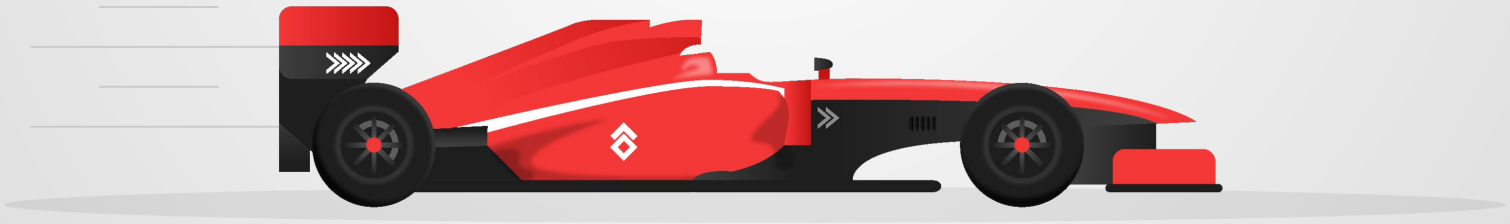
Formula Fun: Racing through Data with KNN

Rachel Roggenkemper, Alex Buntaran, Liam Quach, and Shantanu Singh



Agenda

01

Data Overview

Start your engines! Here's a lap around our data sources, features, and variables driving our predictions

02

Methodology

How we tuned our engines: preprocessing, KNN implementation, and performance metrics explained

03

Results

Crossing the finish line: A pit stop to analyze how well our model predicted F1 outcomes

04

Conclusions

Final thoughts from the podium: What we learned and where the data might take us next

Understanding the Task

Goal

Predict whether a Formula 1 driver finishes in the points (top 10) using machine learning (KNN) based on race and driver data

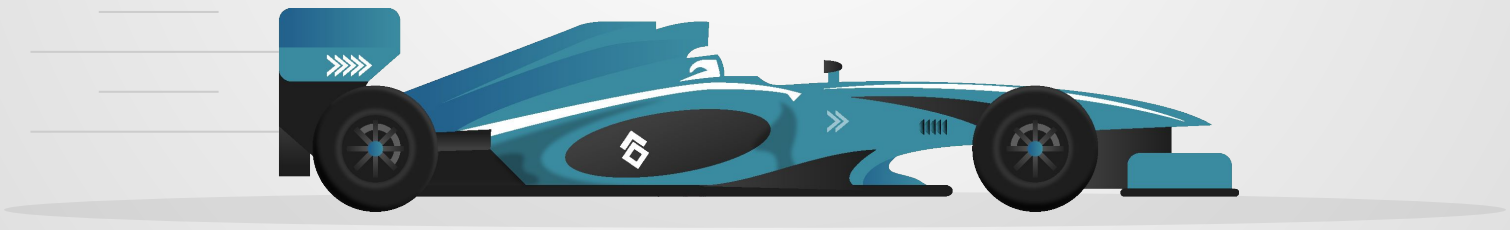
Objective

Develop a robust and accurate model by preprocessing data, implementing KNN, and evaluating model performance for actionable insights

01

Data Overview

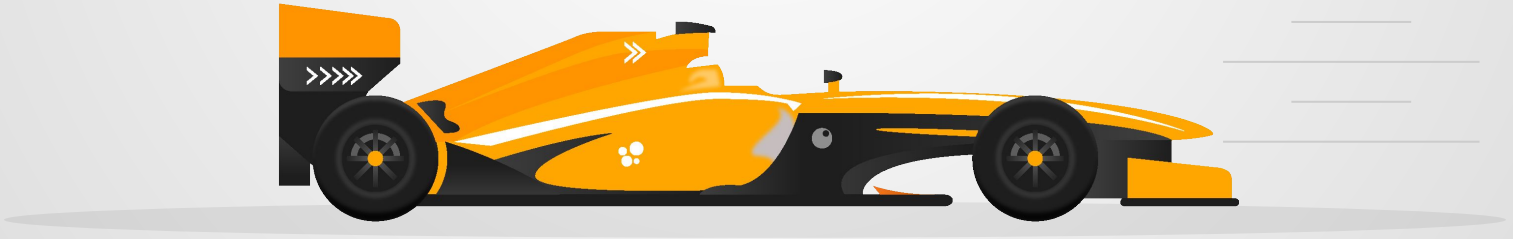
Start your engines! Here's a lap around our data sources, features, and variables driving our predictions



What is Formula 1?

Drivers compete in high-speed races across circuits worldwide where they aim to finish races as high as possible to earn points for themselves and their teams.

- 20 drivers
- 10 teams / constructors (2 drivers per team)
- Points are awarded to the top 10 finishers, with 25 points for the winner, scaling down to 1 point for 10th place



About the Data

Data Source:

- Kaggle: Formula 1 World Championship (1950 - 2024)

Tables Used:

- **Circuits**: Details on the locations of races, including track names, countries, and coordinates
- **Constructors**: Information on the teams that design and build cars for F1 races
- **Drivers**: Biographical data on drivers, including names, nationalities, and dates of birth
- **Pit Stops**: Timing and frequency of pit stops for each driver during races
- **Races**: Details about the races, such as dates, circuits, and seasons
- **Results**: Final race outcomes, including positions, points earned, and other performance metrics

Glimpse of Joined Dataset

Target Variable:

0	Driver did not finish in points (top 10)
1	Driver did finish in points (top 10)

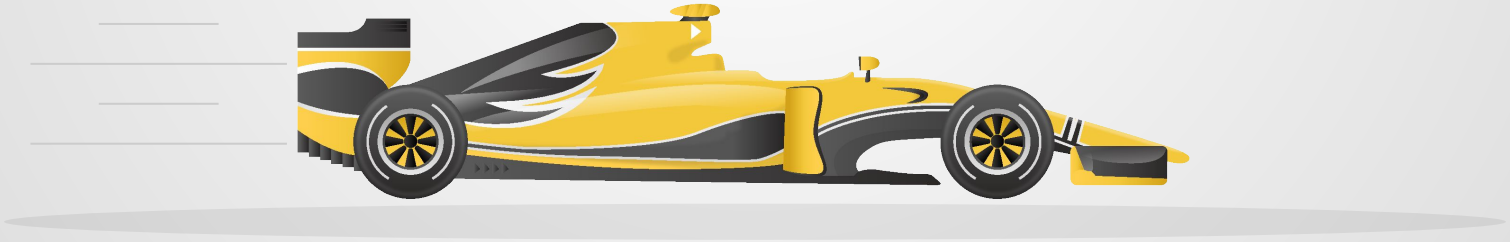
Target	Starting Position	Total Pit Stops	...	Driver	Driver Nationality	Constructor	Circuit
1	1	1	...	Charles Leclerc	Monegasque	Ferrari	Circuit de Monaco
0	2	2	...	Lando Norris	British	McLaren	Red Bull Ring

Each row in the dataset represents an individual driver's performance in a specific race

02

Methodology

How we tuned our engines: preprocessing, KNN implementation, and performance metrics explained



Data Cleaning

1. Joined All Data Frames Together

2. Filter: Only looking data for 2010 season and onward because points system changed starting in 2010

3. Feature Engineering:

- Created total pit stop and total pit duration because original pit stop data is per pit stop
- Created target variable (whether or not driver finished top 10)

Data Preprocessing

Standardized Data

- Standardization ensures that all numeric features are on the same scale, preventing features with larger scales from dominating the KNN distance calculations, which rely on Euclidean distances

Created dummy variables

- Dummy variables were created for categorical features to allow their inclusion in the model, as KNN requires numerical input

80/20 Train/Test Split

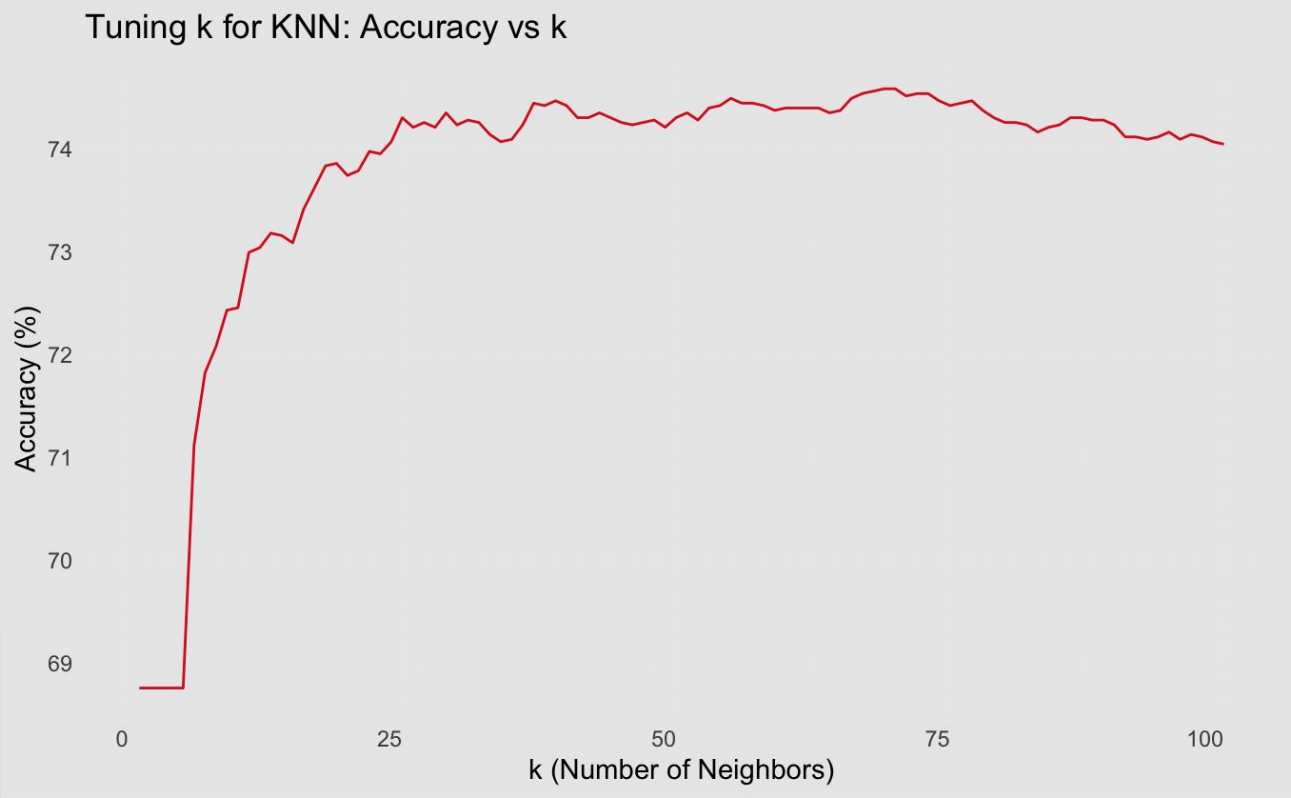
- The data was split into 80% for training and 20% for testing to train the model effectively while retaining a separate dataset to evaluate its performance and generalizability

K-Nearest Neighbors Classifier Algorithm

1. **Calculate Euclidean Distance**
   - The distance between a new data point and all training points is calculated to measure similarity
2. **Find the k Nearest Neighbors**
   - The k closest training points to the new data point are identified based on their distances
3. **Classification**
   - The new data point is assigned the class most frequently occurring among its k nearest neighbors

KNN is non-parametric -> makes no assumptions about form of the data

Hyperparameter Tuning: k

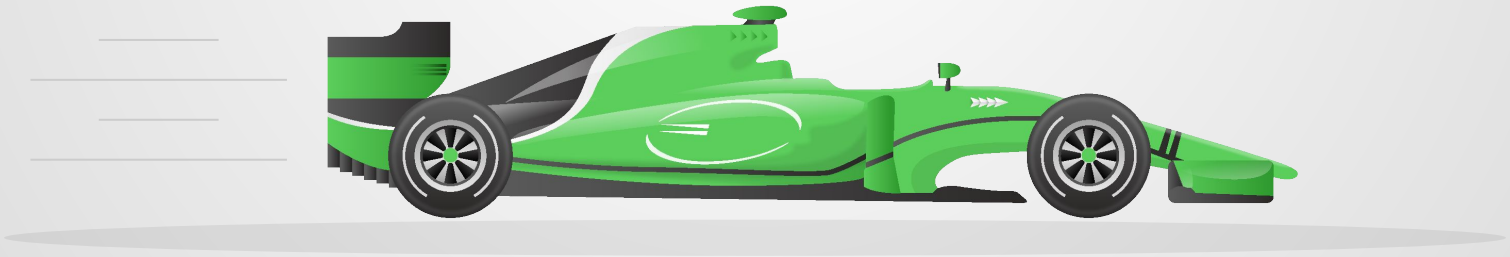


Number of Neighbors that Maximizes Accuracy:
k = 69

03

Results

Crossing the finish line: A pit stop to analyze how well our model predicted F1 outcomes



Confusion Matrix

		Actual: Driver did finish top 10	Actual: Driver did not finish top 10
Predicted	Driver did finish top 10	512	240
	Driver did not finish top 10	35	283

Metrics

Accuracy	Precision	Recall
Percentage correctly classified	Accuracy of positive predictions	Ability to identify all positives
74.3%	68%	93.6%

Obstacles

Overwhelming Data

- Narrowing down relevant predictors from multiple large datasets

Complex Table Joins

- Joining six tables accurately required meticulous care

Filtering for Relevance

- Adjusting data to include only post-2010 for point system consistency

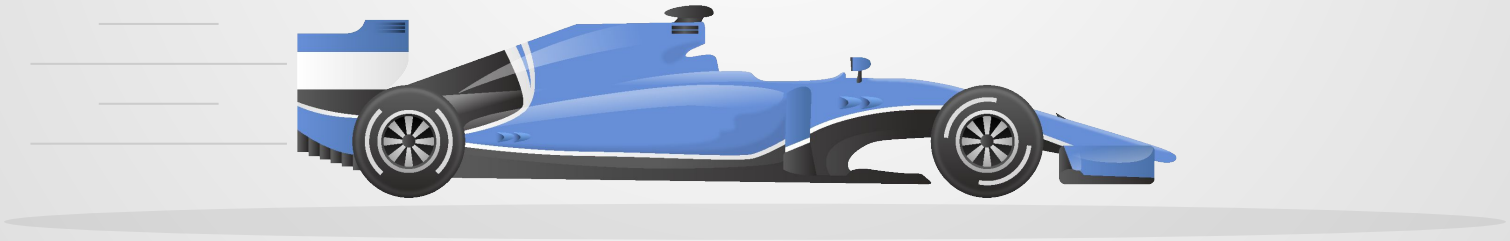
Handling Missing Data

- To ensure consistency and simplicity, we decided to drop rows with missing data, prioritizing data quality

04

Conclusions

Final thoughts from the podium: What we learned and where the data might take us next



Crossing the Finish Line

What We Learned

- Preprocessing and feature engineering are critical for model performance
- KNN is effective but computationally intensive for large datasets
- Hyperparameter tuning significantly improved classification accuracy

Model Insights

- Achieved 74.3% accuracy, with strong recall (93.6%) indicating the model's strength in identifying top 10 finishes
- Trade-off observed between precision and recall.

Future Directions

- Explore more advanced models (e.g., Random Forest) for enhanced performance.
- Incorporate additional features, like weather or tire strategies, for deeper insights.

Thank you!

Any questions?