The Energy University

UNIVERSITI TENAGA NASIONAL

PUTRAJAYA CAMPUS

BACHELOR OF INFORMATION TECHNOLOGY (INFORMATION SYSTEM) (HONS.)

SEMESTER 1 2O22/2023

CISB423:

FINAL YEAR PROJECT 2

Predicting Potential Winner of 2022 FIA Formula One Constructor World Championship

Muhammad Afiq Bin Khushairi
IS0105573

PREDICITING POTENTIAL WINNER OF 2022
FIA FORMULA ONE CNSTRUCTOR WORLD CHAMPIONSHIP

by

MUHAMMAD AFIQ BIN KHUSHAIRI

Project Supervisor: Ts. Dr Aliza Binti Abdul Latif

PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR

THE BACHELOR OF INFORMATION TECHNOLOGY

(INFORMATION SYSTEMS)

(HONS) UNIVERSITI TENAGA NASIONAL

2023

APPROVAL PAGE

TITLE: PREDICITING POTENTIAL WINNER OF 2022 FIA FORMULA ONE CONSTRUCTOR WORLD CHAMPIONSHIP

AUTHOR: MUHAMMAD AFIQ BIN KHUSHAIRI

The undersigned certify that the above candidate has fulfilled the condition of the Final Year Project in partial fulfilment for the Bachelor of Information Technology (Information Systems) (Hons.)

Supervisor: TS. DR. ALIZA BINTI ABDUL LATIF

Signature: ……..

Date: ……..

DECLARATION

I hereby declare that final year project is my original work except for quotations and citations have been duly acknowledged. I also declare that has not been previously and is not concurrently submitted for any degree program at Universiti Tenaga Nasional or at any other institutions. This final year project may be made available within the university library and may be borrowed, consulted, copied or reproduced in accordance with the provision of the UNITEN Library Regulations from time to time made by the Library Committee.

_____

Name: MUHAMMADD AFIQ BIN KHUSHAIRI

Student ID: IS0105573

Date: 15/1/2023

Contents

ACKNOWLEDGEMENT

This project will take a significant amount of time and effort to complete. However, it would not have been possible without the help and support of others. Ts. Dr. Aliza Binti Abdul Latif, my supervisor at UNITEN, deserves special recognition. Her encouragement, constructive criticism, patience with me, and constant supervision ensured the success of my research. I learned a lot from her that I didn't know before. I'd also like to thank my family and friends for their help in getting the job done on time. Without their assistance, I would have been completely lost and unable to move forward with the project due to my lack of background knowledge in this field

CHAPTER 1

INTRODUCTION

1.0 Overview

Chapter 1 includes an overview of the project and a problem statement. Furthermore, the project's objective, scope and expected outcome will be stated. This chapter will also consist of the timeline for this project.

1.1 Background

Formula One is the highest class of international open wheel car racing. A formula One season consists of series of races known as Grand Prix which been held worldwide on multiple circuits. A points system is used at Grands Prix to determine two annual World Championships: one for drivers, the other for constructors. Each driver must hold a valid "Super License" which is the highest class of racing license issued by the FIA. All the races must be run on the tracks graded "1".

Formula One cars are the world's fastest controlled road-course racing cars, owing to their extremely high turning speeds, which are achieved by the development of significant aerodynamic downforce. In 2017, the cars underwent significant upgrades, including wider front and rear wings and wider tyres, resulting in peak cornering forces near 6.5 lateral G-force and top speeds of around 350 km/h. As of 2021, hybrid engines are limited to a maximum speed of 15,000 rpm; the automobiles' performance is determined by electronics, aerodynamics, suspension, and tyres. In 1994, traction control, launch control, automatic

shifting, and other electronic driving aids were prohibited. They were briefly restored in 2001 before being outlawed in 2004 and 2008.

With a yearly operating cost of around US$247 million for designing, building, maintaining, and transporting a team, its financial and political disputes are extensively covered. Liberty Media concluded the $8 billion acquisition of the Formula One Group from private equity firm CVC Capital Partners on 23 January 2017.

1.2 Problem Statement

Since 2017, the competition had been dominated by only one driver and constructor with less challenges from other drivers and constructors. In spite of that in 2021, the championship had been won by a different driver. The competition between two drivers from Mercedes-AMG Petronas and Red Bull Racing to claim the top spot had been high for the season 2021 which had shown potential title challengers for the next season in 2022. Nevertheless, in 2021 Mercedes-AMG Petronas still won the constructor championship.

Between 2014 and 2021, Mercedes-AMG Petronas Formula One Team had dominated the constructor championship by winning 8 times and Mercedes holds the record for the most consecutive constructors' titles. Furthermore, with the major regulation changes in 2022, all top teams such as Red Bull Racing, McLaren Racing and Scuderia Ferrari had been developing the best car to possibly competing and claim the title hold by the Mercedes team on the 2022 season.

1.3 Objectives

1. To predict the potential winner for 2022 constructors' championship

2. To make analysis of constructor's performance from 2019-2022 (after 13 races/round) that have the ability to win the championship 2022.

3. To develop data visualisation to present critical information related to the F1 constructor teams' performance throughout 3 years of the championship.

1.4 Scope

The scope of this project is using dataset from Kaggle.com, the world's largest data science community where you can get many kinds of datasets. The data had been shared by a user name Vopani which all the data being obtain from the formula1.com and being compiled for public views on Kaggle.com. The dataset cover from the year 1950 until 2021 but in this project only will take the data of the championship from 2019-2022(after 13 races/round) in order to make prediction for the 2022 Formula One Championship. The dataset will include the driver performance of each circuit or race, points obtain by drivers and team constructors on each race, driver nationality, driver's age, driver's best qualifying time for each race weekend, starting grid position of each race and driver's race result.

1.5 Expected Outcome

The outcome of this project is to analyse performance for each of top 3 drivers and constructor who competing in the championship in the year 2022 and get the prediction of the potential winner for the 2022 championship that will be held starting from March 2022 till December 2022.

1.6 Project Timeline

The project timeline used for this project are as shown in Table 1.1 as below.

Task Planning	Week															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Supervisor Meeting																
Project Briefing																
Supervisor Meeting																
System Development																
Supervisor Meeting																
Milestone 1 (Progress 1A Presentation)																
System Development																
Supervisor Meeting																
Progress 1B Presentation																
Chapter 1, 2, and 3 Adjustments																
Supervisor Meeting																
Milestone 2 (Progress 2A Presentation)																
Chapter 4, 5, and 6 Documentation																
Supervisor Meeting																
Progress 2B																
Submission of Report and Logbook																
Final Project Presentation																

Table 1.1 Project Timeline

1.7 Chapter Summary

In this chapter, the introduction to this project being explain to give the first idea of what this project is about. The background of the project was written with the problem statement to understand what is needed to be done in the project. From the problem statement, objectives of this project were identified and what were the scope of this project. Finally, the expected outcome and the timeline of the project had been shown in this chapter.

CHAPTER 2

ANALYSIS & SYSTEM REQUIREMENT

2.0 Overview

In this chapter, the purpose of this research is to provide insight into how data analytics may help on making prediction of the potential winner of Formula 1 Championship in 2022. Furthermore, there will be additional information on the current method of analytics approach in this matter.

2.1 Type of Data Analytics

Data is a significant tool that can be used by organisation at a huge scale. When the data being used appropriately, it can help people to make decisions, help organisation to come up with better strategies and make them more productive. Data analytics is the process of looking at data to answer questions, find trends, and get new information. When data analytics is used in a business, it's called "business analytics". People can't get as much and do it as quickly as algorithms and machine learning can, but they can use them to get and analyse more and more data at a faster rate than people can.

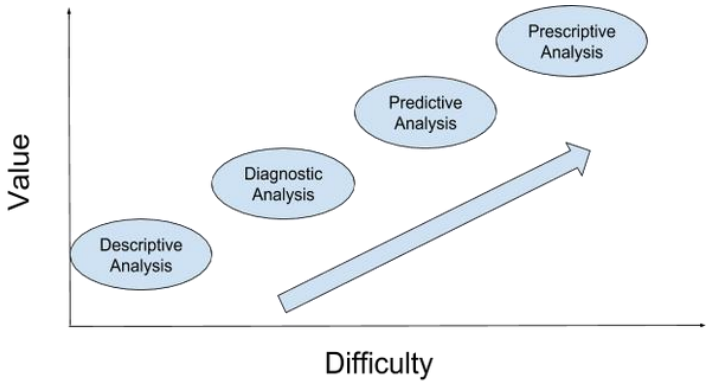

Figure 2.1 Types of data analytics

There are four types of data analytics which is;

I. Descriptive Analytics

- This is the simplest type of analytics and the one that other types are built on. Descriptive analytics is the first type of analytics. It lets you look for trends in raw data and quickly describe what happened or is happening now.
- As an example, a business is analysing its statistics and notices an increase in sales for one of its items. Descriptive analytics may assist in determining which month of the year has the greatest increase in sales.

II. Diagnostic Analytics

- Diagnostic analytics is the process of going one step further with the analysis. There are many types of correlation analysis, including comparing coexisting patterns or movement, identifying correlations between variables, and demonstrating causative correlations when applicable.
- For example, how many patients were admitted to the hospital last month? How many came back in 30 days? After all, some reimbursements are based on readmission rates. Descriptive analytics can quantify occurrences, highlight hospital resource use, and even estimate illness diagnosis rates. Anomalies may be found by comparing the data to historical patterns, and then the search for causative linkages can begin.

III. Predictive Analytics

- Precision analytics is a technique for making educated guesses about future trends or occurrences. By examining past data in conjunction with industry trends, you may make predictions about what the future may hold for your organisation.

- As an example, Predictive analytics may assist financial institutions in forecasting their future health. Through the use of historical financial data and industry data, predictive analytics can forecast sales, profits, and expenses in order to generate a future picture and make decisions.

IV. Prescriptive Analytics

- Prescriptive analytics considers all conceivable variables in a circumstance and makes practical recommendations. This form of analytics is particularly advantageous when it comes to making data-driven judgments.
- For example, through lead scoring, also known as lead ranking, predictive analytics plays a significant role in sales. Lead scoring is the process of giving a point value to different sales funnel activities, allowing you or an algorithm to score leads according to their likelihood to convert into customers.

2.2 How Data Analytics Contribute in Formula One

From NASCAR to Formula One to amateur karting, teams and race organisers are increasingly relying on sophisticated data-driven techniques to take split seconds off lap times and create a more compelling show for fans. Few sports make significant use of data analytics like Formula One does. It has an effect on the design of the vehicles, their handling, and the way the race is shown. Apart from that, telemetry has been used in Formula One since the 1980s to transmit real time data out from vehicle to pit lane engineers.

Behind every car on the circuit is a team of data analysts analysing information from a number of sensors, including lap times, tyre and brake temperatures, airflow, and vehicle performance in order to guide drivers on split-second judgments. Nevertheless, data science and analytics

are critical components of Formula One's business, mostly on the circuits. Max Métral, Formula One's Senior Analytics Manager, is responsible for resolving several important strategic difficulties through increasing commercial value via better and faster decision-making.

Mercedes previously partnered with Tibco Software, a data analytics and integration company, to create a virtual team of data analysts to analyse race data. Every element, such as a car's trajectory, tyre pressure, weight, and track conditions, may be evaluated to discover where a problem occurred. These and a number of other factors are gathered into a data pool for analysts to sort through and decide what went wrong or correctly.

Figure 2.2 Ferrari's pit crew practising pit stop

Pitstop procedures are also optimised with the use of data. When it comes to winning or losing a Formula One race, pit crews have always been critical, with workers trained to conduct any tyre and nose adjustments that are required with absolute precision. A driver might lose a race related to a delay. After practise runs, teams may save time by reviewing video and data from the vehicle and pit equipment.

In Formula One, big data drives massive judgement on decision making. Formula One is certain to benefit from whatever the technology industry develops as the reach and application of data analytics expands throughout the world with cloud computing, predictive analytics, predictive intelligence, machine learning, and prescriptive intelligence all playing a significant role in the sport's future.

Figure 2.3 McLaren’s engineers monitoring the car performance

2.3 Review on Data Analytics

Year of Study Conducted	Article Name	Objective	Methodology Used
2020	Formula 1 Race Predictor	To predict the winner of the next F1 Grand Prix	Machine Learning
2014	How Formula One Teams Are Using Big Data to Get the Inside Edge	To seeing the insight on how data analytics contribute in Formula One	Machine Learning
2020	QlikView Visualization of Formula 1 (F1) data	To explore the relational data model of Formula 1 historical data by visualization	Relational Data Model
2020	Formula One: Extracting and analysing historical results	An overview some exploratory data analysis over a single F1 season	Machine learning
2020	Reinforcement Learning for Formula 1 Race Strategy	Understand how Deep Q-Networks learn to decide pit stops	Reinforcement Learning

Table 2.1 Review on Data Analytics

Based on the table, the articles being shown demonstrates the studies being done towards the implementation of data analytics in Formula One. The most common method used is machine learning to make sense of all the data for viewing and generating real time visualisation of data. Apart from that, there were also studies that using relational data modelling to make visualisation on past data that had been gathered. The final article in the table explaining on usage of the reinforcement learning method to determine the Formula One race strategy.

2.4 Requirement Gathering Technique

The information gathering is by researching on article that related to the implementation of data analytics in Formula One. The research also covering on finding the suitable method and system that can help on achieving the objectives of this project. The information gathering will be done using Google.com as it has a lot of articles that were stored in the search engine and the other website that will be used is towardsdatascience.com which has a lot of information that related can be find in the website.

2.5 Tools and Technology

The tools that will be used in the project will be RapidMiner Studio, Microsoft Excel and Python language. All these tools will be using for data preparation, modelling and cleaning. For data collection, the source of the data used are from formula1.com and from an opensource website Kaggle.com.

2.6 Software Development Methodology

The method of system development would be CRISP-DM, which stand for Cross-Industry Standard Process for data mini. It is a robust and well-proven methodology that provides a structured approach to planning a data mining project. It enables replication of the project, assists in project management and planning, promotes practises, and supports in achieving better outcomes. CRISP-DM entails six distinct stages, which are as follows:

Phase 1: Business Understanding

During this phase, the topic description, project goals, and project requirements are all determined and specified. Since this project relating to study of Formula One, there were 3 objectives that had be identified which is to predict the potential winner for 2022 championship, to make analysis of each driver performance from 2019-2021, and to develop analytics model and a dashboard for this project. Furthermore, we need to understand the hardware and software requirements for completing this project and the sources to collect the data required for this project. The project plan will outline the stages of the project's completion, as well as the tasks and methodologies used.

Phase 2: Data Understanding

In this phase, the first gathering of information and data is made from all accessible sources. The data collected will be from the formula1.com and Kaggle.com. Next, the collected data's qualities are analysed. Afterwards, the information's quality is determined by responses to critical questions about the material's accuracy and veracity.

Phase 3: Data Preparation

This phase is where the data being prepared for the modelling phase. Process of data selection will be carried out where only suitable data will made to the final data set. Next, data cleaning will be performed to remove the missing values that happen to be in the data set. Finally, the data formatting will be applied by rearranging the attributes, reordering the record and rename of attributes and values.

Phase 4: Modelling

The "selection of modelling approach" will be performed on the cleaned data in order to construct the model at this phase. On the dataset, a supervised machine learning approach will

be utilised to the data set. Following that, a few further models are developed. All models are then evaluated to ensure they fit with the project's objectives.

Phase 5: Evaluation

The next phase will be to evaluate the process by going through the procedures used to build the model and ensuring that it is capable of achieving the project goals that had been specified. The goal is to determine if any critical project issues have been overlooked. Based on the results obtained, the best model is chosen for analysis and possible implementation.

Phase 6: Deployment

In this final phase, the approach on deciding how the finding can be applied and how can it be used. The information gathered then must be well organised. Finally, reports and project evaluations must be completed in order to evaluate what aspects of future projects might be improved.

2.7 Chapter Summary

In a nutshell, this chapter justified the analysis and requirement related to the project proposed. Next, the types of data analytics had been explained briefly in this chapter. Apart from that, a better insight on utilization of data analytics in Formula One also being covered throughout this chapter. A review on several related research had been done in order to have better understanding to move further to this project. There were several methodologies had been used on applying data analytics in the motorsports, specifically in Formula One. CRISP-DM will be utilized in this project with the six phase which is business understanding, data understanding, data preparation, modelling, evaluation and deployment that must be done in order to establish the analytics process and determine the most effective methods for completing the project.

CHAPTER 3

RESEARCH METHODOLOGY

3.0 Overview

In this chapter, the three phases of CRISP-DM will be discussed in detail, including data comprehension, data preparation, and modelling. Furthermore, in this chapter will explore the data transformation process and the modelling technique that will be employed. Likewise, the analysis dashboard will be displayed.

3.1 Data Understanding

For data understanding, the dataset used in this project will be explore. The content of the dataset such as attributes and its value will be understood further in order to make a good visualisation fort his project.

3.1.1 Overview of the Dataset

The dataset utilised in this study is a publicly accessible dataset titled "Formula 1 World Championship (1950 - 2022)", which was obtained from the Kaggle website for the data science community. There are 110 columns and almost 20,000 rows in the dataset. The data set being obtain form Kaggle consist of multiple data set which being divided by certain parts.

There are total of 17 datasets available to be used in this project which had been download in compressed zip file from Kaggle. Some of dataset available such as “drivers”, “constructors” and “qualifying”. For this project 1, there are five datasets will be used to visualise some of the data that available. The dataset that will be used are “drivers”, “constructors”, “result”, “races” and “driver_standings”.

For the purpose of this Final Year Project 2, this project will be using 1460 rows and 10 columns of attributes from the 2019 to mid-season 2022 Championship season to be visualise and get the insight of the data available. Some of the attribute being include is “surname” which indicate the name of the driver who is competing in the 2019 to 2022 championship. The next attribute to be included is "name," which pertains to the name of the team being presented by the drivers. Moving on the next attribute is “points” where in this column shows the total points being scored by drivers in each race from 2019 - 2022. There also an attribute name “race” which represent the name of race Grand Prix such as Australian Grand Prix and Monaco Grand Prix. The table below will show all the important attributes available in the dataset

No.	Attribute	Type	Description
1	driver name	Nominal	Name of each driver competing
2	Constructor (Name)	Real	Constructor team name being presented by drivers
3	race	Nominal	Grand Prix name of the race organiser
4	Position order	Real	Finished position of each driver at the end of the race
5	points	Real	Total point scored by the driver each race

6	grid	Real	Position of drivers at the start of the race
7	round	Real	Number race round for the season
10	fastest lap	Nominal	Fastest lap achieved by the driver during the race

Table 3.1 Detail of the dataset used

3.2 Data Preparation

Throughout this stage of the process, the raw dataset that was collected will be processed into a finished dataset that can be applied to the model. In order to make this data transformation, the data set will be going through cleaning process by removing the unwanted data in the in the dataset. There were several tools available to assist in the data cleansing process. The tool that will be using in this project will be Python Jupyter Notebook. The dataset consists of all data regarding Formula One races from 1950 until 2022.

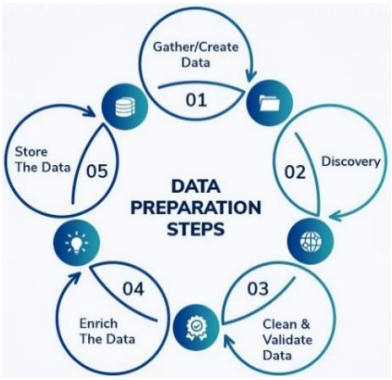

Figure 3.1 Data preparation steps

3.2.1.1 Load Dataset (drivers.csv)

To start with this process, all the datasets that going to be used have to be uploaded into Jupyter Python. After all the datasets being uploaded, the data cleaning process can start in Jupyter.

```
In [17]: import pandas as pd
         import numpy as np

In [18]: df=pd.read_csv("drivers.csv")
         df
```

Out[18]:

	driverId	driverRef	number	code	forename	surname	dob	nationality	url
0	1	hamilton	44	HAM	Lewis	Hamilton	1985-01-07	British	http://en.wikipedia.org/wiki/Lewis_Hamilton
1	2	heidfeld	\N	HEI	Nick	Heidfeld	1977-05-10	German	http://en.wikipedia.org/wiki/Nick_Heidfeld
2	3	rosberg	6	ROS	Nico	Rosberg	1985-06-27	German	http://en.wikipedia.org/wiki/Nico_Rosberg
3	4	alonso	14	ALO	Fernando	Alonso	1981-07-29	Spanish	http://en.wikipedia.org/wiki/Fernando_Alonso
4	5	kovalainen	\N	KOV	Heikki	Kovalainen	1981-10-19	Finnish	http://en.wikipedia.org/wiki/Heikki_Kovalainen
...	...	...	...	...	...	...	...	...	...
849	851	aitken	89	AIT	Jack	Aitken	1995-09-23	British	http://en.wikipedia.org/wiki/Jack_Aitken
850	852	tsunoda	22	TSU	Yuki	Tsunoda	2000-05-11	Japanese	http://en.wikipedia.org/wiki/Yuki_Tsunoda
851	853	mazepin	9	MAZ	Nikita	Mazepin	1999-03-02	Russian	http://en.wikipedia.org/wiki/Nikita_Mazepin
852	854	mick_schumacher	47	MSC	Mick	Schumacher	1999-03-22	German	http://en.wikipedia.org/wiki/Mick_Schumacher
853	855	zhou	24	ZHO	Guanyu	Zhou	1999-05-30	Chinese	http://en.wikipedia.org/wiki/Guanyu_Zhou

Figure 3.2 Load “drivers.csv” dataset in to python

Figure above shown dataset “drivers.csv” being uploaded. The file being uploaded is read as “df” dataframe. The total of rows in this “drivers.csv” are 854 with 9 columns of attributes. This step is applicable to all the four other dataset which will be upload as “constructors.csv”, “races.csv”, “result.csv”, and “driver_standings.csv”.

```
In [1]: import pandas as pd
        import numpy as np

In [2]: df=pd.read_csv("results.csv")
        df
```

Out[2]:

	resultId	raceId	driverId	constructorId	number	grid	position	positionText	positionOrder	points	laps	time	milliseconds	fastestLap	rank	fas
0	1	18	1	1	22	1	1	1	1	10.0	58	1:34:50.616	5690616	39	2	
1	2	18	2	2	3	5	2	2	2	8.0	58	+5.478	5696094	41	3	
2	3	18	3	3	7	7	3	3	3	6.0	58	+8.163	5698779	41	5	
3	4	18	4	4	5	11	4	4	4	5.0	58	+17.181	5707797	58	7	
4	5	18	5	1	23	3	5	5	5	4.0	58	+18.014	5708630	43	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
25455	25461	1076	849	3	6	18	16	16	16	0.0	57	\N	\N	49	18	
25456	25462	1076	4	214	14	10	17	17	17	0.0	57	\N	\N	57	2	
25457	25463	1076	830	9	1	2	\N	R	18	0.0	38	\N	\N	37	6	
25458	25464	1076	20	117	5	17	\N	R	19	0.0	22	\N	\N	17	19	
25459	25465	1076	832	6	55	9	\N	R	20	0.0	1	\N	\N	\N	0	

25460 rows × 18 columns

Figure 3.3 Load “results.csv” dataset in to python

3.2.1.2 Unwanted Attribute and Rows Removal (drivers.csv)

All the unwanted attribute and rows which being included in the dataframe will be remove to make it easier to read and visualise the dataset later on. In the case of the dataframe “drivers.csv”, the attribute “number” with rows containing value “\\N” will be remove from the dataframe by using drop function.

```
In [19]: df.drop(df.index[df['number'] == '\\N'], inplace=True)
         df
```

Out[19]:

	driverId	driverRef	number	code	forename	surname	dob	nationality	url
0	1	hamilton	44	HAM	Lewis	Hamilton	1985-01-07	British	http://en.wikipedia.org/wiki/Lewis_Hamilton
2	3	rosberg	6	ROS	Nico	Rosberg	1985-06-27	German	http://en.wikipedia.org/wiki/Nico_Rosberg
3	4	alonso	14	ALO	Fernando	Alonso	1981-07-29	Spanish	http://en.wikipedia.org/wiki/Fernando_Alonso
7	8	raikkonen	7	RAI	Kimi	Räikkönen	1979-10-17	Finnish	http://en.wikipedia.org/wiki/Kimi_R%C3%A4ikk%C...
8	9	kubica	88	KUB	Robert	Kubica	1984-12-07	Polish	http://en.wikipedia.org/wiki/Robert_Kubica
12	13	massa	19	MAS	Felipe	Massa	1981-04-25	Brazilian	http://en.wikipedia.org/wiki/Felipe_Massa
15	16	sutil	99	SUT	Adrian	Sutil	1983-01-11	German	http://en.wikipedia.org/wiki/Adrian_Sutil
17	18	button	22	BUT	Jenson	Button	1980-01-19	British	http://en.wikipedia.org/wiki/Jenson_Button
19	20	vettel	5	VET	Sebastian	Vettel	1987-07-03	German	http://en.wikipedia.org/wiki/Sebastian_Vettel
153	154	grosjean	8	GRO	Romain	Grosjean	1986-04-17	French	http://en.wikipedia.org/wiki/Romain_Grosjean
154	155	kobayashi	10	KOB	Kamui	Kobayashi	1986-09-13	Japanese	http://en.wikipedia.org/wiki/Kamui_Kobayashi

Figure 3.4 Removed unwanted rows based on missing values

Next, the remaining rows will be filter to the drivers that competing in the championship from the year 2019. The drop function be used again in this stage by selecting the row number of the unwanted rows.

```
In [20]: df.drop([2,12,15,17,154,812,817,819,820,823,826,827,828,830,832,833,834,835,836,837,841,843], axis=0, inplace=True)
         df
```

Out[20]:

	driverId	driverRef	number	code	forename	surname	dob	nationality	url
0	1	hamilton	44	HAM	Lewis	Hamilton	1985-01-07	British	http://en.wikipedia.org/wiki/Lewis_Hamilton
3	4	alonso	14	ALO	Fernando	Alonso	1981-07-29	Spanish	http://en.wikipedia.org/wiki/Fernando_Alonso
7	8	raikkonen	7	RAI	Kimi	Räikkönen	1979-10-17	Finnish	http://en.wikipedia.org/wiki/Kimi_R%C3%A4ikk%C...
8	9	kubica	88	KUB	Robert	Kubica	1984-12-07	Polish	http://en.wikipedia.org/wiki/Robert_Kubica
19	20	vettel	5	VET	Sebastian	Vettel	1987-07-03	German	http://en.wikipedia.org/wiki/Sebastian_Vettel
153	154	grosjean	8	GRO	Romain	Grosjean	1986-04-17	French	http://en.wikipedia.org/wiki/Romain_Grosjean
452	842	gasly	10	GAS	Pierre	Gasly	1996-02-07	French	http://en.wikipedia.org/wiki/Pierre_Gasly
807	807	hulkenberg	27	HUL	Nico	Hülkenberg	1987-08-19	German	http://en.wikipedia.org/wiki/Nico_H%C3%BClkenberg
814	815	perez	11	PER	Sergio	Pérez	1990-01-26	Mexican	http://en.wikipedia.org/wiki/Sergio_P%C3%A9rez
816	817	ricciardo	3	RIC	Daniel	Ricciardo	1989-07-01	Australian	http://en.wikipedia.org/wiki/Daniel_Ricciardo
821	822	bottas	77	BOT	Valtteri	Bottas	1989-08-28	Finnish	http://en.wikipedia.org/wiki/Valtteri_Bottas

Figure 3.5 Removed unwanted rows based on index number

The next step is to drop unwanted columns which in this case the column “driverRef” and “url”. These attributes will not be used in the data visualisation.

```
In [23]: df.drop(["driverRef","url"], axis=1, inplace=True)
         df
```

Out[23]:

	driverId	number	code	forename	surname	dob	nationality
0	1	44	HAM	Lewis	Hamilton	1985-01-07	British
1	4	14	ALO	Fernando	Alonso	1981-07-29	Spanish
2	8	7	RAI	Kimi	Räikkönen	1979-10-17	Finnish
3	9	88	KUB	Robert	Kubica	1984-12-07	Polish
4	20	5	VET	Sebastian	Vettel	1987-07-03	German
5	154	8	GRO	Romain	Grosjean	1986-04-17	French
6	842	10	GAS	Pierre	Gasly	1996-02-07	French
7	807	27	HUL	Nico	Hülkenberg	1987-08-19	German
8	815	11	PER	Sergio	Pérez	1990-01-26	Mexican
9	817	3	RIC	Daniel	Ricciardo	1989-07-01	Australian
10	822	77	BOT	Valtteri	Bottas	1989-08-28	Finnish
11	825	20	MAG	Kevin	Magnussen	1992-10-05	Danish

Figure 3.6 Remove the unwanted attribute

3.2.1.3 Reset Index Column (drivers.csv)

The remaining data are the one that will be used in this project. After that, the index column of the dataframe will reset to see the actual count of the drivers left in the dataframe.

```
In [24]: df.reset_index(drop=True, inplace=True)
         df
```

Out[24]:

	driverId	number	code	forename	surname	dob	nationality
0	1	44	HAM	Lewis	Hamilton	1985-01-07	British
1	8	7	RAI	Kimi	Räikkönen	1979-10-17	Finnish
2	9	88	KUB	Robert	Kubica	1984-12-07	Polish
3	20	5	VET	Sebastian	Vettel	1987-07-03	German
4	154	8	GRO	Romain	Grosjean	1986-04-17	French
5	842	10	GAS	Pierre	Gasly	1996-02-07	French
6	807	27	HUL	Nico	Hülkenberg	1987-08-19	German
7	815	11	PER	Sergio	Pérez	1990-01-26	Mexican
8	817	3	RIC	Daniel	Ricciardo	1989-07-01	Australian
9	822	77	BOT	Valtteri	Bottas	1989-08-28	Finnish
10	825	20	MAG	Kevin	Magnussen	1992-10-05	Danish
11	826	26	KVY	Daniil	Kvyat	1994-04-26	Russian
12	830	33	VER	Max	Verstappen	1997-09-30	Dutch
13	832	55	SAI	Carlos	Sainz	1994-09-01	Spanish
14	840	18	STR	Lance	Stroll	1998-10-29	Canadian
15	841	99	GIO	Antonio	Giovinazzi	1993-12-14	Italian
16	844	16	LEC	Charles	Leclerc	1997-10-16	Monegasque
17	846	4	NOR	Lando	Norris	1999-11-13	British
18	847	63	RUS	George	Russell	1998-02-15	British
19	848	23	ALB	Alexander	Albon	1996-03-23	Thai

Figure 3.7 Reset the column numbering

3.2.2.1 Load Dataset (constructors.csv)

```
In [20]: import pandas as pd
         import numpy as np

In [21]: df=pd.read_csv("constructors.csv")
         df
```

Out[21]:

	constructorId	constructorRef	name	nationality	url
0	1	mclaren	McLaren	British	http://en.wikipedia.org/wiki/McLaren
1	2	bmw_sauber	BMW Sauber	German	http://en.wikipedia.org/wiki/BMW_Sauber
2	3	williams	Williams	British	http://en.wikipedia.org/wiki/Williams_Grand_Pr...
3	4	renault	Renault	French	http://en.wikipedia.org/wiki/Renault_in_Formul...
4	5	toro_rosso	Toro Rosso	Italian	http://en.wikipedia.org/wiki/Scuderia_Toro_Rosso
...	...	...	...	...	...
206	209	manor	Manor Marussia	British	http://en.wikipedia.org/wiki/Manor_Motorsport
207	210	haas	Haas F1 Team	American	http://en.wikipedia.org/wiki/Haas_F1_Team
208	211	racing_point	Racing Point	British	http://en.wikipedia.org/wiki/Racing_Point_F1_Team
209	213	alphatauri	AlphaTauri	Italian	http://en.wikipedia.org/wiki/Scuderia_AlphaTauri
210	214	alpine	Alpine F1 Team	French	http://en.wikipedia.org/wiki/Alpine_F1_Team

211 rows × 5 columns

Figure 3.8 Load “constructors.csv” dataset in to python

The “constructors.csv” will be load into the process. The file being uploaded as “df” dataframe. The dataframe consist of 211 rows and 5 columns after being loaded.

3.2.2.2 Unwanted Attribute and Rows Removal (constructors.csv)

```
In [23]: df.drop(df[(df['constructorId'] >9) & (df['constructorId'] < 51)].index, inplace=True)
         df
```

Out[23]:

	constructorId	constructorRef	name	nationality	url
0	1	mclaren	McLaren	British	http://en.wikipedia.org/wiki/McLaren
1	2	bmw_sauber	BMW Sauber	German	http://en.wikipedia.org/wiki/BMW_Sauber
2	3	williams	Williams	British	http://en.wikipedia.org/wiki/Williams_Grand_Pr...
3	4	renault	Renault	French	http://en.wikipedia.org/wiki/Renault_in_Formul...
4	5	toro_rosso	Toro Rosso	Italian	http://en.wikipedia.org/wiki/Scuderia_Toro_Rosso
...	...	...	...	...	...
206	209	manor	Manor Marussia	British	http://en.wikipedia.org/wiki/Manor_Motorsport
207	210	haas	Haas F1 Team	American	http://en.wikipedia.org/wiki/Haas_F1_Team
208	211	racing_point	Racing Point	British	http://en.wikipedia.org/wiki/Racing_Point_F1_Team
209	213	alphatauri	AlphaTauri	Italian	http://en.wikipedia.org/wiki/Scuderia_AlphaTauri
210	214	alpine	Alpine F1 Team	French	http://en.wikipedia.org/wiki/Alpine_F1_Team

171 rows × 5 columns

Figure 3.9 Removed unwanted rows based on “constructorId” value

Next, unwanted rows will be removed based on attributes “constructorsId” in which the rows that will be kept only the constructor teams that will be competing in 2019-2022 championship. This process will be using df.drop(df[(df['constructorId'] >X) & (df['constructorId']

<X)].index, inplace=True). Then, the unwanted attributes column will be removed which the column name "constructorRef" and "url" from the dataframe.

```
In [28]: df.drop(["constructorRef","url"], axis=1, inplace=True)
         df
```

Out[28]:

	constructorId	name	nationality
0	1	McLaren	British
2	3	Williams	British
3	4	Renault	French
4	5	Toro Rosso	Italian
5	6	Ferrari	Italian
8	9	Red Bull	Austrian
49	51	Alfa Romeo	Swiss
115	117	Aston Martin	British
129	131	Mercedes	German
207	210	Haas F1 Team	American
208	211	Racing Point	British

Figure 3.10 Remove the unwanted attribute

3.2.2.1 Reset Index Column (constructors.csv)

The remaining data in the dataframe are all the constructor's team that will be competing in the 2019. The column of this dataframe will be reset to show the actual count of the remaining data.

```
In [5]: df.reset_index(drop=True, inplace=True)
        df
```

Out[5]:

	constructorId	name	nationality
0	1	McLaren	British
1	3	Williams	British
2	4	Renault	French
3	5	Toro Rosso	Italian
4	6	Ferrari	Italian
5	9	Red Bull	Austrian
6	51	Alfa Romeo	Swiss
7	131	Mercedes	German
8	210	Haas F1 Team	American
9	211	Racing Point	British

Figure 3.11 Reset the column numbering

3.2.3.1 Load Dataset (races.csv)

Dataset “races.csv” are being loaded into jupyter. The file being uploaded as “df” dataframe. The dataframe consist of 1079 rows and 8 columns after being loaded.

```
In [2]: import pandas as pd
        import numpy as np

In [ ]:

In [3]: df=pd.read_csv("races.csv")
        df
```

Out[3]:

	raceId	year	round	circuitId	name	date	time	
0	1	2009	1	1	Australian Grand Prix	29/03/09	6:00:00	http://en.wikipedia.org/wiki/2009_Au
1	2	2009	2	2	Malaysian Grand Prix	05/04/09	9:00:00	http://en.wikipedia.org/wiki/2009_Ma
2	3	2009	3	17	Chinese Grand Prix	19/04/09	7:00:00	http://en.wikipedia.org/wiki/2009_Chi
3	4	2009	4	3	Bahrain Grand Prix	26/04/09	12:00:00	http://en.wikipedia.org/wiki/2009_Bah
4	5	2009	5	4	Spanish Grand Prix	10/05/09	12:00:00	http://en.wikipedia.org/wiki/2009_Spa
...	...	...	...	...	...	...	...	
1074	1092	2022	18	22	Japanese Grand Prix	09/10/22	5:00:00	http://en.wikipedia.org/wiki/2022_Jap
1075	1093	2022	19	69	United States Grand Prix	23/10/22	19:00:00	http://en.wikipedia.org/wiki/2022_Un

Figure 3.12 Load “races.csv” dataset in to python

3.2.3.2 Unwanted Attribute and Rows Removal (races.csv)

```
In [5]: df.drop(df[(df['year'] >1949) & (df['year'] < 2019)].index, inplace=True)

        df
```

Out[5]:

	raceId	year	round	circuitId	name	date	time	
997	1010	2019	1	1	Australian Grand Prix	17/03/19	5:10:00	http://en.wikipedia.org/wiki/2019_Au
998	1011	2019	2	3	Bahrain Grand Prix	31/03/19	15:10:00	http://en.wikipedia.org/wiki/2019_Ba
999	1012	2019	3	17	Chinese Grand Prix	14/04/19	6:10:00	http://en.wikipedia.org/wiki/2019_Chi
1000	1013	2019	4	73	Azerbaijan Grand Prix	28/04/19	12:10:00	http://en.wikipedia.org/wiki/2019_Az
1001	1014	2019	5	4	Spanish Grand Prix	12/05/19	13:10:00	http://en.wikipedia.org/wiki/2019_Spa

Figure 3.13 Removed unwanted rows based on “year” value

For this data frame, the rows that twill be dropped are from the attribute “year”. The function will drop all other rows that contains year less than 2019. Next, the unwanted attributes will be removed which is the attribute “url” for this dataframe (df.drop(["url"], axis=1, inplace=True)).

3.2.3.3 Reset Index Column (races.csv)

The remaining data in dataframe are only the races that being held in 2019-2022. After that, the dataframe index column will be reset.

```
In [9]: df.reset_index(drop=True, inplace=True)
        df
```

Out[9]:

	raceId	year	round	circuitId	name	date	time
0	1010	2019	1	1	Australian Grand Prix	17/03/19	5:10:00
1	1011	2019	2	3	Bahrain Grand Prix	31/03/19	15:10:00
2	1012	2019	3	17	Chinese Grand Prix	14/04/19	6:10:00
3	1013	2019	4	73	Azerbaijan Grand Prix	28/04/19	12:10:00
4	1014	2019	5	4	Spanish Grand Prix	12/05/19	13:10:00
5	1015	2019	6	6	Monaco Grand Prix	26/05/19	13:10:00
6	1016	2019	7	7	Canadian Grand Prix	09/06/19	18:10:00
7	1017	2019	8	34	French Grand Prix	23/06/19	13:10:00
8	1018	2019	9	70	Austrian Grand Prix	30/06/19	13:10:00
9	1019	2019	10	9	British Grand Prix	14/07/19	13:10:00
10	1020	2019	11	10	German Grand Prix	28/07/19	13:10:00
11	1021	2019	12	11	Hungarian Grand Prix	04/08/19	13:10:00

Figure 3.14 Reset the column numbering

3.2.4.1 Load Dataset (result.csv)

```
In [1]: import pandas as pd
        import numpy as np

In [2]: df=pd.read_csv("results.csv")
        df
```

Out[2]:

	resultId	raceId	driverId	constructorId	number	grid	position	positionText	positionOrder	points	laps	time	milliseconds	fastestLap	rank	fas
0	1	18	1	1	22	1	1	1	1	10.0	58	1:34:50.616	5690616	39	2	
1	2	18	2	2	3	5	2	2	2	8.0	58	+5.478	5696094	41	3	
2	3	18	3	3	7	7	3	3	3	6.0	58	+8.163	5698779	41	5	
3	4	18	4	4	5	11	4	4	4	5.0	58	+17.181	5707797	58	7	
4	5	18	5	1	23	3	5	5	5	4.0	58	+18.014	5708630	43	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
25455	25461	1076	849	3	6	18	16	16	16	0.0	57	\N	\N	49	18	
25456	25462	1076	4	214	14	10	17	17	17	0.0	57	\N	\N	57	2	
25457	25463	1076	830	9	1	2	\N	R	18	0.0	38	\N	\N	37	6	
25458	25464	1076	20	117	5	17	\N	R	19	0.0	22	\N	\N	17	19	
25459	25465	1076	832	6	55	9	\N	R	20	0.0	1	\N	\N	\N	0	

25460 rows × 18 columns

Figure 3.15 Load “result.csv” dataset in to python

This operation will load the "driver standings.csv" file. The file being uploaded as dataframe type "df" After loading, the dataframe has 25,460 rows and 18 columns.

3.2.4.2 Unwanted Attribute and Rows Removal (result.csv)

The rows that will be eliminated from this dataframe are those with the "raceId" attribute. The method will remove any rows holding values other than "raceId" between 1010 and 1030 that correspond to races held in 2019. This dataframe will be left with 420 rows.

```
In [3]: df.drop(df[(df['raceId'] >0) & (df['raceId'] < 1010)].index, inplace=True)
        df
```

Out[3]:

	resultId	raceId	driverId	constructorId	number	grid	position	positionText	positionOrder	points	laps	time	milliseconds	fastestLap	rank	fa
24197	24203	1010	822	131	77	2	1	1	1	26.0	58	1:25:27.325	5127325	57	1	
24198	24204	1010	1	131	44	1	2	2	2	18.0	58	+20.886	5148211	57	2	
24199	24205	1010	830	9	33	4	3	3	3	15.0	58	+22.520	5149845	57	3	
24200	24206	1010	20	6	5	3	4	4	4	12.0	58	+57.109	5184434	16	8	
24201	24207	1010	844	6	16	5	5	5	5	10.0	58	+58.203	5185528	58	4	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Figure 3.16 Removed unwanted rows based on “raceId” value

3.2.4.3 Reset Index Column (result.csv)

As with prior dataframes, the index column of this dataframe will be reset.

```
In [4]: df.reset_index(drop=True, inplace=True)
        df
```

0	24203	1010	822	131	77	2	1	1	1	26.0	58	1:25:27.325	5127325	57	1
1	24204	1010	1	131	44	1	2	2	2	18.0	58	+20.886	5148211	57	2
2	24205	1010	830	9	33	4	3	3	3	15.0	58	+22.520	5149845	57	3
3	24206	1010	20	6	5	3	4	4	4	12.0	58	+57.109	5184434	16	8
4	24207	1010	844	6	16	5	5	5	5	10.0	58	+58.203	5185528	58	4

Figure 3.17 Reset the column numbering

3.3 Combine the dataset

After the cleaning process were done, 5 datasets will be combined to make it to 1 dataset using matrix visualization function in Power BI and export the matrix table as a csv file. The attributes that will be combined are year, round, race name, constructor (name), driver surname, grid, position order, fastest lap, points and status. All these attributes are from 5 dataset which are constructor, driver, result, races, and status. The step is shown as in Figure 3.18 until figure 3.21.

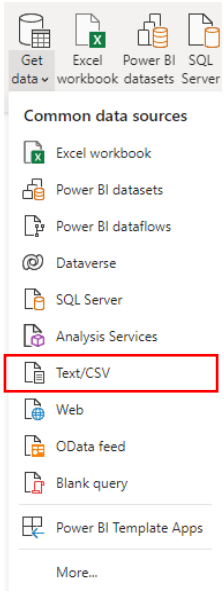

Figure 3.18 Select datasets to be import into Power BI

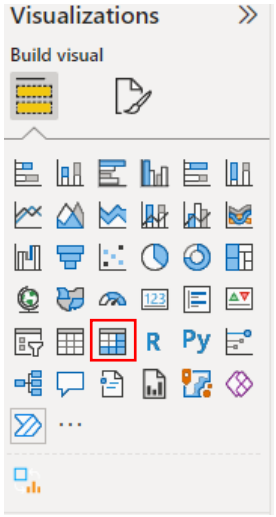

Figure 3.19Use the matrix visualisation function to combined the dataset

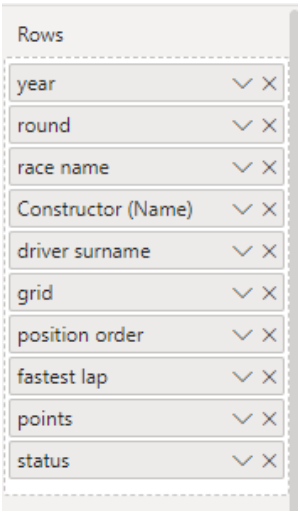

Figure 3.20 Drag the attributes into the rows box

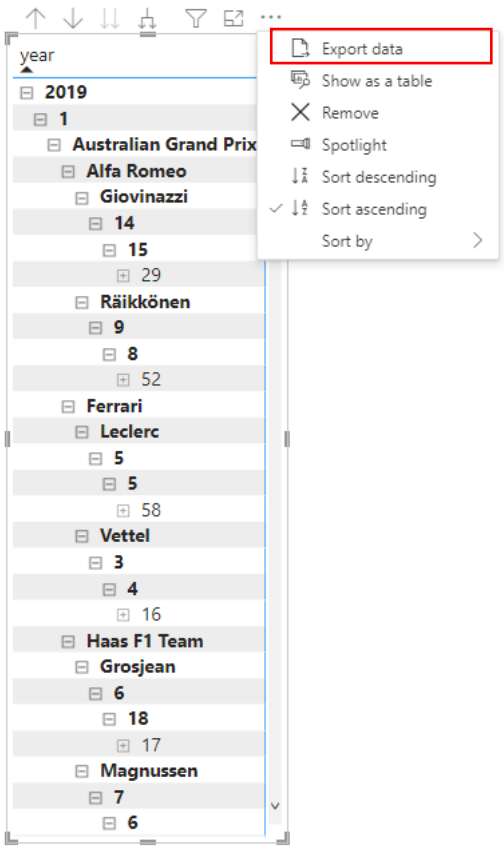

Figure 3.21 Export the matrix table into a csv file

3.4 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a method for analysing datasets in order to highlight their most specific attributes, which typically includes statistical graphics and other data visualisation approaches. Regardless of the fact that a statistical model can be employed, EDA is typically used to explore what the data can tell us outside of formal modelling, and hence varies from conventional hypothesis testing.

3.4.1 EDA Process

```
In [37]: dc.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   year                1460 non-null   int64
 1   round               1460 non-null   int64
 2   race name           1460 non-null   object
 3   Constructor (Name)  1460 non-null   object
 4   driver surname      1460 non-null   object
 5   grid                1460 non-null   int64
 6   position order      1460 non-null   int64
 7   fastest lap         1460 non-null   object
 8   points              1460 non-null   int64
 9   status              1460 non-null   object
dtypes: int64(5), object(5)
memory usage: 114.2+ KB

In [38]: dc.isnull().sum()

Out[38]: year                0
         round               0
         race name           0
         Constructor (Name)  0
         driver surname      0
         grid                0
         position order      0
         fastest lap         0
         points              0
         status              0
         dtype: int64

In [39]: dc.isnull().values.any()

Out[39]: False

In [40]: dc.columns

Out[40]: Index(['year', 'round', 'race name', 'Constructor (Name)', 'driver surname',
                'grid', 'position order', 'fastest lap', 'points', 'status'],
               dtype='object')
```

Figure 3.22 Using info(), isnull() and colums function for EDA process

Figure 3.25 shows the EDA that can check for missing values and the dataset info. The info() function will examine the dataset's Column, Non-Null Count, and Data Type to shows if the data is suitable for analysis. Next, the process also includes with checking for missing values

within the dataset. In figure 3.25, there were no missing values in the dataset. However, there were also another function can be use to check for missing value to make sure the dataset being checked thoroughly for missing values. The finding from the dataset still shows no missing values. Lastly, used the column function to check for column's attributes in the dataset.

3.4.2 Numerical and Categorical Features

```
In [41]: #Numerical features
num_col = dc.select_dtypes(include = np.number).columns.tolist()
print('There are', len(num_col), 'numerics, including:')
print(num_col)

There are 5 numerics, including:
['year', 'round', 'grid', 'position order', 'points']

In [42]: #Categorical features
cat_col = dc.select_dtypes(include = object).columns.tolist()
print('There are', len(cat_col), 'categoricals, including:')
print(cat_col)

There are 5 categoricals, including:
['race name', 'Constructor (Name)', 'driver surname', 'fastest lap', 'status']

In [43]: #Extract details on categorical features
for i in cat_col:
    options_no = dc[i].nunique() #how many options available
    options_name = dc[i].unique().tolist() #options in each category
    print( i, 'has', options_no, 'unique variables, including:' )
    print(options_name,'\n')

race name has 35 unique variables, including:
['Australian Grand Prix', 'Bahrain Grand Prix', 'Chinese Grand Prix', 'Azerbaijan Grand Prix', 'Spanish Grand Prix', 'Monaco Gr
and Prix', 'Canadian Grand Prix', 'French Grand Prix', 'Austrian Grand Prix', 'British Grand Prix', 'German Grand Prix', 'Hunga
rian Grand Prix', 'Belgian Grand Prix', 'Italian Grand Prix', 'Singapore Grand Prix', 'Russian Grand Prix', 'Japanese Grand Pri
x', 'Mexican Grand Prix', 'United States Grand Prix', 'Brazilian Grand Prix', 'Abu Dhabi Grand Prix', 'Styrian Grand Prix', '70
th Anniversary Grand Prix', 'Tuscan Grand Prix', 'Eifel Grand Prix', 'Portuguese Grand Prix', 'Emilia Romagna Grand Prix', 'Tur
kish Grand Prix', 'Sakhir Grand Prix', 'Dutch Grand Prix', 'Mexico City Grand Prix', 'São Paulo Grand Prix', 'Qatar Grand Pri
x', 'Saudi Arabian Grand Prix', 'Miami Grand Prix']

Constructor (Name) has 13 unique variables, including:
['Alfa Romeo', 'Ferrari', 'Haas F1 Team', 'McLaren', 'Mercedes', 'Racing Point', 'Red Bull', 'Renault', 'Toro Rosso', 'William
s', 'AlphaTauri', 'Alpine F1 Team', 'Aston Martin']

driver surname has 29 unique variables, including:
['Giovinazzi', 'Räikkönen', 'Leclerc', 'Vettel', 'Grosjean', 'Magnussen', 'Norris', 'Sainz', 'Bottas', 'Hamilton', 'Pérez', 'St
roll', 'Gasly', 'Verstappen', 'Hülkenberg', 'Ricciardo', 'Albon', 'Kvyat', 'Kubica', 'Russell', 'Ocon', 'Latifi', 'Fittipaldi',
'Aitken', 'Tsunoda', 'Alonso', 'Mazepin', 'Schumacher', 'Zhou']

fastest lap has 78 unique variables, including:
['29', '52', '58', '16', '17', '56', '9', '57', '41', '39', '18', '43', '30', '55', '42', '45', '38', '3', '14', '36', '51', '3
1', '47', '40', '13', '32', '27', '37', '35', '44', '46', '22', '28', '48', '50', '33', '49', '34', '64', '66', '59', '54', '6
5', '6', '61', '67', '72', '78', '76', '69', '60', '63', '4', '62', '53', '5', '\\N', '10', '23', '21', '24', '26', '19', '15',
'68', '70', '25', '71', '8', '11', '20', '80', '85', '75', '74', '2', '12', '7']

status has 48 unique variables, including:
["'+1 Lap", 'Finished', 'Wheel', 'Engine', 'Damage', "'+3 Laps", "'+2 Laps", 'Retired', 'Collision damage', 'Out of fuel', 'Col
lision', 'Power Unit', 'Brakes', 'Transmission', 'Suspension', 'Spun off', 'Accident', 'Power loss', 'Exhaust', 'Water pressur
e', 'Hydraulics', 'Disqualified', 'Overheating', 'Withdrew', 'Oil leak', 'Electronics', 'Fuel pressure', "'+5 Laps", 'Debris',
'Puncture', 'Radiator', 'Gearbox', 'Illness', 'Electrical', 'Driveshaft', 'Wheel nut', 'Turbo', 'Rear wing', 'Mechanical', 'Coo
ling system', 'Water pump', 'Fuel leak', 'Water leak', 'Front wing', 'Vibrations', 'Fuel pump', 'Undertray', "'+6 Laps"]
```

Figure 3.23 Numerical and Categorical features in the dataset

The subsequent phase in EDA involves highlighting the numerical and categorical features of the dataset. There were 5 numerical features in the dataset which is 'year', 'round', 'grid', 'position order' and 'points'. For categorical feature, there were 5 categorical features in the dataset which include 'race name', 'Constructor (Name)', 'driver surname', 'fastest lap' and 'status'. Finally, there are a total of 203 unique variables in the dataset in which 'race name' has 35 unique variables, 'Constructor (Name)' has 13 unique variables, 'driver surname' has 29 unique variables, 'fastest lap' has 78 unique variables and 'status' has 48 unique variables.

3.4.3 Value counts and groupby

```
In [44]: #Check points data
         dc.status.value_counts()
```

```
Out[44]: Finished              759
         '+1 Lap               400
         '+2 Laps               66
         Collision              54
         Accident               22
         Collision damage       16
         Power Unit             15
         Brakes                 13
         Engine                 12
         Gearbox                11
         '+3 Laps               10
         Suspension              8
         Retired                 6
         Power loss              5
         Hydraulics              4
         Disqualified            3
         Fuel pressure           3
         Electronics             3
         Wheel                   3
         Withdrew                3
         Overheating             3
         Water pressure          3
         Spun off                3
         Puncture                3
         Illness                 2
         Undertray               2
         Turbo                   2
         Mechanical              2
         Oil leak                2
         Damage                  2
         Exhaust                 2
         Transmission            2
         Debris                  1
         Water pump              1
         Fuel pump               1
         Vibrations              1
         Front wing              1
         Water leak              1
         Fuel leak               1
         Out of fuel             1
         Cooling system          1
         Radiator                1
         Rear wing               1
         Wheel nut               1
         Driveshaft              1
         Electrical              1
         '+5 Laps                1
         '+6 Laps                1
         Name: status, dtype: int64
```

Figure 3.24 Value count function

```
In [45]: dc.groupby('status').count()
```

Out[45]:

status	year	round	race name	Constructor (Name)	driver surname	grid	position order	fastest lap	points
'+1 Lap	400	400	400	400	400	400	400	400	400
'+2 Laps	66	66	66	66	66	66	66	66	66
'+3 Laps	10	10	10	10	10	10	10	10	10
'+5 Laps	1	1	1	1	1	1	1	1	1
'+6 Laps	1	1	1	1	1	1	1	1	1
Accident	22	22	22	22	22	22	22	22	22
Brakes	13	13	13	13	13	13	13	13	13
Collision	54	54	54	54	54	54	54	54	54
Collision damage	16	16	16	16	16	16	16	16	16
Cooling system	1	1	1	1	1	1	1	1	1
Damage	2	2	2	2	2	2	2	2	2
Debris	1	1	1	1	1	1	1	1	1
Disqualified	3	3	3	3	3	3	3	3	3
Driveshaft	1	1	1	1	1	1	1	1	1
Electrical	1	1	1	1	1	1	1	1	1
Electronics	3	3	3	3	3	3	3	3	3
Engine	12	12	12	12	12	12	12	12	12
Exhaust	2	2	2	2	2	2	2	2	2
Finished	759	759	759	759	759	759	759	759	759
Front wing	1	1	1	1	1	1	1	1	1
Fuel leak	1	1	1	1	1	1	1	1	1
Fuel pressure	3	3	3	3	3	3	3	3	3
Fuel pump	1	1	1	1	1	1	1	1	1
Gearbox	11	11	11	11	11	11	11	11	11
Hydraulics	4	4	4	4	4	4	4	4	4
Illness	2	2	2	2	2	2	2	2	2
Mechanical	2	2	2	2	2	2	2	2	2
Oil leak	2	2	2	2	2	2	2	2	2
Out of fuel	1	1	1	1	1	1	1	1	1
Overheating	3	3	3	3	3	3	3	3	3
Power Unit	15	15	15	15	15	15	15	15	15
Power loss	5	5	5	5	5	5	5	5	5
Puncture	3	3	3	3	3	3	3	3	3

Figure 3.25 Groupby function

Value count function I use to identify the value count on each of the variables in the ‘status’ attribute. In figure 3.27 shows the number of each ‘status’ unique value being used in the dataset. Figure 3.28 shows the groupby function being used to relate column with selected attributes.

3.4.4 Groupby Function

```
In [64]: #counting the number of constructors in dataframe
         dc['Constructor (Name)'].nunique()

Out[64]: 13

In [65]: #counting the 10 highest points scored
         dc.groupby(['Constructor (Name)'], sort=True)['points'].sum().nlargest(10)

Out[65]: Constructor (Name)
         Mercedes          2215
         Red Bull          1719
         Ferrari           1266
         McLaren            709
         Racing Point       283
         AlphaTauri         276
         Renault            272
         Alpine F1 Team     251
         Alfa Romeo         127
         Aston Martin        97
         Name: points, dtype: int64

In [66]: #counting the top 10 highest point in each race circuit
         dc.groupby(['Constructor (Name)', 'race name'], sort=True)['points'].sum().nlargest(10)

Out[66]: Constructor (Name)  race name
         Mercedes            Spanish Grand Prix      150
                             Bahrain Grand Prix      140
                             British Grand Prix      125
                             Hungarian Grand Prix    122
                             Russian Grand Prix      120
                             Austrian Grand Prix     119
         Red Bull            Spanish Grand Prix      118
         Mercedes            French Grand Prix       106
         Red Bull            Monaco Grand Prix       100
         Mercedes            Abu Dhabi Grand Prix     97
         Name: points, dtype: int64
```

Figure 3.26 Steps to groupby selected attributes

In this operation, the similar step is performed in the groupby function. Initially, we employ the nunique() function as shown in line 64 from figure 3.29 to count number of Constructor (Name) in the dataframe. After that, use the groupby function to identify the 10 highest point scored in the Constructor (Name) shown in line 65 from figure 3.29. Finally, the figure shown in line 66 where the top 10 most points scored by Constructor (Name) based on the race circuits.

3.4.5 Random Samples and Summary of EDA

```
In [67]: #random samples in the dataset
         dc.sample(10)
```

Out[67]:

	year	round	race name	Constructor (Name)	driver surname	grid	position order	fastest lap	points	status
218	2019	11	German Grand Prix	Williams	Kubica	18	10	63	1	Finished
292	2019	15	Singapore Grand Prix	Red Bull	Albon	6	6	59	8	Finished
1342	2022	8	Azerbaijan Grand Prix	AlphaTauri	Gasly	6	5	39	10	Finished
1155	2021	20	Qatar Grand Prix	Mercedes	Hamilton	1	1	50	25	Finished
1201	2022	1	Bahrain Grand Prix	Alfa Romeo	Zhou	15	10	39	1	Finished
1121	2021	19	São Paulo Grand Prix	Alfa Romeo	Räikkönen	0	12	54	0	'+1 Lap
869	2021	6	Azerbaijan Grand Prix	Ferrari	Sainz	5	8	42	4	Finished
251	2019	13	Belgian Grand Prix	Racing Point	Stroll	16	10	34	1	Finished
1325	2022	7	Monaco Grand Prix	Alpine F1 Team	Ocon	10	12	50	0	Finished
1273	2022	4	Emilia Romagna Grand Prix	McLaren	Ricciardo	6	18	61	0	'+1 Lap

```
In [68]: dc.describe().T
         # summary of statistics for numerical columns that includes count, mean, standard deviation etc.
         # .T: transpose the table to make it easy to read
```

Out[68]:

	count	mean	std	min	25%	50%	75%	max
year	1460.0	2020.369863	1.079523	2019.0	2019.00	2020.0	2021.00	2022.0
round	1460.0	9.972603	5.795161	1.0	5.00	10.0	14.00	22.0
grid	1460.0	10.067808	5.839074	0.0	5.00	10.0	15.00	20.0
position order	1460.0	10.500000	5.768257	1.0	5.75	10.5	15.25	20.0
points	1460.0	5.060959	7.224567	0.0	0.00	0.0	8.00	26.0

Figure 3.27 Random samples and summary of the final dataset

The random sample being identify by using sample() function to shows the first 10 rows of the dataframe as shown in line 67 from figure 3.30 above. Next, as shown in line 68 the use of describe() function to show the statistical summary of the dataframe for numerical column which includes count, mean and standard deviation. The .T function is use to transpose the table to make it comprehendible.

3.4.6 Final Step in EDA

```
In [69]: #Counting the smallest
         dc.groupby(['Constructor (Name)'], sort=True)['points'].sum().nsmallest(10)

Out[69]: Constructor (Name)
         Williams            27
         Haas F1 Team        62
         Toro Rosso          85
         Aston Martin        97
         Alfa Romeo         127
         Alpine F1 Team     251
         Renault            272
         AlphaTauri         276
         Racing Point       283
         McLaren            709
         Name: points, dtype: int64

In [71]: cols = set(dc.columns) - {'year'}
         df1 = dc[list(cols)]
         df1.describe().T
         # summary of statistics for numerical columns that includes count, mean, standard deviation etc.
         # column year were drop
         # .T: transpose the table to make it easy to read
```

Out[71]:

	count	mean	std	min	25%	50%	75%	max
round	1460.0	9.972603	5.795161	1.0	5.00	10.0	14.00	22.0
points	1460.0	5.060959	7.224567	0.0	0.00	0.0	8.00	26.0
grid	1460.0	10.067808	5.839074	0.0	5.00	10.0	15.00	20.0
position order	1460.0	10.500000	5.768257	1.0	5.75	10.5	15.25	20.0

Figure 3.28 Final step of EDA

In this final step, the groupby() function being used once again to shows the Constructor (Name) which gained the least points. Moving on, the summary of dataframe being display again but without the column ‘year’ and still using the .T function for the table.

3.4 Chapter Summary

In a nutshell, this chapter have described some of the processes that going through the approach of CRISP-DM which are data understanding, data preparation and modelling. It describes the transformation process of the dataset and the chosen modelling approach for the dataset. Finally, this chapter also shown a brief design of the dashboard that will be used in this project.

CHAPTER 4

Modelling

4.0 Overview

Next phase will be moving on to the modelling of data. Data modelling is the technique of developing a visual representation of either a complete data system or parts of it to describe relationships between data items and structures. In this phase, a modelling technique will be chosen for analysis after the training and testing data being done.

4.1 Modelling Technique

In this modelling approach, supervised learning will be applied to this project since it can identify labelled input and output data. In supervised learning, the algorithm “learns” from the training dataset by continuously generating predictions on the data and adjusting for the right solution. Supervised learning also considered as easier to compute compared to unsupervised learning which considered as more difficult.

4.1.1 Encoding the variables

```
In [168]: #Data preprocessing (Turn raw data into clean data set)
          #encoding categorical variables
          from sklearn.preprocessing import OneHotEncoder

          dc_onehot = pd.get_dummies(dc, columns=['race name', 'Constructor (Name)', 'driver surname', 'fastest lap', 'status'], prefix = |
          features=dc_onehot.loc[:, dc_onehot.columns != 'points']
          label=dc['points']
          features.head()
```

Out[168]:

	year	round	grid	position order	race_70th Anniversary Grand Prix	race_Abu Dhabi Grand Prix	race_Australian Grand Prix	race_Austrian Grand Prix	race_Azerbaijan Grand Prix	race_Bahrain Grand Prix	...	status_Transmission	status_Turbo	st
0	2019	1	14	15	0	0	1	0	0	0	...	0	0	
1	2019	1	9	8	0	0	1	0	0	0	...	0	0	
2	2019	1	5	5	0	0	1	0	0	0	...	0	0	
3	2019	1	3	4	0	0	1	0	0	0	...	0	0	
4	2019	1	6	18	0	0	1	0	0	0	...	0	0	

5 rows × 207 columns

Figure 4.1 Encoding categorical variables

The process to encode categorical data is to turn the categorical data into integers to make it easier for the data to be compute before in can be fit to evaluate the model. The categorical data that will be involve are 'race name', 'Constructor (Name)', 'driver surname', 'fastest lap', 'status'.

4.1.2 Drop selected column

```
In [212]: features = features.drop(['year'], axis=1)

In [213]: features.head()

Out[213]:
```

	round	grid	position order	race_70th Anniversary Grand Prix	race_Abu Dhabi Grand Prix	race_Australian Grand Prix	race_Austrian Grand Prix	race_Azerbaijan Grand Prix	race_Bahrain Grand Prix	race_Belgian Grand Prix	...	status_Transmission	status_Tu
0	1	14	15	0	0	1	0	0	0	0	...	0	
1	1	9	8	0	0	1	0	0	0	0	...	0	
2	1	5	5	0	0	1	0	0	0	0	...	0	
3	1	3	4	0	0	1	0	0	0	0	...	0	
4	1	6	18	0	0	1	0	0	0	0	...	0	

5 rows × 206 columns

```
In [214]: features.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Columns: 206 entries, round to status_Withdrew
dtypes: int64(3), uint8(203)
memory usage: 323.8 KB
```

Figure 4.2 Drop 'year' column for encoding

The 'year' column is going to be removed because it is considered to be of less significance than the other columns. This is due to the fact that the data type for 'year' is numerical, which means that it cannot be counted and yet be integrated into the model. On the other hand, the info() function shows that there were 206 columns that already been encode from ‘round’ to ‘status’.

4.1.3 Scaling Features

```
In [172]: #scalling features

          from sklearn.preprocessing import MinMaxScaler
          scaler=MinMaxScaler()
          features=scaler.fit_transform(features)

In [173]: features

Out[173]: array([[0.        , 0.7       , 0.73684211, ..., 0.        , 0.        ,
                  0.        ],
                 [0.        , 0.45      , 0.36842105, ..., 0.        , 0.        ,
                  0.        ],
                 [0.        , 0.25      , 0.21052632, ..., 0.        , 0.        ,
                  0.        ],
                 ...,
                 [0.57142857, 0.5       , 0.        , ..., 0.        , 0.        ,
                  0.        ],
                 [0.57142857, 0.85      , 0.84210526, ..., 0.        , 0.        ,
                  0.        ],
                 [0.57142857, 0.95      , 0.89473684, ..., 0.        , 0.        ,
                  0.        ]])

In [174]: dc.head()

Out[174]:
```

	year	round	race name	Constructor (Name)	driver surname	grid	position order	fastest lap	points	status
0	2019	1	Australian Grand Prix	Alfa Romeo	Giovinazzi	14	15	29	0	'+1 Lap
1	2019	1	Australian Grand Prix	Alfa Romeo	Räikkönen	9	8	52	4	'+1 Lap
2	2019	1	Australian Grand Prix	Ferrari	Leclerc	5	5	58	10	Finished
3	2019	1	Australian Grand Prix	Ferrari	Vettel	3	4	16	12	Finished
4	2019	1	Australian Grand Prix	Haas F1 Team	Grosjean	6	18	17	0	Wheel

Figure 4.3 Scaling technique

As can be observed, the given dataset has a variety of characteristics with varying magnitudes, units, and ranges as a result of the scaling techniques. In the calculations of distance, features with high magnitudes will be weighed more heavily than those with low magnitudes. As a method of mitigating this effect, the magnitude must be normalised with all characteristics. Scaling can be utilised to achieve this goal.

4.2 Train and Test Data

The dataset is going to be split up into two distinct datasets, with one being used for training and the other being used for testing. Due to the fact that training the model often requires a significant amount of data points, the data are frequently divided in an unequal manner. The 70/30 split is the one that is most commonly used for training and testing. It is the initial dataset that the Machine Learning algorithm is trained on in order to learn and make accurate predictions.

The dataset will be utilised in the performance analysis of the machine learning algorithm that was developed using the training dataset. It is not viable to keep repeating the training dataset in the testing step since the Machine Learning algorithm would already "know" the predicted outcome. The data from the tests make up thirty percent of the total.

4.2.1 Training Data

```
In [175]: #Training data
          from sklearn.model_selection import train_test_split
          train_data, test_data, train_labels, test_labels = train_test_split(features, label, test_size=0.3, random_state=42)
```

Figure 4.4 Training data

Sklearn is the function being use to train this dataset. It provides a wide range of efficient tools for machine learning and statistical modelling, such as classification, regression, and clustering, via a consistent Python interface. Importing the train test split function from sklearn is the first step of training data. Furthermore, this method eliminates the need to manually split the dataset. By default, sklearn train test split will divide the dataset at random into the two specified parts.

4.2.2 Model Comparison

```
In [176]: #Model comparison
from sklearn.metrics import r2_score
def compare_models(model):
    model_name = model.__class__.__name__
    fit=model.fit(train_data, train_labels)
    y_pred=fit.predict(test_data)
    r2=r2_score(test_labels, y_pred)
    return([model_name, r2])

from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn import svm
from sklearn.tree import DecisionTreeRegressor

models = [
    RandomForestRegressor(n_estimators=200, max_depth=3, random_state=0),
    LogisticRegression(),
    LinearRegression(),
    svm.SVR(),
    DecisionTreeRegressor()
]
```

Figure 4.5 Comparing all the model suggested

Figure above shown the step being used to comparing the model. In this phase, the Random Forest Regressor, Decision Tree Regressor, Support Vector Regression (SVR), Logistic Regression, and Linear Regression models are imported. The models will be trained, and the results will indicate what type of model is suited for this dataset's modelling approach.

The R2 regression score function will be used to assess how much of the overall variation is related to each attribute when evaluating the measure. In statistical parlance, the R2 score indicates how well the data points correspond to a curve or line. This assessment measure will be utilised later in the chapter titled "Evaluation".

4.2.3 Calculation of trained model

```
['RandomForestRegressor', 0.9374873069252018]
['LogisticRegression', 0.7803712056139949]
['LinearRegression', -5.782558575461003e+18]
['SVR', 0.8781295333918295]
['DecisionTreeRegressor', 0.9054455255338173]
```

Figure 4.6 Trained and comparing the model suggested

In figure 4.6, the trained model is depicted together with all the results that the model predicted. Random Forest Regressor obtains the highest score among the outcomes, 93.7%. In contrast, Decision Tree Regressor has a score of 90.5% and SVR has a score of 87.8%.

4.2.4 One Hot Encoding

```
In [180]: dc_onehot = dc_onehot.drop(['year'], axis=1)
          dc_onehot.head()
```

Out[180]:

	round	grid	position order	points	race_70th Anniversary Grand Prix	race_Abu Dhabi Grand Prix	race_Australian Grand Prix	race_Austrian Grand Prix	race_Azerbaijan Grand Prix	race_Bahrain Grand Prix	...	status_Transmission	status_Turbo	s
0	1	14	15	0	0	0	1	0	0	0	...	0	0	
1	1	9	8	4	0	0	1	0	0	0	...	0	0	
2	1	5	5	10	0	0	1	0	0	0	...	0	0	
3	1	3	4	12	0	0	1	0	0	0	...	0	0	
4	1	6	18	0	0	0	1	0	0	0	...	0	0	

5 rows × 207 columns

Figure 4.7 One Hot Encoding before test the data

Figure 4.7 depicts the step after the train and comparison of the One Hot Encoding model. Before examining the data, the dataset demonstrates that this is the most appropriate model. Aside than that, the dataset has 5 rows and 207 columns.

4.2.5 Testing Data

```
In [185]: #Testing the data
#Setting test data to columns from DF and excluding 'points' values
test = pd.DataFrame(test_data, columns=dc_onehot.loc[:, dc_onehot.columns != 'points'].columns)

#Using stack function to return a reshaped DF by pivoting the columns of current df
sts = test[[col for col in test.columns if 'status' in col]].stack()[test[[col for col in test.columns if 'status' in col]].stack
stslist = list(pd.DataFrame(sts).index.get_level_values(1))
status = [i.split("_")[1] for i in stslist]

rc = test[[col for col in test.columns if 'race' in col]].stack()[test[[col for col in test.columns if 'race' in col]].stack()>0
rclist = list(pd.DataFrame(rc).index.get_level_values(1))
races = [i.split("_")[1] for i in rclist]

In [186]: test.head()
```

Out[186]:

	round	grid	position order	race_70th Anniversary Grand Prix	race_Abu Dhabi Grand Prix	race_Australian Grand Prix	race_Austrian Grand Prix	race_Azerbaijan Grand Prix	race_Bahrain Grand Prix	race_Belgian Grand Prix	...	status_Transmission	status
0	0.285714	0.40	0.210526	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	
1	0.809524	0.95	0.631579	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	
2	0.952381	0.10	0.052632	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	
3	0.238095	0.50	0.421053	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	
4	0.619048	0.40	0.210526	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	

5 rows × 206 columns

Figure 4.8 Testing Data

The stack function has been set up to return to columns from the Data Frame while ignoring data for 'points.' After we have finished setting the data, we will use the stack function to produce a reshaped Data Frame. This will be accomplished by pivoting the columns of the existing Data Frame.

4.2.6 Testing the Tested Data

```
In [187]: test.drop([col for col in test.columns if 'race' in col], axis=1, inplace=True)
          test.drop([col for col in test.columns if 'status' in col], axis=1, inplace=True)
          test.head()
```

Out[187]:

	round	grid	position order	Constructor_Alfa Romeo	Constructor_AlphaTauri	Constructor_Alpine F1 Team	Constructor_Aston Martin	Constructor_Ferrari	Constructor_Haas F1 Team	Constructo
0	0.285714	0.40	0.210526	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.809524	0.95	0.631579	0.0	0.0	1.0	0.0	0.0	0.0	
2	0.952381	0.10	0.052632	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.238095	0.50	0.421053	0.0	1.0	0.0	0.0	0.0	0.0	
4	0.619048	0.40	0.210526	0.0	0.0	0.0	0.0	0.0	0.0	

5 rows × 123 columns

```
In [188]: test ['race'] = races
          test['status'] = status
          test.head()
```

Out[188]:

	round	grid	position order	Constructor_Alfa Romeo	Constructor_AlphaTauri	Constructor_Alpine F1 Team	Constructor_Aston Martin	Constructor_Ferrari	Constructor_Haas F1 Team	Constructo
0	0.285714	0.40	0.210526	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.809524	0.95	0.631579	0.0	0.0	1.0	0.0	0.0	0.0	
2	0.952381	0.10	0.052632	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.238095	0.50	0.421053	0.0	1.0	0.0	0.0	0.0	0.0	
4	0.619048	0.40	0.210526	0.0	0.0	0.0	0.0	0.0	0.0	

5 rows × 125 columns

Figure 4.9 Drop and add the variables

Figure 4.9 depict the process of removing the ‘race’ and ‘status’ attributes to reshaped back the data frame. This dataset will be used for evaluation and modelling result.

4.3 Chapter Summary

In this chapter provide information regarding CRISP-DM phase 4 which is modelling were used in Python. In order to construct the model, a number of modelling techniques were tested and evaluated. This chapter explains the training and testing of data as well as the training and comparison of modelling techniques.

CHAPTER 5

EVALUATION AND RESULT

5.0 Overview

This chapter will discuss on the evaluation of the model that had been determined in Chapter 4. The results of the model will be discussed in detail in this chapter.

5.1 Random Forest Regressor

Decision Trees are utilised for both regression and classification challenges. A random forest is a nonlinear estimator that fits a number of categorising decision trees to different subsamples of the dataset and utilises averaging to increase predicted accuracy and prevent overfitting. In this model there is a process called bootstrapping where sampling of subsets from the dataset are randomly choose over a given number of iteration and a given number of variables. The outcome will be averaged to obtain an impactful result.

5.2 Evaluation

After the completion of modelling phase, the model's performance to get the view whether the objective have been met such as identifying the failures and reviewing the unexpected paths. If the model works perfectly, the model will be deployed to see the performance whether if it will match during the test.

5.2.1 R^2 Score

```
In [166]: logmodel = RandomForestRegressor()
          logmodel.fit(train_data,train_labels)

Out[166]: RandomForestRegressor()

In [167]: pred = logmodel.predict(test_data)

In [168]: clf = RandomForestRegressor()
          model = clf.fit(train_data, train_labels)

          test["predicted_finished"] = model.predict(test_data)
          test["actual_finished"] = pd.DataFrame(test_labels)["position order"].tolist()
          test_group = test.groupby("constructor")
          test_group.apply(lambda x: r2_score(x.actual_finished, x.predicted_finished))

Out[168]: constructor
          Alfa Romeo        0.810874
          AlphaTauri        0.834210
          Alpine F1 Team    0.934277
          Aston Martin      0.774538
          Ferrari           0.966955
          Haas F1 Team      0.563414
          McLaren           0.979566
          Mercedes          0.960440
          Racing Point      0.953609
          Red Bull          0.966348
          Renault           0.942555
          Toro Rosso        0.809916
          Williams          0.526182
          dtype: float64
```

Figure 5.1 Evaluation by using R^2 score

The R^2 score, which ranges from 0 to 1, indicates how closely one regression line matches the data from the dataset. It is impossible to accurately anticipate data variance using models that have a 0 or a 1 as their starting point.

5.3 Model Results

The result from modelling will provide the outcome of the model which associated with data cleansing and modelling. The outcome will be plot which will become a bar plot.

5.3.1 Features That Important to the Model

```
In [685]: #Model Results
          varimp= {'Significance':model.feature_importances_,'Name':dc_onehot.columns[dc_onehot.columns!="position order"]}

In [686]: #Plotting the features importance
          a4_dims = (8.27,16.7)

          fig, ax = plt.subplots(figsize=a4_dims)
          dc = pd.DataFrame.from_dict(varimp)
          dc.sort_values(ascending=False,by=["Significance"],inplace=True)
          dc = dc.dropna()
          sns.barplot(x="Significance", y="Name", palette="vlag", data=final, orient="h", ax=ax);
```

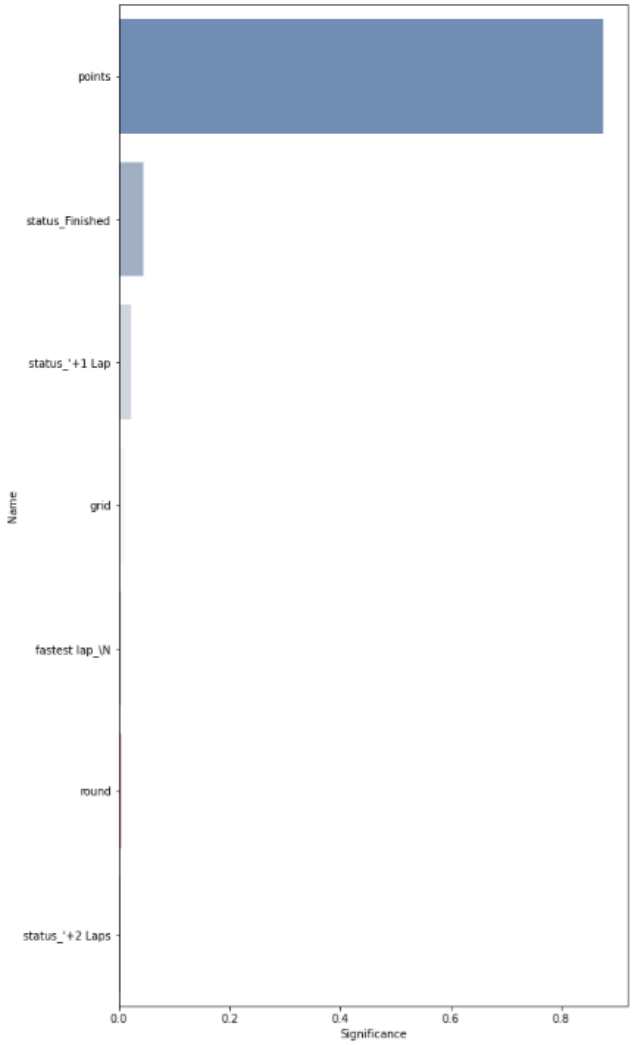

Figure 5.2 Bar plot from the model result

Figure 5.2 depicts the bar plot of the model's significant features. The significance of the feature's value is crucial to the model's decision-making process. In the visualisation dashboard, the three most vital aspects will be represented.

5.4 Visualisation Dashboard

Dashboards are data visualisation tools that track, monitor, analyse, and show Key Performance Indicators (KPIs), metrics, and key data points. Multiple data visualisation outputs (graphs) are assembled to enable users to actively engage in the analytics process. The cleanedd dataset that had been combined will be import into Power BI to start the visualisation process for this project.

5.4.1 Main Dashboard

Figure 5.3 Interactive dashboard for constructor performance analysis

Figure 5.3 shows the interactive dashboard to analysed the constructors team performance over the year from 2019 to 2022. This ddashboard contain multiple type of visualisation to make the audience comprehend better on the data that will be presented.

5.4.2 Points Gained by F1 Constructor Team

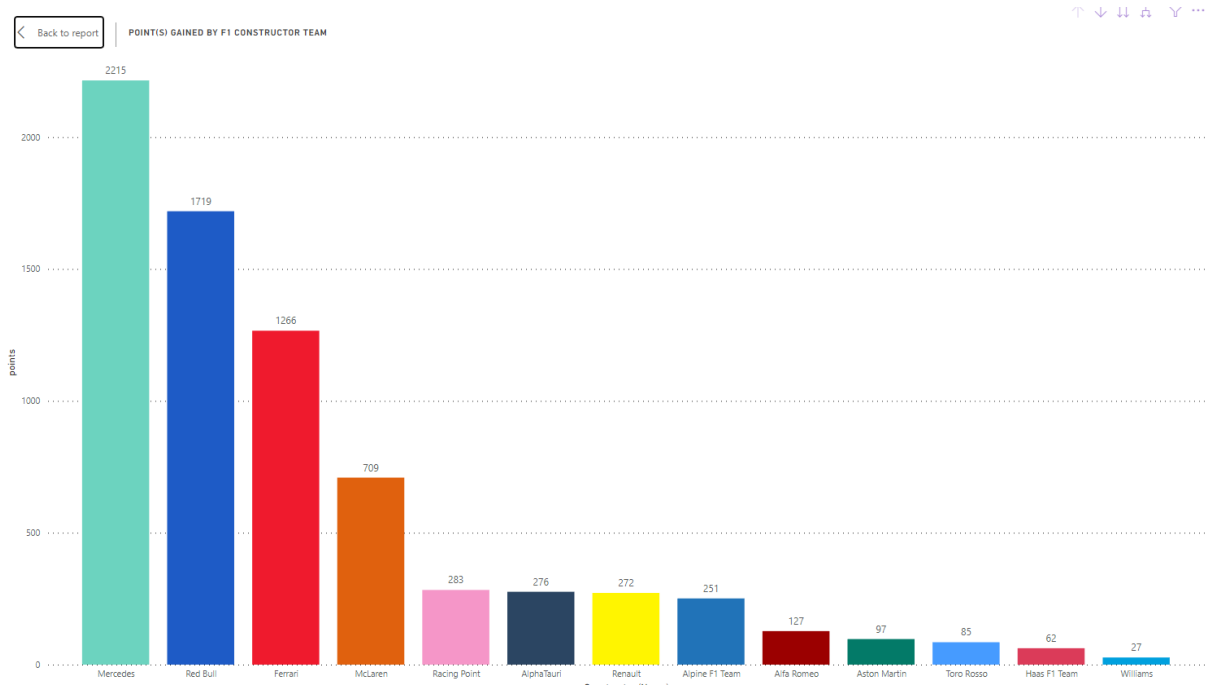

Figure 5.4 Total points scored by constructor team from 2019-2022

The column chart in figure 5.4 visualised the total point scored by the constructors team from 2019-2022. The team that scored the most points is Mercedes team with 2215 points then followed by Red Bull team with 1719 points.

5.4.3 Points Gained by F1 Constructor Team

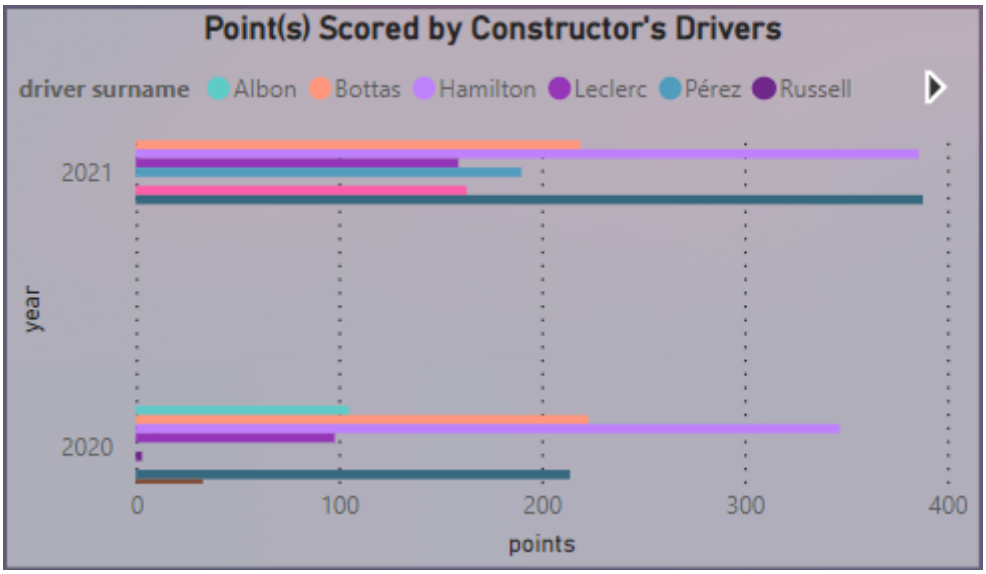

Figure 5.5 Points scored by constructor's drivers in 2020 and 2021

Figure 5.5 depicts in a clustered bar chart to visualised the points being scored by top 3 team in the championship for year 2020 and 2021. As shown in the chart, Hamilton had a consistent point scored for 2 years in a row.

5.4.4 Most Performed Circuit

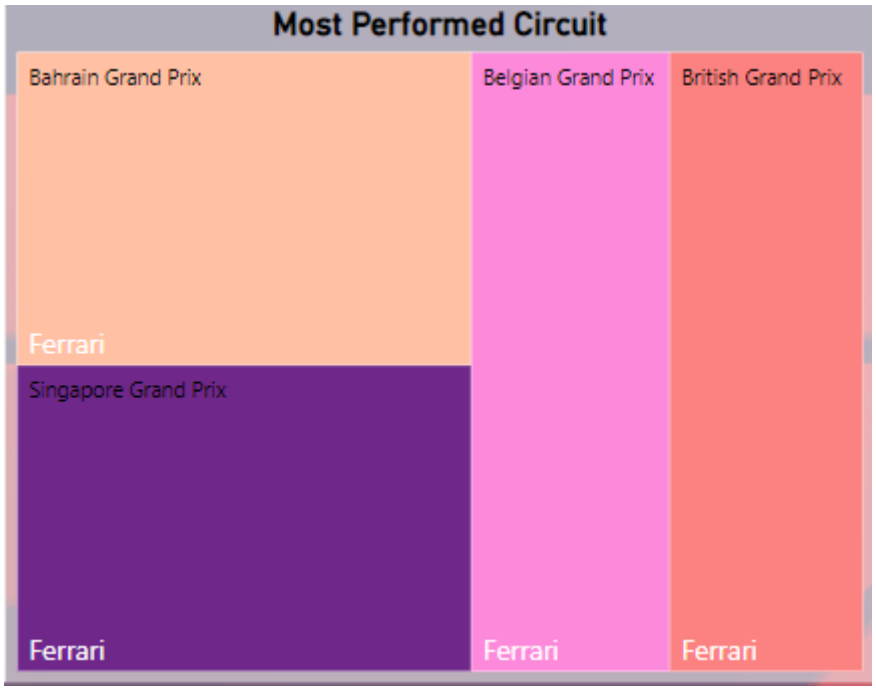

Figure 5.6 Most performed circuits by Ferrari team

Figure 5.6 shows a treemap of most performed circuit by Ferrari team for the year 2019 to 2022. This charts shows that the Ferrari team had the most points scored in these 4 Gran Prix (Race).

5.4.5 Constructor Team Performance by Year

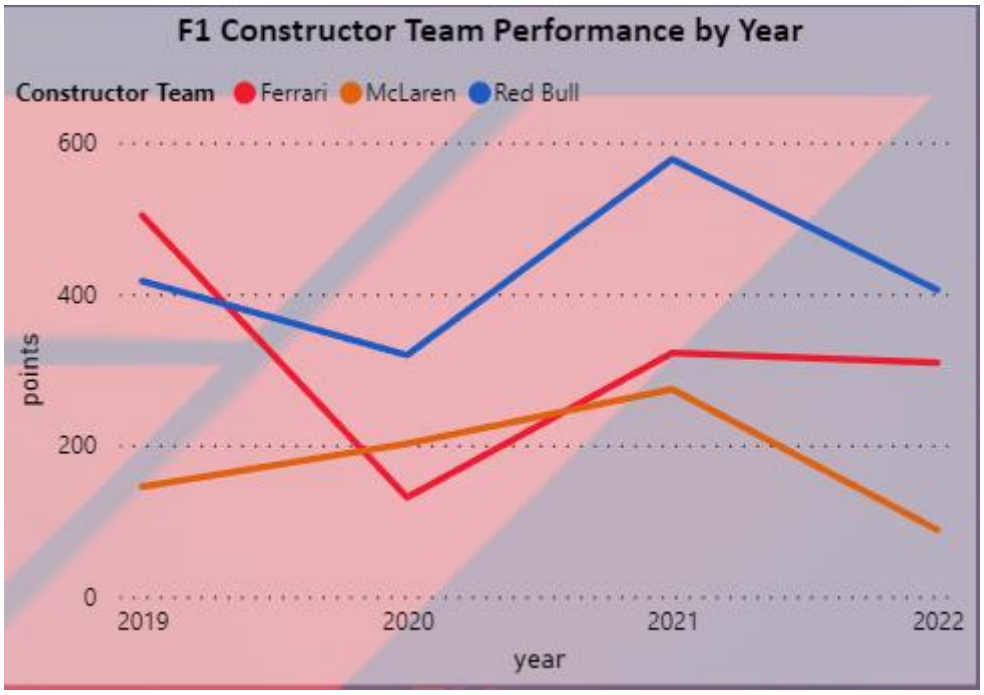

Figure 5.7 Ferrari, Red Bull, and McLaren performance based on points scored each year

Figure 5.7 shows a line chart to visualise the trend of Ferrari, Red Bull and McLaren team performance each year from 2019 to 2022. As visualised in the figure, Ferrari had a downfall in performance in 2020 where the team failedd to score as much points from the previous year while the Re Bull team had a great improvement in performance to secure more points for the last 2 years.

5.4.6 Count of Race Finished

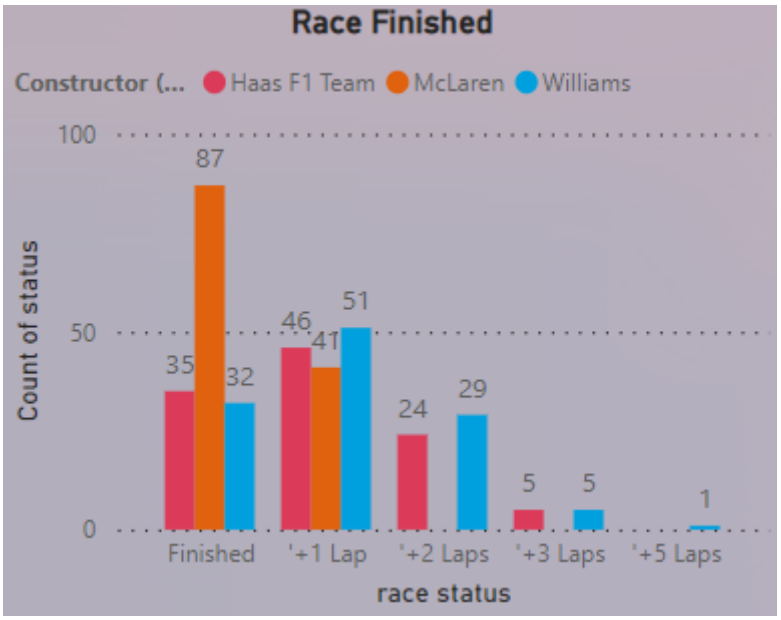

Figure 5.8 Number of races finished by Haas, McLaren, and Williams constructor teams

Figure 5.8 depicts the number of races being finished by Haas, McLaren, and Williams constructor teams from year 2019 to 2022. The Williams team had the worst performance where the team once finished the race with more than 5 laps remaining.

5.4.7 Count of Race Did Not Finished (DNF)

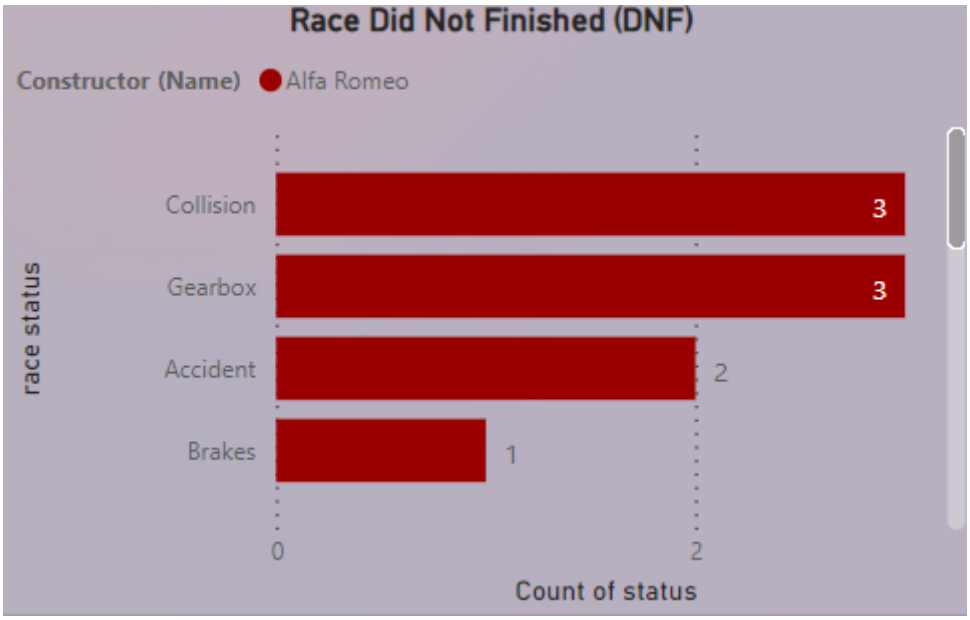

Figure 5.9 Number of race the Alfa Romeo team did not finish in 4 years

The figure 5.9 shown the number of races the Alfa Romeo team DNFs with the causes of the DNFs. The most causee for Alfa Romeo to not finished a race causs by collision and technical problem with the gearbox.

5.5 Chapter Summary

This chapter explained on how the evaluation of the model that had been trained and test in chapter 4 performed. Apart from that, the visualisation dashboard also had been improved from project 1. The visualisation being implement in the project had been explaine in this chapter. Prediction model also being explained in the chapter.

CHAPTER 6

CONCLUSION

6.0 Overview

This project will be concluded in this chapter. This chapter will be summarised all the chapter from Chapter 1 to Chapter 5, the project outcome, challenges and limitation will be discussed in this chapter.

6.1 Summary of Every Chapter

6.1.1 Chapter 1

In Chapter 1, the context of this project being discussed related to the Formula 1 motorsport. The project's problem statement and objectives had been determined. The project's scope and expected outcome had also been clarified in this chapter. Finally, the project timeline had been shown in this chapter to show the progression of the project for this semester.

6.1.2 Chapter 2

Analysis and system requirements were covered in this chapter where the research and analysis on this project being conducted. Research on the similar and most recent studies regarding the topic being explore in various form of data analytics that had been performed. Furthermore, the impact of data analytics in Formula 1 had been discussed in this chapter. Next, this chapter also covered the development methodology had been explain in detail to highlight the process. Finally, the techniques, methods, tools and technology being implement in this project being mentioned.

6.1.3 Chapter 3

In this chapter, data understanding and data preparation will be explained in detail to relate with the topic of this project. These phases helped to understand all the data that had been gathered to complete this project. The data preparation phase had included data cleaning process and Exploratory Data Analysis (EDA) to prepare the data before modelling phase.

6.1.4 Chapter 4

This chapter determined the project's modelling technique that suitable to be used for this project. The data that had been prepared in chapter 3 will be test and train to choose the most suitable modelling technique to be implement.

6.1.5 Chapter 5

This chapter had done the evaluation of the model being compute in chapter 4 previously. Furthermore, a visualisation dashboard also being shown and discuss on what the visualisation represents from the data.

6.2 Outcome of Project 1 and 2

Previously in Project 1, the problem statement of the project had been identified and objectives of the project had been discussed in Chapter 1 as an introduction of the project. Furthermore, after some research being done for this project in Chapter 2, we get to know data analysis had long been used in Formula 1 sport. Apart from that, in Chapter 3 we had the opportunity to see all the dataset and get to went through data understanding phase and data preparation phase to get to know the data that will be implement in the project.

In this Project 2, The dataset gained during Project 1had been improvised and enhanced to fit the data into the project objectives. Further analysis had been done through the data and the most important attributes had been chosen to create a model for the data. Other than that, the tools being used in the development of the project such as Power BI used to visualised the data and combining the dataset, Python also being used in this project as a tool for data preparation and modelling. Finally, a full report had been written for this Project 2 and a visualisation dashboard had been created to complete this project.

6.3 Problem Encounter

The problem being encounter during the development of this project such as lack of understanding for this project. Apart from that, the dataset being used in this project has many attributes that can be used to be implement in this project. In order to fit the data into this project, the scope of dataset had been downsized to make the project less confusing. It took some time to figure which attributes are the most suitable to be used in modelling phase. Next, with this semester have too many public holidays had been affected the project progress a little. Finally, with little knowledge on using the tools such as Power BI had made the progress of this project slower since a lot need to be understood in order to start using the software with all the function available.

6.4 Limitation

The were some limitations being encounter during the development of this project such as lack of certain function in Power BI to make the visualisation more interactive. Since the version of Power BI provided is a student version, some of advance function were not able to be used. Furthermore, with lack of experience and knowledge in data analytics had been one of the limitations that had been encountered during implementation of the project.

6.5 Chapter Summary

Overall, this chapter had covered all the summary from Chapter 1 to Chapter 5 which explained all the crucial information of each chapter in this documentation of the project. Other than that, the outcome of the Project 1 and Project 2 also being explained briefly. Finally, the problem and limitation during the implementation of the project also being discussed in this chapter.

REFERENCE

1. Catherine Cote, (2021) 4 types of data analytics to improve decision making. Retrieved from: https://online.hbs.edu/blog/post/types-of-data-analysis
2. Priya Pedamkar, Types of Data Analysis. Retrieved from: https://www.educba.com/types-of-data-analysis/
3. Emidio Amadebai, The 4 Types of Analytics Explained (With Examples). Retrieved from: https://www.analyticsfordecisions.com/types-of-analytics/
4. Bernard Marr, Big Data in Motorsports: How F1 And NASCAR Compete on Analytics. Retrieved from: https://www.smartdatacollective.com/big-data-motorsports-how-f1-and-nascar-compete-analytics/
5. Hyperight, (2020) Winning the race of data analytics: Formula 1. Retrieved from: https://hyperight.com/winning-the-race-of-data-analytics-formula-1/
6. Tom Robertson, (2021) Formula 1 & Big Data Analytics. Retrieved from: https://medium.com/analytics-vidhya/formula-1-big-data-analytics-cf333ddb6779
7. Frank Bi, (2014) How Formula One Teams Are Using Big Data To Get The Inside Edge. Retrieved from: https://www.forbes.com/sites/frankbi/2014/11/13/how-formula-one-teams-are-using-big-data-to-get-the-inside-edge/?sh=1e96d1005588
8. Ciarán Cooney, (2020) Formula One: Extracting and analysing historical results. Retrieved from: https://towardsdatascience.com/formula-one-extracting-and-analysing-historical-results-19c950cda1d1
9. Jiahui Wang, (2020) QlikView Visualization of Formula 1 (F1) Relational Data Model. Retrieved from: https://towardsdatascience.com/qlikview-visualization-of-formula-1-f1-relational-data-model-82e8b46c9f71

10. Zipporah Luna, (2021) CRISP-DM Phase 4: Modeling Phase. Retrieved from: https://medium.com/analytics-vidhya/crisp-dm-phase-4-modeling-phase-b81f2580ff3