# 22PT22 – PAVAN KUMAR S G F1 Analysis App - Tools, Libraries, and Frameworks

## Overview

The F1 Analysis App is a powerful data-driven application designed to analyze Formula 1 data and provide insights into driver and team performance, race strategies, and predictions for future seasons. Built using Streamlit, this interactive web application integrates data science, machine learning, and advanced visualization techniques to present an intuitive and informative user experience. Users can explore in-depth analyses of driver performances, compare qualifying vs race statistics, and evaluate team performances through dynamic and interactive visualizations.

## Technologies Used

### Streamlit

A Python framework used to build the interactive web application, allowing for rapid development of data-driven applications with minimal effort. Streamlit is known for its simplicity, enabling developers to create intuitive dashboards and visualizations without extensive front-end development.

### Pandas

A robust data analysis and manipulation library in Python, essential for handling large datasets efficiently. Pandas provides functionalities such as data cleaning, merging, reshaping, and advanced indexing, making it indispensable for extracting meaningful insights from raw F1 race data.

### Matplotlib & Seaborn

These libraries are used for static, animated, and interactive data visualizations. Matplotlib provides low-level control over plots, while Seaborn offers high-level interfaces for creating aesthetically pleasing statistical graphics, crucial for analyzing race trends and performance metrics.

### Scikit-learn

A comprehensive machine learning library in Python that facilitates model training, evaluation, and deployment. The app utilizes Scikit-learn for regression models, clustering algorithms, and classification techniques to predict F1 outcomes, analyze race strategies, and compare team performances.

### NetworkX

An advanced network analysis library used for representing complex relationships in F1, such as driver movements across teams and historical team associations. This helps visualize driver career trajectories and team dynamics using graph-based analytics.

### Plotly

A widely used library for interactive visualizations, allowing users to engage with the data dynamically. Plotly enables zooming, filtering, and real-time updates, enhancing the analysis of qualifying vs race performances, pit stop efficiencies, and championship trends.

## Installation

To run the F1 Analysis App, you need to have Python 3.7 or higher installed. Follow the steps below to set up the environment.

### Prerequisites

Ensure you have Python installed on your system. It is recommended to use a virtual environment to manage dependencies.

### Clone the Repository

Use the following command to clone the repository:

git clone https://github.com/yourusername/f1-analysis-app.git

cd f1-analysis-app

### Install Dependencies

Install the required libraries using the following command:

pip install -r requirements.txt

### Run the Application

Start the Streamlit application with the command:

streamlit run src/main.py

## Project Structure

The project is structured as follows:

f1-analysis-app/  
│  
├── data/ # Directory containing dataset files  
├── src/ # Source code directory  
│ ├── models/ # Contains various analysis models  
│ │ ├── best\_team\_lineup.py  
│ │ ├── champion\_age\_trends.py  
│ │ ├── championship\_retention.py  
│ │ ├── driver\_consistency.py  
│ │ ├── driver\_movements.py  
│ │ ├── driver\_performance.py  
│ │ ├── head\_to\_head.py  
│ │ ├── hypothetical\_swaps.py  
│ │ ├── lap\_time\_efficiency.py  
│ │ ├── pit\_stop\_strategies.py  
│ │ ├── predict\_2025.py  
│ │ ├── qualifying\_vs\_race.py  
│ │ ├── struggling\_teams.py  
│ │ ├── team\_performance.py  
│ │ └── track\_struggles.py  
│ ├── utils/  
│ │ └── data\_loader.py # Utility script for loading data  
│ └── main.py # Main application script  
├── requirements.txt # List of dependencies  
└── README.md # Project documentation

## Analysis and Predictions

### Driver Performance

Analyzes the performance of drivers based on their race results, including metrics such as total races, wins, podiums, and points.

### Qualifying vs Race Performance

Compares qualifying positions with final race positions to identify patterns and correlations.

### Pit Stop Strategies

Examines pit stop counts and durations to assess their impact on race performance.

### Team Performance

Evaluates team performances based on historical data, considering average finishing positions.

### Best Team Lineup

Determines the most effective team lineup using statistical methods to optimize driver pairings.

### Struggling Teams

Identifies underperforming teams based on their historical and seasonal performance.

### Predictions for 2025 Season

Uses linear regression models to forecast driver and team performances for the upcoming F1 season.

### Championship Retention

Analyzes the probability of drivers retaining their championship titles based on past trends.

### Champion Age Trends

Studies the age distribution of F1 champions over the years to determine trends in championship-winning ages.

### Hypothetical Driver Swaps

Simulates potential outcomes of swapping drivers between teams to assess how different lineups could impact results.

### Driver Movements

Visualizes driver career trajectories and movements between teams using network analysis.