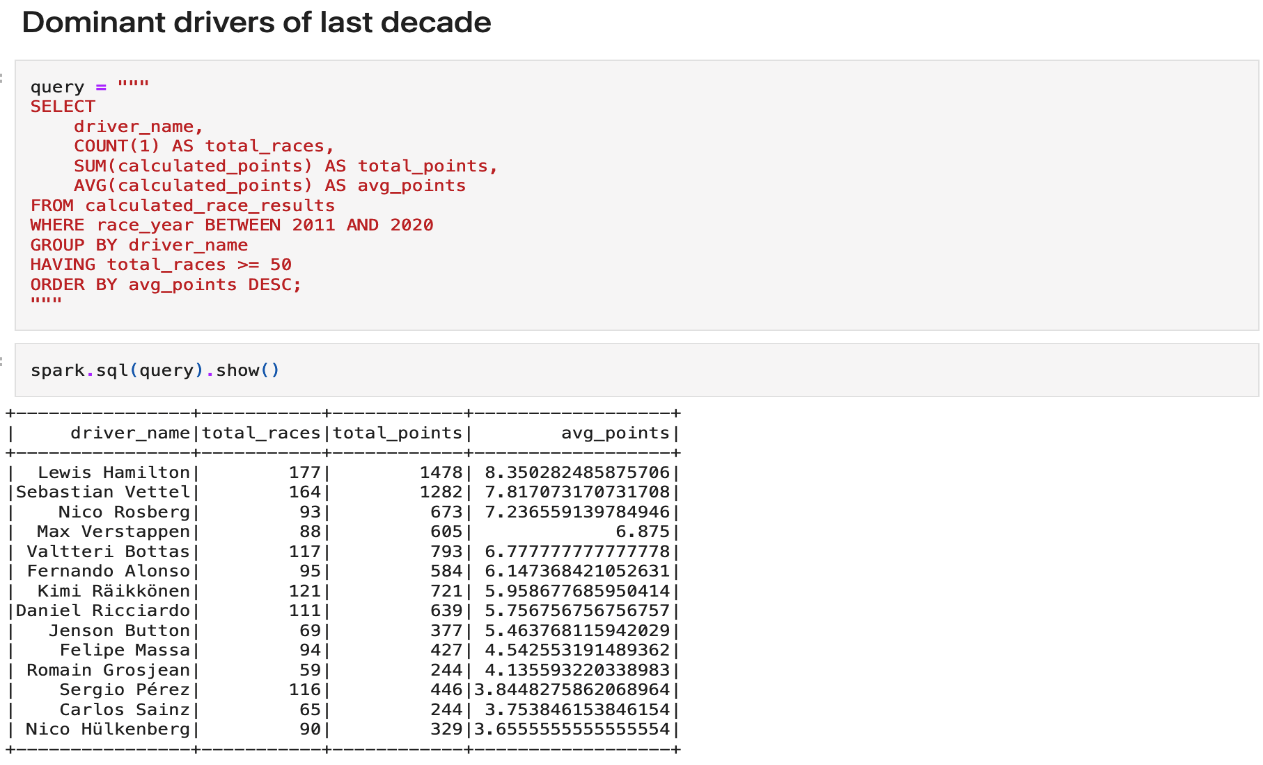
11. **GitHub**: A version control platform that helped to manage and collaborate on the project’s source code. It's a central place for storing and tracking changes to codebase.

12. **Streamlit**: A Python library used to create interactive web applications for data visualization and exploration. It allowed to showcase the project's results and insights in an accessible and user-friendly manner.

**Chapter 5 Results**

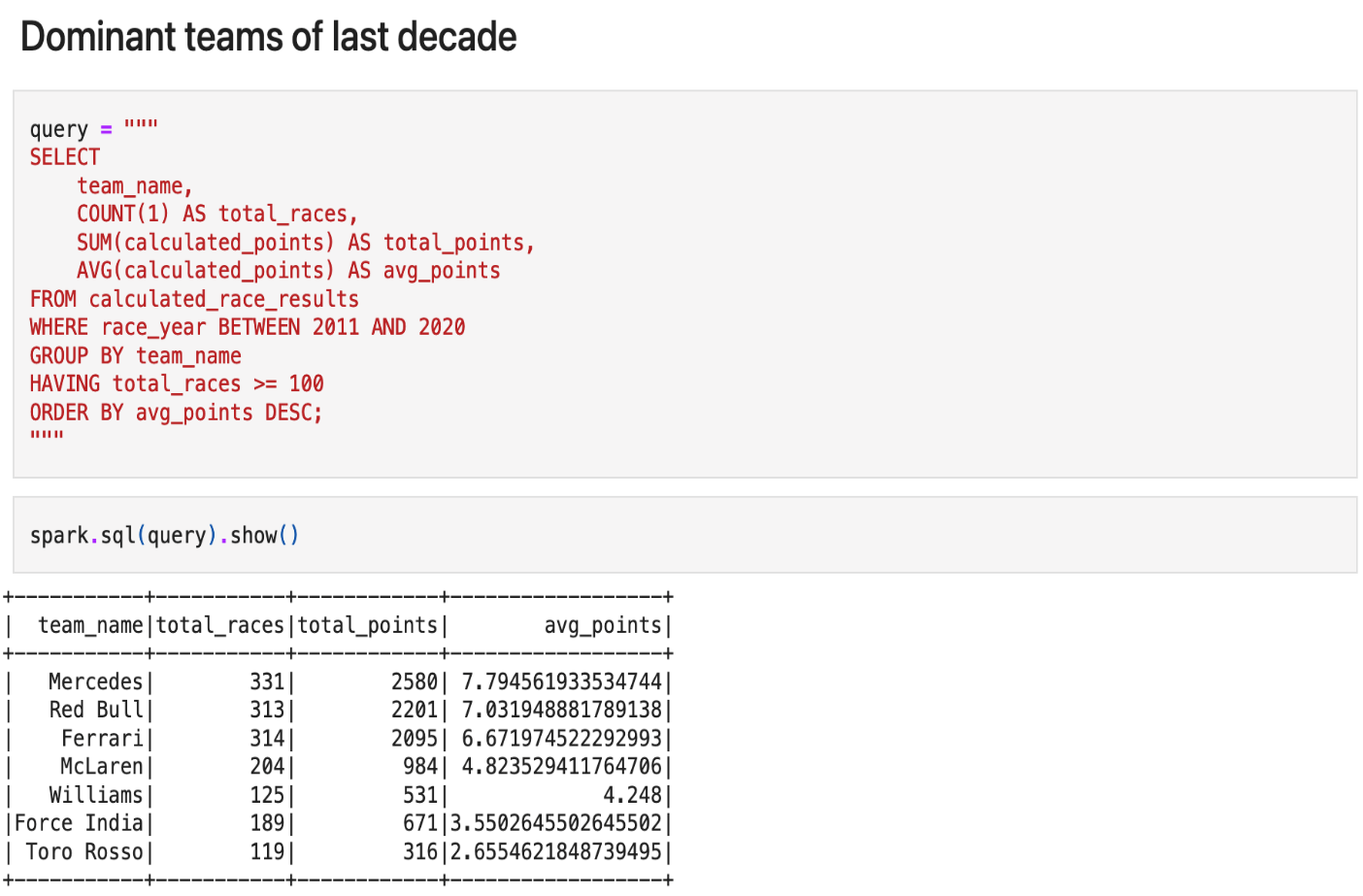
**5.1 Analysis**

**5.1.1 Dominant drivers of last decade**

****

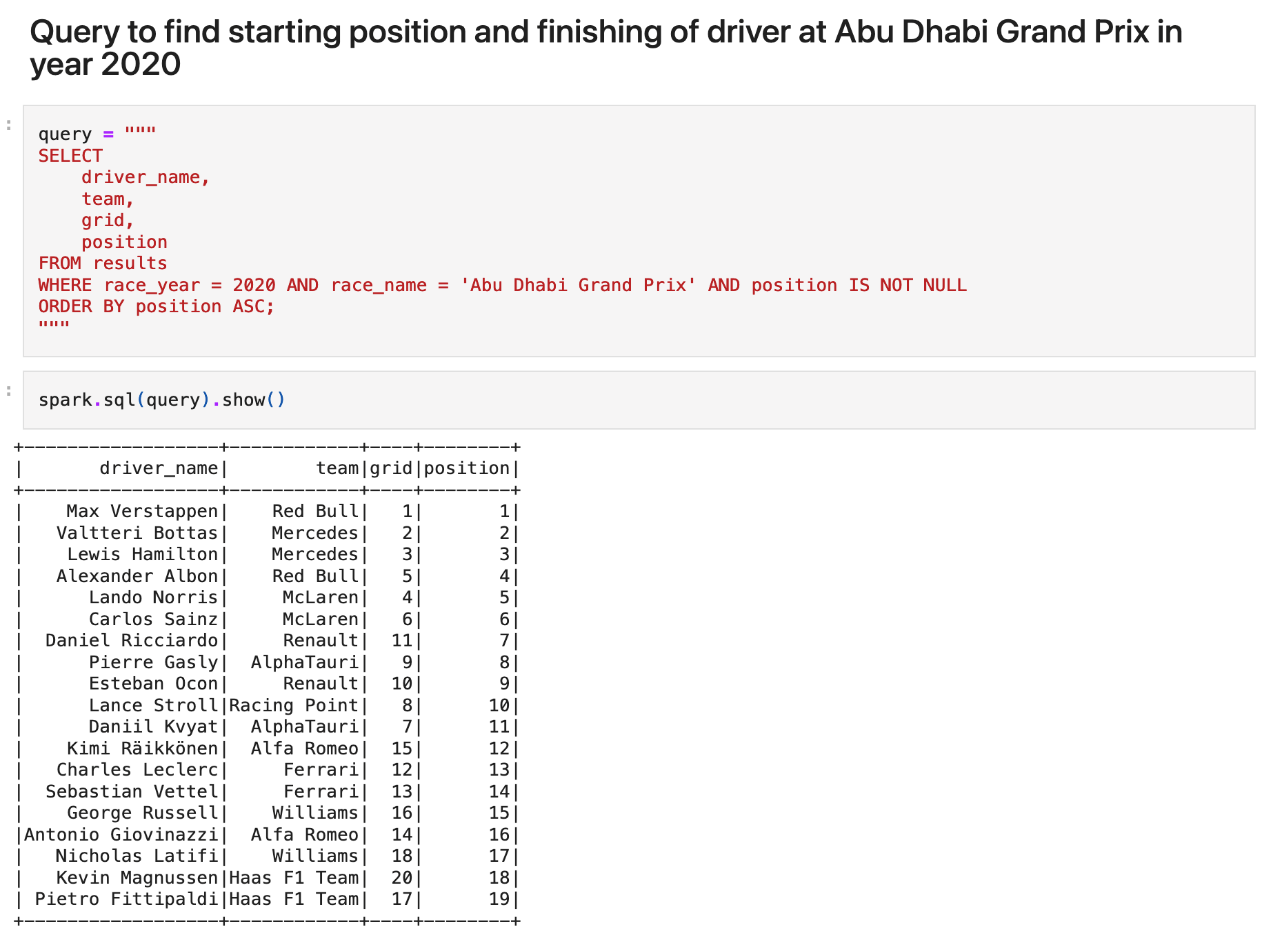
**Fig. 4 Query and output**

**5.1.2 Dominant teams of last decade**

****

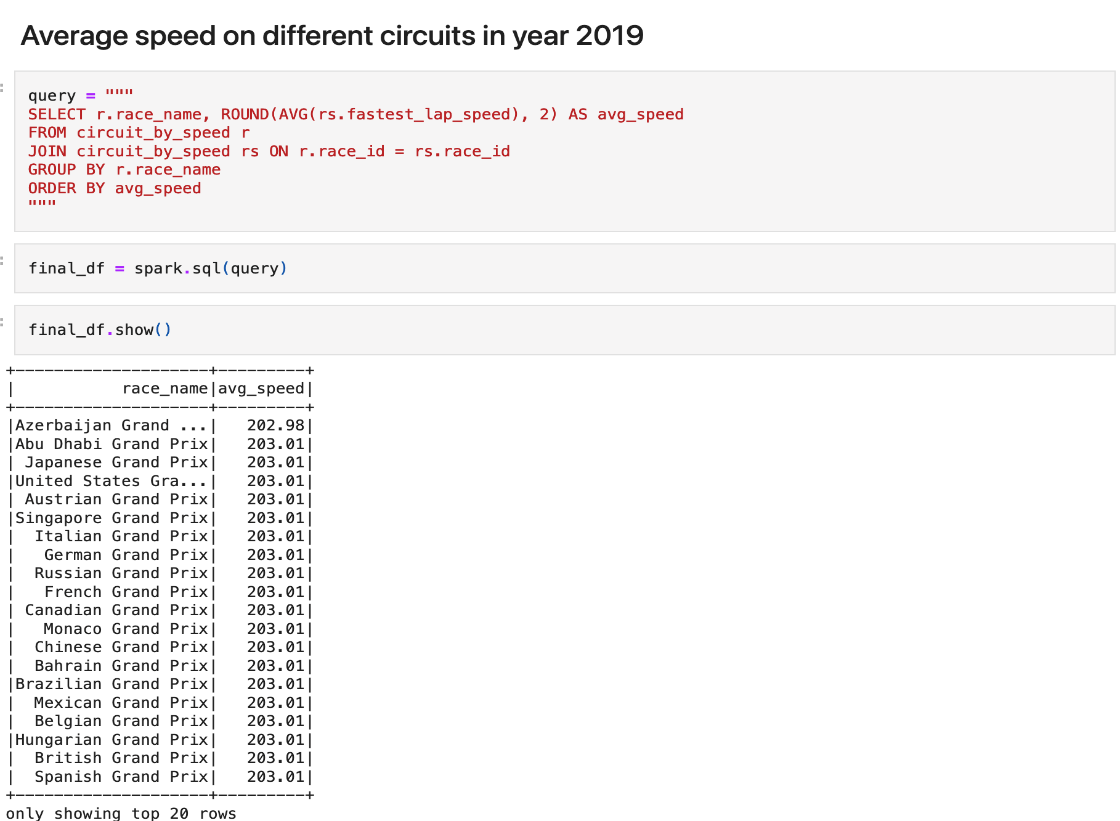
**Fig. 5 Query and output**

**5.1.3 Query to find starting position and finishing of driver at Abu Dhabi Grand Prix in year 2020**

****

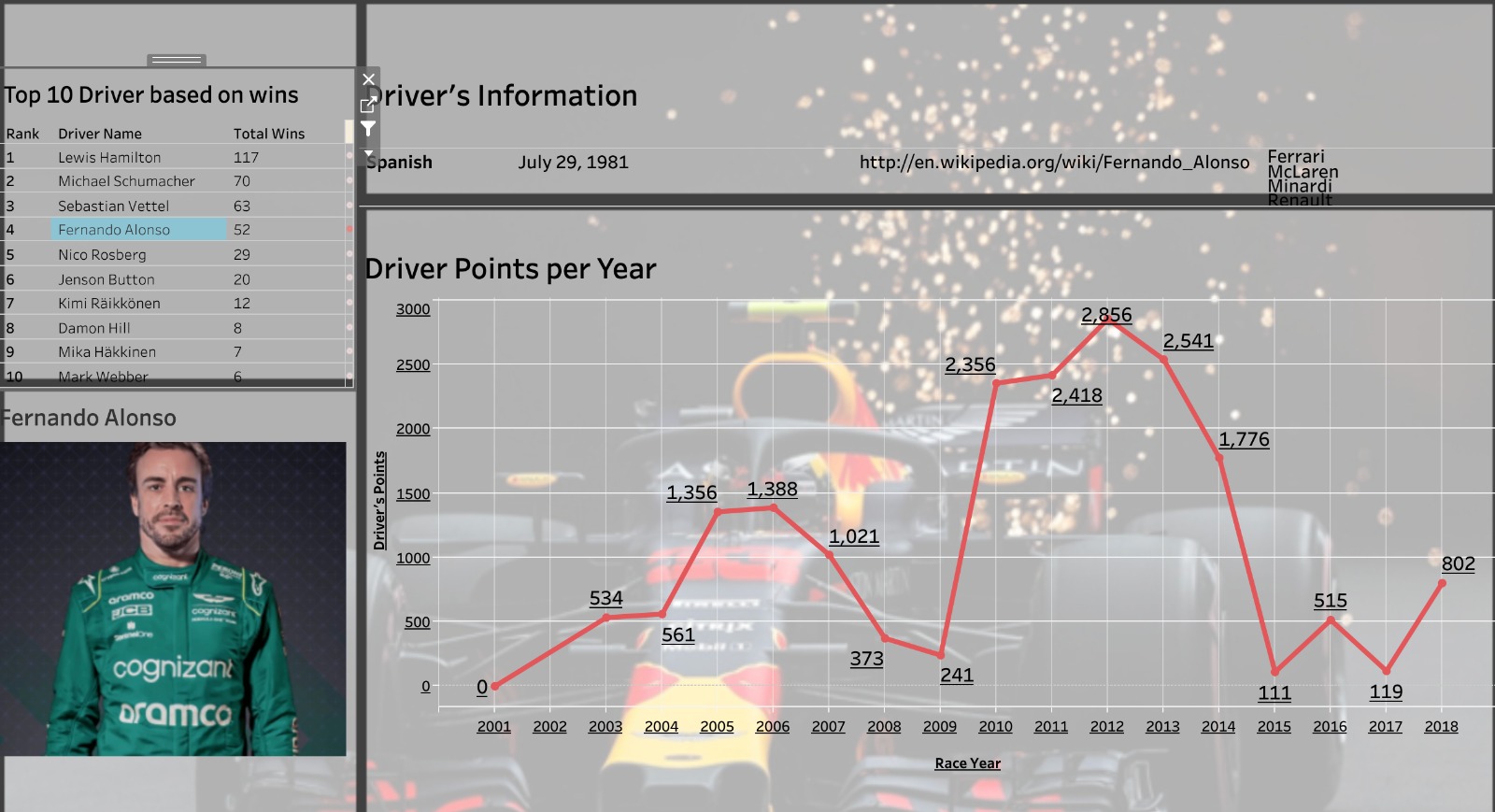
**Fig. 6 Query and output**

**5.1.4 Average speed on different circuits in year 2019**

****

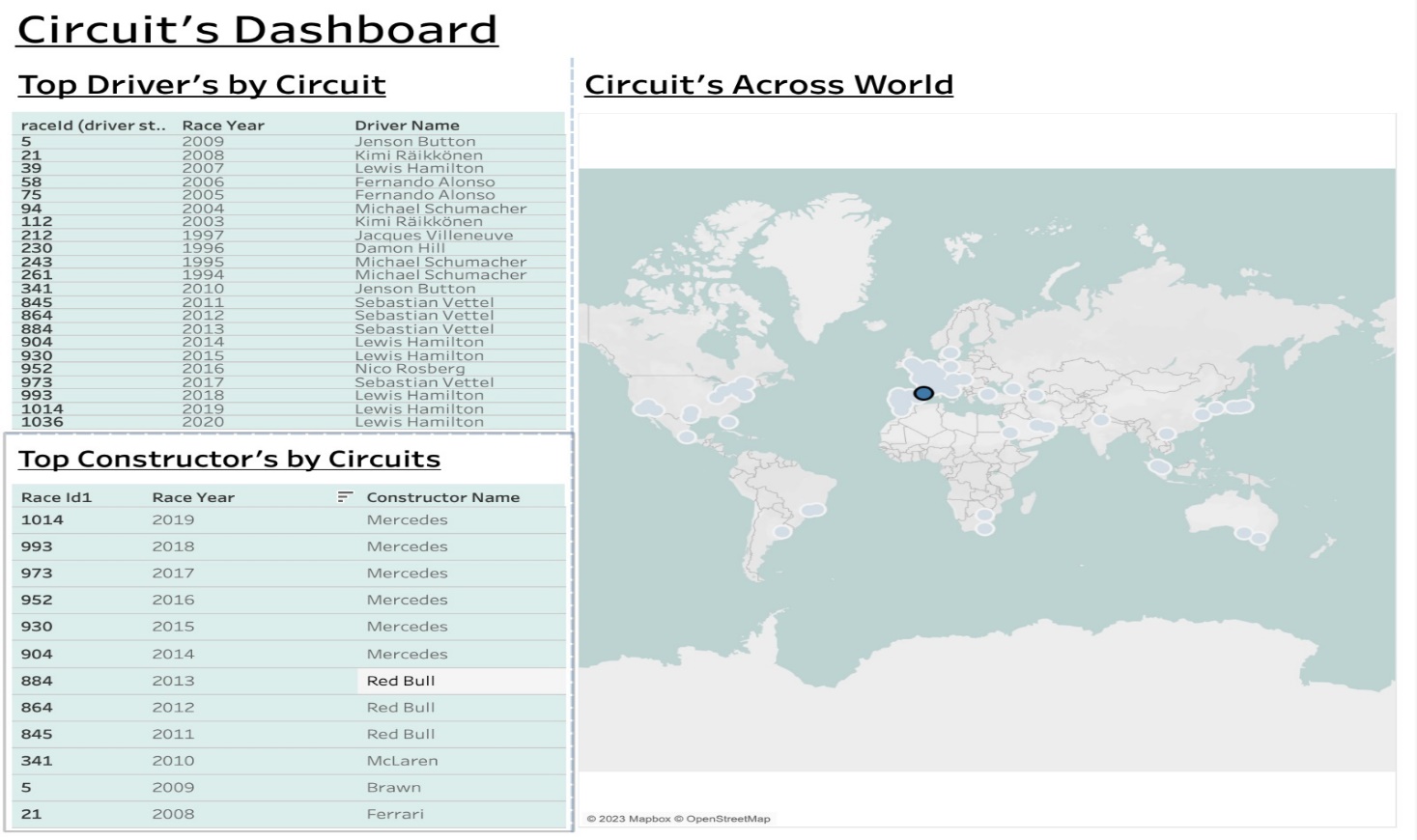
**Fig. 7 Query and output**

**5.2 Tableau**

**5.2.1 Dashboard 1**

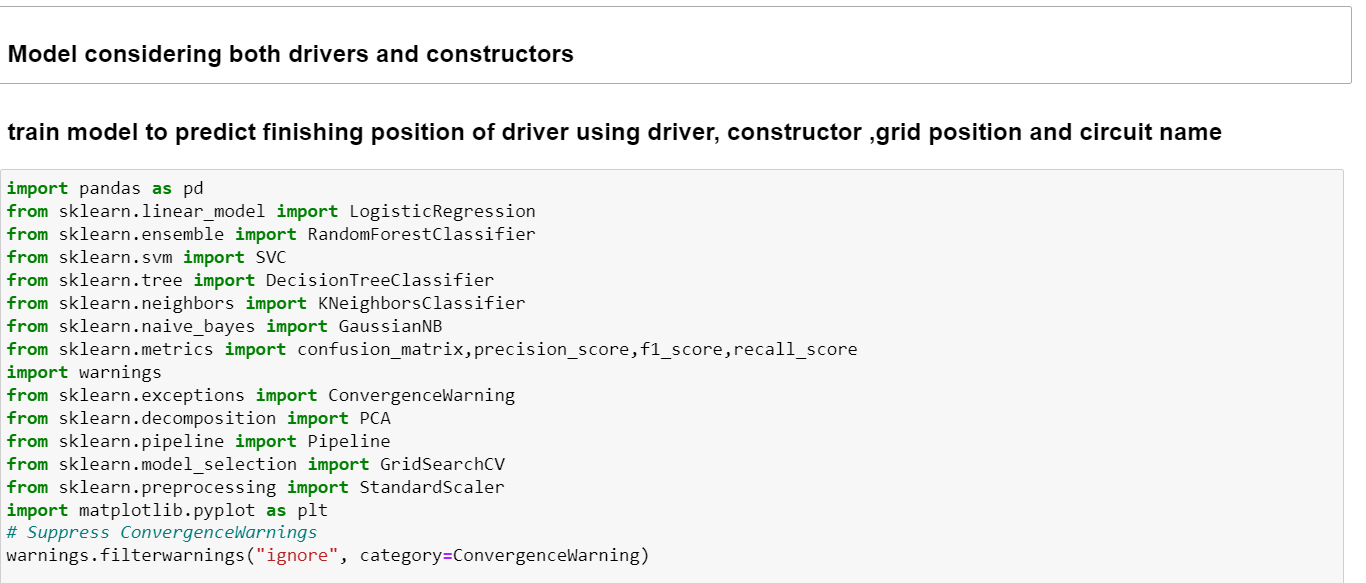
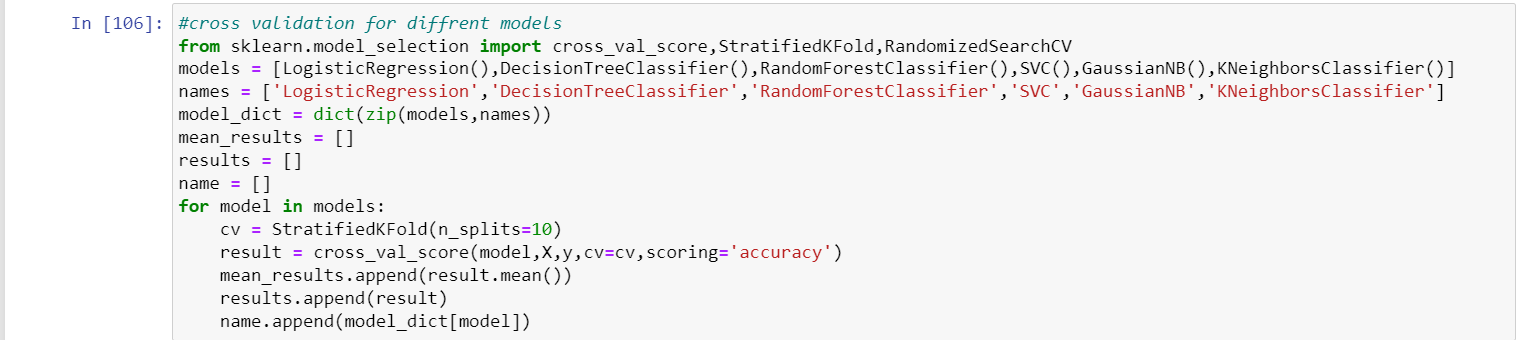
**Fig. 8 Dashboard for Driver**

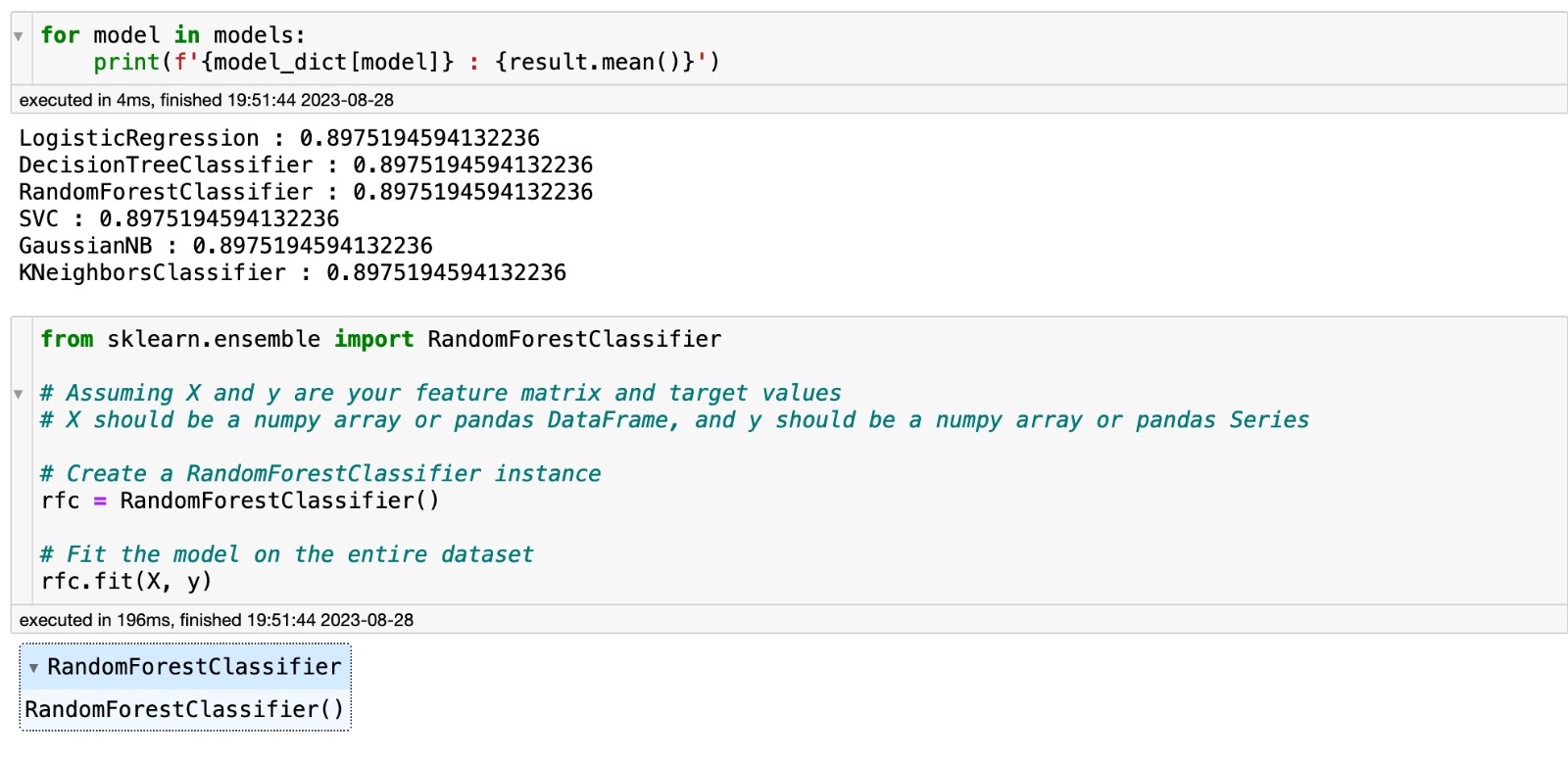
**5.2.2 Dashboard 2**

****

**Fig. 9 Dashboard for Circuit**

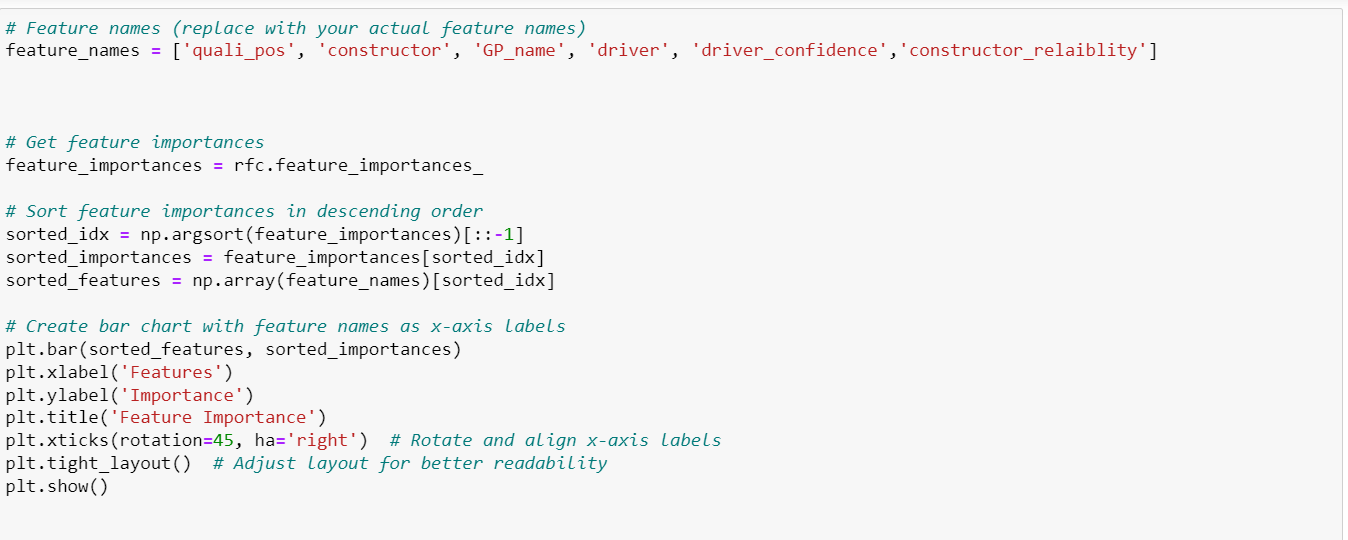
**5.3 Machine Learning**

**5.3.1 Model considering both drivers and constructors**

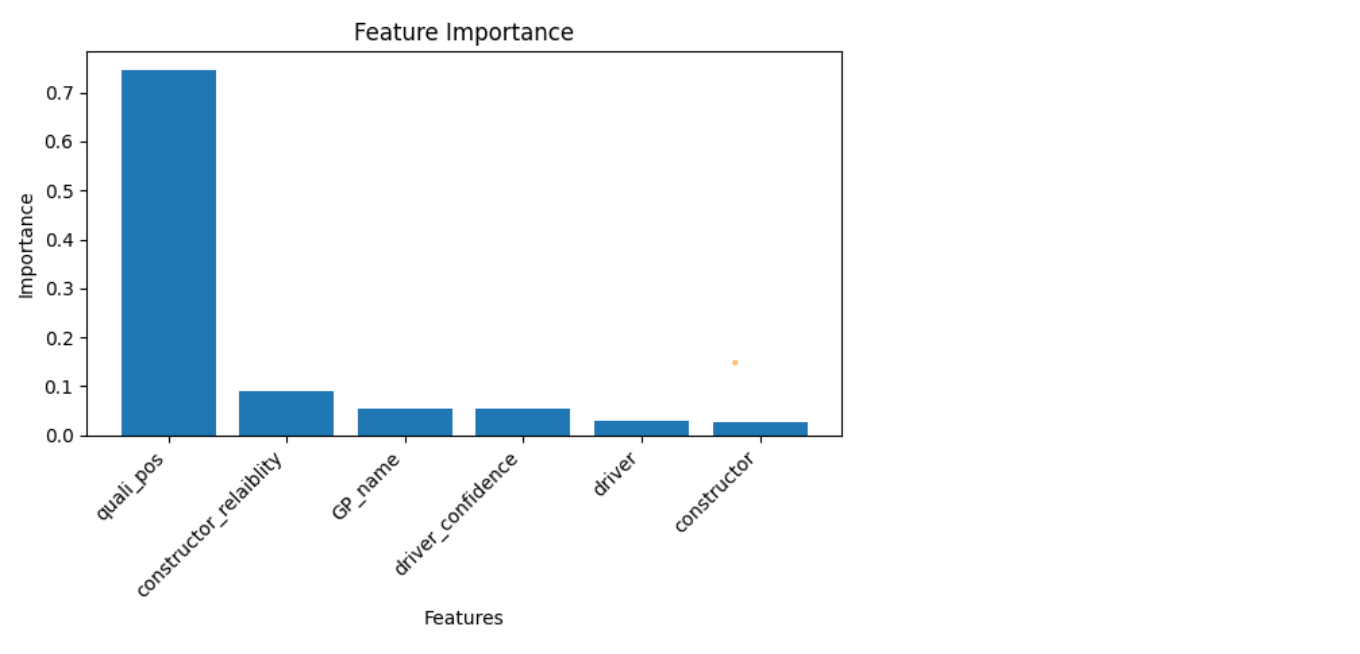
****

****

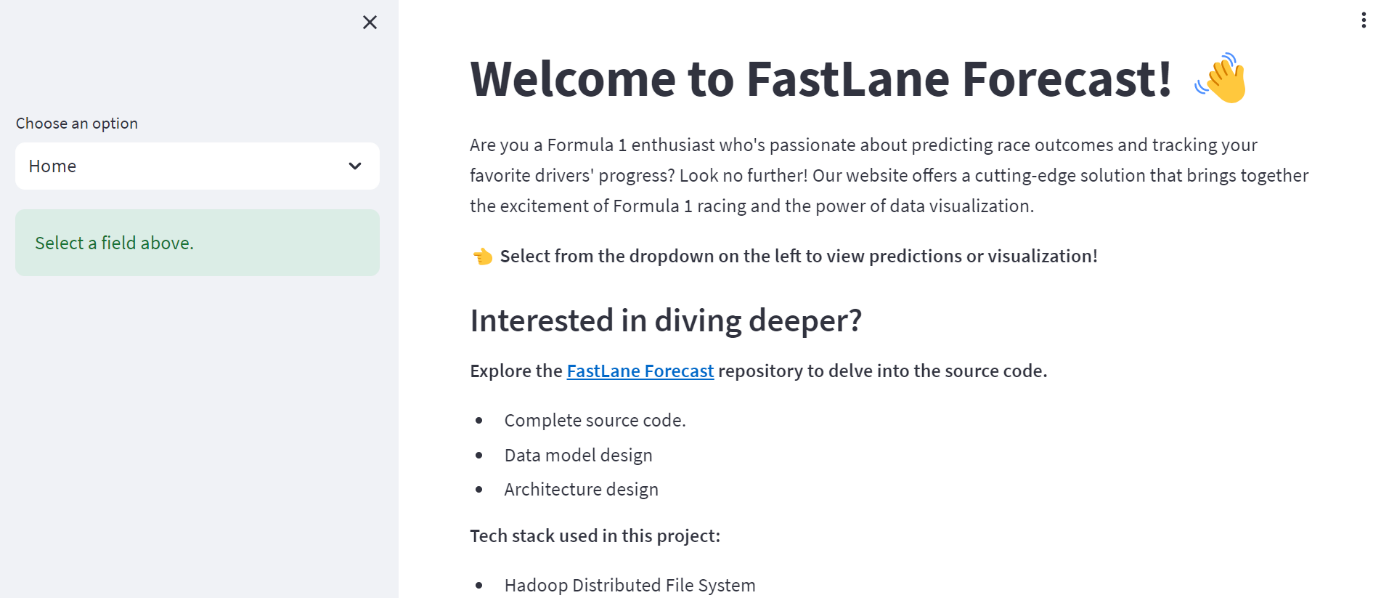
**Fig. 10 Code for training model**

**5.3.2 Feature Importance**

**Fig. 11 Code for Feature Importance**

****

**Fig. 12 Feature Importance**

**5.4 Deployment**

**Fig 13 Home page**

# 

# Fig 14 Driver’s position prediction

# 

# Fig 15 Tableau Dashboard 1

# Fig 14 Tableau Dashboard 2

Link of deployment: <https://fastlaneforecast-90d27975bbe0.herokuapp.com/>

# Chapter 6 Future Enhancement

Here is list of potential future enhancements for the "Fast Lane Forecast" project:

1. Real-time Data Integration: Implement live race data streaming for up-to-the-minute predictions.

2. Weather and Track Conditions: Include historical weather and track data for more accurate predictions.

3. Driver and Team Strategies: Predict optimal strategies like pit stops and tire changes.

4. Feature Engineering: Create domain-specific features to enhance prediction insights.

5. Ensemble Techniques: Explore combining predictions from multiple models for higher accuracy.

These enhancements can advance the project's capabilities and offer more value to Formula 1 stakeholders.

# Chapter 7 References

[1] Patil, A., Jain, N., Agrahari, R., Hossari, M., Orlandi, F., Dev, S. (2023). A Data-Driven Analysis of Formula 1 Car Races Outcome. In: Longo, L., O’Reilly, R. (eds) Artificial Intelligence and Cognitive Science. AICS 2022. Communications in Computer and Information Science, vol 1662. Springer, Cham. <https://doi.org/10.1007/978-3-031-26438-2_11>

[2] <https://towardsdatascience.com/formula-1-race-predictor-5d4bfae887da>

[3] Heilmeier, A., Thomaser, A., Graf, M., & Betz, J. (2020). Virtual Strategy Engineer: Using Artificial Neural Networks for Making Race Strategy Decisions in Circuit Motorsport. Applied Sciences, 10(21), 7805. <https://doi.org/10.3390/app10217805>

[4] Sicoie, H. (2022). Machine Learning Framework for Formula 1 Race Winner and Championship Standings Predictor <http://arno.uvt.nl/show.cgi?fid=157635>

[5] Liu, Y., Wang, Y., & Zhang, J. (2012). New machine learning algorithm:Random forest. In International conference on information computing and applications (pp. 246–252)