National College of Ireland

Using Supervised Machine Learning to Predict the Final Rankings of the 2021 Formula One Championship

MSc Research Project

Master of Science in Data Analytics

Emma O'Hanlon

Student ID: 19210451

School of Computing

National College of Ireland

Supervisor: Zahid Iqbal

National College of Ireland

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Emma O'Hanlon
Student ID:	X19210451
Programme:	Master of Science in Data Analytics **Year:** 2022
Module:	MSc Research Project
Supervisor:	Zahid Iqbal
Submission Due Date:	12/12/2022
Project Title:	Using Supervised Machine Learning to Predict the Final Rankings of the 2021 Formula One
Word Count:	9,227 **Page Count** 19 excluding references, 23 including

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: 

Date: 12/12/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Using Supervised Machine Learning to Predict the Final Rankings of the 2021 Formula One

Emma O'Hanlon

19210451

1 Abstract

Formula One motor car racing is one of the most data-driven sports in the world. Decisions led by data are used to develop every strategy, from tactics for the entire season to in-event racing. It can be challenging for researchers to find the information that these teams are gathering internally in the public domain. The objective of this study is to gather publicly available data and use machine learning to forecast the results of the 2021 Grand Prix Championship. It should further the rapidly expanding field of motorsport forecasting and facilitate a clearer comprehension of how machine learning may be used to forecast sporting outcomes. This will be achieved using both artificial neural networks for regression and a multiple linear regression. Initial feature selection is carried out using the linear regression model and the learnings are then applied to the neural network. The model's performance metrics are compared, and the results show that with an R^2 of 96%, the neural network fared better than the linear regression model overall. However, the accuracy fluctuates as the grid positions of the drivers are ranked. When comparing the results of the top 10 grid positions, the regression model fared better than the neural network.

2 Introduction

The world of motorcar racing has become one of the most data driven industries in competitive sports today. Formula One (F1) happens to be at the pinnacle of single-seat car racing. The premise is simple, the drivers race around a track against other drivers and the fastest wins. There are ten constructor teams with two drivers each. Of course, there are many rules and regulations surrounding a Grand Prix but at its core, drivers and their constructor teams are battling it out on the racetrack to be crowned both driver and constructor champion for that season. This project aims to use both regression with Artificial Neural Networks (ANN) and multiple linear regression to predict a driver's final ranking in the 2021 Formula One Grand Prix through publicly available data. Throughout the literature review, neural networks have become an increasingly popular method in sports prediction. However, the classic regression techniques are still one of the most popular techniques, used for their simplicity and interpretability. Using both will allow for a comparison of the two to determine which is more suited for F1 result prediction.

2.1 Motivation

F1 is a data rich sport. A car produces 1,500 data points across 120 sensors in a single race (Kavishwara, 2021). However, there is little published research that combines both Machine Learning (ML) and F1 result prediction. While the area of motorsport is layered with complexities and unpredictability, there is noticeably more literature on National Association for Stock Car Auto Racing (NASCAR) and result prediction available. The learning from that research can be applied here as well as the abundance of online blogs and forums where F1 fans use analytics for race results prediction. This research will add to the growing area of predictive analytics with motorsport using supervised ML.

2.2 Variables

F1 racing is known for its unpredictable nature. The outcome of a race can be affected by a variety of factors such as weather conditions, engine health, driver mentality, constructor budget, among others. With this in mind, using data that is somewhat consistent is required. A thorough investigation of any data outliers is essential, not just in terms of data points but in the changing of regulations throughout the years. For example, from 2010 refilling of gas at a pit stop was banned due to safety regulations. This means the time it takes to complete a pit stop dramatically reduced. Constructor teams have changed names frequently throughout the years as well as the number of races that are held every season has increased since 2010. For these reasons, the data used in this project will contain years from 2010 to 2021 inclusively.

2.3 Research Question and Research Objectives

The research question addressed in this study is, *"What supervised machine learning algorithm that can best predict the driver rankings of a Formula One Grand Prix Championship and additionally how does regression using artificial neural networks compare against multiple linear regression for sports result prediction?"*

To answer said question, the below objectives will be examined:

- Examine how much significance each feature has to the regression model to reduce the number of redundant features used in the algorithms
- Evaluate comparable performance measures from both models to help determine the results and models integrity

2.4 Contribution

The motorsport regression models should contribute to the sport result prediction research field where the emphasis is to use open-sourced data. The research will contain reproducible coding that will benefit not just the academic community but also those who are interested in predictive analytics in combination with F1 racing sport result prediction.

2.5 Structure of Paper

The structure of the paper continues with the literature review. It focuses on ML and result prediction in not just on motorsports but a variety of different team and solo sports. Section 4 details the methodology and framework used in this research and how the models were developed and trained. Section 5, 6, and 7 describe the design specification and implementation of the models. Section 8 and 9 summaries the results of the models and how the models are compared to determine which is best to predict the ranking of the F1 2021 Grand Prix. The concluding section, section 10, will reiterate the aim of this paper, conclude the findings and evaluations, and end with a piece of the future of this work.

3 Related Work

Predictive analytics in conjunction with F1, have a limited number of published papers. For this reason, this literature review will go back as far 20 years to get a fully comprehensive review of sport result prediction. As mentioned, ML and NASCAR holds a larger number of published materials over F1. While there are differences between the two motorsports, they have common traits. Such as, both determine winners by drivers who complete the race the fastest. Grid placements are determined by a type of qualifying race, and both use an accumulator to establish the champion and driver rankings at the end of the racing season (Read, 2014). As such, taking the learnings from research within NASCAR can be applied to this research paper.

3.1 Machine Learning and Motorsport

The combination of ML and motorsport leads to research carried out by Graves, Shane, and Fitzgerald (2003), who use a probability model to predict the finishing driver positions of a NASCAR race. Bradley-Terry and Luce and Stern's models are combined, and order data is ranked to determine the driver's final position in the race. What is interesting about this model is that it can also predict how successful the driver will be in future races based on their ability in past races. Predicting the finishing positions within the race works by drivers dropping out as the race proceeds, leaving only the more successful ones to place near the top in this type of backwards ranking system. This thinking can also apply to F1 since as the season progresses, the best drivers are usually place near to the top of the rankings.

The same topic of driver ranking in NASCAR is explored in Pfitzner and Rishel (2005) paper. The authors use a correlation analysis of just 14 races from the 2003 season. The results show that there are numerous features that have a moderate to strong relationship with the target variable. For example, the final driver position had a positive correlation with the speed of car in the race and in qualifying. Another finding shows that there is a relationship between the car and driver combined with winning streaks. While this research did not find this correlation, there is a link with driver and circuit name in relation to drivers final standing position. A weakness of this study is the use of a small sample of 14 races out of a possible 38 which could deem the results inconclusive. An expansion of this work lead to Allender (2011) using an empirical model on 38 races of the 2002 season. The aim is to identify what features are most important in relation to driver performance in NASCR. They found through a regression model that on top of the other features used, only one interaction term is required. This is not the case for this research where three interaction terms were found to be most significant to the model performance. However, like Pfitzner and Rishel, the dataset is small and limited in terms of number of features used.

Continuing to investigate the correlation between features, Silva and Silva (2010) use Spearman's rank correlation coefficients in conjunction with chi-square tests of independence on 2009 NASCAR and F1 data. The driver's finishing positions with driver performance from practice, qualifying, and past races shows positive correlations with the NASCAR data. Only the driver qualifying, and past race performance has a positive correlation with finishing position for the F1 data. This research also shows that qualifying times has a positive impact on the driver's final podium position of a race. While this research does focus on a driver's individual performance and does not necessarily delve into the connections between NASCAR and F1 data, still, it is encouraging to see F1 data utilised in ML.

3.2 In-Event Decision Making and Motorsport

In-event decisions using ML is a widespread practice for motor racing teams. Data centres churn out massive amount of data to help the strategic teams make decisions on pit stop and tyre strategy, whether to push to overtake, or when to preserve energy, to name but a few (Choo, 2015). Nonetheless it is difficult to find published articles on this topic. This could be due to the availability of the data and how valuable and restricted it is to constructor teams. However, a research paper by Tulabandhula and Rudin (2014) predicts when the driver will change positions in NASCAR race during the period between tyre changes. Multiple ML algorithms are used across one hundred features that includes current position of the driver, position of the driver in previous races, and statistics on how many tyres were changed in each pit stop. The Support Vector Machine (SVM) and Least Absolute Shrinkage and Selection Operator

(LASSO) regression gave the highest R^2 ranging from 0.4 to 0.5. The paper suggests that a better R^2 may not be possible in their model because of the complexities around motorsport.
Heilmeier et al. (2020) does make use of publicly available F1 data to find the best pit stop strategy by using ML and simulation. There are many controls around the type of trye, minimum number of changes required, and number of tyres available to a team. The paper discusses the development of a Virtual Strategy Engineer (VSE) built using two ANN's with data from 2014 to 2019. The first neural network is a feedforward which identifies if a driver should make a pit stop or not within that lap. A recurrent neural network is then used to determine which of the available tyres should be put on the car. Using the real data generated from the races by a Monty Carlo simulation, it will generate random events, then store the data until it is loaded at the beginning of each simulation. This paper gave favourable results when run on an example race and the use of two neural networks elevated the simulation which is impressive given it is run using publicly available data.

3.3 Neural Networks and Sport Prediction

There has been an increase in the use of ANN models to predict sports-based results as the popularity of Neural Networks (NN) grows over time. One of the earliest studies in this literature review by Purucker (1996), uses data from the National Football League (NFL) to compare statistical categories using NN. The aim is to predict the winners of NFL games by using both supervised and unsupervised NN algorithms such as Adaptive Resonance Theory (ART), Kohonen Self- Organizing Map (SOM), and Back-Propagation (BP). The author deems BP as the most successful model. However, an issue with this paper is that even though BP positively identified the winners of 11 out of 14 NFL games, the number of features used are modest. The dataset contains results from eight rounds with only five features. More data is likely required to make a definitive statement about BP.
Kahn (2003) uses data from 208 games from the 2003 NFL season. Still, the number of features used is still low at just five. A BP ANN model is used to classify whether a home team will win or lose while playing away. By using a Multi-Layer Perceptron Neural Network (MLP) of 10-3-2 the author achieves an accuracy of 75%. This result is a considerably higher accuracy in comparison to NFL experts who in 2009 on ESPN, achieved an accuracy of 67% (Blaikie et al., 2011). One pitfall of this study is that no other methods were used which may have resulted in a higher accuracy or model performance. NN, Naïve Bayes, and LogitBoost are just some of the methods that Hucaljuk and Rakipović (2011) uses to predict the result of The Champions League football matches. When reducing the number of features the final model uses, the authors method could be deemed somewhat naïve. By using personal domain knowledge, the authors themselves chose what they believe are the best predictors for the model and achieve an accuracy of 60% with NN. While domain knowledge is commonly used as a dimension reduction technique, it can be somewhat time consuming and too specific to a certain dataset (Umayaparvathi and Iyakutti, 2017).
Continuing with ANN for sport result prediction, McCabe and Trevathan (2008) analyse results from multiple football and rugby competitions. A MLP with an ANN structure of 20-10-2 is trained using conjugative-gradient algorithms and BP. Just like F1 and NASCAR, there are differences between these sports, but the authors can use features that all the named sports have in common. The final model reached an overall accuracy of 67.5%. It is encouraging to see that the model is transferrable between different sports. One pitfall of the research is that upon inspection of the features uses, they seem to have a high level of dependency on one another. There could be presence of multicollinearity which is not referenced in the report. Focusing on

individual or solo competitions, Davoodi and Khanteymoori (2010) uses ANN to predict the finishing times of each horse and jokey in a horse race. Over a range of different models, again BP achieves the best accuracy with an average 77%. What is interesting is that some features used are like those in F1 such as track condition and race distance. The learning here can be applied to the ANN used for this research with encouraging results.

3.4 Neural Networks Versus Classical Models in Sport Prediction

Even with the rise of ANN for predictions, statistical regression models still have their place due to their clarity and reliability (Smith and Mason, 2010). A similar objective to this research comes from Maszczyk et al. (2014) who aims to compare the performance of NN and regression models in sport result prediction. The dataset contains information of 70 javelin throwers to help predict how far each athlete can throw the javelin. Starting with a correlation matrix along with a regression analysis, four significant features were found. A non-linear regression resulted in an absolute error of 29.45 meters. However, a MLP neural network with a structure of 4-3-1 achieved an absolute error of 16.77 meters. Therefore, the author inferred that the NN was the superior model. Still, the NN lessens in predictive power as the javelin travels further. This could be the result of an imbalance in the data that was not outlined in the paper. Using MLP and regression models, Edelmann-Nusser, Hohmann and Henneberg (2002) assessed the time taken for an elite female swimmer to finish a backstroke swim. Starting with a regression analysis, the model shows how statistically significant features can affect the timings of completing the backstroke. Still, just as Maszczyk et al. found, the MLP fared better than the regression model with a prediction error of just 0.05 seconds. Just as in this report, the use of a regression model first to and then aid the NN model has seen success.

Again, a MLP is used to predict the results of a European Premier League football match but is then compared against the results of a SVM, gaussian Naïve Bayes, and random forest. Rudrapal et al. (2020) analyses over 11,400 match statistics between 2000 and 2016 and includes 40 features, as well as the location of the match. In terms of accuracy and F1 scores, the MLP model scored highest with an accuracy of 73.5%. While this is the largest dataset used in this review so far, there is no evidence to suggest all the features were required and benefitted the model. A learning from this literature review is the importance of feature selection. A data-driven feature selection technique combined with expert domain knowledge, seems to perform well with Bayesian Networks as one of the more popular techniques in elite sport performance. In a study by Richter, O'Reilly, and Delahunt (2021), the authors say that the features selected could almost be as important as the size of the data.

3.5 Betting and Sport Prediction

Sports analysts and researchers have continually studied prediction and forecasting models in order to determine what odds bookmakers will offer. Using both internal and publicly available data, betting companies engineer profitable betting odds on sports or sporting figures. It has been proposed that a hybrid model based on publicly available data and betting odds data could perform better or the same than just betting odds data alone (Tax and Joustra, 2015). The authors use 13 seasons of data from a Dutch football competition, Dutch Eredivisie, to predict the final result of each match. The features are selected by a combination of domain knowledge and Principal Component Analysis (PCA). However, the highest accuracy of 55.3% was achieved on the lone betting dataset with a Fuzzy Unordered Rule Induction Algorithm (FURIA) classifier. Still, the authors argue that the higher result is not statistically significant against the ANN model which used both datasets. This paper presents an important step in using publicly available data for sports prediction. For in-play betting on the Professional Golfers' Association tour (PGA), study by Wiseman (2016) attempts to predict the winning score after the first round of golf is played by using linear regression and feature selection.

Along with linear regression models, decision trees, and NN were also modelled. In contrast to the findings above, the best predictors were the regression models and not NN. Even though this research into F1 result prediction is not motivated by the betting industry, Wiseman's findings that linear regression can be just as effective as NN is in line with the findings of in this paper.

4 Research Methodology

To execute a comprehensive and sound machine learning research project, it is essential that a proven industry framework is chosen as a project guide. The chosen framework for this research project is the Sport Result Prediction Cross Industry Standard Process for Data Mining otherwise known as SPR-CRISP-DM, see Figure 1. It is largely based on the classic cross-industry standard process for data mining (CRISP-DM) but is solely focused on sport result prediction (Bunker and Thabtah, 2019).

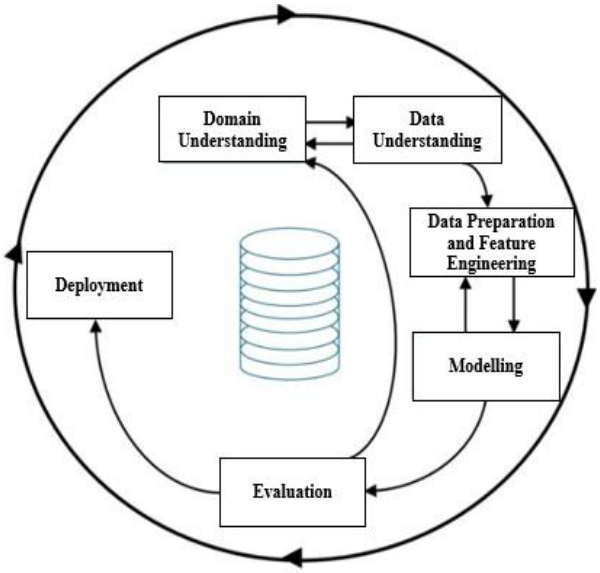

Figure 1 - Phases of SPR-CRISP-DM adapted from Chapman et al. (2000)

4.1 Domain Understanding

The domain understanding phase of this project is covered in the Related Work section of this report. While this research is invaluable, a prior understanding of the sport in relation to rules, regulations, and statistics is also beneficial. This is to better understand the projects problem statement and objectives. There are multiple ways to gain this type of knowledge such as watching the Grand Prix live and availing of the numerous motorsport documentaries that are available online which can give an insight into the history of the sport.[1] [2] F1 has a rich resource of blogs where not just fans, but former drivers give their insights into the season proceedings.[3] Lastly, the F1 official website holds all historical and current race results, as well as future Grand Prix timelines.[4] It is important to note that the 2020 season of F1 fell during the COVID-19 pandemic and races from March to June were cancelled leaving just 17 Grand Prix over the recent average of 21.

[1] *Drive To Survive*, online film recording, Netflix, <https://www.netflix.com/search?q=drive%20to%20survive>
[2] https://f1tv.formula1.com/page/5351/legends-of-f1
[3] https://www.youtube.com/channel/UCtLZ6qQgB-EwQy5HWIo3X-w
[4] https://www.formula1.com/

4.2 Data Understanding

The final dataset used for this research project is collected from two sources, the open source Ergast Developer API and Wikipedia. The Ergast API contains historical race data from 1950 to 2021 retrieved from the F1 website. Wikipedia is used to get the weather information race days. All the data was collected using Python. The data from the Ergast API is in the form of five different datasets. Both a data dictionary and Entity Relationship Diagram (ERD) are used to better understand how the datasets could be joined together in Python to create the final dataset. The dictionary allows full transparency over the type of data that is being collected which is integral to code reproduction for an end user. It also removes any ambiguity as it can be used as a centralised repository for the data (Rashid *et al.*, 2020). This is important for this project as newer results data can be retrieved and stored from the Python code. An ERD is derived from the data dictionary. This type of diagram is a commonly used technique to establish data structures and create data designs (Watt and Nelson, 2014). Figure 2 is a visual of the relationship between the six main datasets and their attributes. The main table is the race results where all other tables are linked. The features *season* and *round* are common to all datasets. Using these two features and one additional feature means a primary key can be created to link the table to the main *race_results* table.

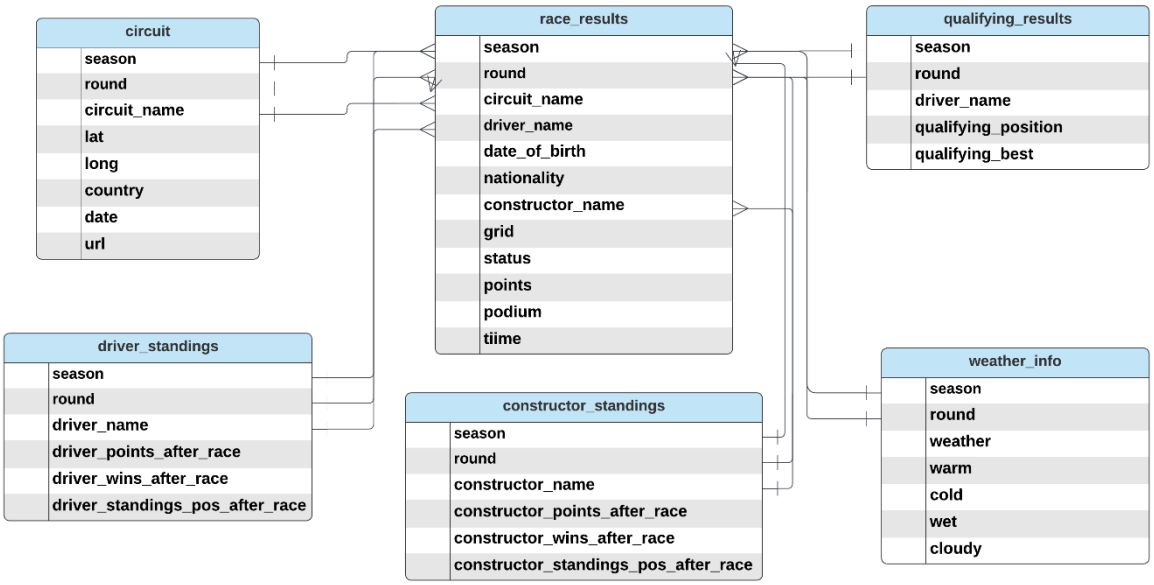

Figure 2 - Entity Relationship Diagram

4.2.1 Data Exploration

Exploratory Data Analysis (EDA) is one of the most fundamental steps in the data understanding phase. To ensure that the most relevant and useful information is gathered from EDA, the problem statement should be at the heart of the analysis. This project aims to understand what features best describe the most amount of variance in the dataset to reduce the number of redundant features used in the algorithms. It is vital that one does not shape the EDA to suit a biased outcome. The data should be fairly represented and not contorted in a way that aims to prove in favour of the expected outcome (Martinez, Martinez, and Solka, 2017). Figure 3 is a correlation plot of all the quantitative variables in the dataset. It is a useful plot to help identify any multicollinearity issues that may be present between the features. The Pearson Product Moment correlation coefficient are between -1 and 1. The coefficient 0.9 suggests a strong association, a weak correlation is around 0.2, and 0 suggest no association between the

two variables (Sullivan, 2016). The highest correlation can be seen between *constructor_points_after_race* and *driver_points_after_race* with a coefficient of 1. This means these two variables have a perfect positive correlation which would make sense as the two are dependent on one another. This strong association can be seen across all the constructor and driver data recorded after the race. This may cause an issue with multicollinearity when modelling the data. The next highest correlation can be seen between the variables *podium* and *qualifying_position*. This suggests that the finished position from the qualifying has a direct link with the finishing position of the race. However, it should be noted that this correlation is only at 0.6 so this relationship would most likely not cause problems in a regression analysis (Hastie, Tibshirani and Friedman, 2001). The weather data does not have any correlation which may suggest the variables are redundant, nevertheless, further analysis is required.

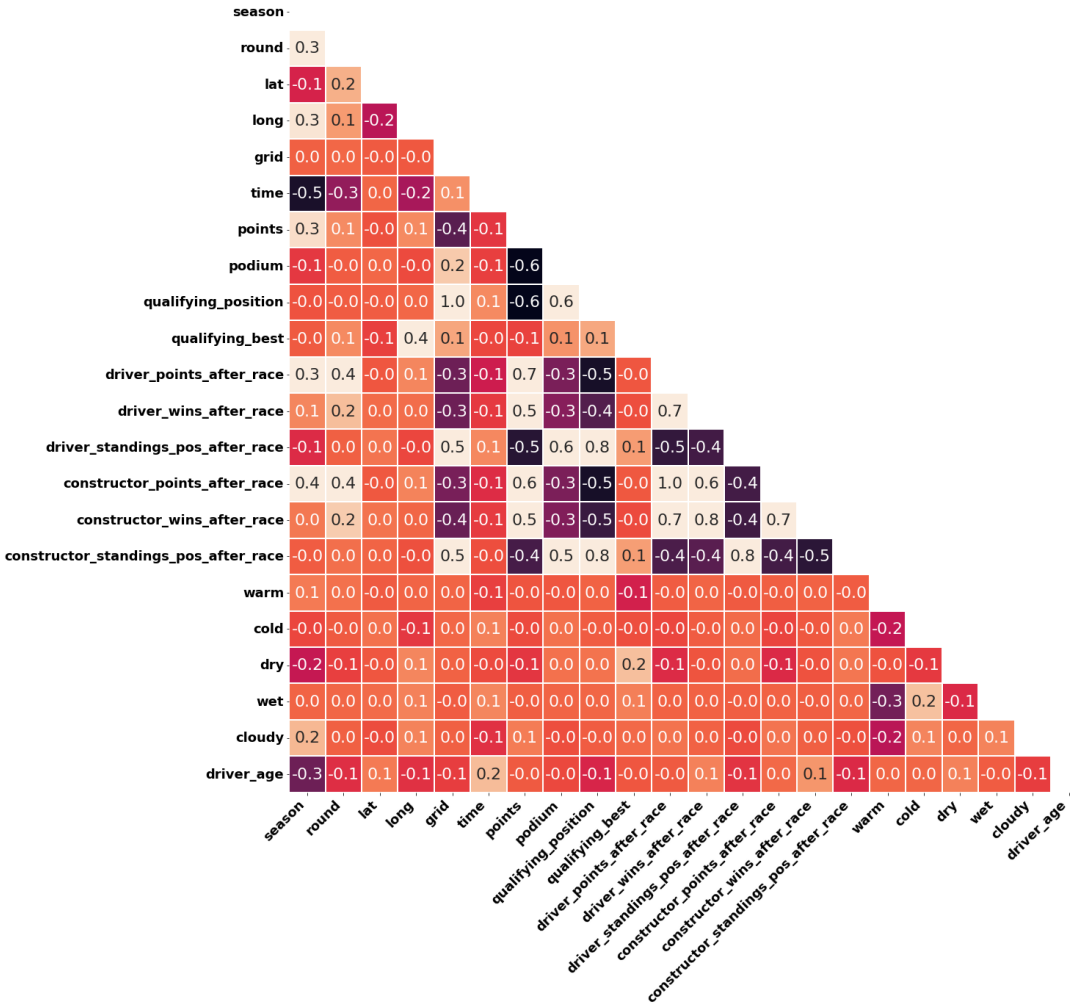

Figure 3 - Correlation plot

Figure 4 is a series of histogram where a distribution of the data points can be examined on the quantitative features. *qualifying_best*, *driver_age*, and *time* seem to follow a normal distribution. All others are skewing to the right. This could be due to outliers in the data that pull the distribution in a positive direction. Still, the number of cases in these non-normal distributions is greater than 30, meaning the central limit theory applies. Even though the normality assumption for regression is violated, the large sample size of over 5,000 observations allows for the application of a sound linear model (Andersson and Olofsson, 2012).

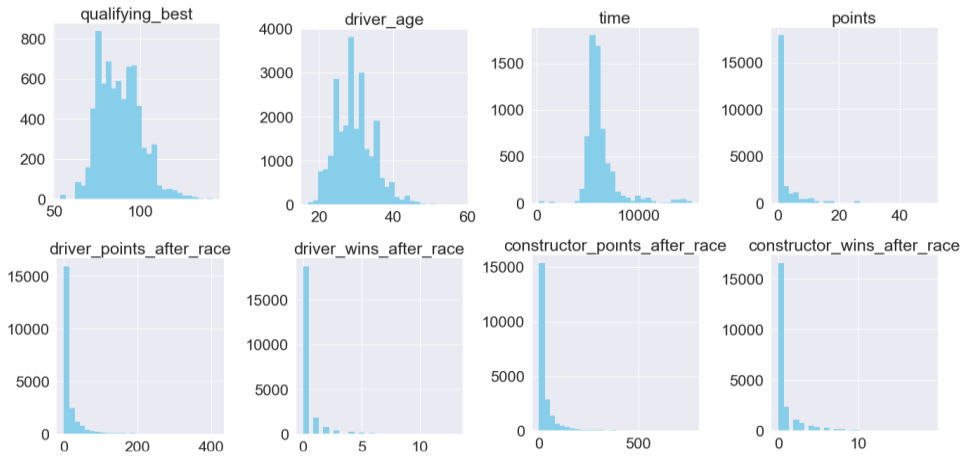

Figure 4 – Histogram of qualitative features

Figure 5 is a bar chart that shows the accumulation of points earned from1950 to 2021. From 2010 onwards, the number of awarded points has dramatically increased. This is due to a change in the points system proposed by the F1 Commission. The points for 1st place increased from 10 to 25. 2nd place increased from 8 to 18 and 3rd from 6 to 15. The awarded points from 4th to 10th also increased. The Fédération Internationale de l'Automobile, believed that the extended gap of points between the positions would create a more ruthless racing attitude. This way of thinking is supported by Judde, Booth and Brooks, (2013) whose paper on how regulation change has had significant impact on championship uncertainty.

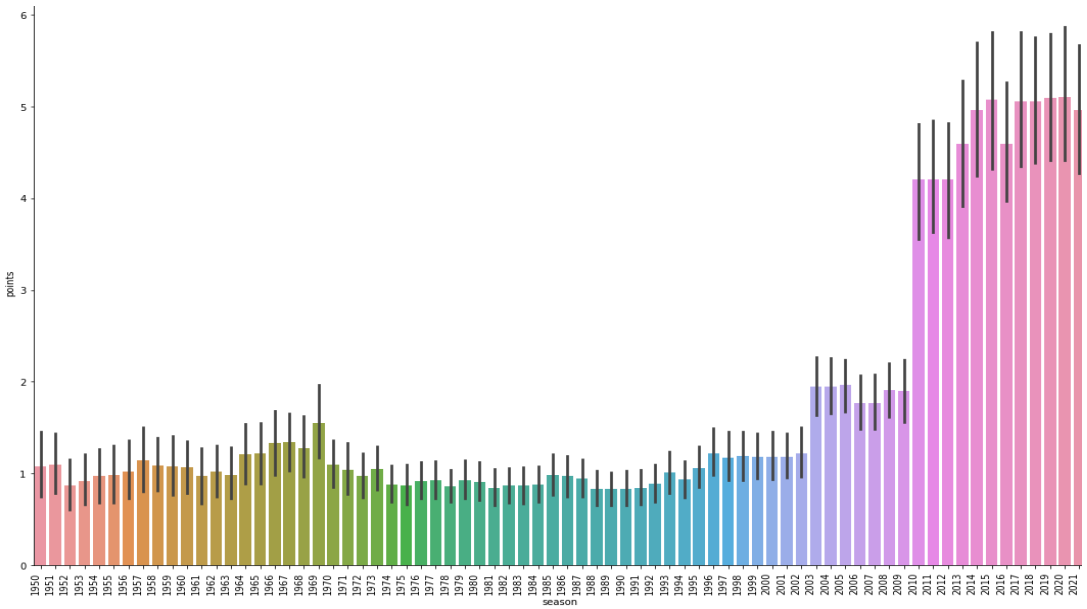

Figure 5 - Accumulated points per season

4.3 Data Preparation & Feature Extraction

Five out of the six data frames were gathered from the Ergast Application Programming Interface (API) using Python. The first data frame which holds the circuit information from 1950 to 2021, is used as the main table that all the other tables will be linked to. The data is gathered from an API query using a GET request. The data is stored with Python in a JSON format and then saved as a Pandas data frame. Using *round* and *year* from the circuits data

frame, the API iterated through these two variables to retrieve the race results data. This process is used on all subsequent data frames. The awarded points for drivers, Verstappen, Hamilton, and Sainz, came through as null for the Grand Prix in Spa in 2011. The nulls were replaced with the correct points retrieved from the F1 website. The third data frame contains the driver standings information but since the points and number of wins are awarded after the race, a lookup function is used to shift the points and wins up one race. The issue of missing data appears again in the 2021 season for the same three drivers, so the correct data was imputed from round 12 to 22.
The fourth data frame is like the driver standings but contains constructor standings data from only 1958 onwards. A lookup function is also used to shift the points and wins for Red Bull, Ferrari, and Mercedes in 2021 for round 12 to 22.
The fifth data frame hold the qualifying results data. This data is only available from 2003. The qualifying sessions have changed over the years. Since 2006 the qualifying takes place on the Saturday where cars battle through 3 knockout rounds. Drivers try to set the fastest lap time in each of the three rounds. The drivers below a certain threshold are knocked out. This is what determines the drivers place on the grid for the race day on Sunday. The data before 2006 only consists of two rounds. Instead of using the race time from the two or three rounds in qualifying, a "best" qualifying time is created. The driver's fasted qualifying time is then used as a predictor and other qualifying times are dropped from the data frame.
The sixth data frame contains the weather information for the race day. Weather can have an impact on the race depending on if it is dry or wet, what the pit stop strategy should be and what tyres should be used at what point in the race. Iterating through the Wikipedia link in the first data frame, the weather data is scraped by using Selenium WebDriver and stored in a dictionary. The weather data is mapped into five categories, *dry*, *wet*, *cold*, *warm*, and *cloudy*.
All six datasets were then merged in Python as described in the Data Understanding section using common keys. The final data set holds information from 2010 to 2021 with 5,129 observations and 28 variables. Data from 2010 onwards is chosen is because it holds the least amount of variation of rules and regulations for drivers and constructor teams.

4.3.1 Natural Language Processing

Before the weather data could be stored in a dictionary and mapped into categories, the column with the raw string data went through a series of cleaning:

- Removing punctuation and links
- Splitting the string into words and tokens – tokenising
- Stop words removed using the NLTK library
- Words were stripped back to root form – lemmatising (Shah, 2020).

The results consisted of a dictionary where a lambda function could be applied to categorise the data into the five buckets described earlier and create new columns. The five features were tagged with 0 or 1 where 1 represented the presence of that weather type.

4.3.2 Further Cleaning

- The *time* variable is converted to milliseconds to match the qualifier time
- The age of the driver for each round in each year is calculated by using the "dateutil.relativedelta" package and using the *date* and *date_of_birth* variables
- Some constructor names changed over the years. The old names are replaced with their most recent names from 2021 using a function. For example, the old names of Force India and Lotus F1 were replaced with Aston Martin (Fairman, 2021)
- The *status* column contains 71 distinct values for the reasons why a driver did not finish a race. These were grouped into broader categories which reduced to 33 statuses
- For NN any factors were converted into numerical. This is not required for MLR

4.3.3 Null Values

The variable *time* holds 52% null values due to how the data is recorded. If a driver finishes more than one lap more than the winner, the time is recorded as "+1 Lap" or "+2 Laps" etc… The actual finishing time is not available in a time format, so an estimate of the time is needed. The number of additional laps is multiplied by the best qualifying time, since the qualifying time represents the how long it takes for that driver to do one lap of the circuit. This is then added to the winners completed time. This steps below outline this process:

- Extract the drivers who finished +1lap or more from each season and each Grand Prix and create a copy data frame named *split_df_plus_laps*
- Extract the number from the string and save as a numeric variable named *num_laps*
- Create a variable where the number of laps is multiplied by the best qualifying time named *add_time*
- Create a data frame with the winners of each season and round named winners_df
- Left join *split_df_plus_laps* to the winners_df on *season* and *circuit_name*
- Create a new variable *new_time* that adds together *add_time* to *time* from winners_df
- Add the *new_time* variable to the original data frame and drop the old "time" column

The % of null vales in the *new_time* column reduced to 19%. The reason for the remaining nulls is due to the driver not finishing a race. The reasons are recorded in the *status* column such as engine failure or accident. The nulls are changed to 9,999 milliseconds. The number chosen must be high to represent that the driver did not finish the race. Other null values were replaced with the median of their group. These represented less than 1% of the data.

5 Design Specification

Reproducible will be possible through data automation. The configuration manual attached to this project will outline in detail how this is possible. Figure 6 is a representation of the three main tiers makeup of the project design, the client tier, the application tier, and the data presentation tier. The data is collected using Spyder Python in Anaconda on a local machine. The files are in JSON format and stored locally in CSV format. However, storing locally is not necessarily required, it is possible to run the code and collect the information through the APIs and save in Python as a Pandas data frame. Any transforming, cleaning, and exploring were initially carried out in Python. EDA is explored in both Python and RStudio. RStudio is used to run the models and store the results, see Figure 6.

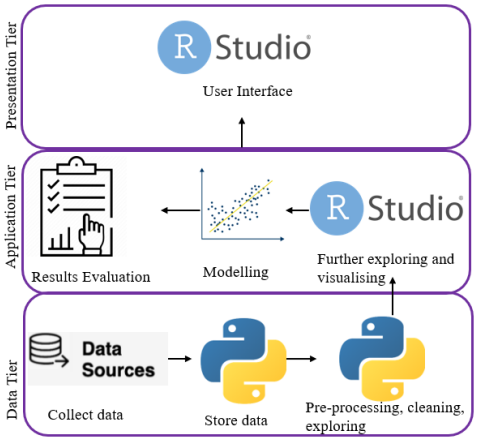

Figure 6 – Process flow chart

6 Implementing a Multiple Linear Regression Model

Both the MLR and ANN were built and ran using R Studio. The switch to R programming from Python is more of a personal preference than a belief that R programming is superior to Python in relation to predictive analysis. Sudhaka (2018) has concluded in their work regarding their differences and similarities between the two, that they can work collectively when trying to achieve a common goal. The MLR is built using the lm() function. This function fits a line using the minimising least squares method by estimating the intercept and the slope coefficients (James, G., Witten, D., Hastie, T., Tibshirani, 2013). To achieve a model fit for purpose, several assumptions need to be met:

- The dependant and independent variables should have a linear relationship
- Homoscedasticity is present - equal variances of residuals
- Distribution of errors is normal
- Outliers should not influence analysis
- No multicollinearity is present - predictors are not highly correlated
- Is independent of errors - no relationship between the residuals and the variable

The first model uses all available features along with the interaction of those variables. The interaction is the combination of two or more independent variables. The interaction effect is when this combination has a greater effect on the target variable as opposed to the individual variables on their own. It explores how independent variables interact with each other and how that effects the dependant variable. For example, the combination of *driver_wins_after_race* and *constructor_wins_after_race* may have a larger effect on the target variable of *driver_standings_pos_after_race* rather than just one of the variables on their own (Frost, 2017). model_1_mlr, while having a favourable result of an adjusted R^2 (correct goodness of fit) of 0.95, is quite long and difficult to interpret. The strategy for this MLR model is to get the best accuracy measure while being understandable and useable to an end user. The second model ran the features but with no interaction. The adjusted R^2 of .90 is a good result but when reviewing the model results, not all variables were significant and so did not add any benefit to the model. Focusing on the p-values, variables were removed that had a value greater than 5%. The first variables to be removed in the third model were longitude and latitude. This information is already captured in the circuit and country information. model_3_mlr had the same adjusted R^2 of model_2_mlr. *circuit_name* and country had a p value greater than 5% but if these were removed too, the would be no indication of the location. So, for model_4 an interaction is created between circuit and country. This led to an adjusted R^2 of 0.90, similar to the previous two models but now *circuit_name:country* has a p-value of 6%. While still higher than the cut off of 5%, the next model focused on those variables with a higher p-value. *circuit_name* and *driver_name* is combined for model_5_mlr which resulted in a lower p-value of less than 5% and an adjusted R^2 of 0.90.

While p-values are a strong indication if a feature is contributing to a model, checking the Variance Inflation Factors (VIF) will help build a model with little to no multicollinearity. A VIF of 5 to 10 indicates that there is a problem with multicollinearity. Using the R library "olsrr", *driver_points_after_race* and *constructor_points_after_race* is found to have a VIF of 24 and 31 respectively so will be removed for from model_6_mlr. With the highly corelated variables removed, the adjusted R^2 reduced to 0.86, possible due to the variables containing some independent information that the model deems is necessary to train on and could be possibly combined to reduce correlation (Curtis and Ghosh, 2011). Thus, removing these variables will have a negative impact on the adjusted R^2.

model_7_mlr variable *grid* had a VIF score of 11 and *qualifying_position* had a VIF score of 13. In most cases the grid position will be the same as the driver's *qualifying_position* unless the driver had to take a penalty or some sort. Therefore, the final grid position on race day is would logically have more of an impact on the final position over the final *qualifying_position*. *qualifying_position* is therefore removed for model_7_mlr. The result is an adjusted R^2 of 0.9027. Using the driver's nationality may prove to be troublesome in terms of racial biases so is removed from model_8 (Zliobaite, 2015). Similar result to model_7_mlr of 0.9024 but with bias removed. Running the VIF calculation again on model_8_mlr shows that podium has a score of 9, which could indicate a problem, however after checking the adjusted R^2 results of model_9_mlr, the reduced score of 0.90 shows that it is valuable to the prediction. Reviewing the p-values again showed that the weather characteristics of cold and warm were not significant for the prediction and so were removed for model_10_mlr. The results gave an adjusted R^2 of 0.90. The VIF score were still high for *driver_points_after_race* and *constructor_points_after_race* and were also high for *driver_wins_after_race* and *constructor_wins_after_race*. An interaction between the two sets were created for model_11_mlr. This lowered the adjusted R^2 to 0.89. Going back to model_10_mlr, podium and points were combined due to their high VIF score. With an adjusted R^2 of .90 the multicollinearity is removed but has little effect on the adjusted R^2.

6.1 Testing For Assumptions

A series of tests were applied to model_12_mlr based on the assumptions referenced in the section 6.

- Homoscedasticity and Linearity - Figure 7Figure 7 is a plot of the residuals of the model on the vertical axis and the fitted values on the horizontal axis. This scatter plot is useful for identifying unwanted residuals. These unwanted residuals can lead to homoscedasticity which violates one of the assumptions. This plot can also test for the assumption of linearity. Once there is no clear presence of a pattern of the values, the model is said to be linear (Field, 2018). Figure 7 does not have a clear pattern. The residuals are somewhat symmetrically distributed and are clustering towards the middle of the plot. Overall, this model passes the assumptions of homoscedasticity and linearity.
- Normal Distribution and Outliers – For a normal distribution of errors, the points should follow a straight line. The Quantile – Quantile (Q-Q) plot in Figure 7 has some skewness on the right-hand side which indicates that there is an over-dispersion relative to the normal distribution. This could be due to outliers.

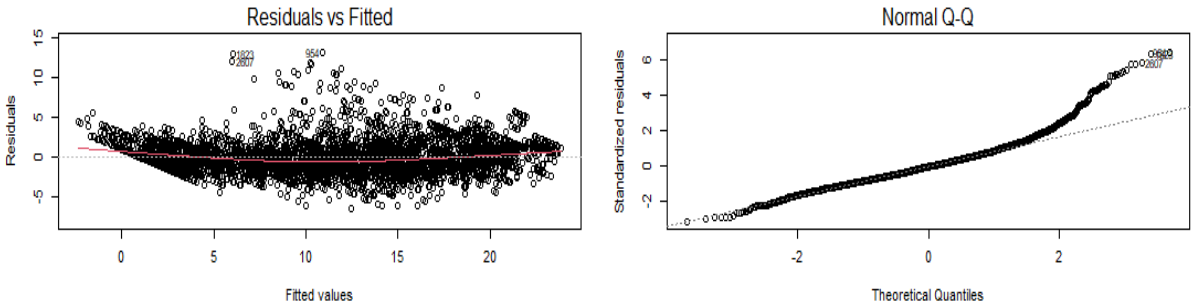

Figure 7 –Residuals vs fitted plot and Q-Q plot of mlr_model_12

Since both plots showed evidence of outliers, these were investigated and removed by using Cook's distance. To identify the influential data points this report, a threshold of 4/(n-k-1) where n is the sample size, k is the number of independent variables is used (Nguyenova, 2020). 226 influential data points were identified by using this method. This represents 5% of the total observations. These were removed and the data run-on model_13_mlr. This resulted in an adjusted R^2 of 0.90. Same as model_12_mlr but with the assumption of normal distribution of errors and outliers passed.

- Independence of Errors- The Durbin-Watson (DW) test was executed using the durbinWatsonTest function form the car package. This tests for autocorrelation in the residuals. The DW statistic should be between 0 and 4 and a value close to 2 indicates independence of errors (Kenton, 2019). model_13_mlr had a result of 1.97 so can infer that this model has passed this assumption.
- Multicollinearity – this assumption is checked through the building of the models. Regarding the high VIF for *constructor_points_after_race* and *driver_points_after_race*, initially the research suggested to leave them in the model. However, running the model with the test dataset proved that removing these two variables increased the accuracy. See Table 1 for the Adjusted R^2 results of each model.

Table 1 - Model results

Model	**Adjusted R^2**	**Model**	**Adjusted R^2**
model_1_mlr	0.9481	model_8_mlr	0.9024
model_2_mlr	0.9033	model_9_mlr	0.8992
model_3_mlr	0.9033	model_10_mlr	0.9025
model_4_mlr	0.9034	model_11_mlr	0.8932
model_5_mlr	0.9033	model_12_mlr	0.9006
model_6_mlr	0.8639	model_13_mlr	0.9001
model_7_mlr	0.9027	model_13_mlr _test	0.9252

Once all the assumptions were checked and corrected, it can be assumed the final model does not violate any principal assumptions for linear regression.

7 Implementing an Artificial Neural Network for Regression

The ANN model is build using R Programming language and Tensorflow. The response variable remained the same as in MLR, *driver_standings_pos_after_race.* The predictors are the same as those used in model_13_mlr but any interactions between variables were removed as they cannot be processed correctly in a neural network model when using the keras library. Figure 8 is a visualisation of the structure of the learning model. It is a first look at a NN model that has hidden layers consisting of 12 neurons in the first layer and 7 neurons in the second layer. The input layer has 17 neurons, and the output has just one neuron. Deciding on the number of hidden layers is usually just a process of trial and error since there is no exact science on the exact right number. Using the neuralnet package allows for deeper understating of the architecture of the model (Khandelwal, 2022).

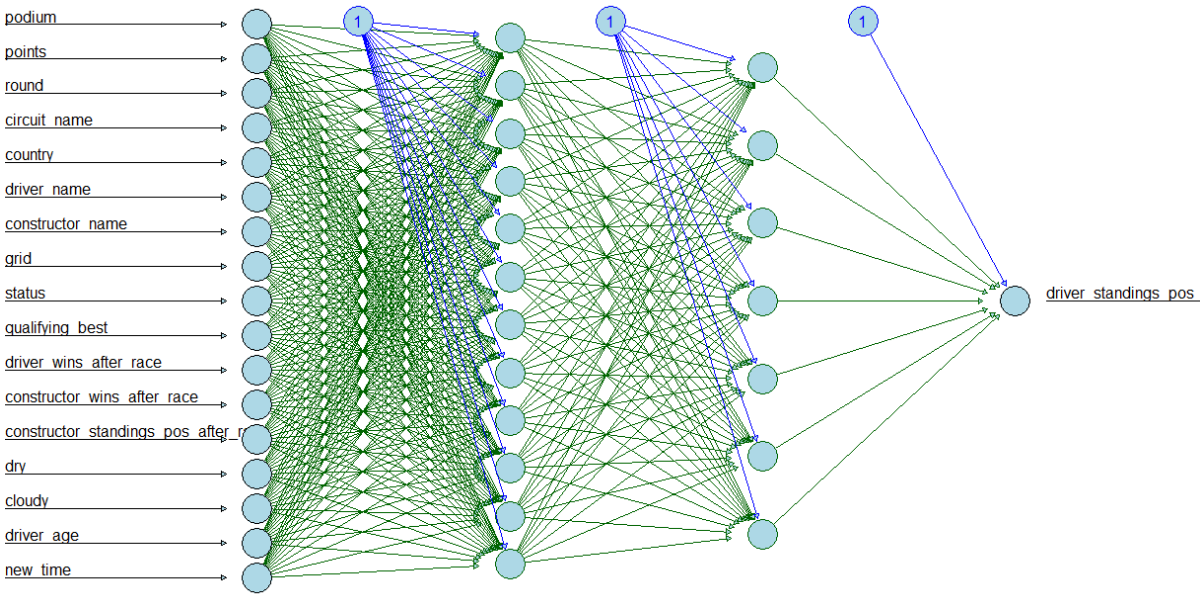

Figure 8 - Visualisation of model architecture

The following several steps prepare the data for the model:

1. Split the data frame into a test and training set. The same test and training sets will be used as for the MLR where the test set is all the data from 2021 and the training set consists of data from 2010 to 2020
2. The data frames are converted to a matrix
3. The training and test set are split further into independent and dependant variables where the independent variable is the *driver_standings_pos_after_race* and the dependant are the 17 variables that can be seen in Figure 8
4. The data requires scaling to a common scale for better prediction. Data that has different ranges can contribute differently to a model. It can most likely lead to bias in the model where one variable has a different weight to another variable. The scale function in R allows for standardisation by subtracting the value of each column in the matrix by the matching mean value and dividing it by the standard deviation. The resulting scale will be a value between zero and one.[5]

The neural network is developed using a layering methodology. The sequential feedforward network is build using keras_model_sequential() with dense layers. model_1_ANN is built with 1 hidden layer, 5 neurons, 17 predictor variables, and 1 output layer. The activation function used is ReLU - rectified linear unit. This function transforms the weighted sum of the input neurons and biases to determine if a neuron can be fired/activated or not (Castaneda, Morris and Khoshgoftaar, 2019). ReLU is one of the most simplistic activations and has become the default function for deep learning (Nair and Hinton, 2010).
model_1_ANN, ran with a mini-batch gradient descent size of 32 and the epoch hyperparameter set to 100. The validation data is set to 20% of the training data. Figure 9 contains four graphs with the loss plotted against the value loss and the Mean Absolute Error (mae) plotted against the value mae for model_1_ANN on the left-hand side and subsequent model_2_ANN on the right. The loss and mae of the model_1_ANN training sets are following

[5] https://www.r-bloggers.com/

a similar trend to the value loss and value mae of the validation set. Initial error is quite high for both sets before drastically decreasing around epoch seven. The final loss is 2.98 with a mae of 1.32 for model_1_ANN. This indicates that overfitting has not occurred. However, to be prudent a dropout rate is added to the next model, so it does not rely too heavily on one neuron.

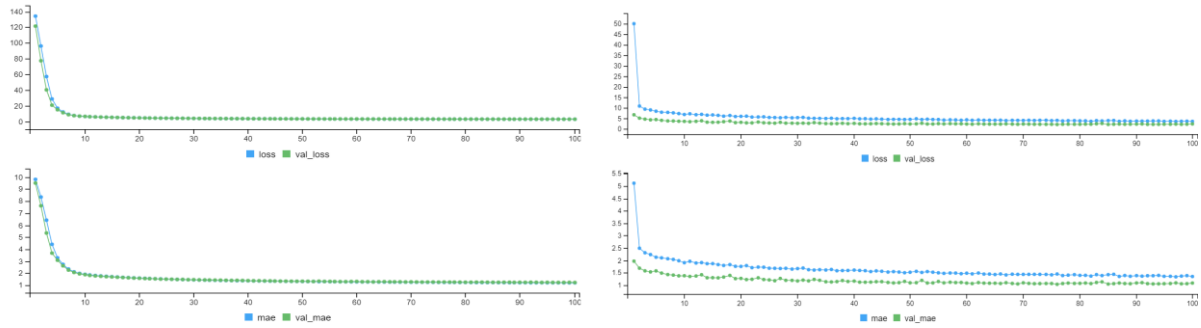

Figure 9 - Error & loss value for model_1_ANN and model_2_ANN

Since the model is sequential, layers can be added linearly. One type of standard layer is a dense layer used in model_1_ANN. Each input node of the dense layer is joined to an output node. A random number of activations from the prior layer is set to zero, this helps reduce overfitting. model_2_ANN has 3 dense layers with 100, 50, and one numeric vector respectively. The dropout has been set to .4 and .2. The right graph in Figure 9 shows that the difference in loss and mae is larger than in model_1_ANN. The pattern of the plots also indicates that there is no overfitting. The final loss for model_2_ANN is 3.22 with a mae of 1.31. It suggests that the first model is a better fit and adding in additional dense layers and dropout layers did not improve the model. The model_1_ANN will be used to validate the results against the test set.

8 Results - Multiple Linear Regression

Model_13_mlr had a R^2 value of 0.93 and adjusted R^2 of 0.93. These results indicate that the 93% of the variability in model_13_mlr _test can be explained suggesting an accuracy of 93%. The F-statistic result is 397.6 on 15 and 444 degrees of freedom. The p-value of less than 0% show that the model's coefficients have a significant contribution to the prediction of a driver's final position in the 2021 championship. The intercept which represents the expected points a driver's standing position after a race when all the model's predictors are held constant. The intercept with a position of 0.07 together with *qualifying_best, driver_wins_after_race, cloudy, new_time, circuit_name:country* and *podium:points* all have a negative effect on the average position of the driver. For example, the *driver_wins_after_race* has an estimate of -0.43 so the less the number of accumulated drivers wins, the lower the final position of a race. Whereas the driver position on the grid has a positive estimate of 0.05 so suggests the higher the position on the grid the more likely the final position will be high too.

To ensure that the model yields an unbiased result, the test dataset is used to predict the driver's standing position of the 2021 Formula One Championship. The Root Mean Square Error (RMSE) of model_13_mlr is 1.57. This means that the model calculates a difference between the predicted driver's position and the actual driver's position of 1.57 positions. The R^2 of 0.93 with the test set is 3 percentage points higher than the training set. So, with the test set, the model can predict the driver's position with 93% accuracy.

Table 2 shows the results of the model when using the test and training set with model_13_mlr. The test set does produce more positive results than the training set which could suggest no overfitting occurred on the training dataset and since the difference is not too large, underfitting is likely not occurred either.

Table 2 - Results Comparison MLR

Metric	Train set	Test set
R^2	0.90	0.93
RMSE	2.04	1.57

Figure 10 is a plot of the predicted driver positions versus the actual driver positions. The plot shows that the model looks to have an even accuracy across the different positions from 1st to 20th. Even though the influential data points were already removed, there does seem to be some outliers in the 10th to 15th positions. However, the data does not seem to be skewed in that direction, so they have no impact on the dependant variable.

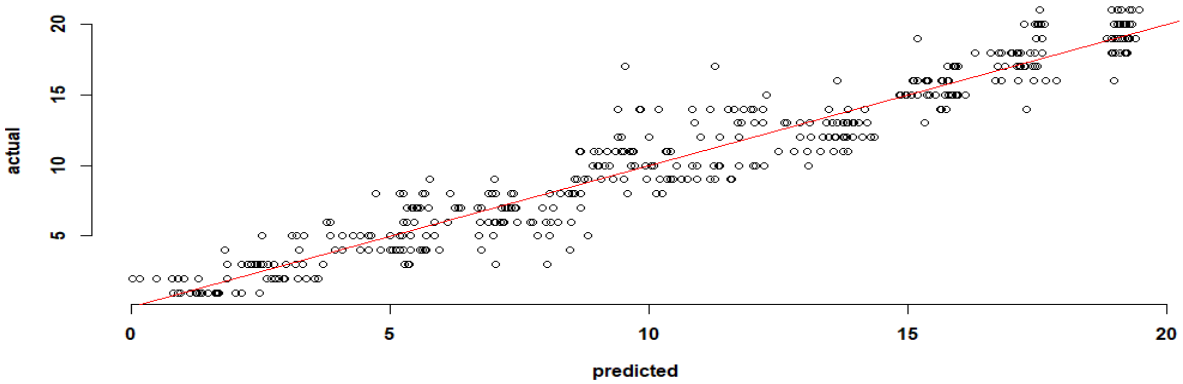

Figure 10 - Predicted versus actual values – MLR

9 Results - Artificial Neural Network for Regression

model_1_ANN had a R^2 value of 0.96 and implies that 96% of the variability in model_1_ANN can be explained. The RMSE of 1.66 shows that the model calculates the predicted values within 1.66 positions. Both the R^2 and the RMSE are higher in the test set than the training set, so overfitting is unlikely to have occurred, but the difference is also not higher enough to indicate underfitting, see Table 3. The R^2 results for ANN are 0.3 percentage points higher in the test than for the MLR model. The RMSE results also fare better.

Table 3 - Results comparison for ANN

Metric	Train set	Test set
R^2	0.94	0.96
RMSE	1.66	1.24

When reviewing the actual versus predicted values of the ANN in Figure 11, the trend is similar to the actuals versus predicted values of the MLR. The ANN model does seem to have less variability and seems to evenly predict each position to the same degree.

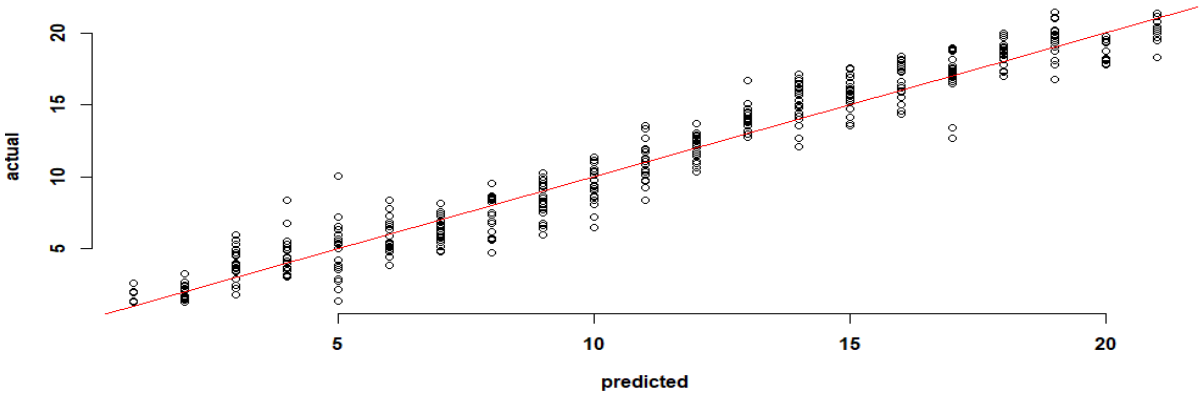

Figure 11 - Predicted versus actual values – ANN

9.1 Model Comparison

Table 4 holds the result of the MLR and ANN model with the actual and predicted results for the top 10 positions. In the 2021 F1 season, Verstappen won the championship in the last race after a battle with Hamilton. The MLR model predicted that Hamilton would win the championship. Going into the very last Grand Prix, Verstappen and Hamilton were on equal points, so the championship would be decided in that race. The MLR model may have predicted Hamilton as the champion due the significance it places on the *constructor_points_after_race feature*. Hamilton's Mercedes-AMG Petronas had a lead over Verstappen's Oracle Red Bull Racing team in the constructor championship. The ANN model correctly predicted Verstappen as the champion and Hamilton in second place. Bottas, Perez, and Sainz were all predicted correctly in the MLR model. While Bottas is correctly placed third in the ANN model, it places Perez in eighth place. This is four places away from the actual position. Perez did not actual finish the last race of season. The ANN model could have placed more weight on the status feature than the MLR model. It is difficult to understand the feature importance of the ANN model as interpreting neural networks is still in its infancy and reviewing feature importance has not been develop in a user-friendly way (Montavon, Samek and Müller, 2018). Evaluation of MLR is more developed so we see that the *new_time* feature is one of the least significant for the model. This could be why Perez is placed correctly in 4th with the MLR model.

Table 4 - Actual versus predicted results

Driver	**Actual Position**	**Predicted Position Ranked - MLR**	**Predicted Position - MLR**	**Predicted Position Ranked- ANN**	**Predicted Position - ANN**
Verstappen	1	2	1.25	1	1.34
Hamilton	2	1	0.02	2	1.50
Bottas	3	3	3.03	3	3.74
Perez	4	4	5.68	8	8.51
Sainz	5	5	5.69	6	6.71
Norris	6	7	7.44	4	5.56
Leclerc	7	6	6.36	7	6.46
Ricciardo	8	8	8.33	9	7.76
Gasly	9	10	10.74	5	6.12
Alonso	10	9	9.72	10	9.08

The MLR model switches positions for Norris and Leclerc as well as Gasly and Alonso. Ricciardo is predicted correctly. The ANN model incorrectly predicts Sainz, Norris, Ricciardo, and Alonso while Leclerc is predicted correctly. While the performance measures are better for the ANN model, for the top 10, the MLR model is more accurate. It is never more than one position out. The ANN struggles to predict from position four to ten. The large discrepancy can also be seen with Gasly where he is placed 5th but came 9th. In last race of the season, Gasly did indeed finish 5th under the *podium* feature but 9th in the overall championship. While it is unclear what weight is given to this feature in ANN, it is only statistically significant in MLR when combined with *points*. It may hold more weight in ANN and therefore pushes Gasly higher up the ranks. Reviewing the remaining 11th to 20th positions, ANN did have a 100% prediction accuracy. The MLR struggles more with the 11th to 20th positions and is out by an average of -0.29 positions.

10 Conclusion and Future Work

Two regression models are used to predict the final driver rankings of the 2021 F1 Championship, MLR and regression using ANN. Data from the official F1 website and weather information from Wikipedia are extracted using Python and the models are run in R Studio. The results show that ANN has the best performance measures in terms of R^2 and RMSE. The top 3 podium finishes are correctly predicted in the ANN model whereas in the MLR model, 1st and 2nd place were inverted. However, when reviewing the top 10 predictions for both models, MLR fared better than ANN. From 4th to 10th the MLR achieves better over accuracy with an error of just one position on average. The ANN model incorrectly placed Perez in 8th place when his actual position was 4th. This large error meant that the remaining positions up until 10th were incorrectly predicted by minimum one and maximum four positions. It can be suggested that the *new_time* feature weighs higher for ANN than MLR and so the actual finishing times for that last race has a bigger impact on the overall championship. The result of ANN having an overall better prediction result than MLR has also been seen in research carried out by (Maszczyk *et al.*, 2014) and (Edelmann-Nusser, Hohmann and Henneberg, 2002). The deciding factor on what is the more suitable model would be what importance one places on the different positions of the leader board. The top 10 drivers for the 2021 season are more accurately predicted by the MLR model and since these are the only positions that yield points for both the driver and constructor, it maybe that MLR is preferred. However, evaluating all positions on the leader board, ANN is a better overall predictor with 100% accuracy on the 11th to 20th position. Interpretability is also an important factor in deciding what model is best to use. If the researcher is most concerned with how to explain and decipher model results, one could argue that the MLR model is simpler to use that ANN. Based on these findings, the research question of what supervised machine learning algorithm that can best predict the outcome of a Grand Prix F1 championship with respect to driver ranking and additionally how does neural networks compare against multiple linear regression for sports result prediction has been address and answered.

The data used in this project can be seen as simplistic, which is probably to the benefit of the model's accuracy. However, to create a more sophisticated model that could be implemented in the motor racing industry, better quality data is required. While it is difficult to get open-source data on F1, this project could benefit from additional information regarding the number of pit stops and time of each pit stop. It is critical piece of information which can determine the final position of a driver. It can be seen in the literature review that in-event decision-making regarding pit stop strategy can affect the outcome of the race. This research could also be extended to other forms of motor racing such as NASCAR and IndyCar since the number of published papers on result prediction with motor racing overall is quite limited.

References

Allender, M. (2011) 'Predicting the Outcome of NASCAR Races: The Role of Driver Experience', *Journal of Business & Economics Research (JBER)*, 6(3), pp. 79–84. Available at: https://doi.org/10.19030/JBER.V6I3.2403.

Andersson, M. and Olofsson, P. (2012) 'Chapter 4: Limit Theorems - Probability, Statistics, and Stochastic Processes, 2nd Edition', in *publisher logo Probability, Statistics, and Stochastic Processes, 2nd Edition*. 2nd edn. New Jersey: John Wiley & Sons, p. Chapter 4. Available at: https://learning.oreilly.com/library/view/Probability,+Statistics,+and+Stochastic+Processes,+2nd+Edition/9781118231326/chapter04.html#c04anchor-3 (Accessed: 7 March 2021).

Bunker, R.P. and Thabtah, F. (2019) 'A machine learning framework for sport result prediction', *Applied Computing and Informatics*, 15(1), pp. 27–33. Available at: https://doi.org/10.1016/j.aci.2017.09.005.

Castaneda, G., Morris, P. and Khoshgoftaar, T.M. (2019) 'Evaluation of maxout activations in deep learning across several big data domains', *Journal of Big Data*, 6(1). Available at: https://doi.org/10.1186/S40537-019-0233-0.

Chapman, P. *et al.* (2000) *CRISP-DM 1.0, CRISP-DM Consortium*. CRISP-DM consortium.

Choo, C.L.W. (2015) *Real-time decision making in motorsports: analytics for improving professional car race strategy*. Massachusetts Institute of Technology. Available at: https://dspace.mit.edu/handle/1721.1/100310 (Accessed: 13 November 2022).

Curtis, S.M.K. and Ghosh, S.K. (2011) 'A Bayesian approach to multicollinearity and the simultaneous selection and clustering of predictors in linear regression', *Journal of Statistical Theory and Practice*, 5(4), pp. 715–735. Available at: https://doi.org/10.1080/15598608.2011.10483741.

Davoodi, E. and Khanteymoori, A. (2010) 'Horse racing prediction using artificial neural networks | Semantic Scholar', in. Available at: https://www.semanticscholar.org/paper/Horse-racing-prediction-using-artificial-neural-Davoodi-Khanteymoori/019119dff662e50678563bb50bccca13d433bab2 (Accessed: 2 March 2022).

Edelmann-Nusser, J., Hohmann, A. and Henneberg, B. (2002) 'Modeling and prediction of competitive performance in swimming upon neural networks', *European Journal of Sport Science*, 2(2), pp. 1–10. Available at: https://doi.org/10.1080/17461390200072201.

Fairman, K. (2021) *These Are All The F1 Team Changes In The Last Decade – WTF1*, *wtf1.com*. Available at: https://wtf1.com/post/these-are-all-the-f1-team-changes-in-the-last-decade/ (Accessed: 23 September 2022).

Field, A.P. (2018) *Discovering statistics using IBM SPSS statistics: 5th edition*, *ProtoView*. Available at: http://library1.nida.ac.th/termpaper6/sd/2554/19755.pdf.

Frost, J. (2017) *Understanding Interaction Effects in Statistics*, *www.statisticsbyjim.com*. Available at: https://statisticsbyjim.com/regression/interaction-effects/ (Accessed: 12 March 2021).

Graves, T., Shane, C. and Fitzgerald, M. (2003) 'Hierarchical models for permutations: Analysis of auto racing results', *Journal of the American Statistical Association*, 98, p. 282.

Hastie, T., Tibshirani, R. and Friedman, J. (2001) *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. 2nd edn, *Springer Series in Statistics*. 2nd edn. Stanford: Springer Series in Statistics. Available at: https://doi.org/10.1007/978-1-4419-9863-7_941.

Heilmeier, A. *et al.* (2020) 'Virtual strategy engineer: Using artificial neural networks for making race strategy decisions in circuit motorsport', *Applied Sciences (Switzerland)*, 10(21), pp. 1–32. Available at: https://doi.org/10.3390/APP10217805.

Hucaljuk, J. and Rakipović, A. (2011) *Predicting football scores using machine learning techniques, 2011 Proceedings of the 34th International Convention MIPRO*.

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013) *An Introduction to Statistical Learning - with Applications in R | Gareth James | Springer*. Available at: https://www.springer.com/gp/book/9781461471370%0Ahttp://www.springer.com/us/book/9781461471370.

Judde, C., Booth, R. and Brooks, R. (2013) 'Second Place Is First of the Losers: An Analysis of Competitive Balance in Formula One', *Journal of Sports Economics*, 14(4). Available at: https://doi.org/10.1177/1527002513496009.

Kahn, J. (2003) *Neural Network Prediction of NFL Football Games*. University of Wisconsin. Available at: https://docplayer.net/21763052-Neural-network-prediction-of-nfl-football-games-joshua-kahn.html (Accessed: 24 February 2022).

Kavishwara, R. (2021) *Formula 1 & Machine Learning. There aren't many things in the… | by Ravindu Kavishwara | Becoming Human: Artificial Intelligence Magazine*, *www.becominghuman.ai*. Available at: https://becominghuman.ai/formula-1-and-machine-learning-62d1f7166c41 (Accessed: 9 April 2022).

Kenton, W. (2019) *Durbin Watson Statistic Definition, www.investopedia.com/*. Available at: https://www.investopedia.com/terms/d/durbin-watson-statistic.asp (Accessed: 13 March 2021).

Khandelwal, R. (2022) *Visualizing Deep Learning Model Architecture | by Renu Khandelwal | AIGuys | Medium*, *www.https://medium.com/*. Available at: https://medium.com/aiguys/visualizing-deep-learning-model-architecture-5c18e057b73e (Accessed: 15 October 2022).

Martinez, W.L., Martinez, A.R. and Solka, J.L. (2017) *Exploratory Data Analysis with MATLAB ® Third Edition*. 3rd edn. Chapman and Hall.

Maszczyk, A. *et al.* (2014) 'Application of Neural and Regression Models in Sports Results Prediction', *Procedia - Social and Behavioral Sciences*, 117, pp. 482–487. Available at: https://doi.org/10.1016/J.SBSPRO.2014.02.249.

McCabe, A. and Trevathan, J. (2008) 'Artificial intelligence in sports prediction', in *Proceedings - International Conference on Information Technology: New Generations, ITNG 2008*, pp. 1194–1197. Available at: https://doi.org/10.1109/ITNG.2008.203.

Montavon, G., Samek, W. and Müller, K.R. (2018) 'Methods for interpreting and understanding deep neural networks', *Digital Signal Processing: A Review Journal*. Elsevier Inc., pp. 1–15. Available at: https://doi.org/10.1016/j.dsp.2017.10.011.

Nair, V., and Hinton, G.E. (2010) 'Rectified Linear Units Improve Restricted Boltzmann Machines', *Proceedings of the 27th international conference on machine learning (ICML-10)* [Preprint].

Nguyenova, L. (2020) *A little closer to Cook's distance | by Ly Nguyenova | Medium*, *www.medium.com/*. Available at: https://medium.com/@lymielynn/a-little-closer-to-cooks-distance-e8cc923a3250 (Accessed: 14 March 2021).

Pfitzner, B. and Rishel, T. (2005) 'Do Reliable Predictors Exist for the Outcomes of NASCAR Races? – The Sport Journal', *The Sports Journal*, 8(2), pp. 1–9. Available at: https://thesportjournal.org/article/do-reliable-predictors-exist-for-the-outcomes-of-nascar-races/ (Accessed: 5 March 2022).

Purucker, M.C. (1996) 'Neural network quarterbacking', *IEEE Potentials*, 15(3), pp. 9–15. Available at: https://doi.org/10.1109/45.535226.

Rashid, Sabbir M *et al.* (2020) 'The Semantic Data Dictionary – An Approach for Describing and Annotating Data', *Data Intelligence*, 2(4), pp. 443–486. Available at: https://doi.org/10.1162/DINT_A_00058.

Read, D. (2014) *F1 vs NASCAR: which is better? | Top Gear*, *topgear.com*. Available at: https://www.topgear.com/car-news/motorsport/f1-vs-nascar-which-better (Accessed: 13 November 2022).

Richter, C., O'Reilly, M. and Delahunt, E. (2021) 'Machine learning in sports science: challenges and opportunities', *Sports Biomechanics*, pp. 1–7. Available at: https://doi.org/10.1080/14763141.2021.1910334.

Rudrapal, D. *et al.* (2020) 'A Deep Learning Approach to Predict Football Match Result', *Advances in Intelligent Systems and Computing*, 990, pp. 93–99. Available at: https://doi.org/10.1007/978-981-13-8676-3_9.

Shah, P. (2020) *Sentiment Analysis using TextBlob*, *www.towardsdatascience.com*. Available at: https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524 (Accessed: 4 April 2021).

Silva, K.M. and Silva, F.J. (2010) *Practice, Qualifying, and Past Success in NASCAR and F1 1 A Tale of Two Motorsports: A Graphical--Statistical Analysis of How Practice, Qualifying, and Past Success Relate to Finish Position in NASCAR and Formula One Racing* *. University of Redlands. Available at: http://newton.uor.edu/FacultyFolder/Silva/NASCARvF1.pdf (Accessed: 31 January 2022).

Smith, A., and Mason, A.K. (2010) 'Cost Estimation Predictive Modeling: Regression Versus Neural Networks', *http://dx.doi.org/10.1080/00137919708903174*, 42(2), pp. 137–161. Available at: https://doi.org/10.1080/00137919708903174.

Sudhaka, K. (2018) 'Python vs. R Programming Language', *International Journal of Management, IT and Engineering*, 8(8), pp. 70–79. Available at: https://www.indianjournals.com/ijor.aspx?target=ijor:ijmie&volume=8&issue=8&article=009 (Accessed: 24 September 2022).

Sullivan, L. (2016) *Correlation and Linear Regression*, *Boston University School of Public Health*. Available at: https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Correlation-Regression/BS704_Correlation-Regression_print.html (Accessed: 7 March 2021).

Tulabandhula, T. and Rudin, C. (2014) 'Tire changes, fresh air, and yellow flags: Challenges in predictive analytics for professional racing', *Big Data*, 2(2), pp. 97–112. Available at: https://doi.org/10.1089/BIG.2014.0018/ASSET/IMAGES/LARGE/FIGURE11.JPEG.

Umayaparvathi, V. and Iyakutti, K. (2017) 'Automated Feature Selection and Churn Prediction using Deep Learning Models', *International Research Journal of Engineering and Technology*, 4(3), pp. 1846–1854. Available at: www.irjet.net.

Watt, A. and Nelson, E. (2014) *Chapter 8 The Entity Relationship Data Model*. BCcampus. Available at: https://opentextbc.ca/dbdesign01/chapter/chapter-8-entity-relationship-model/ (Accessed: 8 December 2021).

Wiseman, O. (2016) *Using Machine Learning to Predict the Winning Score of Professional Golf Events on the PGA Tour*. National College of Ireland. Available at: http://norma.ncirl.ie/2493/ (Accessed: 4 March 2022).

Zliobaite, I. (2015) 'A survey on measuring indirect discrimination in machine learning', *ACM Journal Name*, 0(0). Available at: https://doi.org/10.48550/arxiv.1511.00148.