National College of Ireland

Computing

Data Analytics

2021/2022

Andrew Kelly

X18212158

x18212158@student.ncirl.ie

Analysing the impact of the COVID 19 Pandemic on Association Football's Top 5 Leagues and its finances.

Technical Report

Contents

Executive Summary

This document will break down the approach that I have taken to complete the project. This project focuses on a variety of datasets that focus on the likes of transfer fees and market values of football players and also the revenues and expenditures of clubs in what are Europe's top 5 leagues. The primary objective was to analyse whether the Covid-19 pandemic had an impact on the spending and revenues of football clubs and to investigate whether other leagues were affected more than the English premier league.

This document will provide the reader with an understanding of what the project set out to achieve and the work that was done in order to complete it such as the languages and applications used to help in its completion.

On the 23rd of December 2021, the midpoint documentation was submitted along with a video recording that broke down what work had been completed thus far along with the work that is left to be completed. Not all the relevant techniques had been learnt early on in the project and were only picked up in our second semester through our data and web mining class and our advanced business data analysis class.

The projects final submission is due on the 15th of May 2022. All datasets were web scraped using Python from sources such as Transfermarkt.com and statista.com. The techniques used and analysis carried out allowed me to draw conclusions about how each league was affected by the Covid-19 pandemic and also use modelling to predict where transfer fees are going in future for football clubs in Europe's top 5 leagues. Analysis allowed me to find out public opinion in regard to transfer fees.

1.0 Introduction

1.1. Background

There were a variety of reasons as to why I undertook this particular project. For as long as I can remember I have loved football and it has been a noticeably big part of my life. I joined my local club at just 4 years of age and continued to play up until my early 20's where I had to stop due to other commitments but continued to play 5 a side at my local Astro weekly along with organising tournaments with groups of friends from other areas and schools to play against each other during the summer. As I have gotten older, analytics has become as big part of football analysis for both recruitment and assessment of a club's results. My interest in this area has grown over the years and with its growing prominence within the game I wanted to learn more about it and felt that this project would be the perfect time to marry the skills I have picked up throughout my 4 years in college with my love for football (Soccerment Research, 2020).

Throughout the Covid-19 pandemic much was made by journalists and football clubs about how they will struggle to survive the pandemic and sign players for their clubs. I was sceptical to these claims, particularly for clubs in the English premier league and is often described as a commercial giant and considered too big to fail. These claims of financial ruin were more believable for clubs in other leagues such as in France where they are struggling to secure a buyer for their TV rights along with Italy where it is routinely said how little money most clubs have in their league (MacInnes, 2021). The Deloitte football money report routinely shows the Premier League is the richest of them all in football (Deloitte, 2022).This project was the perfect time for me to investigate this and see if there was substance to the above claims. I knew of numerous resources online from previous research into the transfer fees of players and my following of football financial blogs such The Swiss Ramble and YouTube channels like HITC Sevens and the James Lawrence Allcott Channel. My social media also follows a lot of people who are in the football analytics space which allowed me to gain knowledge of some reliable and respected resources for the data that I would be collected.

Through my previous 3 years of learning combined with what I acquired from my internship and 4th year modules, I felt well equipped to take on this project and was confident that I would be able to achieve my goals and answer the questions set out at the start of the project.

1.2. Aims

This project aims to investigate the transfer fees from football clubs in Europe’s top 5 leagues have been both before, during, and after the COVID-19 pandemic. I will look at whether the English premier league clubs managed to deal with the pandemic better than the other 4 major European leagues which was the narrative driven by the media during the pandemic. I will use regression models to predict player prices while also using an LSTM model to predict player prices based off of historical transfer fees. I also intend to find out what public sentiment is about some of the world’s most high-profile transfers on the last year through the use of sentiment analysis.

This will be achieved by progressing with the tutorials and the classes that we have during our 4th year and carrying out my own extra study in order to gain the necessary skills to complete this project. Once this is done, I will apply these skills in order to obtain from sources that I have identified such as scraping transfer data for the fees paid by clubs for them along with various sources that will data contain for the likes of tv revenues gained by clubs and expenditures for each club. When this data is scraped, I can clean the dataset through methods such as removing null values and correcting any incorrect character types in that dataset. When the dataset is cleansed, I will apply numerous modelling techniques like regression and LSTM which I have learnt myself for the completion of this project. The technologies used for this project will be Google Colab, Pycharm, Python, and Excel.

When the different models are complete, I can then begin to interpret the different results that have been gained from then. When interpretations are made, I can then draw conclusions from then and decide whether the questions we have asked in the outset are answered. We will find out whether the English premier league coped better with the pandemic than their European counterparts and where transfer fees, revenues, and incomes are going (Taylor & Conn, 2020).

1.3. Technology

The technologies used for this project will be Google Colab, Pycharm, Python, SPSS, Excel, and Tableau.

Google Colab – This is an online IDE provided by Google which allows a user to write and execute Python code through their browser. It is often used for data analytics and machine learning. This will be an especially useful IDE for me when completing my project due to its Jupyter Notebook style and its ability to run Python code.

Pycharm – This is an IDE that is used in computer programming primarily for the Python programming language. This is an especially useful program as it provides code assistance, has a built-in debugger and integrated unit testing. This program will allow me to run any Python code that I have and create visualisations while also being an alternative IDE to Google Colab when any code does not run as wanted on it.

Python – Python is a high-level OOP. It has an easy to learn syntax with high level data structures that is especially useful. It is compatible with numerous operating systems. It is very commonly used in both data analytics and machine learning. Another benefit of Python is that is particularly good at discovering bugs in the code. This will allow me to run various models that I will use along with being the language that I will use to scrape my data and create my visualisations.

Excel – This is a spreadsheet program developed by Microsoft. It has a range of features including calculation, creating graphs, and computation. For this project I will be using Excel in the form of CSV files in order to store my data. These spreadsheets will be created by writing any data scraped to a csv file using Python. This will be especially useful for this project.

1.4 Structure

Section 2 – This is a breakdown of the datasets used in the project. How I went about finding this data and choosing it. How I will go about using the data in the project.

Section 3 – This will breakdown how I went about conducting my analysis. I will go step by step in to how each analysis technique was carried out and how I scraped the data. It will break down how the data is pre-processed and cleansed for use.

Section 4 – This will outline approaches that I took to analysing my data. The different options available to me and why I picked what I picked. I will justify any decisions that I made in regard to this.

Section 5 – This is where I will display the results of my analysis. This will be divided up by section and what they were trying to achieve, and each result will be supported by context.

Section 6 – This is where conclusions will be drawn from the findings. I will outline the strengths and limitations of my project.

Section 7 – This will discuss what the project could be or where I could with it with more access to time and resources and where I would like to take the project in future and how it may lead me down paths in future.

Section 8 – Here is where I will include the references of any sources used throughout the project. Harvard referencing will be used for the project.

Section 9 – This section is for the appendices of the project. This will contain the likes of my project proposal and my monthly reflective journals throughout the project.

2.0 Data

For this report there are a variety of different datasets being used. The primary dataset is a dataset I made by merging multiple datasets that I scraped from transfermarkt.com. This dataset contains details of transfers made by football clubs since the 2015/2016 season. The data that I had scraped contained transfer data from all leagues in the world that transfermarkt.com had data for, in my case I only needed data from Europe's top 5 leagues as these leagues give you the best idea as to where transfer fees will go in future opposed to the lower fees spent on players by leagues outside of the top 5. I would have to do some cleansing in order to remove these leagues and only retain the leagues that were relevant to this report. We were only retaining transfers that came in to the top 5 leagues, transfers from the top 5 leagues to a league outside of this category would be removed from the dataset. Another field that was removed from the dataset would be that of free transfers and loans as these only make up a small portion of the market and would impact our results of the likes of our LSTM model and regression testing. Free transfers and loans will not help give an idea of where transfer fees are going are much less used by clubs in the top 5 leagues opposed to their smaller league counterparts. Due to some issues with writing all transfers from each season from 2015/2016 to 2021/2022 to the same csv file, each season would have to be scraped individually to its own csv file and I would then later merge all csv files to one as it would make the modelling process more efficient. The reasoning behind using transfermarkt.com as my data source for the transfer data is due to it being renowned as the gold standard for details in relation to transfers in football and valuations of players along with having data spanning a long period of time which could be useful in future for using even larger sample sizes of data. This dataset will contain fields such as the age of the player transferred, the club they are transferred from, the club they are transferred to, the market value of the player, the players name, the transfer fee paid for the player, the league that the player has transferred to, and the season that the transfer occurred. The primarily used fields for my analysis will be the age, market value, transfer fee and the year that the transfer occurred. These fields will all be especially useful in our regression and LSTM models.

Other datasets include data that has been scraped from statista.com. This data primarily consisted of financial data of clubs in Europe's top 5 league's which includes their revenues and incomes. This will allow me to compare how incomes of the different leagues were affected by the Covid-19 pandemic as the data spans from both before, during, and after the pandemic. We will also be able to see the top earning clubs, what clubs spend the most along with what clubs make the most from kit sponsors and what leagues make the most from TV revenue where you will expect to see English premier league clubs earning the most. Fields in these datasets will include tv revenue earned, ticket sales, commercial earnings, top earning clubs, and kit sponsors. These fields will be particularly useful in showing us how different leagues were affected by the Covid-19 pandemic and being able to compare them over periods of time. These datasets will be scraped using Python and will be stored in csv files.

The final dataset used will be tweets that are collected for sentiment analysis. Python code will be run, and API key is used from twitters developer platform. Tweets will be extracted from Twitter and stored in a csv file. The csv file will then be read by code to analyse it using the Vader lexicon library which can interpret the likes of emojis in order to be able to assign a polarity score of positive, negative, or neutral. These values will be stored in a new csv file which can then be read by Python code and executed to produce visualisations showing how well received transfer fees from specific transfers were received by the public. This will be highly effective in finding out what public opinion is in regard to where transfer fees are headed in future as some people believe that the money involved in these transfers is far too high.

2.0.1 Tables breaking down details of each dataset collected

Table 1

Table Breaking Down Each Dataset		
Dataset	***Top Transfers***	***Sentiment Test***
File Format	CSV	CSV
Structure Type	Structured	Structured
Size	67 KB	148 KB
Number of instasnces	765	700
Number of attributes	9	8
Type	Web Scraped	Web Scraped

Table 2

Table Breaking Down Each Dataset		
Dataset	***20 Highest Revenue***	***PL Rev***
File Format	CSV	CSV
Structure Type	Structured	Structured
Size	1 KB	1 KB
Number of instasnces	20	20
Number of attributes	2	2
Type	Web Scraped	Web Scraped

Table 3

Table Breaking Down Each Dataset		
Dataset	***PL Salary***	***PL Sponsor***
File Format	CSV	CSV
Structure Type	Structured	Structured
Size	1 KB	1 KB
Number of instasnces	20	20
Number of attributes	2	2
Type	Web Scraped	Web Scraped

Table 4

Table Breaking Down Each Dataset		
Dataset	***Ligue 1 Salary***	***Serie A Salary***
File Format	CSV	CSV
Structure Type	Structured	Structured
Size	1 KB	1 KB
Number of instasnces	20	20
Number of attributes	2	2
Type	Web Scraped	Web Scraped

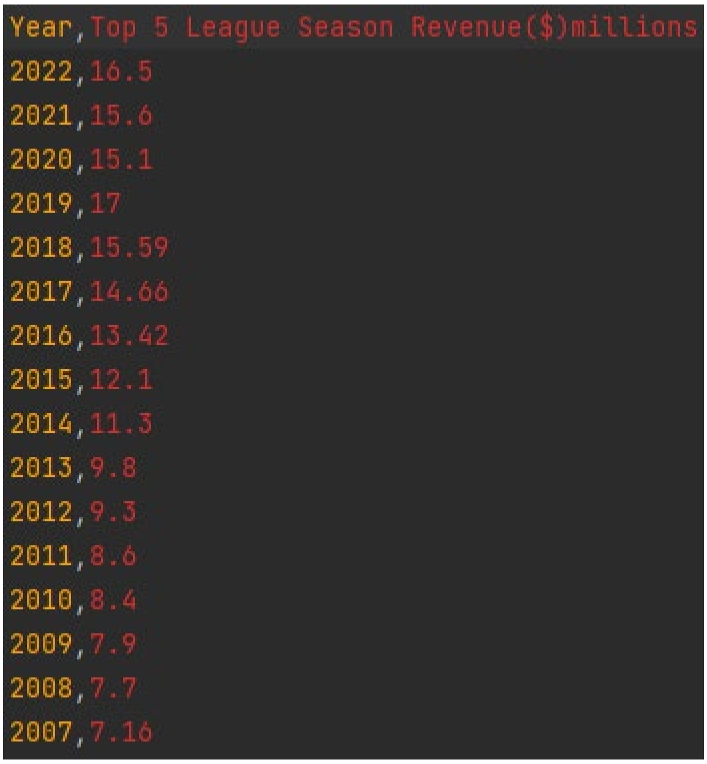
Table 5

Table Breaking Down Each Dataset		
Dataset	***Bundesliga Salary***	***La Liga Salary***
File Format	CSV	CSV
Structure Type	Structured	Structured
Size	1 KB	1 KB
Number of instasnces	18	20
Number of attributes	2	2
Type	Web Scraped	Web Scraped

Table 6

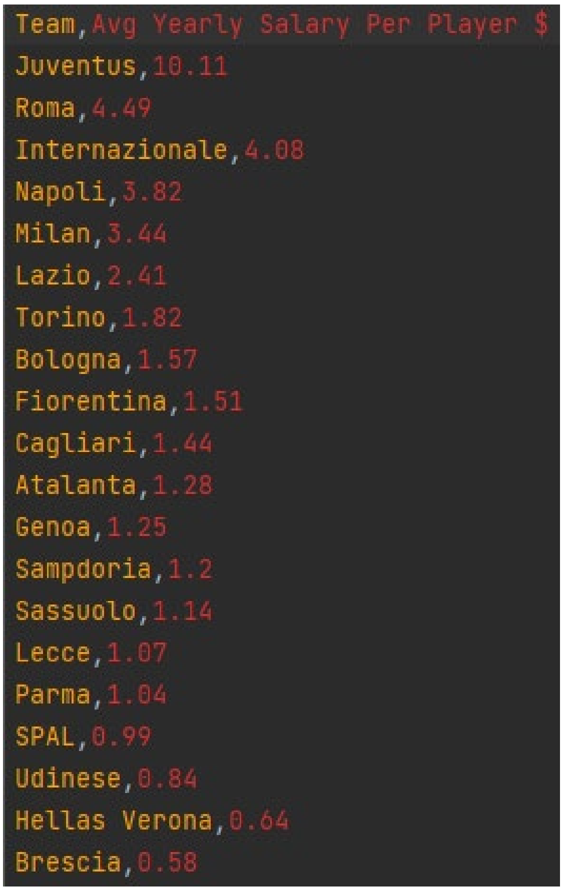
Table Breaking Down Each Dataset	
Dataset	***Top 5 League Total Revenue***
File Format	CSV
Structure Type	Structured
Size	1 KB
Number of instasnces	16
Number of attributes	2
Type	Web Scraped

2.0.2 Sample screenshots of datasets collected



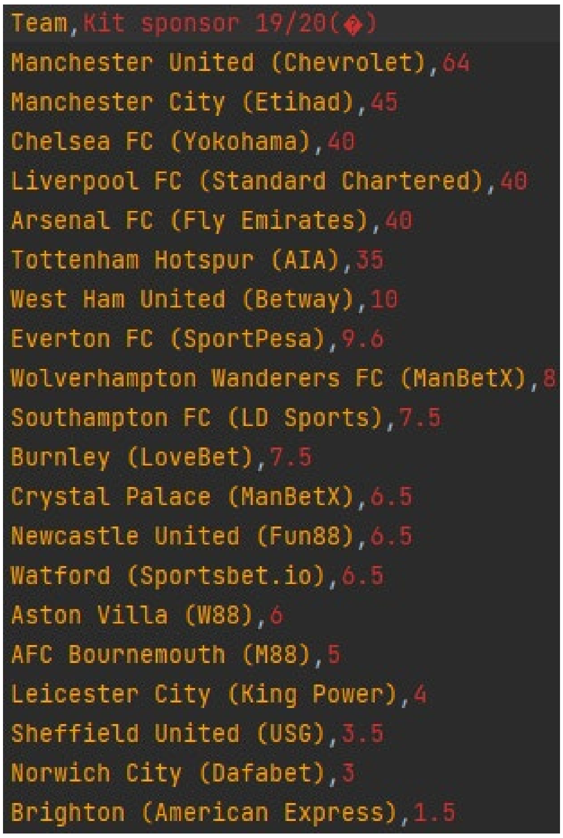
```
Year,Top 5 League Season Revenue($)millions
2022,16.5
2021,15.6
2020,15.1
2019,17
2018,15.59
2017,14.66
2016,13.42
2015,12.1
2014,11.3
2013,9.8
2012,9.3
2011,8.6
2010,8.4
2009,7.9
2008,7.7
2007,7.16
```

Figure 1



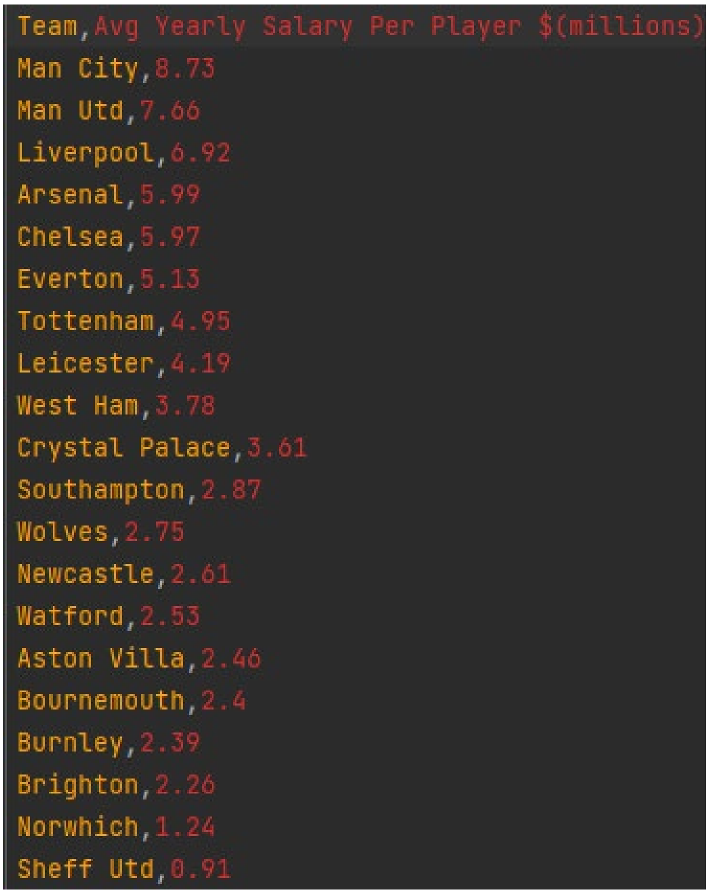
```
Team,Avg Yearly Salary Per Player $
Juventus,10.11
Roma,4.49
Internazionale,4.08
Napoli,3.82
Milan,3.44
Lazio,2.41
Torino,1.82
Bologna,1.57
Fiorentina,1.51
Cagliari,1.44
Atalanta,1.28
Genoa,1.25
Sampdoria,1.2
Sassuolo,1.14
Lecce,1.07
Parma,1.04
SPAL,0.99
Udinese,0.84
Hellas Verona,0.64
Brescia,0.58
```

Figure 2



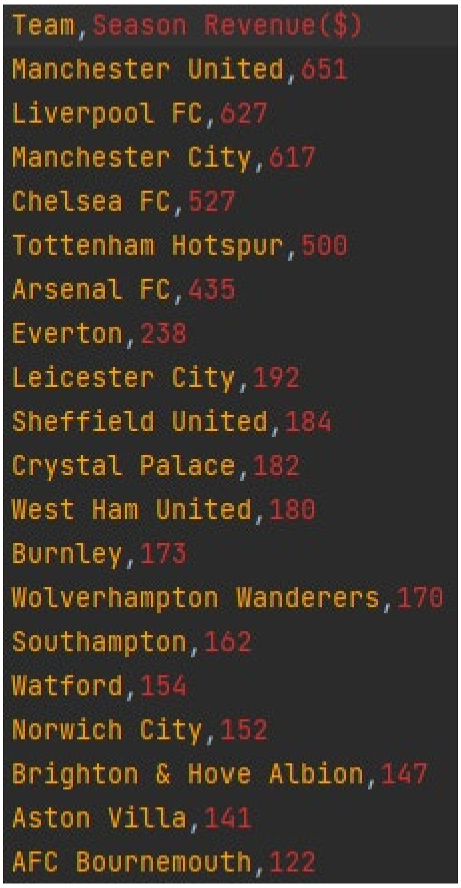
```
Team,Kit sponsor 19/20(�)
Manchester United (Chevrolet),64
Manchester City (Etihad),45
Chelsea FC (Yokohama),40
Liverpool FC (Standard Chartered),40
Arsenal FC (Fly Emirates),40
Tottenham Hotspur (AIA),35
West Ham United (Betway),10
Everton FC (SportPesa),9.6
Wolverhampton Wanderers FC (ManBetX),8
Southampton FC (LD Sports),7.5
Burnley (LoveBet),7.5
Crystal Palace (ManBetX),6.5
Newcastle United (Fun88),6.5
Watford (Sportsbet.io),6.5
Aston Villa (W88),6
AFC Bournemouth (M88),5
Leicester City (King Power),4
Sheffield United (USG),3.5
Norwich City (Dafabet),3
Brighton (American Express),1.5
```

Figure 3



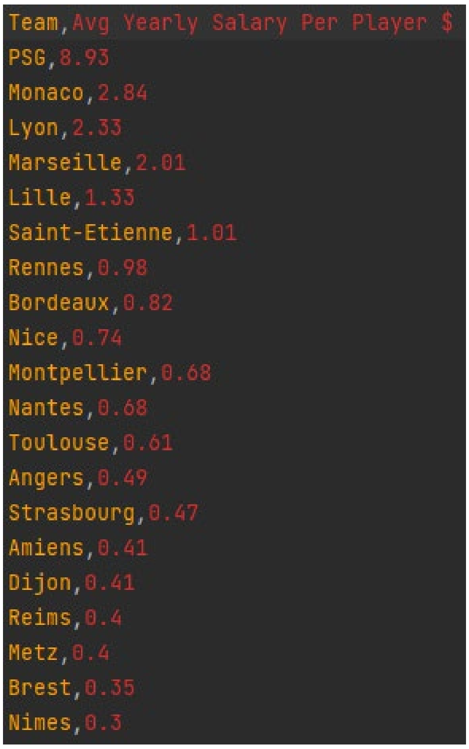
```
Team,Avg Yearly Salary Per Player $(millions)
Man City,8.73
Man Utd,7.66
Liverpool,6.92
Arsenal,5.99
Chelsea,5.97
Everton,5.13
Tottenham,4.95
Leicester,4.19
West Ham,3.78
Crystal Palace,3.61
Southampton,2.87
Wolves,2.75
Newcastle,2.61
Watford,2.53
Aston Villa,2.46
Bournemouth,2.4
Burnley,2.39
Brighton,2.26
Norwhich,1.24
Sheff Utd,0.91
```

Figure 4



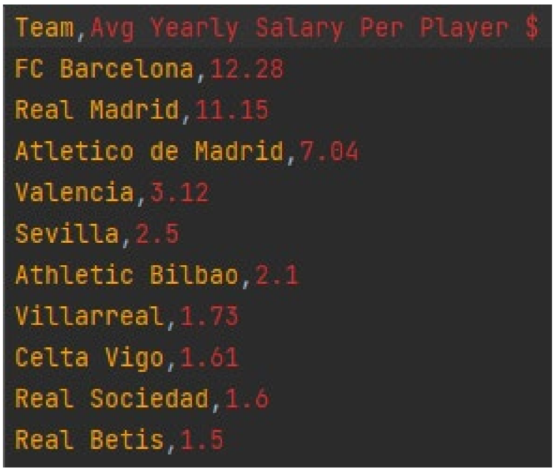
```
Team,Season Revenue($)
Manchester United,651
Liverpool FC,627
Manchester City,617
Chelsea FC,527
Tottenham Hotspur,500
Arsenal FC,435
Everton,238
Leicester City,192
Sheffield United,184
Crystal Palace,182
West Ham United,180
Burnley,173
Wolverhampton Wanderers,170
Southampton,162
Watford,154
Norwich City,152
Brighton & Hove Albion,147
Aston Villa,141
AFC Bournemouth,122
```

Figure 5



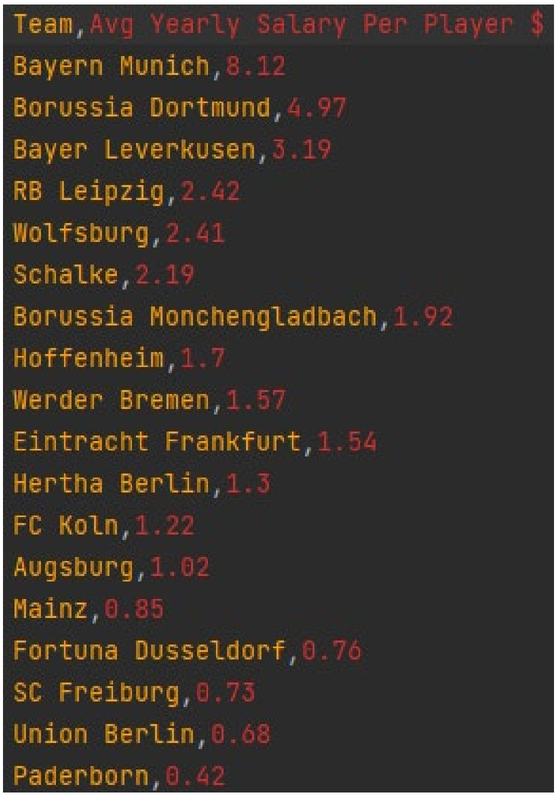
```
Team,Avg Yearly Salary Per Player $
PSG,8.93
Monaco,2.84
Lyon,2.33
Marseille,2.01
Lille,1.33
Saint-Etienne,1.01
Rennes,0.98
Bordeaux,0.82
Nice,0.74
Montpellier,0.68
Nantes,0.68
Toulouse,0.61
Angers,0.49
Strasbourg,0.47
Amiens,0.41
Dijon,0.41
Reims,0.4
Metz,0.4
Brest,0.35
Nimes,0.3
```

Figure 6



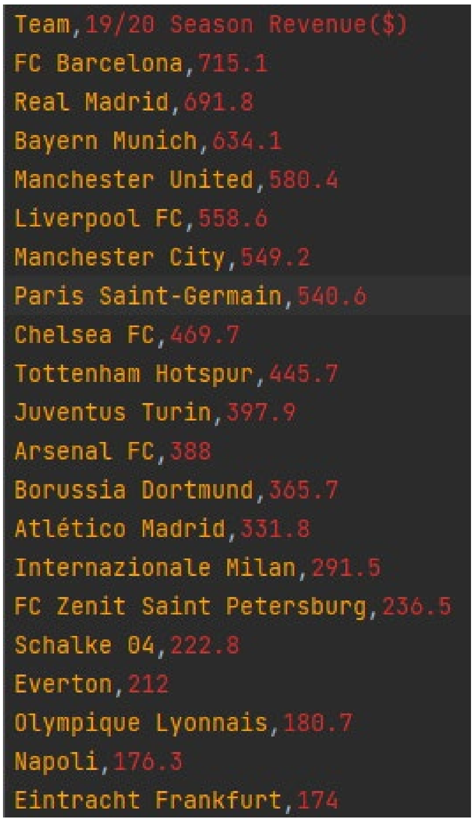
```
Team,Avg Yearly Salary Per Player $
FC Barcelona,12.28
Real Madrid,11.15
Atletico de Madrid,7.04
Valencia,3.12
Sevilla,2.5
Athletic Bilbao,2.1
Villarreal,1.73
Celta Vigo,1.61
Real Sociedad,1.6
Real Betis,1.5
```

Figure 7



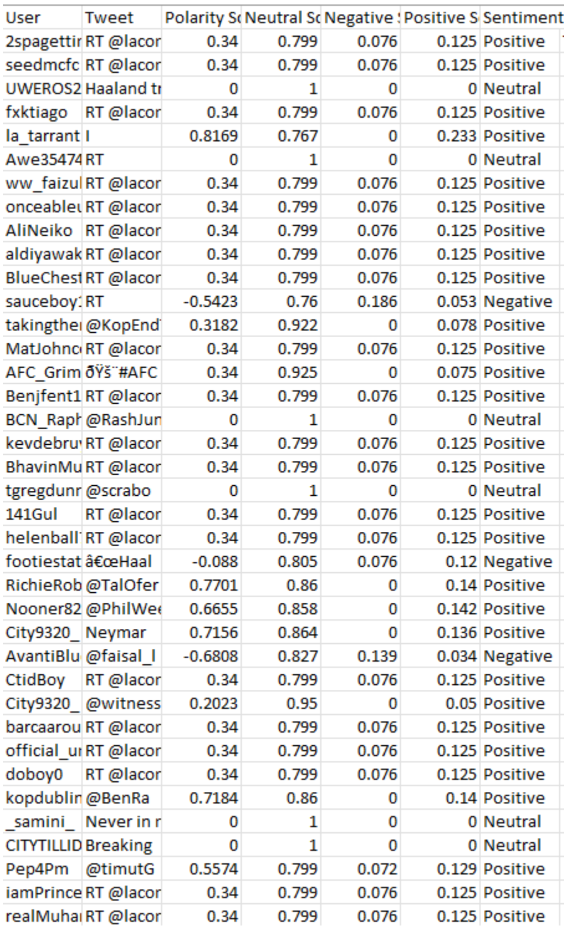
```
Team,Avg Yearly Salary Per Player $
Bayern Munich,8.12
Borussia Dortmund,4.97
Bayer Leverkusen,3.19
RB Leipzig,2.42
Wolfsburg,2.41
Schalke,2.19
Borussia Monchengladbach,1.92
Hoffenheim,1.7
Werder Bremen,1.57
Eintracht Frankfurt,1.54
Hertha Berlin,1.3
FC Koln,1.22
Augsburg,1.02
Mainz,0.85
Fortuna Dusseldorf,0.76
SC Freiburg,0.73
Union Berlin,0.68
Paderborn,0.42
```

Figure 8



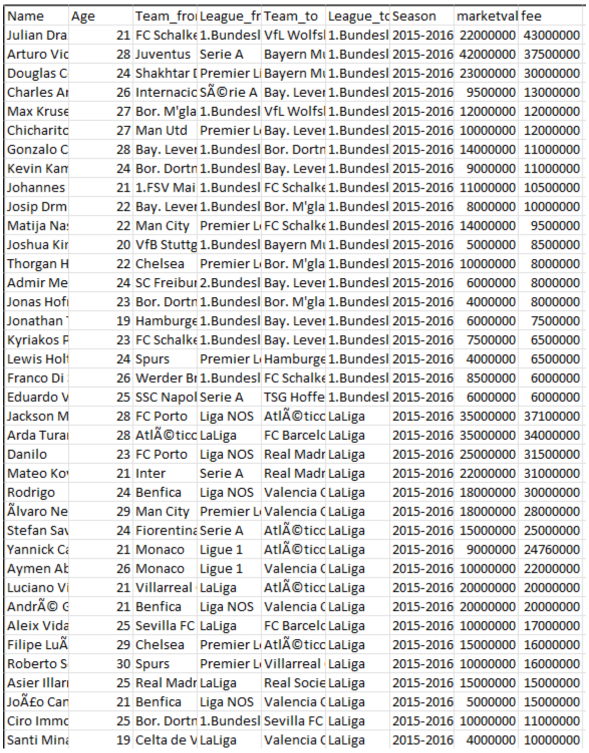
```
Team,19/20 Season Revenue($)
FC Barcelona,715.1
Real Madrid,691.8
Bayern Munich,634.1
Manchester United,580.4
Liverpool FC,558.6
Manchester City,549.2
Paris Saint-Germain,540.6
Chelsea FC,469.7
Tottenham Hotspur,445.7
Juventus Turin,397.9
Arsenal FC,388
Borussia Dortmund,365.7
Atlético Madrid,331.8
Internazionale Milan,291.5
FC Zenit Saint Petersburg,236.5
Schalke 04,222.8
Everton,212
Olympique Lyonnais,180.7
Napoli,176.3
Eintracht Frankfurt,174
```

Figure 9



User	Tweet	Polarity Sc	Neutral Sc	Negative :	Positive S	Sentiment
2spagettir	RT @lacor	0.34	0.799	0.076	0.125	Positive
seedmcfc	RT @lacor	0.34	0.799	0.076	0.125	Positive
UWEROS2	Haaland tr	0	1	0	0	Neutral
fxktiago	RT @lacor	0.34	0.799	0.076	0.125	Positive
la_tarrant	I	0.8169	0.767	0	0.233	Positive
Awe35474	RT	0	1	0	0	Neutral
ww_faizul	RT @lacor	0.34	0.799	0.076	0.125	Positive
onceableu	RT @lacor	0.34	0.799	0.076	0.125	Positive
AliNeiko	RT @lacor	0.34	0.799	0.076	0.125	Positive
aldiyawak	RT @lacor	0.34	0.799	0.076	0.125	Positive
BlueChest	RT @lacor	0.34	0.799	0.076	0.125	Positive
sauceboy:	RT	-0.5423	0.76	0.186	0.053	Negative
takingthe	@KopEnd	0.3182	0.922	0	0.078	Positive
MatJohnc	RT @lacor	0.34	0.799	0.076	0.125	Positive
AFC_Grim	ðŸš¨#AFC	0.34	0.925	0	0.075	Positive
Benjfent1	RT @lacor	0.34	0.799	0.076	0.125	Positive
BCN_Rapl	@RashJur	0	1	0	0	Neutral
kevdebru	RT @lacor	0.34	0.799	0.076	0.125	Positive
BhavinMu	RT @lacor	0.34	0.799	0.076	0.125	Positive
tgregdunr	@scrabo	0	1	0	0	Neutral
141Gul	RT @lacor	0.34	0.799	0.076	0.125	Positive
helenball	RT @lacor	0.34	0.799	0.076	0.125	Positive
footiestat	â€œHaal	-0.088	0.805	0.076	0.12	Negative
RichieRob	@TalOfer	0.7701	0.86	0	0.14	Positive
Nooner82	@PhilWe	0.6655	0.858	0	0.142	Positive
City9320_	Neymar	0.7156	0.864	0	0.136	Positive
AvantiBlu	@faisal_l	-0.6808	0.827	0.139	0.034	Negative
CtidBoy	RT @lacor	0.34	0.799	0.076	0.125	Positive
City9320_	@witness	0.2023	0.95	0	0.05	Positive
barcaarou	RT @lacor	0.34	0.799	0.076	0.125	Positive
official_u	RT @lacor	0.34	0.799	0.076	0.125	Positive
doboy0	RT @lacor	0.34	0.799	0.076	0.125	Positive
kopdublin	@BenRa	0.7184	0.86	0	0.14	Positive
samini	Never in r	0	1	0	0	Neutral
CITYTILLID	Breaking	0	1	0	0	Neutral
Pep4Pm	@timutG	0.5574	0.799	0.072	0.129	Positive
iamPrince	RT @lacor	0.34	0.799	0.076	0.125	Positive
realMuha	RT @lacor	0.34	0.799	0.076	0.125	Positive

Figure 10



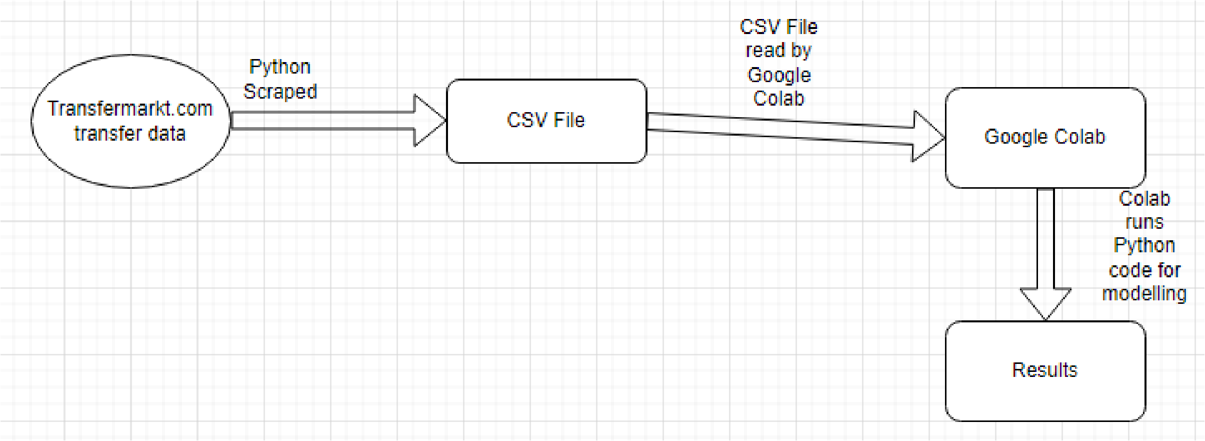
Name	Age	Team_fro	League_fr	Team_to	League_tc	Season	marketval	fee
Julian Dra	21	FC Schalke	1.Bundesl	VfL Wolfsl	1.Bundesl	2015-2016	22000000	43000000
Arturo Vic	28	Juventus	Serie A	Bayern Mu	1.Bundesl	2015-2016	42000000	37500000
Douglas C	24	Shakhtar D	Premier Li	Bayern Mu	1.Bundesl	2015-2016	23000000	30000000
Charles Ar	26	Internacic	SÃ©rie A	Bay. Lever	1.Bundesl	2015-2016	9500000	13000000
Max Kruse	27	Bor. M'gla	1.Bundesl	VfL Wolfsl	1.Bundesl	2015-2016	12000000	12000000
Chicharitc	27	Man Utd	Premier L	Bay. Lever	1.Bundesl	2015-2016	10000000	12000000
Gonzalo C	28	Bay. Lever	1.Bundesl	Bor. Dortn	1.Bundesl	2015-2016	14000000	11000000
Kevin Kam	24	Bor. Dortn	1.Bundesl	Bay. Lever	1.Bundesl	2015-2016	9000000	11000000
Johannes	21	1.FSV Mai	1.Bundesl	FC Schalke	1.Bundesl	2015-2016	11000000	10500000
Josip Drm	22	Bay. Lever	1.Bundesl	Bor. M'gla	1.Bundesl	2015-2016	8000000	10000000
Matija Na:	22	Man City	Premier L	FC Schalke	1.Bundesl	2015-2016	14000000	9500000
Joshua Kir	20	VfB Stuttg	1.Bundesl	Bayern Mu	1.Bundesl	2015-2016	5000000	8500000
Thorgan H	22	Chelsea	Premier L	Bor. M'gla	1.Bundesl	2015-2016	10000000	8000000
Admir Me	24	SC Freibur	2.Bundesl	Bay. Lever	1.Bundesl	2015-2016	6000000	8000000
Jonas Hofi	23	Bor. Dortn	1.Bundesl	Bor. M'gla	1.Bundesl	2015-2016	4000000	8000000
Jonathan T	19	Hamburge	1.Bundesl	Bay. Lever	1.Bundesl	2015-2016	6000000	7500000
Kyriakos P	23	FC Schalke	1.Bundesl	Bay. Lever	1.Bundesl	2015-2016	7500000	6500000
Lewis Holt	24	Spurs	Premier L	Hamburge	1.Bundesl	2015-2016	4000000	6500000
Franco Di	26	Werder Br	1.Bundesl	FC Schalke	1.Bundesl	2015-2016	8500000	6000000
Eduardo V	25	SSC Napol	Serie A	TSG Hoffe	1.Bundesl	2015-2016	6000000	6000000
Jackson M	28	FC Porto	Liga NOS	AtlÃ©ticc	LaLiga	2015-2016	35000000	37100000
Arda Tura	28	AtlÃ©ticc	LaLiga	FC Barcelc	LaLiga	2015-2016	35000000	34000000
Danilo	23	FC Porto	Liga NOS	Real Madr	LaLiga	2015-2016	25000000	31500000
Mateo Ko	21	Inter	Serie A	Real Madr	LaLiga	2015-2016	22000000	31000000
Rodrigo	24	Benfica	Liga NOS	Valencia C	LaLiga	2015-2016	18000000	30000000
Ãlvaro Ne	29	Man City	Premier L	Valencia C	LaLiga	2015-2016	18000000	28000000
Stefan Sav	24	Fiorentina	Serie A	AtlÃ©ticc	LaLiga	2015-2016	15000000	25000000
Yannick Ca	21	Monaco	Ligue 1	AtlÃ©ticc	LaLiga	2015-2016	9000000	24760000
Aymen Ab	26	Monaco	Ligue 1	Valencia C	LaLiga	2015-2016	10000000	22000000
Luciano Vi	21	Villarreal	LaLiga	AtlÃ©ticc	LaLiga	2015-2016	20000000	20000000
AndrÃ© G	21	Benfica	Liga NOS	Valencia C	LaLiga	2015-2016	20000000	20000000
Aleix Vida	25	Sevilla FC	LaLiga	FC Barcelc	LaLiga	2015-2016	10000000	17000000
Filipe LuÃ	29	Chelsea	Premier L	AtlÃ©ticc	LaLiga	2015-2016	15000000	16000000
Roberto S	30	Spurs	Premier L	Villarreal	LaLiga	2015-2016	10000000	16000000
Asier Illari	25	Real Madr	LaLiga	Real Socie	LaLiga	2015-2016	15000000	15000000
JoÃ£o Can	21	Benfica	Liga NOS	Valencia C	LaLiga	2015-2016	5000000	15000000
Ciro Immc	25	Bor. Dortn	1.Bundesl	Sevilla FC	LaLiga	2015-2016	10000000	11000000
Santi Mina	19	Celta de V	LaLiga	Valencia C	LaLiga	2015-2016	4000000	10000000

Figure 11

2.0.3 Data Dictionary

Details of every dataset can be seen above in Table's 1-6 above.

Implementation – The datasets were scraped using Python and stored in csv files that could be opened and viewed in my IDE's which were Google Colab and PyCharm along with Excel. These datasets were cleansed in preparation for the modelling and analysis phase of the project. Analysis of the datasets can be carried out using Python code and can create visualisations to display our results for interpretation. Analysis of the datasets will be carried out in Python through Google Colab, PyCharm, and SPSS along with the potential for some extra visualisations to be created with Tableau if the Python library does not exist for such visualisations.


Transfermarkt.com transfer data
Python Scraped
CSV File
CSV File read by Google Colab
Google Colab
Colab runs Python code for modelling
Results

3.0 Methodology

For this project I will be using the KDD methodology. Steps seen in Figure 12.

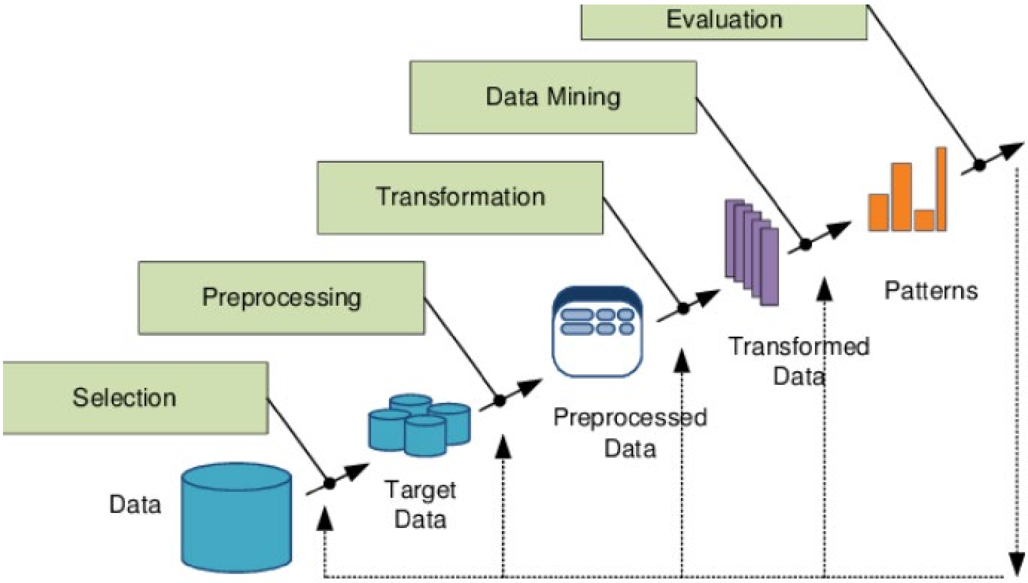


Figure 12

(Costaglio, et al., 2009).

3.0.1 Data Selection

This is part of the KDD methodology where the researcher will find any data that could be relevant to what they are trying to find out and to the topic at hand. Once the relevant data is found it can then be scraped in this case or the files can be downloaded should you find a pre-existing dataset from a source such as Kaggle or GitHub. In this instance all datasets will be scraped from various sources. The sources of data for this project would be transfermarkt.com and statista.com. The primary pages within this dataset that will be explored is the transfers section of the site where there are separate web pages for each season containing every transfer. Due to potential difficulty in filtering out leagues that are not necessary for the analysis in the scraping phase, it is likely that all transfer will be scraped and then cleansed removing it of any leagues that are not needed for modelling. It was likely that the transfermarkt data set that will be scraped will contain values such as "free transfer" and "loan transfer" which is something that the researcher would want to account for when completing their analysis and these would not be suitable for the analysis conducted. The dataset could also contain incorrect characters so they could need to be converted but due to the likes of player names, where these characters will mostly occur, not being a particularly necessary field for the analysis being conducted, this may not be a factor.

It did not require much research to find both of these data sources. Transfermarkt.com is considered a standard by people within football for accurate and extensive data on transfers of football players and the details of these transfers along with data relating to the contract status of a player. Statista is not as well-known but has an extensive set of data relating to financial figures in football including the likes of incomes and expenditures by clubs. This data would be suitable to be scraped and used for analysis in relation to money made and spent by clubs in Europe's top 5 leagues before, during, and after the Covid-19 pandemic.

All of the above data sources contained the relevant information with suitable numbers of variables and instances to produce significant results.

3.0.2 Pre-processing

In this section of the project the researcher will pre-process their data. The objective is to get the dataset in to an ideal state for analysis by cleansing and removing any unnecessary data. Once the relevant datasets were chosen and scraped, the researcher can then commence with cleansing the dataset.

The first step was to remove any rows that in the "league_to" did not contain one of "1.Bundesliga," "Premier League," "Serie A," "La Liga," and "Ligue 1". This is due to our analysis only being based around these 5 leagues while the scraped data contained all major leagues around the world. A sample of the dataset containing only these leagues can be seen above in Figure 11. This cleansing would be completed by uploading the dataset to IDE and executing Python code to remove these rows.

The next step would be to remove to remove any rows that contain "free transfer" and "loan transfer" in the "transfer fee" column. These value types were not relevant for the analysis as we only wanted permanent transfer where a monetary transfer fee was exchanged between two clubs. A sample of the dataset containing on monetary, permanent transfers can be seen above in Figure 11. This cleansing would be completed by uploading the dataset to IDE and executing Python code to remove these rows.

The last step in the “toptransfers.csv” cleansing would be to remove any rows containing empty values or insufficient values such as “-“. Not removing these empty values could lead to errors when running modelling along with providing potentially skewed results should they not cause an initial error. The researcher will want to have as accurate models as possible. This cleansing would be completed by uploading the dataset to IDE and executing Python code to remove these rows. A sample of this can be seen in Figure 11 above.

Due to not needing the “name,” “team_from,” and “team_to” for the analysis and modelling portions of the project I decided to not go ahead with converting any characters that were in a different format for this section, but this could be carried out with Python code.

For the “sentimenttest.csv” file no manual cleansing was necessary for completing analysis on it. All code for scraping tweets is from Twitter’s developer platform. Only tweets in English are pulled down. The processing of the tweets is done through the use of the NLTK and Vader lexicon libraries. Vader lexicon is able to interpret the likes of emojis so these tweets can be kept and understood while manually entered emojis such as “:)” can be understood by the library and assigned a positive, negative, or neutral value. No other pre-processing is needed for this dataset. A sample of this dataset can be seen above in Figure 10.

For the “top5rev.csv” dataset, no pre-processing was needed. A sample of this dataset can be seen above in Figure 1.

For the “serieasalary.csv” dataset, no pre-processing was needed. A sample of this dataset can be seen above in Figure 2.

For the “premsponsor.csv” dataset, no pre-processing was needed. A sample of this dataset can be seen above in Figure 3.

For the “PLSalary.csv” dataset, no pre-processing was needed. A sample of this dataset can be seen above in Figure 4.

For the “PLRev.csv” dataset, no pre-processing was needed. A sample of this dataset can be seen above in Figure 5.

For the “ligue1salary.csv” dataset, no pre-processing was needed. A sample of this dataset can be seen above in Figure 6.

For the “laligasalary.csv” dataset, no pre-processing was needed. A sample of this dataset can be seen above in Figure 7.

For the “bundessalary.csv” dataset, no pre-processing was needed. A sample of this dataset can be seen above in Figure 8.

For the “top20rev.csv” dataset, no pre-processing was needed. A sample of this dataset can be seen above in Figure 9.

3.0.3 Transformation

This is transforming the data in way that will be ideal for mining the data. This can be done in order to remove noise from the dataset. This can be achieved by using the Pandas library within Python which is built on top of another library known as Numpys. The use of these libraries can assist in normalisation, filling, and cleansing the data for later analysis and visualisation creation.

The next step would be to aggregate the data. This is the process of storing and presenting your data in a summary format. In the case of the transfermarkt.com data, we are gathering data from multiple URL’s which is a separate URL for each season from 2015/2016 up to 2021/2022. This allowed me to obtain high quality data of transfers with over 700 instances and containing 9 different attributes, not all of which were necessary for analysis but useful for context of the transfer.

This same process was applicable to the datasets acquired in Figures 1-9.

3.0.4 Data Mining

This is the process of searching for patterns in the data and representing it in a particular way in order to show such trends. This can also be used for machine learning purposes which was used in the case of this project.

In this project I applied the LSTM model in order to try and predict transfer fees in future. Although not learnt in class, I endeavoured to learn it myself as I believed it to provide some great insight along with be an interesting new technique to learn and work with.

This data will also be used for other analysis such as regression models which were learnt in class and then expanded upon through my own learning outside of class in order to gain the understanding of it as possible and to best apply it to my project for best results and implementation.

The last stage of modelling that the data was identified for was that of sentiment analysis. This is another analysis type that we did not learn in any classes, but I had interest in learning about and applying to my project, so I was determined to learn about it and implement it into my analysis and was the perfect use case for the tweets that had been extracted from Twitter using its developer platform.

At this stage all the transfer data datasets will be merged together to create the “toptransfers.csv” dataset. These datasets were obtained through Python using the Beautifulsoup library which will parse the URL and extract the data. In order to avoid getting IP banned for scraping too much data or not scraping data ethically, a user-agent is used which calls another machine, browser, and IP address in order to extract this data. You tell beautifulsoup that you are scraping from a grid-view within a responsive table. The next step is to state all columns that are to be taken from the table. You then create an array list from this scraped data which is then passed to a csv file which is created at the start of the code, and you tell to write to.

For the “PLSalary.csv” dataset, it is scraped again through the beautifulsoup library but this time with the addition of the Pandas library within Python. Beautifulsoup takes the URL and parses it. The table within the URL is read by the code and creates a new data frame with Pandas. The data frame contains the data from the 2 columns that are featured in the table on statista.com. This data frame is then written to the csv file that is created at the start of the code which can be read via Excel and interpreted and modelled by the IDE which in this case is Google Colab.

For the “top5rev.csv” dataset, it is scraped again through the beautifulsoup library but this time with the addition of the Pandas library within Python. Beautifulsoup takes the URL and parses it. The table within the URL is read by the code and creates a new data frame with Pandas. The data frame contains the data from the 2 columns that are featured in the table on statista.com. This data frame is then written to the csv file that is created at the start of the code which can be read via Excel and interpreted and modelled by the IDE which in this case is Google Colab.

For the “SerieASalary.csv” dataset, it is scraped again through the beautifulsoup library but this time with the addition of the Pandas library within Python. Beautifulsoup takes the URL and parses it. The table within the URL is read by the code and creates a new data frame with Pandas. The data frame contains the data from the 2 columns that are featured in the table on statista.com. This data frame is then written to the csv file that is created at the start of the code which can be read via Excel and interpreted and modelled by the IDE which in this case is Google Colab.

For the “BundesSalary.csv” dataset, it is scraped again through the beautifulsoup library but this time with the addition of the Pandas library within Python. Beautifulsoup takes the URL and parses it. The table within the URL is read by the code and creates a new data frame with Pandas. The data frame contains the data from the 2 columns that are featured in the table on statista.com. This data frame is then written to the csv file that is created at the start of the code which can be read via Excel and interpreted and modelled by the IDE which in this case is Google Colab.

For the “Ligue1Salary.csv” dataset, it is scraped again through the beautifulsoup library but this time with the addition of the Pandas library within Python. Beautifulsoup takes the URL and parses it. The table within the URL is read by the code and creates a new data frame with Pandas. The data frame contains the data from the 2 columns that are featured in the table on statista.com. This data frame is then written to the csv file that is created at the start of the code which can be read via Excel and interpreted and modelled by the IDE which in this case is Google Colab.

For the “LaLigaSalary.csv” dataset, it is scraped again through the beautifulsoup library but this time with the addition of the Pandas library within Python. Beautifulsoup takes the URL and parses it. The table within the URL is read by the code and creates a new data frame with Pandas. The data frame contains the data from the 2 columns that are featured in the table on statista.com. This data frame is then written to the csv file that is created at the start of the code which can be read via Excel and interpreted and modelled by the IDE which in this case is Google Colab.

For the “Top20Rev.csv” dataset, it is scraped again through the beautifulsoup library but this time with the addition of the Pandas library within Python. Beautifulsoup takes the URL and parses it. The table within the URL is read by the code and creates a new data frame with Pandas. The data frame contains the data from the 2 columns that are featured in the table on statista.com. This data frame is then written to the csv file that is created at the start of the code which can be read via Excel and interpreted and modelled by the IDE which in this case is Google Colab.

For the “PLSponsor.csv” dataset, it is scraped again through the beautifulsoup library but this time with the addition of the Pandas library within Python. Beautifulsoup takes the URL and parses it. The table within the URL is read by the code and creates a new data frame with Pandas. The data frame contains the data from the 2 columns that are featured in the table on statista.com. This data frame is then written to the csv file that is created at the start of the code which can be read via Excel and interpreted and modelled by the IDE which in this case is Google Colab.

For the “PLRev.csv” dataset, it is scraped again through the beautifulsoup library but this time with the addition of the Pandas library within Python. Beautifulsoup takes the URL and parses it. The table within the URL is read by the code and creates a new data frame with Pandas. The data frame contains the data from the 2 columns that are featured in the table on statista.com. This data frame is then written to the csv file that is created at the start of the code which can be read via Excel and interpreted and modelled by the IDE which in this case is Google Colab.

3.0.5 Evaluation

This is the part of the KDD methodology where the researcher will interpret their data and the models that they have made for the analysis. This analysis will yield visualisations which allow the researcher to interpret the results and draw conclusions from them. All modelling for the project will be completed using Python as the programming language, Google Colab and PyCharm as the IDE’s, while some extra analysis may be carried out with SPSS and Tableau should Python libraries not be suitable for certain visualisations. All visualisations will be created programmatically through Python and its libraries such as Matplotlib. I have gained the knowledge of these libraries and the visualisations that can be created with them through the classes I have undertaken this year and through my own learning outside of class time. The modelling processes that have been used in this project have been learnt through classes in 4th year along with my own learning of other models in my free time in order to get the best analysis possible.

The “toptransfers.csv” dataset, being the most prominently used and most important dataset for this project will provide us with numerous different visualisations and modelling results to interpret due to the wide range of tests that will be run on it.

All other datasets will have results of their own to interpret and a range of visualisations to talk about, but they will not be as prominent or in depth as the top transfers dataset.

Sentiment analysis will be carried out on tweets of public opinion in relation to transfer fees on specific players in order to gauge what public perception is in regard to the state of transfer fees paid for footballers in 2021. We will be given polarity scores of each tweet along with a value of positive, negative, or neutral based on the polarity score assigned to the tweet. We can then have visualisations showing the overall sentiment of the transfer fee and whether people feel more positive or negative about it.

4.0 Analysis

The goal of this report is to carry out analysis on a number of different datasets that have been scraped using Python. The analysis will cover Europe's top 5 leagues and look at areas such as transfers of the clubs in these leagues since 2015/2016 to 2021/2022 along with looking into the revenues earned by these clubs since 2015/2016 to 2021/2022 and to lastly analyse how the average wage that teams in each league pay their players differs from each other to find out if the money that is in the English premier league exceeds that of their European counterparts. Another area that will be looked at is public what public sentiment is like in regard to transfer fees in 2022 and if fans feel favourably or negatively about these large fees paid for players. This analysis will be carried out using a variety of different techniques in order to draw the best conclusions and get the best results possible.

4.0.1 Techniques Used

Simple Linear Regression – In this project simple linear regression was used in order to work out the correlation between different variables in the in the “toptransfers.csv” file had on the transfer fee of a player and how strong they are. We wanted to see if factors such as a players age and their market value would have an impact on the transfer fee that is paid for them. This regression model will allow us to see what the transfer fee for example would be for a 25-year-old who has a market value of €40 million. This output will be given through inputting predictor variables. The reasoning behind using simple linear regression is that it is considered to be a scientific and reliable way of predicting future values if the regression model has a good R value. It is also quite easy and fast to train a regression model which was ideal for the use of it in this project.

The age and market value of players in this case were used as the independent variables while I used the transfer fee as the dependent variable for this model as that is what we were trying to predict.

Multi Linear Regression – This was used in this project in order to how strong the relationship is between two or more independent variables and one dependent variable. For use in this project the dependent variable would be the transfer fee for the player and the independent variables would be age and market value. Opposed to simple linear regression, which is still useful, multi linear allows you to predict the outcome with several predictor variables opposed to the single predictor value of simple linear regression. It can tell us what the value is of the dependent variable at a certain value of the independent variable. For a good multi linear regression model you want to see a linear relationship between both the dependent variable and the independent variables. The two independent variables do not have too high a correlation with each other. Though for this project I wanted to use both simple and multi linear regression, multi linear is a more optimal model to use of the two due to the increased number of independent variables with no correlation to each other. Libraries such as sklearn in Python will be used to create the model.

Sentiment Analysis – In this project sentiment analysis was used in order to gain an idea of what public sentiment was towards some of the biggest transfers of the last two years to see what the public think of the fees and the way transfer fees are heading in football. Sentiment analysis identifies and extracts information from a scraped source and uses libraries to break down the overall polarity of the text. In the case of this project the data source were tweets from Twitter containing keywords about the transfers of specific footballers such as Jadon Sanchos transfer to Manchester United in the summer of 2021. The sentiment analysis breaks down tweets into the different categories of positive, negative, and neutral. Each tweet is ranked by the polarity score that it acquires when a library is run to assign its score. The polarity score can rank from less than 0 which would be negative, equal to 0 which would be neutral and greater than 0 which would be positive. I felt that this would be a great method to use in order to gather knowledge about what people think about transfer fees and felt that Twitter would be a great place to collect the data source from due to the volume of tweets that are created each second and which is a great platform for football fans to give their opinion. The reasoning behind choosing tweets about the players that I picked is due to them being some of the most high-profile footballers in the world that commanded some of the highest transfer fees we have ever seen. The Vader lexicon library which is used for sentiment analysis was chosen for its ability to clearly understand what is being said in English along with its ability to interpret emojis which can add a lot of contexts to a tweet and if not considered could affect the polarity score assigned. Libraries such as NLTK and Vader lexicon will be used in assisting with the sentiment analysis.

LSTM – This stands for long short-term memory and is an artificial neural network. It used for machine learning as a prediction model and is typically used for attempting to predict stock prices. In this case I was interested to see if the time series model used by LSTM would be of value and be able to predict the future value of transfers based on historic transfer fees, market value and age of a player. LSTM was also used due to its ability to forget and not use the less important or significant information put into it while being able to store, use and learn from the more relevant information to the prediction.

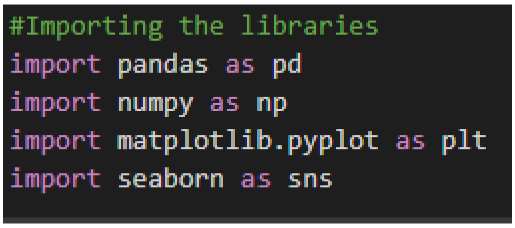
Visualisation – A range of visualisations in this project will be used in order to best display and interpret the results that I have gotten from the modelling. All visualisations have been created programmatically through Python and through libraries such as Matplotlib and Plotly. The types of visualisations used are bar charts, line graphs, scatter plots, radars, distribution plots, and Pearson correlation matrices. All of the above visualisations were used as I felt they best fit what I was trying to achieve along with some being necessary to effectively run a model such as the distribution plots and Pearson correlation matrices for regression models while the likes of a radar worked best for displaying the results of the sentiment analysis. Bar charts were primarily used in order to plot the financial figures of the leagues due to the ease of comparing them to each other.

5.0 Implementation

This section will outline how the models were implemented to the project and how they were used.

5.1.1 Simple Linear Regression

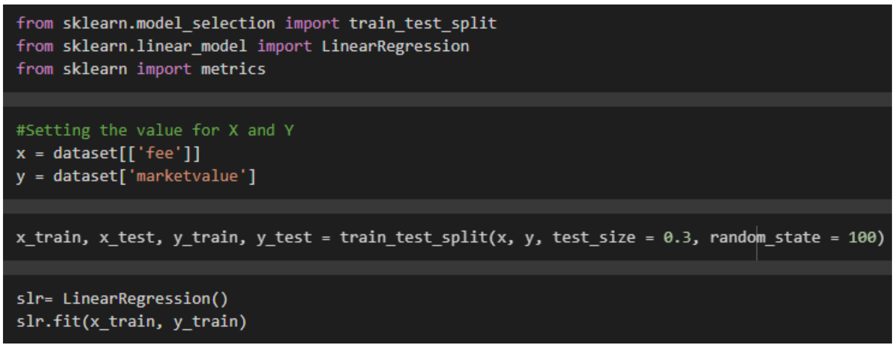
This was implemented using the pandas, numpy, matplotlib, seaborn and sklearn libraries within Python. Code telling the model to run boxplots is executed using seaborn. Code is then run to produce distribution charts and a scatter plot of the variables. A Pearson correlation matrix is then created with the seaborn library to show correlation between variables. Seen in figure 13 below.



```
#Importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Figure 13

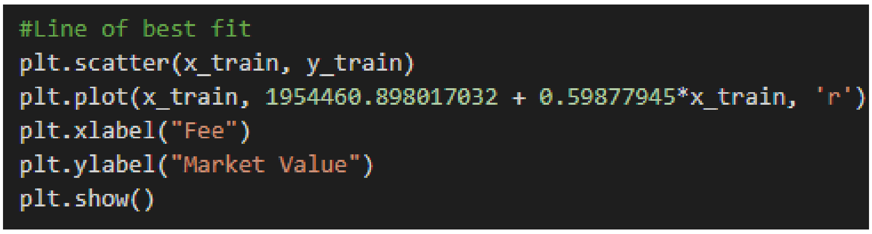
The next phase is to train and test the model which is done through the sklearn library. For this project I used a test size of 0.3 and a train size of 0.7. 30% of the data is used for testing which means that 70% of the data is used for training. The train and split variables are then split, and you then predict the output by passing the test variable. You then print the array of predictions made. A data frame is then created to store the predictions and save them to a csv file. Seen in Figure 14 below.



```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics

#Setting the value for X and Y
x = dataset[['fee']]
y = dataset['marketvalue']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 100)

slr= LinearRegression()
slr.fit(x_train, y_train)
```

Figure 14

Line of best fit is implemented to a scatter plot that plots the scatter plot with the new values created by the model which gathered from the intercept and coefficient values. Seen in Figure 15 below.

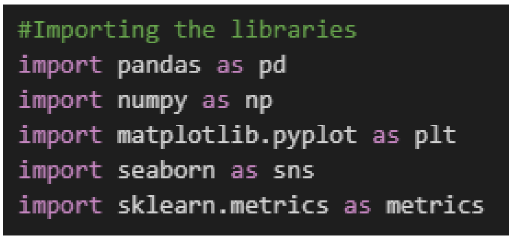


```
#Line of best fit
plt.scatter(x_train, y_train)
plt.plot(x_train, 1954460.898017032 + 0.59877945*x_train, 'r')
plt.xlabel("Fee")
plt.ylabel("Market Value")
plt.show()
```

Figure 15

5.1.2 Multi Linear Regression

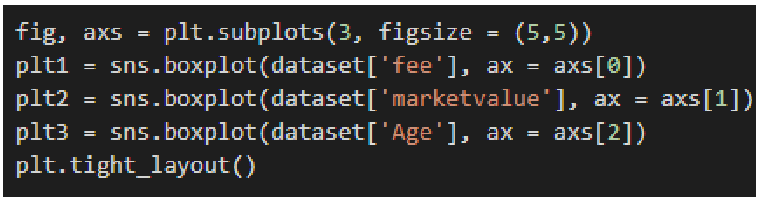
This was implemented using the pandas, numpy, matplotlib, seaborn and sklearn libraries within Python. Seen in Figure 16 below.



```
#Importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn.metrics as metrics
```

Figure 16

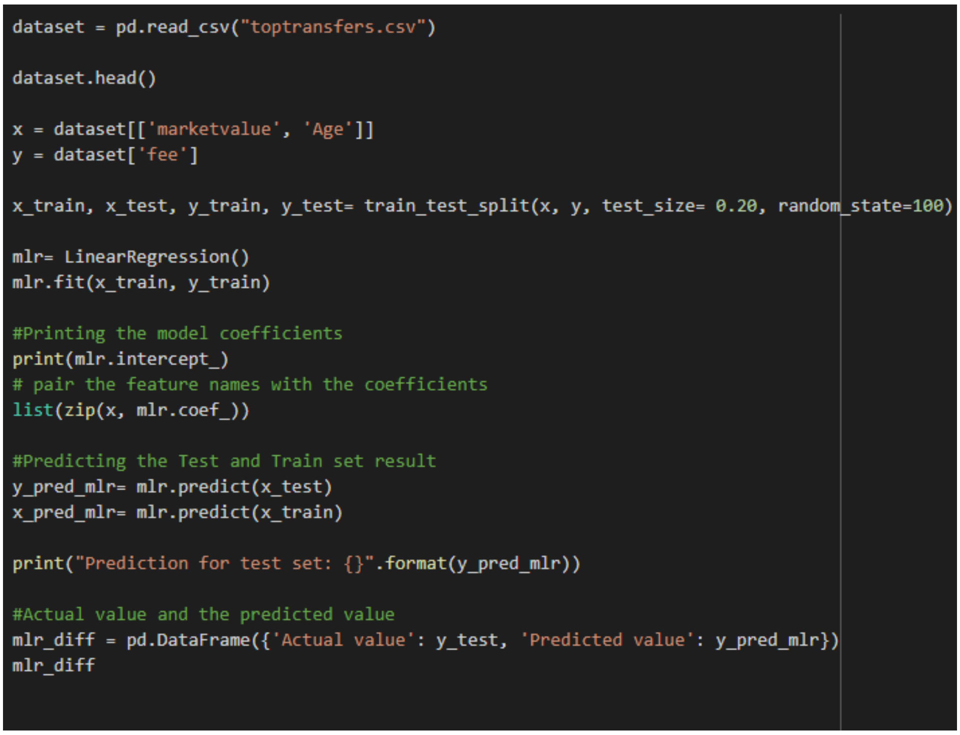
Boxplots using seaborn were plotted for the 2 independent and the dependent variable. Distribution plots through seaborn are made along with scatter plots and a Pearson correlation matrix. Example of this in Figure 17 below.



```
fig, axs = plt.subplots(3, figsize = (5,5))
plt1 = sns.boxplot(dataset['fee'], ax = axs[0])
plt2 = sns.boxplot(dataset['marketvalue'], ax = axs[1])
plt3 = sns.boxplot(dataset['Age'], ax = axs[2])
plt.tight_layout()
```

Figure 17

Sklearn is then used again for creating the model and we use a 0.2 test split and a 0.8 test split in this case. The train and split variables are split you again predict the output by passing the test variable and print it. A data frame is then created to store the predictions and save them to a csv file. This can be seen in Figure 18 below.

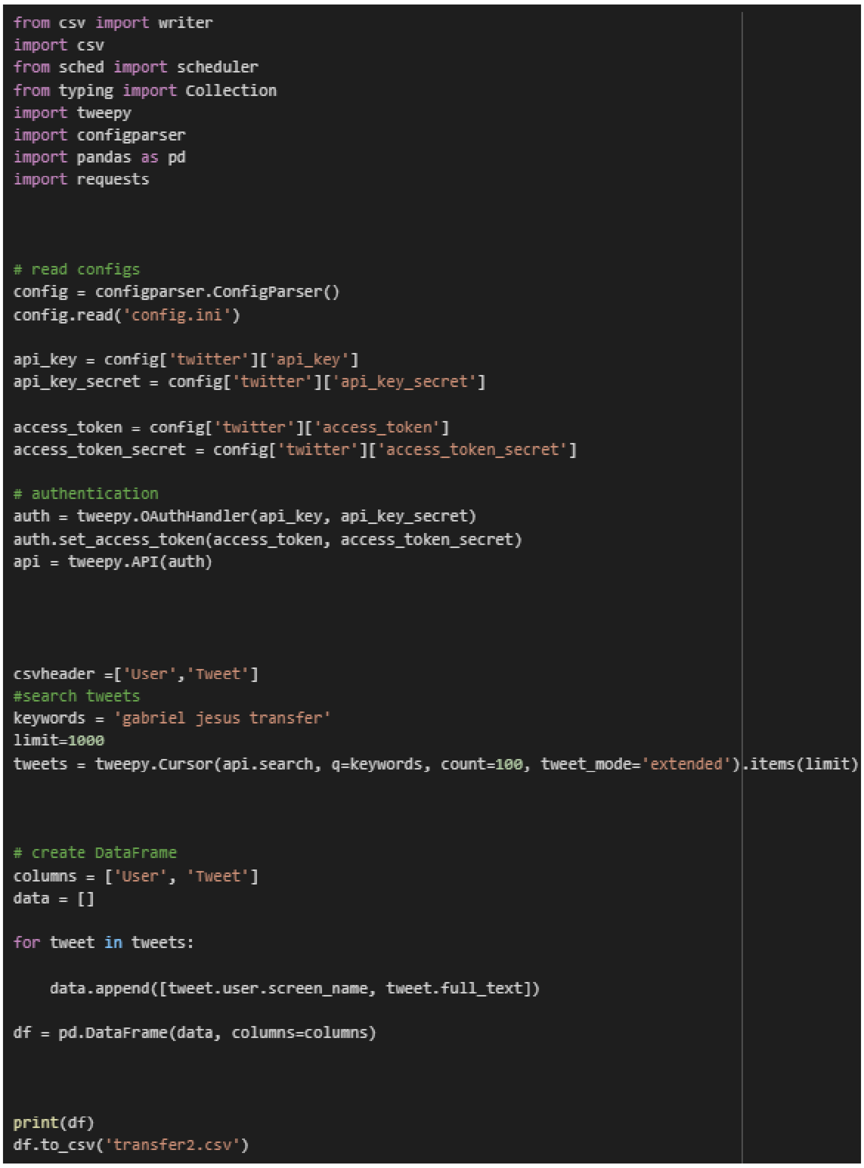


```
dataset = pd.read_csv("toptransfers.csv")

dataset.head()

x = dataset[['marketvalue', 'Age']]
y = dataset['fee']

x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.20, random_state=100)

mlr= LinearRegression()
mlr.fit(x_train, y_train)

#Printing the model coefficients
print(mlr.intercept_)
# pair the feature names with the coefficients
list(zip(x, mlr.coef_))

#Predicting the Test and Train set result
y_pred_mlr= mlr.predict(x_test)
x_pred_mlr= mlr.predict(x_train)

print("Prediction for test set: {}".format(y_pred_mlr))

#Actual value and the predicted value
mlr_diff = pd.DataFrame({'Actual value': y_test, 'Predicted value': y_pred_mlr})
mlr_diff
```

Figure 18

5.1.3 Sentiment Analysis

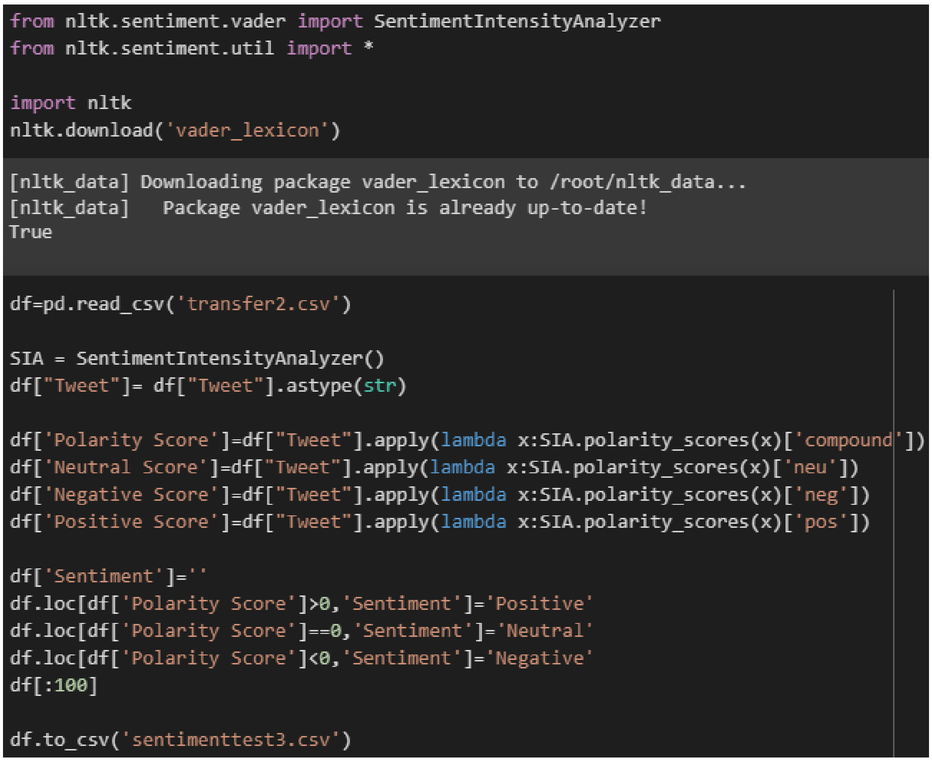
This was implemented using the libraries of tweepy, scheduler, Collection, configparser and pandas. The config while along with the API keys that you were given by Twitter's developer platform is read by the Python script to gain access to scrape the tweets. The tweepy API will then be authenticated. The Python code is then told to scrape tweets under the columns of 'user' and 'tweet' with a section for keywords which tells it what tweets to scrape e.g., "Gabriel Jesus transfer fee." A data frame is then created to store the tweets and saved to a csv file. Seen in Figure 19 below.



```
from csv import writer
import csv
from sched import scheduler
from typing import Collection
import tweepy
import configparser
import pandas as pd
import requests



# read configs
config = configparser.ConfigParser()
config.read('config.ini')

api_key = config['twitter']['api_key']
api_key_secret = config['twitter']['api_key_secret']

access_token = config['twitter']['access_token']
access_token_secret = config['twitter']['access_token_secret']

# authentication
auth = tweepy.OAuthHandler(api_key, api_key_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)




csvheader =['User','Tweet']
#search tweets
keywords = 'gabriel jesus transfer'
limit=1000
tweets = tweepy.Cursor(api.search, q=keywords, count=100, tweet_mode='extended').items(limit)



# create DataFrame
columns = ['User', 'Tweet']
data = []

for tweet in tweets:

    data.append([tweet.user.screen_name, tweet.full_text])

df = pd.DataFrame(data, columns=columns)



print(df)
df.to_csv('transfer2.csv')
```

Figure 19

The nltk library and vader lexicon library are then used to find the sentiment in each tweet collected. The new csv file that is created is then passed into new Python code which uses these libraries to find out the sentiment of each tweet based on a value of positive, negative, and neutral which is obtained through the use of the polarity score assigned by the vader lexicon library. This new file is then saved to a csv file to be used for analysis. Seen below in Figure 20.



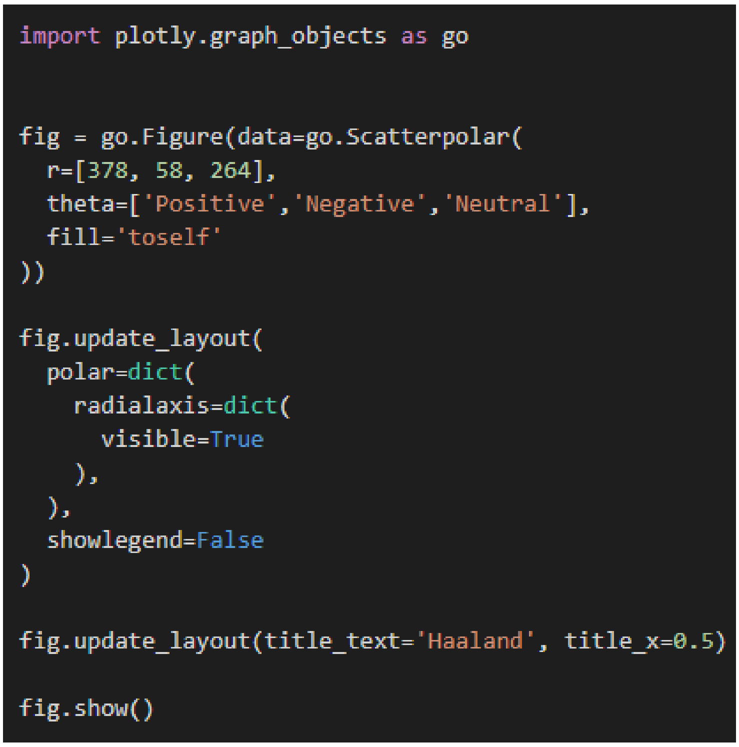
```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.sentiment.util import *

import nltk
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
True
```

```
df=pd.read_csv('transfer2.csv')

SIA = SentimentIntensityAnalyzer()
df["Tweet"]= df["Tweet"].astype(str)

df['Polarity Score']=df["Tweet"].apply(lambda x:SIA.polarity_scores(x)['compound'])
df['Neutral Score']=df["Tweet"].apply(lambda x:SIA.polarity_scores(x)['neu'])
df['Negative Score']=df["Tweet"].apply(lambda x:SIA.polarity_scores(x)['neg'])
df['Positive Score']=df["Tweet"].apply(lambda x:SIA.polarity_scores(x)['pos'])

df['Sentiment']=''
df.loc[df['Polarity Score']>0,'Sentiment']='Positive'
df.loc[df['Polarity Score']==0,'Sentiment']='Neutral'
df.loc[df['Polarity Score']<0,'Sentiment']='Negative'
df[:100]

df.to_csv('sentimenttest3.csv')
```

Figure 20

To plot the radar graph that represents the sentiment analysis, I used the Plotly library. The total count of positive, negative, and neutral tweets is then plotted to a radar graph. This whole process is then repeated but with different keywords to find tweets about different transfers. Seen below in Figure 21.

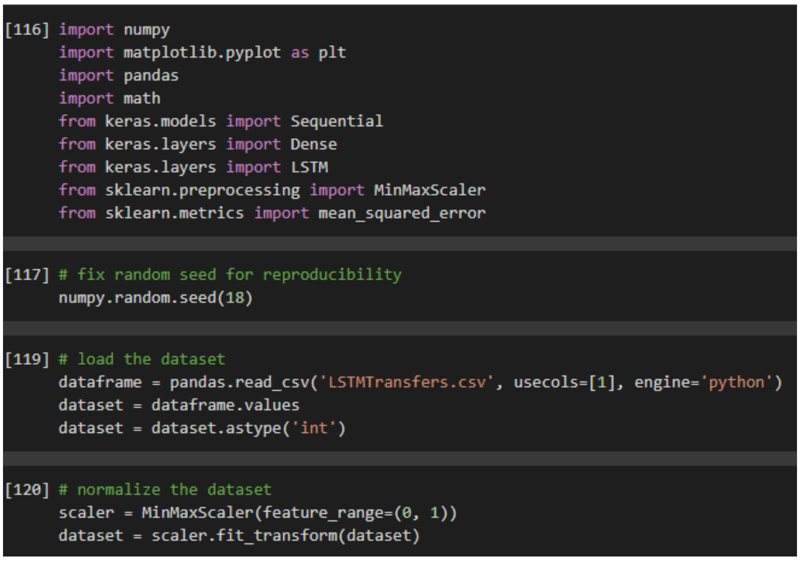


```
import plotly.graph_objects as go


fig = go.Figure(data=go.Scatterpolar(
  r=[378, 58, 264],
  theta=['Positive','Negative','Neutral'],
  fill='toself'
))

fig.update_layout(
  polar=dict(
    radialaxis=dict(
      visible=True
    ),
  ),
  showlegend=False
)

fig.update_layout(title_text='Haaland', title_x=0.5)

fig.show()
```

Figure 21

5.1.3 LSTM

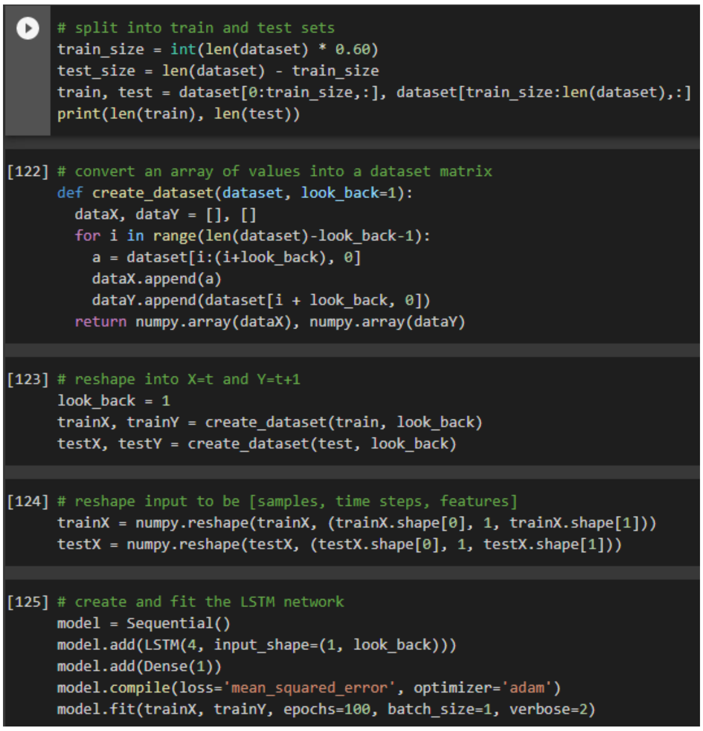
The LSTM model was implemented through libraries in Python such as numpy, matplotlib, pandas, math, keras and sklearn. The first step is to import all these libraries. You then assign a random side for reproducibility. The dataset is loaded into the IDE and the dataset is normalised. This is seen in Figure 22 below.



```
[116] import numpy
      import matplotlib.pyplot as plt
      import pandas
      import math
      from keras.models import Sequential
      from keras.layers import Dense
      from keras.layers import LSTM
      from sklearn.preprocessing import MinMaxScaler
      from sklearn.metrics import mean_squared_error

[117] # fix random seed for reproducibility
      numpy.random.seed(18)

[119] # load the dataset
      dataframe = pandas.read_csv('LSTMTransfers.csv', usecols=[1], engine='python')
      dataset = dataframe.values
      dataset = dataset.astype('int')

[120] # normalize the dataset
      scaler = MinMaxScaler(feature_range=(0, 1))
      dataset = scaler.fit_transform(dataset)
```

Figure 22

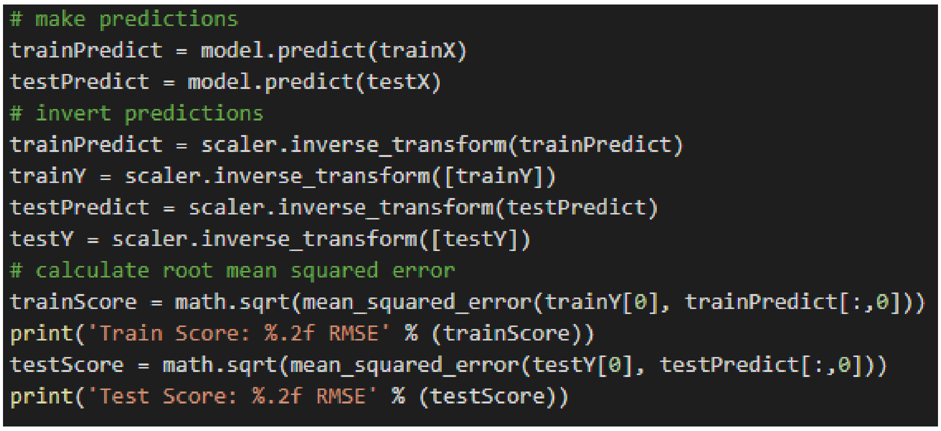
A test and train split are made of 60% train and 40% testing for this model. A dataset matrix will then be created by converting an array of values from the dataset. The dataset is then reshaped by creating a dataset for the train and test variables. The LSTM network is created and fitted by creating a sequential model. Seen in Figure 23 below.



```
      # split into train and test sets
      train_size = int(len(dataset) * 0.60)
      test_size = len(dataset) - train_size
      train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
      print(len(train), len(test))

[122] # convert an array of values into a dataset matrix
      def create_dataset(dataset, look_back=1):
        dataX, dataY = [], []
        for i in range(len(dataset)-look_back-1):
          a = dataset[i:(i+look_back), 0]
          dataX.append(a)
          dataY.append(dataset[i + look_back, 0])
        return numpy.array(dataX), numpy.array(dataY)

[123] # reshape into X=t and Y=t+1
      look_back = 1
      trainX, trainY = create_dataset(train, look_back)
      testX, testY = create_dataset(test, look_back)

[124] # reshape input to be [samples, time steps, features]
      trainX = numpy.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))
      testX = numpy.reshape(testX, (testX.shape[0], 1, testX.shape[1]))

[125] # create and fit the LSTM network
      model = Sequential()
      model.add(LSTM(4, input_shape=(1, look_back)))
      model.add(Dense(1))
      model.compile(loss='mean_squared_error', optimizer='adam')
      model.fit(trainX, trainY, epochs=100, batch_size=1, verbose=2)
```

Figure 23

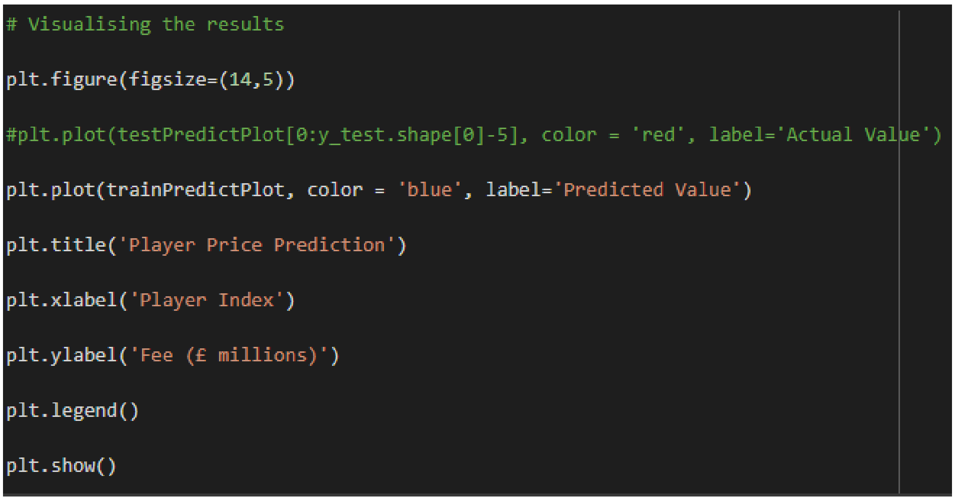
The next step is to acquire the train and test root mean squared error which is done by telling the model to make predictions, then to take these predicted values and invert them. Once this is done you can use the math library in order to work out the root mean squared error of the train and test set. This can be seen below in Figure 24.



```
# make predictions
trainPredict = model.predict(trainX)
testPredict = model.predict(testX)
# invert predictions
trainPredict = scaler.inverse_transform(trainPredict)
trainY = scaler.inverse_transform([trainY])
testPredict = scaler.inverse_transform(testPredict)
testY = scaler.inverse_transform([testY])
# calculate root mean squared error
trainScore = math.sqrt(mean_squared_error(trainY[0], trainPredict[:,0]))
print('Train Score: %.2f RMSE' % (trainScore))
testScore = math.sqrt(mean_squared_error(testY[0], testPredict[:,0]))
print('Test Score: %.2f RMSE' % (testScore))
```

Figure 24

Matplotlib is then used to visualise the results by stating the plot size, the variables in the plot and assigning labels and legends to the graph which can be seen in Figure 25 below.



```
# Visualising the results

plt.figure(figsize=(14,5))

#plt.plot(testPredictPlot[0:y_test.shape[0]-5], color = 'red', label='Actual Value')

plt.plot(trainPredictPlot, color = 'blue', label='Predicted Value')

plt.title('Player Price Prediction')

plt.xlabel('Player Index')

plt.ylabel('Fee (£ millions)')

plt.legend()

plt.show()
```

Figure 25

6.0 Results

In this section I will cover the results gained from my statistical analysis and modelling while breaking it down in to sections for ease of understanding and reading for the reader.

6.1 Net spend by league since 2017/2018

This area will look at how the net spend on transfers has changed in each league from the 2017/2018 season to the 2021/2022 season. We are looking to see if there is a trend across each league and whether any leagues stand out compared to the rest in regard to net spend and their trend. This is an easy way to interpret the spending of each league and give a reader a clear idea as to how each league is.

6.1.1 Premier League

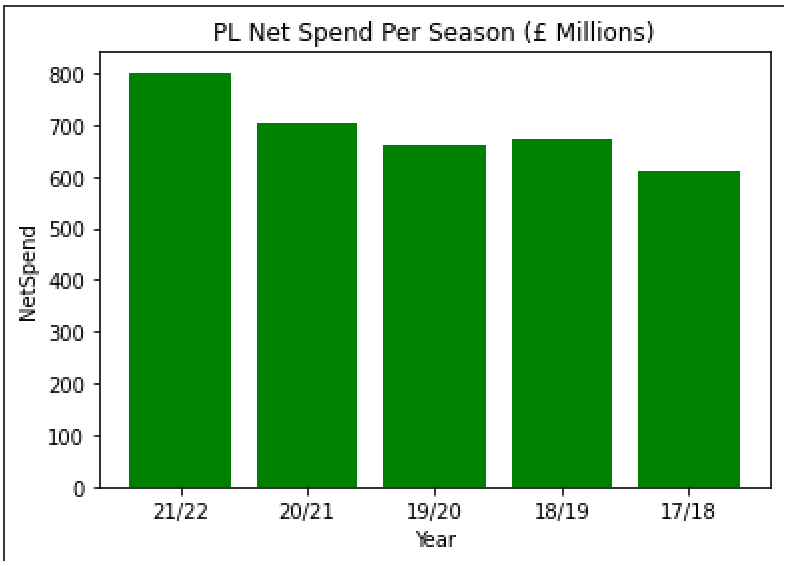


Figure 26

Figure 26 above shows the total net spend by the Premier Leagues clubs since 2017/2018. For context in this graph any figure above the X-axis is means they have spent more on transfers than they earned while any figure below the X-axis means that they have earned more from transfers than they have spent. All figures given are in millions and GBP. From this bar chart we can see that the Premier League clubs have spent more money each year barring a slight dip in 2019/2020 which was the year the pandemic started but had not affected football clubs spending as the transfer windows for this season had already closed. In 2020/2021, which was a season massively impacted by Covid such as all football being stopped and fans no longer being able to attend games, we see an increase from 2019/2020 spending. This already shows how little impact the pandemic had on the spending of the Premier League and in 2021/2022 where the sport was out of the pandemic, we see another large increase in net spend. In this 5-year period before and after the pandemic we see spending go from just over £600 million pounds to over £800 million.

6.1.2 La Liga

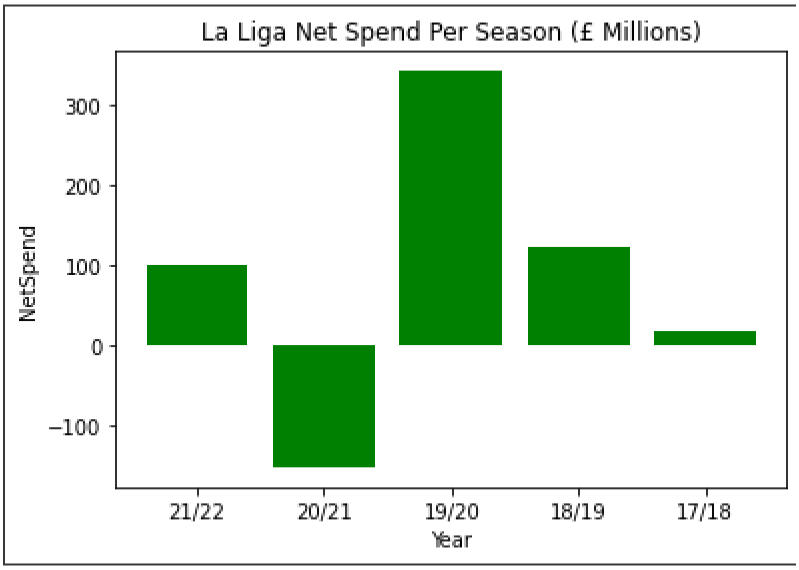


Figure 27

Figure 27 above shows the total net spend by the La Liga clubs since 2017/2018. For context in this graph any figure above the X-axis is means they have spent more on transfers than they earned while any figure below the X-axis means that they have earned more from transfers than they have spent. All figures given are in millions and GBP. From this bar chart we can see that the spending habits of La Liga clubs differs considerably from the Premier League. La Liga's highest net spend since 2017/2018 was in 2019/2020 where it comes in at roughly £350 million opposed to the Premier Leagues lowest being just over £600 million in this period. While some clubs may have been operating at a net loss over these years, we see that in 2020/2021, the year football was most heavily affected by the pandemic, that La Liga as a whole made a net gain of over £100 million. La Liga spending was trending upwards at a significant rate pre pandemic and is now near 2018/2019 levels so it will be interesting to see in future sections if this trend is likely to continue.

6.1.3 Serie A

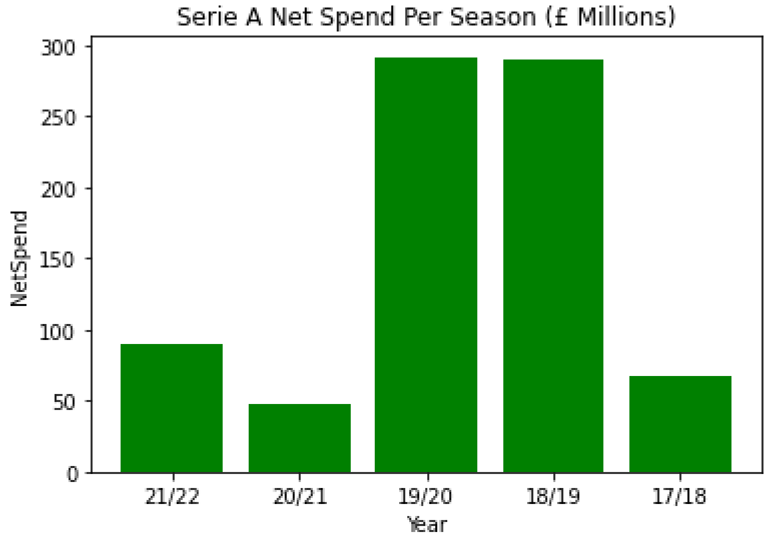


Figure 28

Figure 28 above shows the total net spend by the Serie A clubs since 2017/2018. For context in this graph any figure above the X-axis is means they have spent more on transfers than they earned while any figure below the X-axis means that they have earned more from transfers than they have spent. All figures given are in millions and GBP. From this bar chart we can see that every season, including 2020/2021, Serie A clubs as whole spend more on transfers than they earn. Pre pandemic there was strong trend of money spent by these clubs increasing which then took a sharp dip during the Covid-19 hit season of 2020/2021. Interestingly the 2020/2021 spending was not far short of the net spend 2017/2018. Since the pandemic, the net spend has gone up yet again and it will be interesting to see in future sections how this will go in future. Similarly, to La Liga, Serie A's highest net spend of just shy of £300 million is dwarfed by the Premier Leagues lowest net spend of £600 million.

6.1.4 Bundesliga

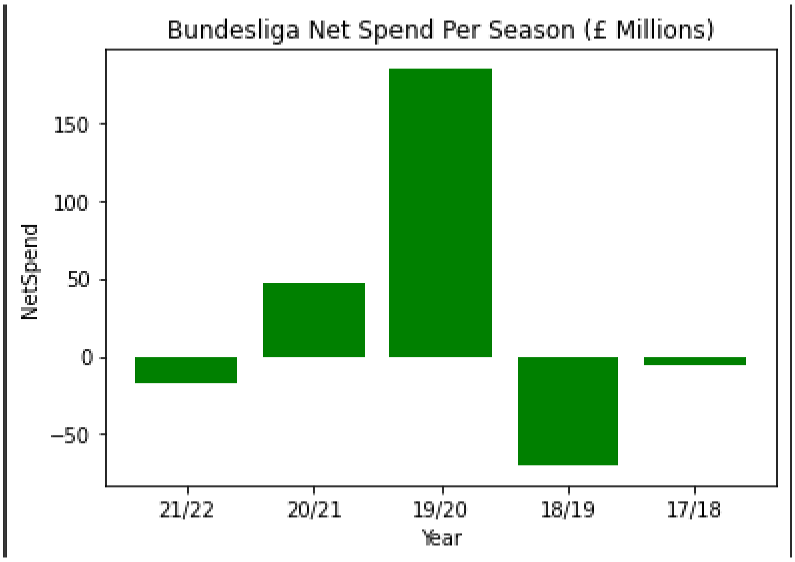


Figure 29

Figure 29 above shows the total net spend by the Bundesliga clubs since 2017/2018. For context in this graph any figure above the X-axis is means they have spent more on transfers than they earned while any figure below the X-axis means that they have earned more from transfers than they have spent. All figures given are in millions and GBP. From the above bar chart, we can see that in 3 of the last 5 seasons, Bundesliga clubs have earned more from transfers than they have spent occurring in 2017/2018, 2018/2019 and 2021/2022. The only 2 seasons in which they spent more were just pre pandemic in 2019/2020 where they had a net spend of over £150 million and the pandemic hit season of 2020/2021 where they had a net spend of roughly £50 million. Interestingly the post pandemic season where it was deemed clubs would be not so severely affected, Bundesliga clubs made their second biggest profit. There is no recognisable upward trend in the Bundesliga spending habits and if it continues this way, they will be making more profit on transfers each season, but it will be interesting to see if this is the case in future sections. Yet again we see that the Premier Leagues lowest net spend dwarfs the highest net spend season of the Bundesliga.

6.1.5 Ligue 1

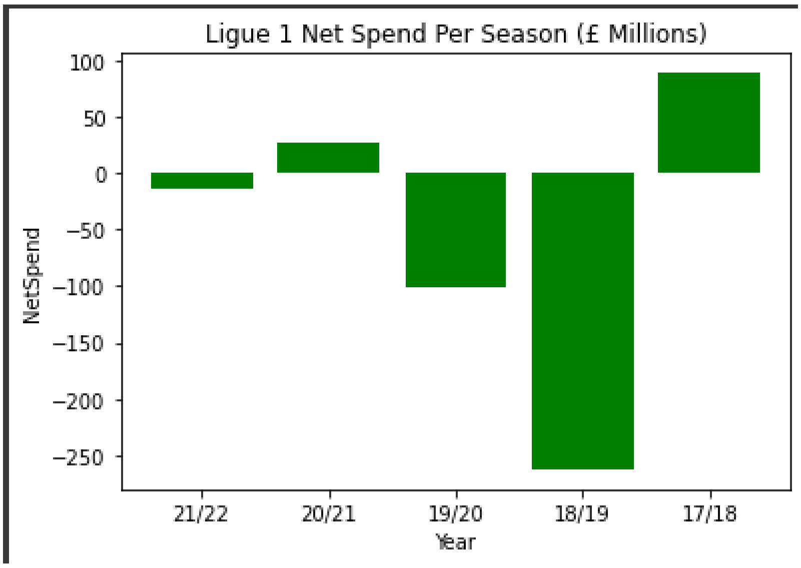


Figure 30

Figure 30 above shows the total net spend by the Ligue 1 clubs since 2017/2018. For context in this graph any figure above the X-axis is means they have spent more on transfers than they earned while any figure below the X-axis means that they have earned more from transfers than they have spent. All figures given are in millions and GBP. From the above bar chart, we can see that 3 of the last 5 seasons, Ligue 1 clubs have earned more from transfers than they have spent with these occurring in 2018/2019, 2019/2020, and 2021/2022. Interestingly the seasons where they spent more were long before the pandemic and then the pandemic hit season of 2020/2021 while the season after, which you would expect them to increase spending, yielded a small profit. It is hard to distinguish any real trend in the spending habits of Ligue 1 clubs as from 2018/2019 up to 2020/2021 we see an increase in spending but then 1 year either side of the we again see them decrease spending. It will be interesting to see in future sections where this spending may go. Yet again even the Premier Leagues lowest net spend year is significantly larger than the highest spending year of Ligue 1 which is just shy of £100m and came all the way back in 2017/2018.

6.2 Average Salary of Players by club per League

This section will look at the average salary of every club in each of the top 5 leagues for the 2020/2021 season. This section will be broken down by each league and each section will contain a chart showing the average salary per player from each club. While not in the modelling phase just yet, this is a good way to visualise how each teams pay their players and if there is a difference in the gaps between the leagues compared to the net spend section.

6.2.1 Premier League

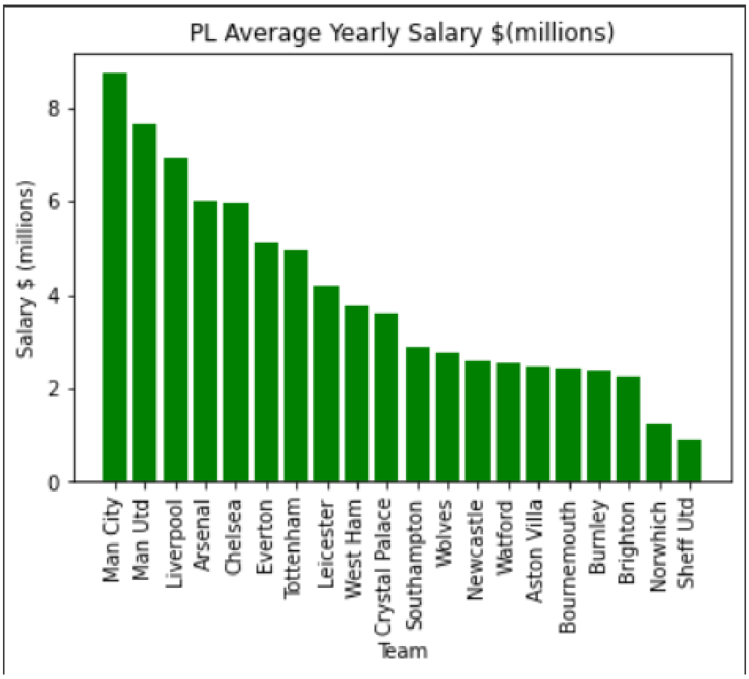


Figure 31

In Figure 31 above we see the average yearly salary paid to each player by Premier League clubs in the 2020/2021 season. All figures in this case are in millions and US dollars. In the above image we see that there is a gradual curve from the lowest average salary club to the highest average salary club with the lowest being roughly $1 million per year and the highest being roughly £9 million per year. A large spread of salaries like this is to be expected due to varying natures of commercial revenues and sponsors different teams make along with the wealth of their owners and the different competitions that more successful clubs can gain prize money from allows them to spend more on player wages.

6.2.2 La Liga

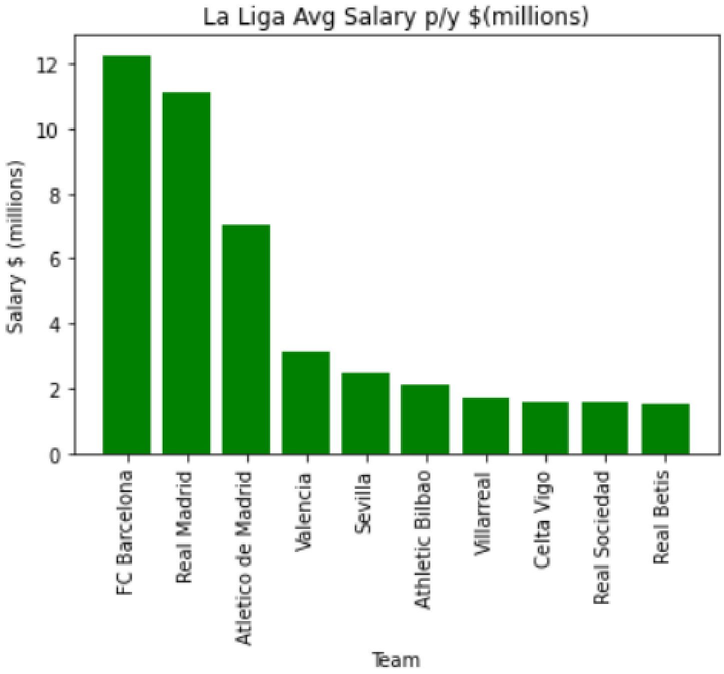


Figure 32

In figure 32 above we see the average yearly salary paid to each player by La Liga clubs in the 2020/2021 season. All figures in this case are in millions and US dollars. Due to a lack of information on the salaries of the 10 lowest La Liga we only had access to that of the top 10 highest playing clubs, but we can still get a good idea of the leagues wage structure and how different clubs are able to spend. We can see that the lowest wage spenders in the top 10 are just shy of $2 million and up to 4th place which is just below $4 million. The top 3 clubs in Spain dwarf the spending ability of the rest and even 3rd place is some way off the top 2 who spend $12 million and $11 million, respectively. This even larger than the spend of the biggest Premier League clubs but every Premier League club bar the bottom 2 in spending, spend more than all but 4 of La Liga. This shows the immense spending power of La Liga's biggest clubs but shows the overall power of the Premier League.

6.2.3 Serie A

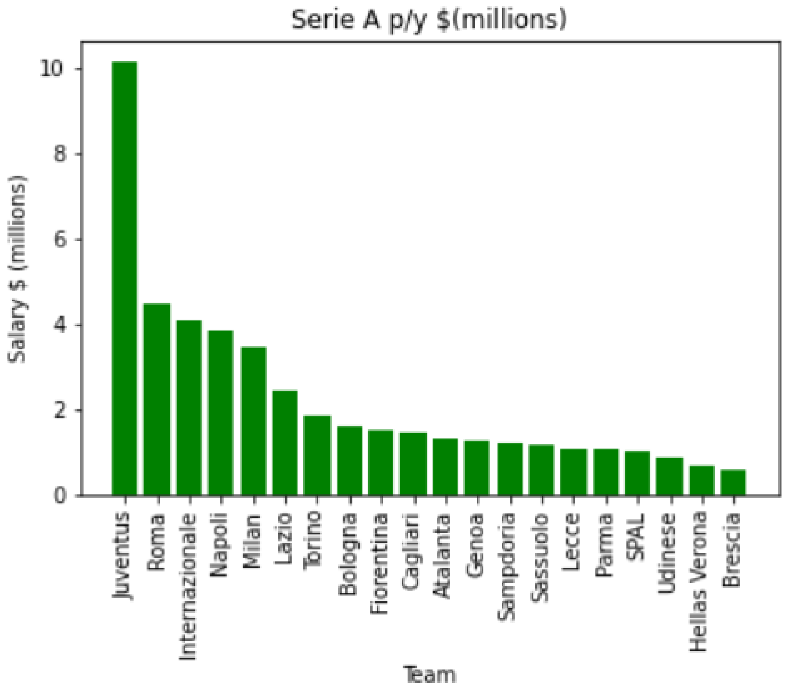


Figure 33

In figure 33 above we see the average yearly salary paid to each player by Serie A clubs in the 2020/2021 season. All figures in this case are in millions and US dollars. We can see from the bar chart above that yet again there is an outlier in terms of salary paid to players in a league. In this case we can see that Juventus pay $10 million per year on but there is a drop off of almost $6 million to the next highest spender. This league is quite even in regard to the spending of all clubs aside from Juventus and you see a steady gradual increase in wages as you move up the chart from lowest to highest. Another trend that we see is another league who cannot compete with the Premier League on a whole with the exception of Juventus. The 2nd to 5th highest spenders in Serie A would only be in the middle of the pack in the Premier League while every team below 5th is outspent by virtually every Premier League side.

6.2.4 Bundesliga

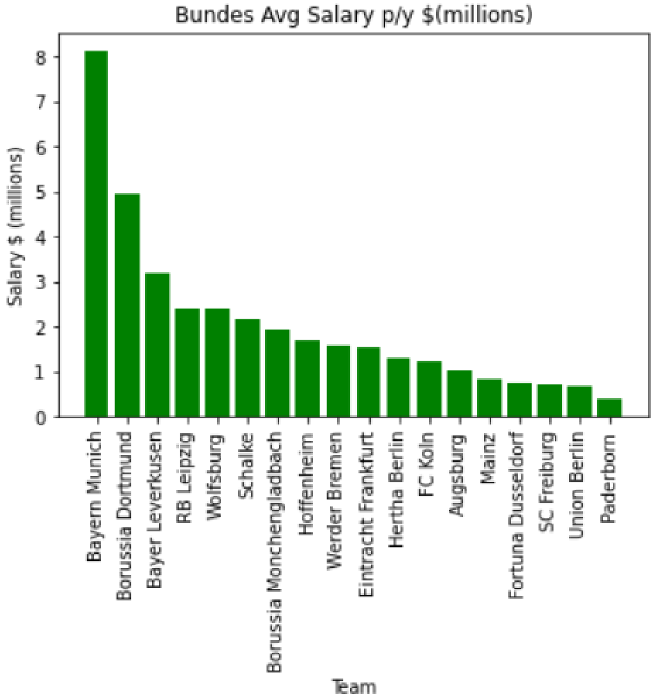


Figure 34

In figure 34 above we see the average yearly salary paid to each player by Bundesliga clubs in the 2020/2021 season. All figures in this case are in millions and US dollars. Similarly, to Serie A before, we see one big outlier in terms of wage expenditure in Bundesliga which is Bayern Munich at $8 million per year. After that there is Borussia Dortmund who are also a bit of an outlier $5 million and after this, we see the pack tighten back up for a more even distribution. It is another case of where we see the top 2 or 3 teams can compete for wages with the big Premier League clubs, but any other club would fall into the lower end of the Premier League or in most cases, spend less than all Premier League sides.

6.2.3 Ligue 1

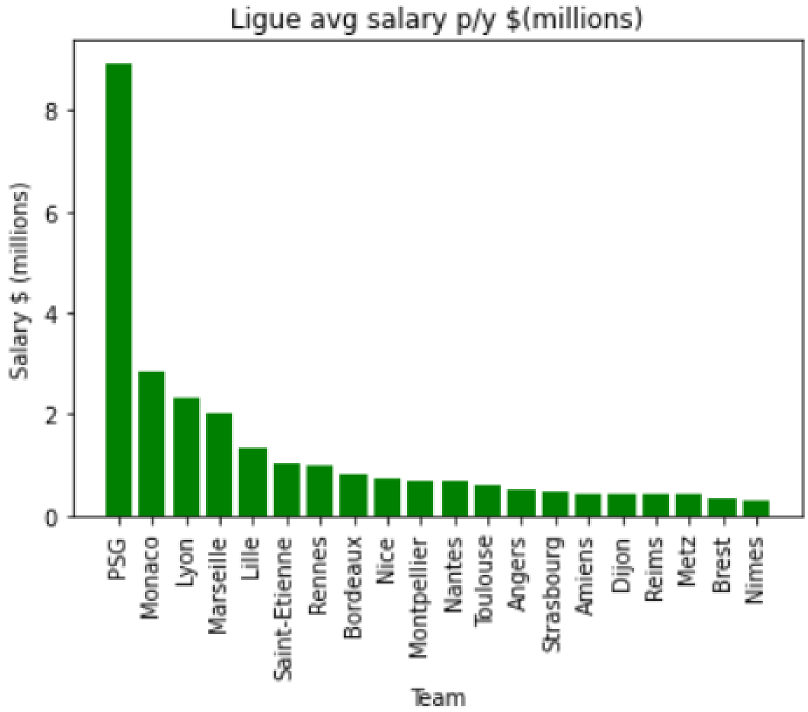


Figure 35

In figure 35 above we see the average yearly salary paid to each player by Ligue 1 clubs in the 2020/2021 season. All figures in this case are in millions and US dollars. As has been the case with all previous leagues, with the exception of the Premier league, we see one outlier in the Ligue 1 spending significantly more than any side in the league. In this case we see that PSG spend almost $10 million which compares to the big spenders in the other, but the next highest spenders are Monaco who spend roughly $3 million while every other would be comfortably outspent by their Premier Leagues counterparts.

6.3 Top 20 Highest Earning Clubs

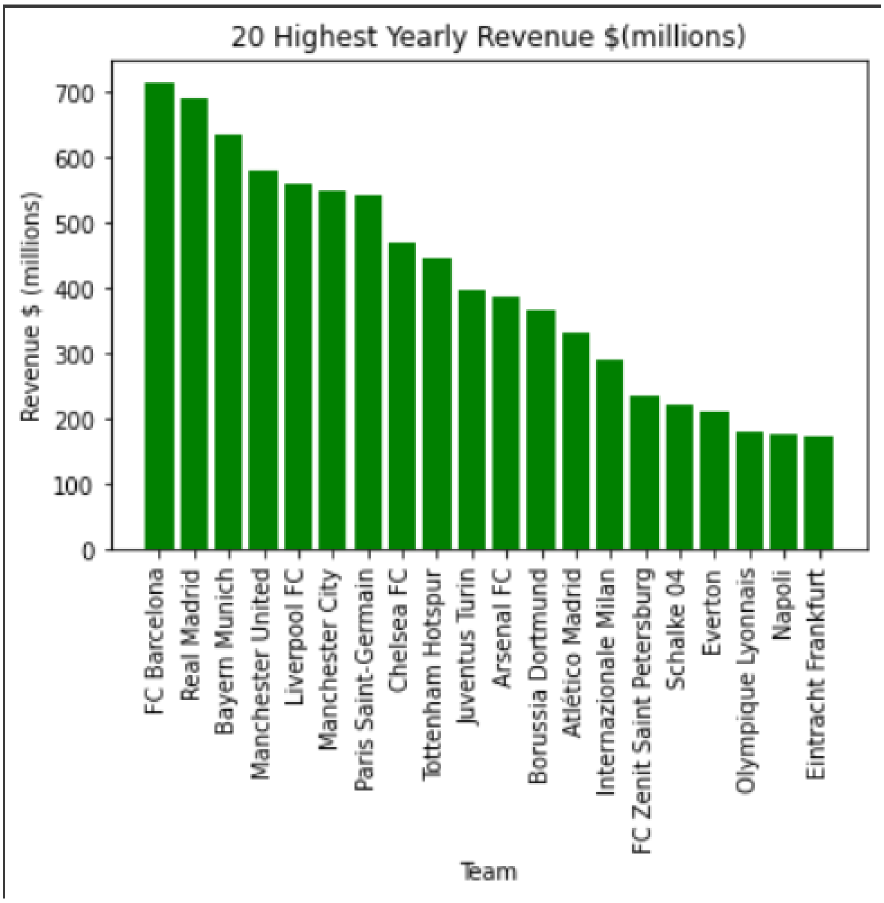


Figure 36

In Figure 36 above we can see a bar chart of the 20 clubs in Europe who generated the most revenue in 2020/2021. This visualisation is presented in millions of US dollars. A trend we can see continue in regard to this is the earning power of the Premier League clubs where see 7 of the 20 listed are from there, we see 3 from La Liga, 4 from the Bundesliga, 2 from Ligue 1 and 3 from Serie A with the one remaining team coming from a Russian League. While the Premier League does not occupy any of the top 3 spots, a Premier League club occupies 6 of the top 11 spots. Where every other league is represented a maximum of twice this is only in the instance of La Liga.

6.4 Fee paid for player on scatter plot against the market value

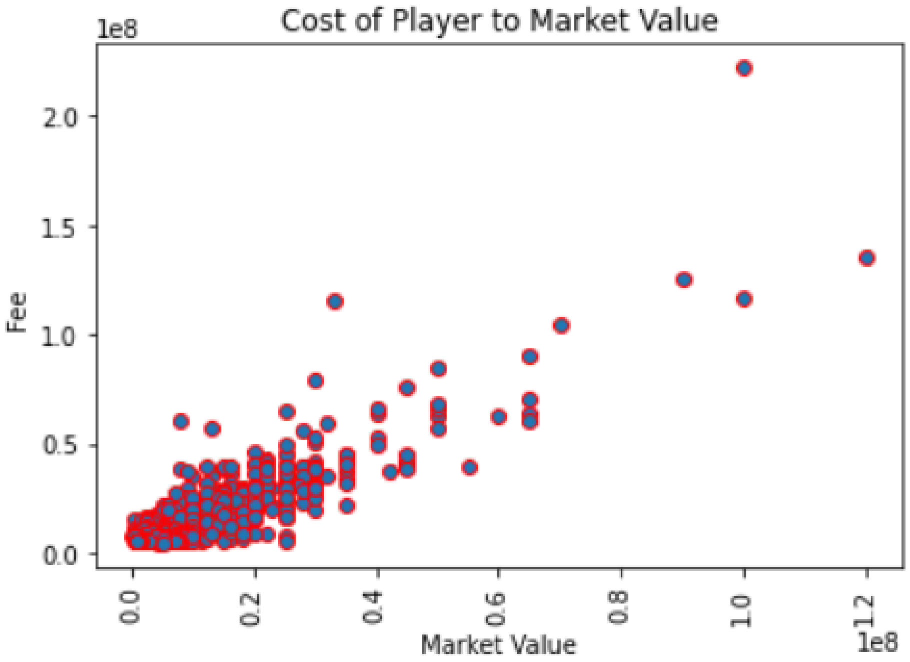


Figure 37

In Figure 37 above we can see a scatter plot. This plots the market value of a player against the transfer fee paid them and we hope to be able to see a correlation with few outliers. What we see is that with the higher market value players is that their transfer is near the top right where it is expected to be while we would expect to see the lower value players in the bottom left which is the case. We would not expect to see a player with a high market value in the bottom right corner of the graph and alternatively we would not expect to see a lower market value player in the top left corner which would indicate an exceedingly high transfer fee for a low value player and could represent bad value for a club. Seeing a high market value player in the bottom right corner would indicate good business by the buying club and poor business by the selling club. This scatter plot shows us that there is a strong linear relationship between the two variables.

5.5 Regression Models

This section will break down the results from both the simple linear regression model and the multi linear regression models. It covers the prediction model that was used in order to predict the transfer fee of a player with the independent variable of market value in the case of the simple linear regression. For multi linear regression the independent variables were age and market value with the transfer fee being the dependent variable. All results will be provided with screenshots.

6.5.1 Simple Linear Regression

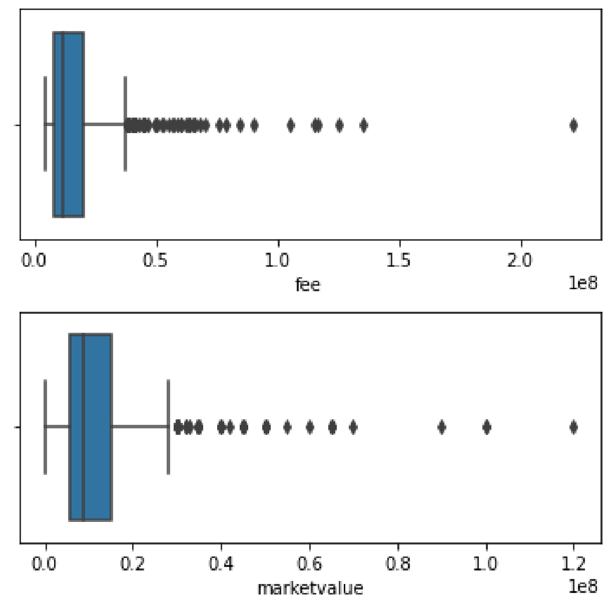


Figure 38

In Figure 38 above we see the first result from our simple linear regression analysis. This image is of a horizontal boxplot that is produced. This boxplot displays the distribution of the variables in the dataset. In this case it is the independent variable and the dependent variable that we are using which is the fee and market value of a player. We see the upper and lower extremes of the boxplot along with quartiles 1 to 3. The last thing we see that is indicated by the line in the blue box which is the median of variable. We see that the median for the fee is roughly 0.1 which is £10 million and median for the market value is 0.7 which is £7 million.

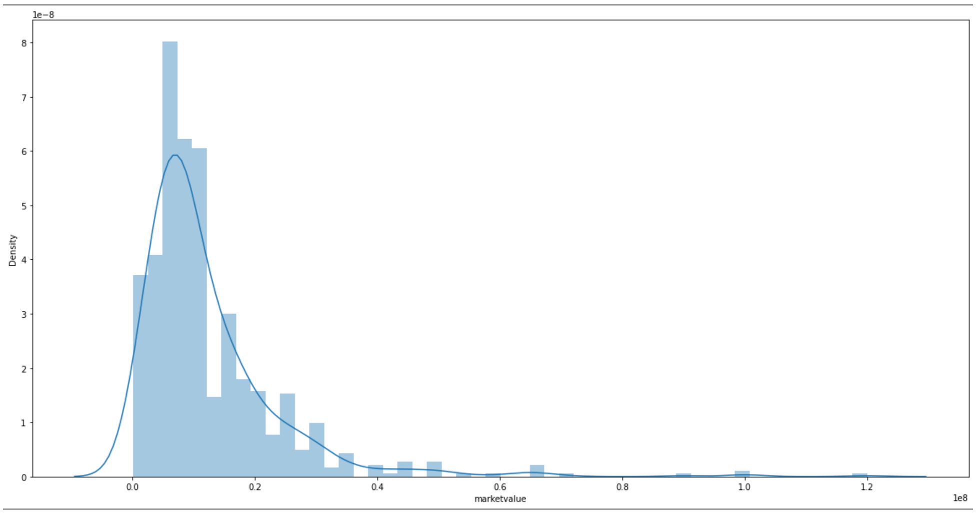


Figure 39

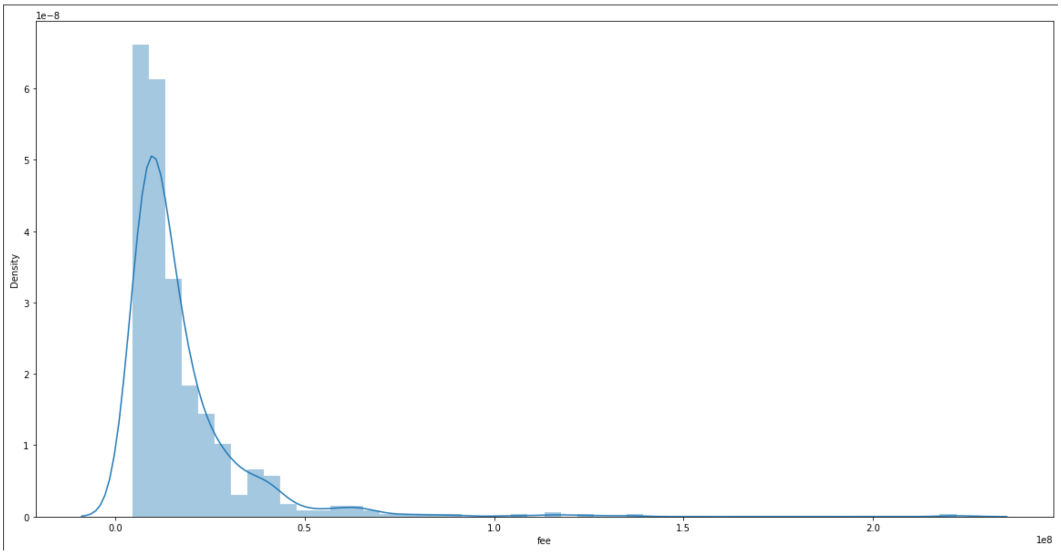


Figure 40

In Figure 39 and Figure 40 above we see the distribution plots for the market value and fee of a player. We can see from the results of these plots that the results match that of the distribution plots above in Figure 38.

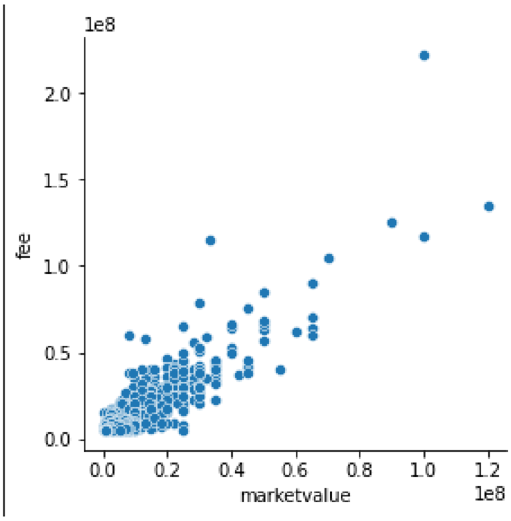


Figure 41

In Figure 41 above we see a scatter plot which plots market value against fee. This image is for reference when looking at the line of best fit being applied later in the model.

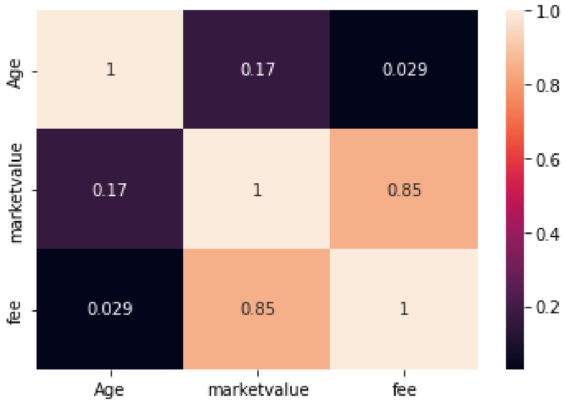
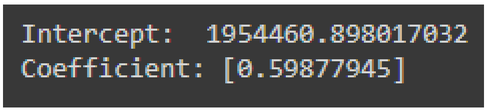


Figure 42

In Figure 42 above we see a Pearson correlation matrix. This image shows the strength of correlation between the variables in the dataset. As expected, every variable has a very strong positive correlation as each other with 1. We then look at how the dependent variable which is fee correlates to market value which is the independent variable in simple regression and age which is the additional independent variable for multi linear regression which we will see later in this section. Fee has a fairly strong positive to correlation to market value with a value of 0.85. The correlation between fee and age along with the correlation between market value and age will be covered later.



```
Intercept:  1954460.898017032
Coefficient: [0.59877945]
```

Figure 43

In Figure 43 above we have the next result of the simple linear regression model. This image shows the coefficients of the model that were achieved from training and testing the model. We are given an intercept value which represents the mean value of the response variable when all predictor models in the model are equal to 0. In this case our intercept is 1954460.90 when rounded to two decimal places. With a coefficient value, if the coefficient is positive, it means as the independent variable increases, the dependent variable will also increase while a negative coefficient would tell you that as the independent variable increases, the dependent variable will decrease. In this model we can see that we have a positive coefficient of 0.60 when rounded to two decimal places. This tells us that there is a moderate positive relationship between the variables when the model is trained and tested.

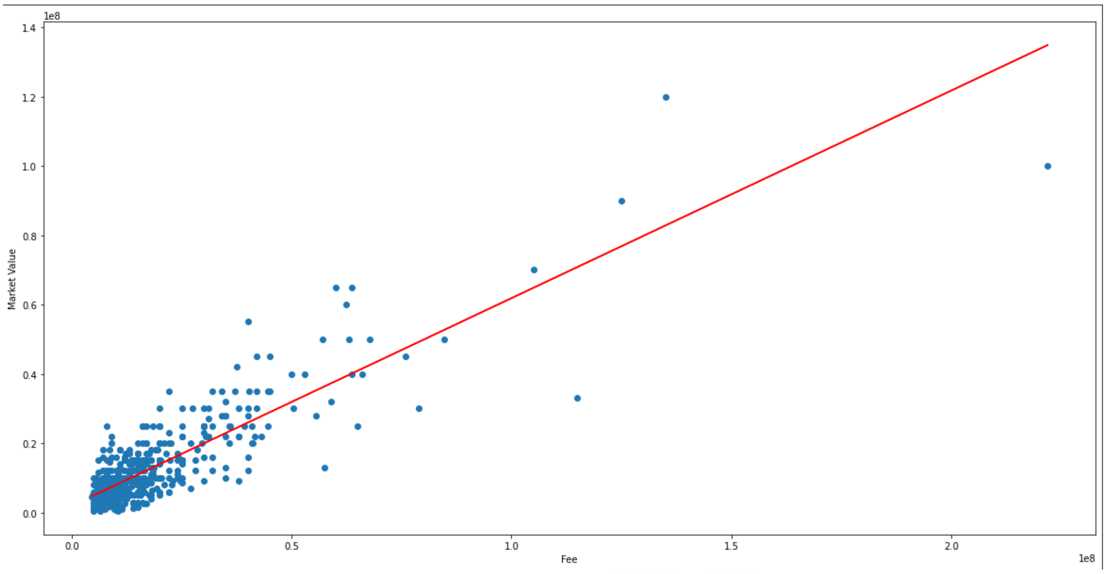
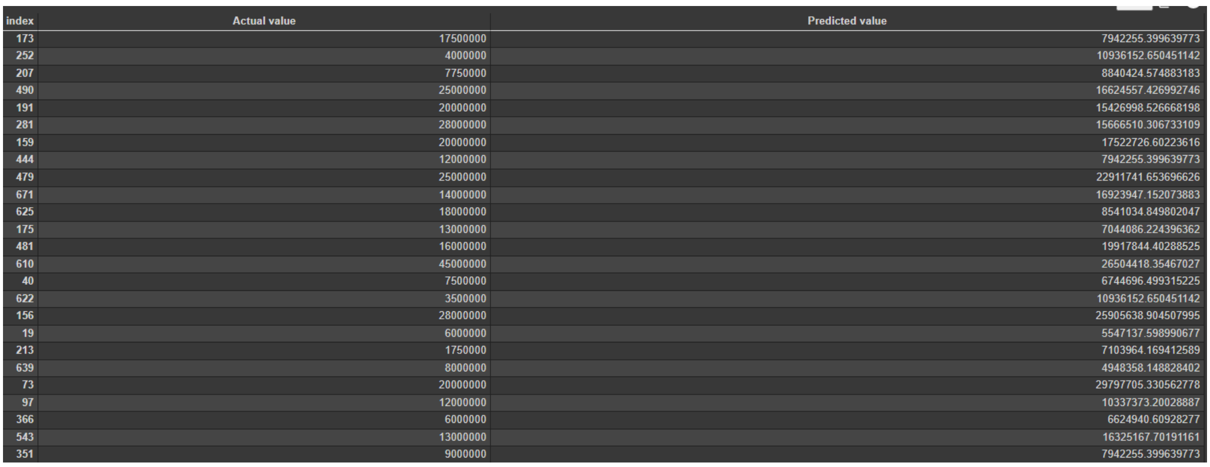


Figure 44

In Figure 44 above a scatter plot is created when the train and testing is applied to the dataset. A line of best fit is also plotted. In this plot a figure of 0.6 would equal £60 million while 1.2 would equal £120 million. The line of best fit allows you to see if there is a relationship between the variables while also allowing you to be able to predict future values based on the line. An example of this would be a player with a market value of £60 million costing £100 million.



index	Actual value	Predicted value
173	17500000	7942255.399639773
252	4000000	10936152.650451142
207	7750000	8840424.574883183
490	25000000	16624557.426992746
191	20000000	15426998.526668198
281	28000000	15666510.306733109
159	20000000	17522726.60223616
444	12000000	7942255.399639773
479	25000000	22911741.653696626
671	14000000	16923947.152073883
625	18000000	8541034.849802047
175	13000000	7044086.224396362
481	16000000	19917844.40288525
610	45000000	26504418.35467027
40	7500000	6744696.499315225
622	3500000	10936152.650451142
156	28000000	25905638.904507995
19	6000000	5547137.598990677
213	1750000	7103964.169412589
639	8000000	4948358.148828402
73	20000000	29797705.330562778
97	12000000	10337373.20028887
366	6000000	6624940.60928277
543	13000000	16325167.70191161
351	9000000	7942255.399639773

Figure 45

In Figure 45 above we see a table displaying a sample of predictions made by the model showing the actual fee for a player to the predicted value. An example of a result is in index 207, the actual value £7.75 million while the predicted value was £8.84 million. In a later testing section, I will test the accuracy of the model.

6.5.2 Multi Linear Regression

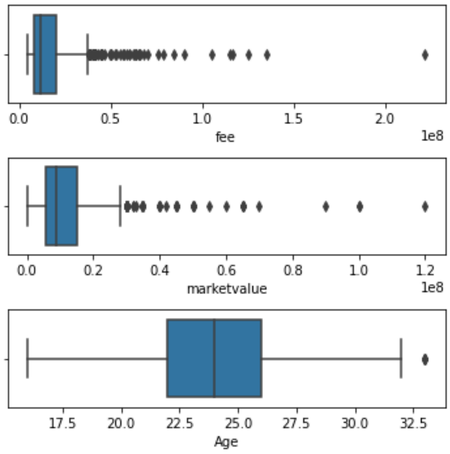


Figure 46

In Figure 46 above see a similar boxplot to that of Figure 24 but with an added boxplot for Age which is the additional independent variable. This shows a median age of 24 in the dataset.

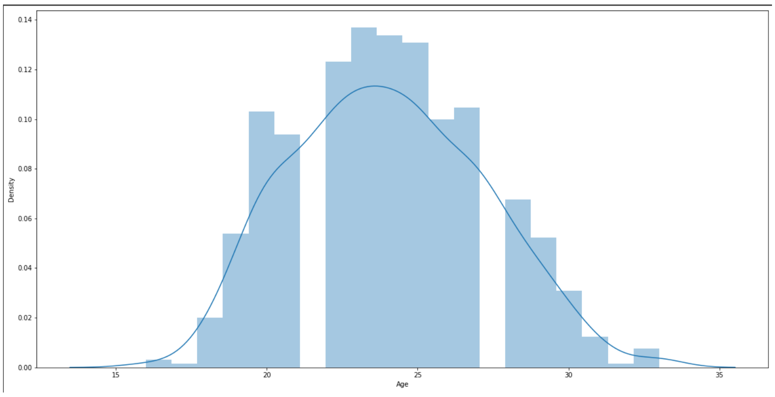


Figure 47

Figure 47 above shows the distribution plot of age in the dataset which is comparable to the boxplot in Figure 46.

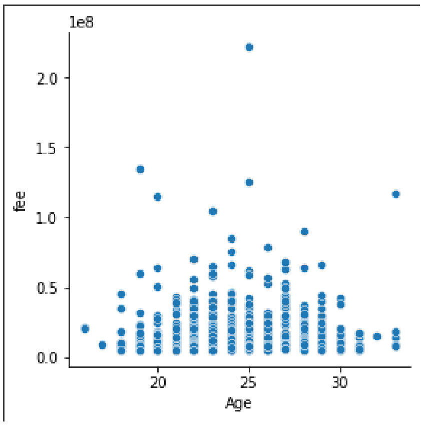


Figure 48

Figure 48 above shows a scatter plot of age to fee.

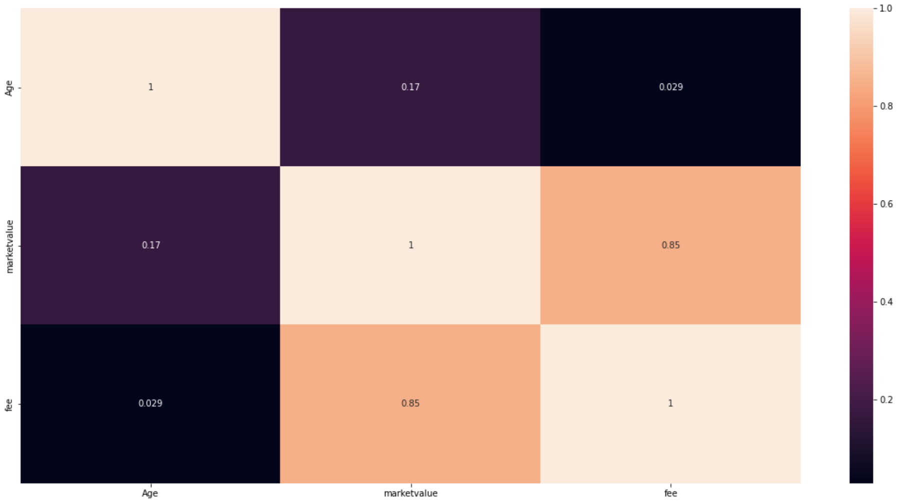
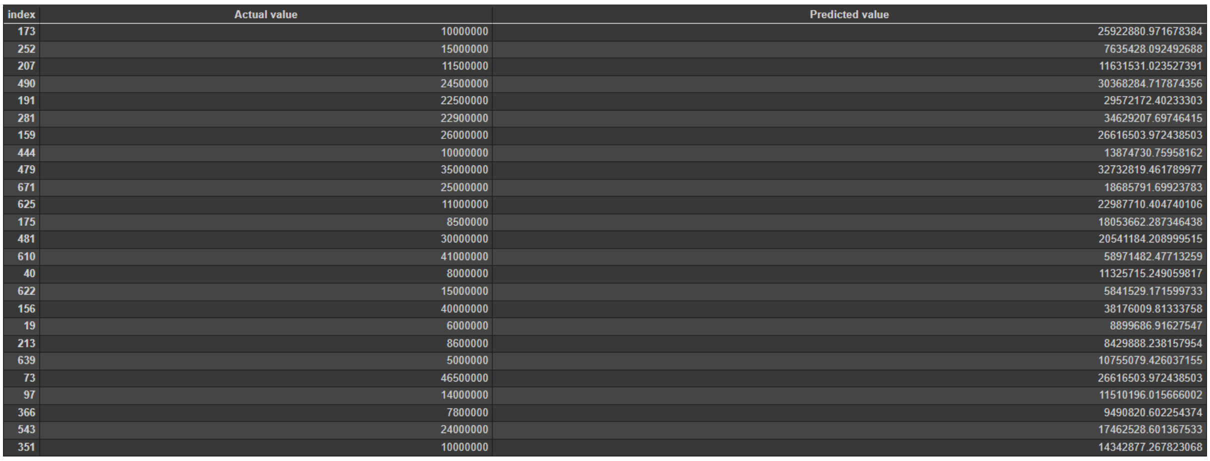


Figure 49

Figure 49 shows the Pearson correlation matrix that we saw in Figure 28. There is a negligible correlation between fee and age while there is also a negligible relationship between market value and age.



index	Actual value	Predicted value
173	10000000	25922880.971678384
252	15000000	7635428.092492688
207	11500000	11631531.023527391
490	24500000	30368284.717874356
191	22500000	29572172.40233303
281	22900000	34629207.69746415
159	26000000	26616503.972438503
444	10000000	13874730.75958162
479	35000000	32732819.461789977
671	25000000	18685791.69923783
625	11000000	22987710.404740106
175	8500000	18053662.287346438
481	30000000	20541184.208999515
610	41000000	58971482.47713259
40	8000000	11325715.249059817
622	15000000	5841529.171599733
156	40000000	38176009.81333758
19	6000000	8899686.91627547
213	8600000	8429888.238157954
639	5000000	10755079.426037155
73	46500000	26616503.972438503
97	14000000	11510196.015666002
366	7800000	9490820.602254374
543	24000000	17462528.601367533
351	10000000	14342877.267823068

Figure 50

Figure 50 shows the actual fee paid for a player to the predicted value of a player when the model has been applied. An example of this is in index 207, the actual fee paid was £11.5 million while the model predicted a value of £11.6 million.

6.5.3 Sentiment Analysis

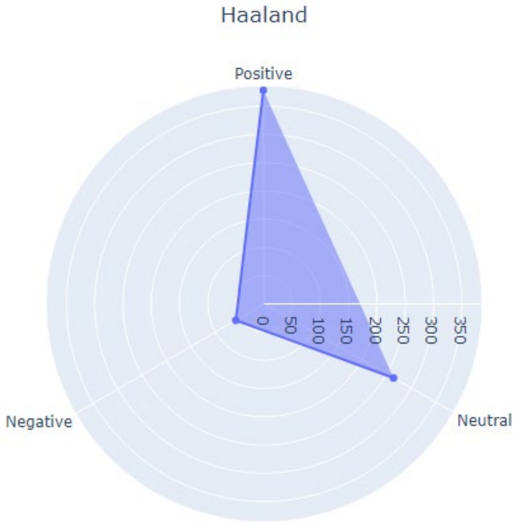


Figure 51

Figure 51 above shows a radar of the overall sentiment to the Erling Haaland transfer cost that will be occurring this summer which will be one of the most polarising deals of the summer. We see that there were between 350 and 400 positive tweets, between 250 and 300 neutral tweets, and 50 and 100 negative tweets about this transfer. This tells us that the overall sentiment to this transfer fee was positive. This data was obtained when scraping tweets that contained the words "Erling Haaland transfer fee."

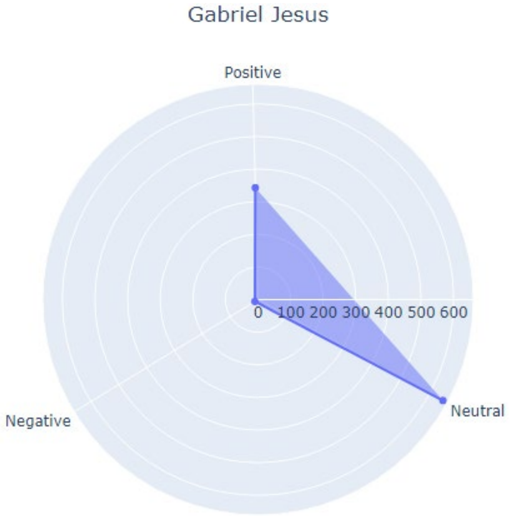


Figure 52

Figure 52 above shows a radar of the overall sentiment to the Gabriel Jesus transfer cost that is rumoured occur this summer which there will be much talk about due to it being a high-profile player. We see that there were between 300 and 400 positive tweets, between 600 and 700 neutral tweets, and 0 and 100 negative tweets about this transfer. This tells us that the overall sentiment to this transfer fee was neutral with lots of positive tweets and very few negative. This data was obtained when scraping tweets that contained the words "Gabriel Jesus transfer fee."

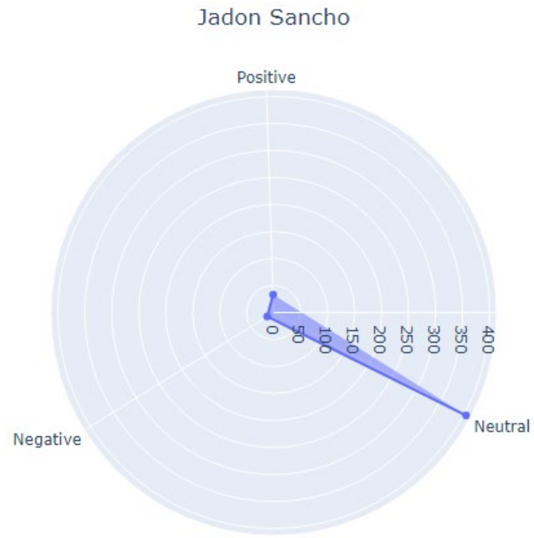


Figure 53

Figure 53 above shows a radar of the overall sentiment to the Jadon Sancho transfer cost that occurred last summer which there was much talk about due to it being one of the most high-profile transfers at the time. We see that there were between 0 and 50 positive tweets, between 400 and 450 neutral tweets, and 0 and 50 negative tweets about this transfer. This tells us that the overall sentiment to this transfer fee was neutral with very few positive and negative tweets. This data was obtained when scraping tweets that contained the words "Jadon Sancho transfer fee."

6.5.4 LSTM

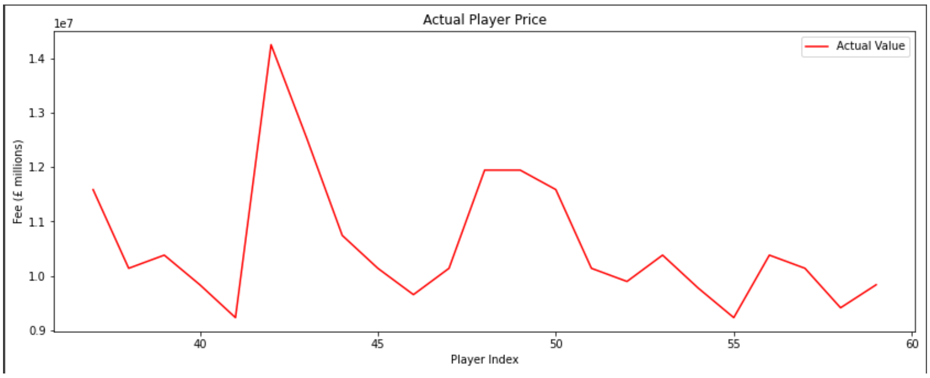


Figure 54

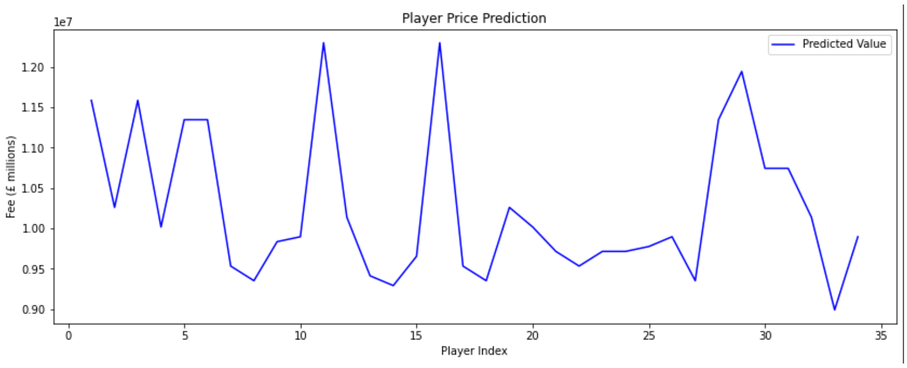


Figure 55

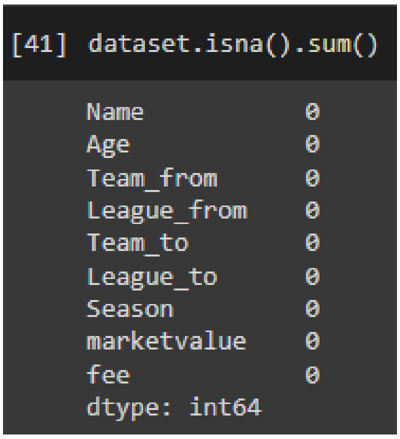
The above images in Figures 54 and 55 show the results of the LSTM model. Figure 40 shows the actual fee paid for a 21-year-old on the Y-axis and X-axis is showing what player index number on the list it is. Figure 55 shows the predicted fee paid for a 21-year-old on the Y-axis and X-axis is showing what player index number on the list it is. We can see that the model has predicted lower fees than the actual fee paid for a player.

7.0 Testing

This section will cover of how the models used in the project were tested.

7.1 Simple Linear Regression

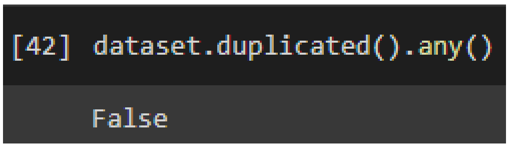
The first test to be run on this model was to see if there were any missing values in the dataset which was achieved by the piece of code and resulting output in Figure 56. This shows us that there are 0 rows with empty or null values.



```
[41] dataset.isna().sum()

     Name           0
     Age            0
     Team_from      0
     League_from    0
     Team_to        0
     League_to      0
     Season         0
     marketvalue    0
     fee            0
     dtype: int64
```

Figure 56

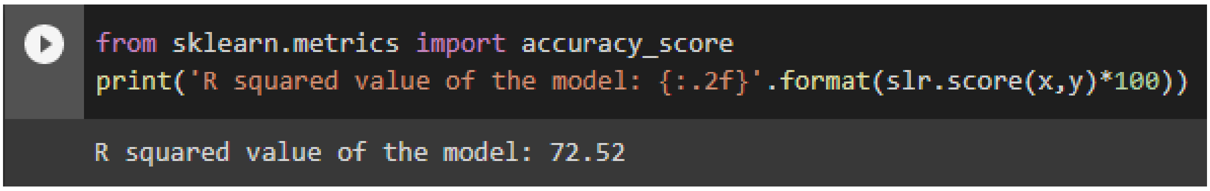
The next set of testing to be carried out on this model was to see if there were any duplicated rows in the dataset before running the model. We can see from Figure 57 below that were no duplicates of rows.



```
[42] dataset.duplicated().any()

     False
```

Figure 57

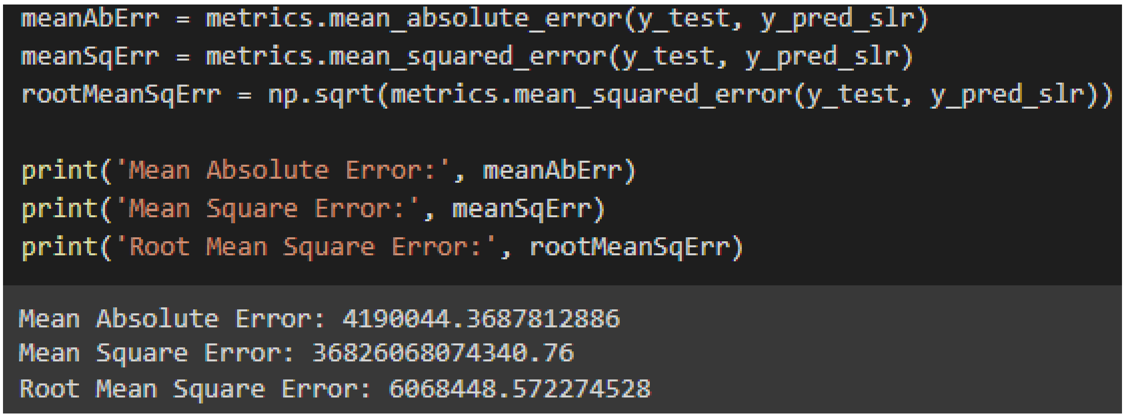
The next step after the model is complete is to test the accuracy of the model. This is achieved by using the accuracy score within sklearn in order to find the R squared value. As seen in Figure 44 below the R squared value of the model is 72.52 which means that it has an accuracy of 72.52% which is a satisfactory number. Seen in Figure 58.



```
from sklearn.metrics import accuracy_score
print('R squared value of the model: {:.2f}'.format(slr.score(x,y)*100))

R squared value of the model: 72.52
```

Figure 58

We then want to find the mean absolute error, mean square error and the root mean square error of the model. This can be seen in Figure 59 below.



```
meanAbErr = metrics.mean_absolute_error(y_test, y_pred_slr)
meanSqErr = metrics.mean_squared_error(y_test, y_pred_slr)
rootMeanSqErr = np.sqrt(metrics.mean_squared_error(y_test, y_pred_slr))

print('Mean Absolute Error:', meanAbErr)
print('Mean Square Error:', meanSqErr)
print('Root Mean Square Error:', rootMeanSqErr)
```

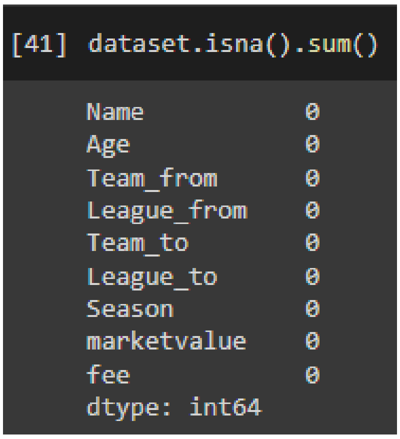
```
Mean Absolute Error: 4190044.3687812886
Mean Square Error: 36826068074340.76
Root Mean Square Error: 6068448.572274528
```

Figure 59

Although you would generally expect much lower numbers in these values, that is because of the range of numbers in the dataset. Due to this dataset only containing values in the tens of millions, these are acceptable values. If the range were for example 0 to 1 then there would be a major problem with the model.

7.2 Multi Linear Regression

The first test to be run on this model was to see if there were any missing values in the dataset which was achieved by the piece of code and resulting output in Figure 60. This shows us that there are 0 rows with empty or null values.

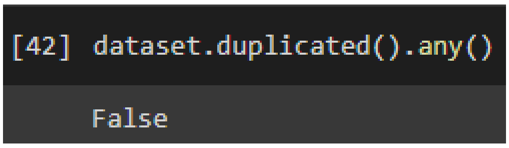


```
[41] dataset.isna().sum()
```

```
Name            0
Age             0
Team_from       0
League_from     0
Team_to         0
League_to       0
Season          0
marketvalue     0
fee             0
dtype: int64
```

Figure 60

The next set of testing to be carried out on this model was to see if there were any duplicated rows in the dataset before running the model. We can see from Figure 61 below that were no duplicates of rows.

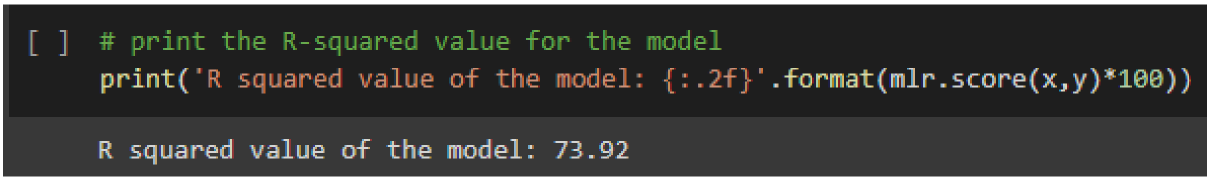


```
[42] dataset.duplicated().any()
```

```
False
```

Figure 61

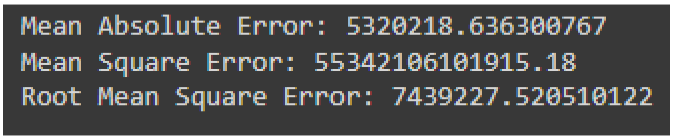
The next step after the model is complete is to test the accuracy of the model. This is achieved by using the accuracy score within sklearn in order to find the R squared value. As seen in Figure 53 below the R squared value of the model is 73.92 which means that it has an accuracy of 73.92% which is a satisfactory number. Seen below in Figure 62.



```
[ ]  # print the R-squared value for the model
     print('R squared value of the model: {:.2f}'.format(mlr.score(x,y)*100))

     R squared value of the model: 73.92
```

Figure 62

We then want to find the mean absolute error, mean square error and the root mean square error of the model. This can be seen in Figure 63 below.



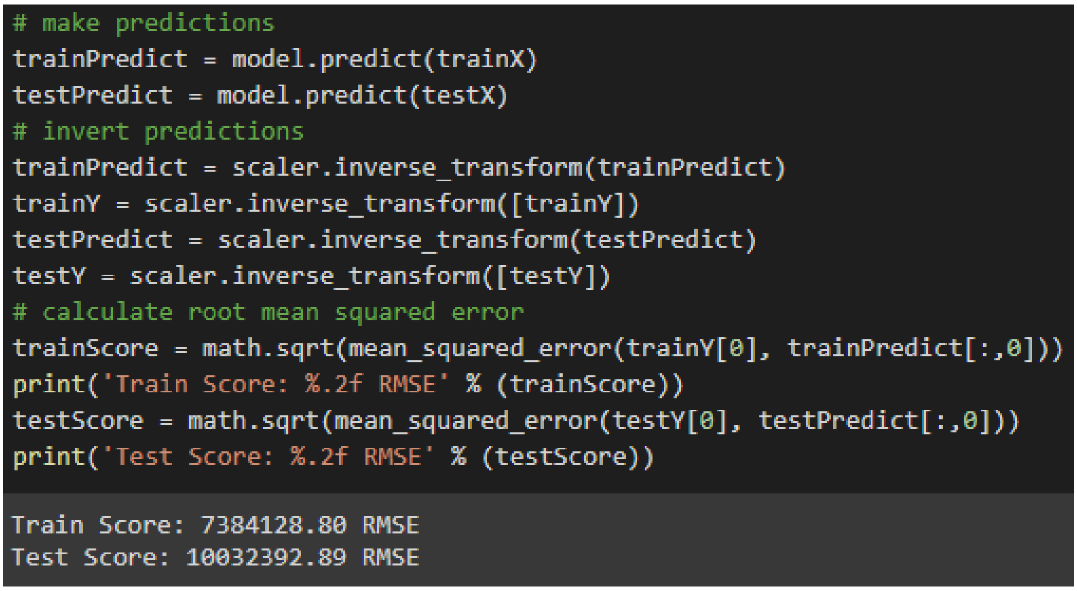
```
Mean Absolute Error: 5320218.636300767
Mean Square Error: 55342106101915.18
Root Mean Square Error: 7439227.520510122
```

Figure 63

Although you would generally expect much lower numbers in these values, that is because of the range of numbers in the dataset. Due to this dataset only containing values in the tens of millions, these are acceptable values. If the range were for example 0 to 1 then there would be a major problem with the model.

7.3 LSTM

The LSTM model was tested by working out the root mean squared error for both the train and test set. The result of the mean squared error for the train and test will be seen in Figure 64 below. Generally, you would expect to see an RMSE of between of 0 and 1 but due to the nature of data in this dataset these figures are acceptable.



```
# make predictions
trainPredict = model.predict(trainX)
testPredict = model.predict(testX)
# invert predictions
trainPredict = scaler.inverse_transform(trainPredict)
trainY = scaler.inverse_transform([trainY])
testPredict = scaler.inverse_transform(testPredict)
testY = scaler.inverse_transform([testY])
# calculate root mean squared error
trainScore = math.sqrt(mean_squared_error(trainY[0], trainPredict[:,0]))
print('Train Score: %.2f RMSE' % (trainScore))
testScore = math.sqrt(mean_squared_error(testY[0], testPredict[:,0]))
print('Test Score: %.2f RMSE' % (testScore))

Train Score: 7384128.80 RMSE
Test Score: 10032392.89 RMSE
```

Figure 64

8.0 Conclusions

To conclude, I believe that from the data acquired for this project and the visualisations and modelling used for analysis that a lot of the key aims were met from the start of the project. From the spending visualisations along with the earnings visualisations, it is safe to say that the Premier League was able to cope much better with the financial restraints of the Covid-19 pandemic than their European counterparts. The Premier Leagues lowest spending season almost doubles that of the highest spending season for each of the other top 4 leagues. For the season during the pandemic the Premier League experienced their second highest ever spending season and the following season which is considered post pandemic was the highest ever spending season. Contrast this to the other four European leagues, which had some of their lowest ever spending seasons with some even making profits in this time period from transfers. Another way to see the financial power in the Premier League is in the chart showing the top 20 highest revenue earning clubs in Europe where 6 of the top 11 spots were occupied by Premier League clubs and 7 of the top 20 while no other league had more than 4 clubs in the top 20. I believe that this clearly shows that the Premier League is financially much more powerful than the other four major leagues as a whole. There are larger clubs from these leagues who can compete and even surpass the Premier League's best but, on the whole, it is stronger financially. This opinion is again seen in the average salary per year where we see the Premier League clubs largely outspending their counterparts and the wage structure of a club is a very good way to spot their financial power opposed to pure spending on transfers. I believe that these conclusions are solid from the analysis and visualisations created and clear trends can be seen but a weakness is a lack of descriptive statistics which would have provided some useful context like mean and median along with variance and standard deviation between these datasets.

The sentiment analysis was quite surprising in results as I expected far more negative public opinion on the state of transfer fees from high-profile transfers. I was happy with how the model turned out where we saw one transfer with overall positive sentiment and two with overall neutral and I believe the radar graphs presented this in a pleasing and easy to interpret manner. A limitation to the analysis carried out is when collecting tweets, if around the time of collecting tweets, a large news organisation such Sky News tweet about a transfer and it gains a lot of retweets, these retweets are collected in the dataset opposed to people's own opinions. This may skew results as the retweet may not give full context about a fans view. Overall, I am happy with how this model worked and would perform with even more refinement such as adding slightly negative and slightly positive to the model to get better variety in results.

The regression models were both a success with accuracies of 72.52% and 73.92% which are more than acceptable figures for a regression model. There is not much that can be changed about the simple linear model, but I believe a limitation of the multi linear model was the fact that only 2 independent variables were used, and a better model could have been created with more. The model was very successful and produced a line of best along with actual and predicted values to look at how player prices could change.

The last part to talk about is the LSTM model which was difficult to learn and implement but was a fulfilling and exciting challenge to complete. Like the regression models above, ideally there would be more independent variables to improve the model but for something I learned and implemented in my spare time I am happy with. The model had good root mean squared error scores relative to the figures in the dataset and the model was able to produce a prediction that resembled the actual values.

9.0 Further Development or Research

Given more time and resources I would apply more descriptive statistics to sections 6.1 to 6.3 while being able to obtain even more historical data to get an idea of how spending as been for a longer period of time and to see if the Premier League has always been the most dominant financially and if not, what caused the change. I would also like to expand this analysis beyond Europe’s top 5 leagues.

As stated above I would like to have been able to apply more independent variables to my regression models such as factoring in the position that a player and some metrics on how well they performed opposed to just their age and market value to predict a transfer fee, which in their own right are useful but a better model could be created with more independent variables. This could again be extended beyond Europe’s top 5 leagues while also being extended to transfers over a longer period of time than just the 2015/2016 to the 2021/2022 season.

With more time and resources, I would love to be able to learn more about LSTM and refine my skills with it while also providing the model with even more variables to learn from in order to get an even more accurate and efficient model, but I am happy with my own progress that had been made.

With the sentiment analysis I would be able to refine what kind of tweets more precisely can be collected in order to get a better array of tweets rather than just a mix of tweets and the same retweets. I would also like to be able to have a wider range of categorisation for polarity scores opposed to the three that were featured in this project. The new ranking could include, negative, somewhat negative, neutral, somewhat positive, and positive. I believe that this would give a more accurate representation of public sentiment. Context is also needed where in the case of Erling Haaland’s transfer fee, though it was very expensive, it was considered a bargain for the quality of player which means the deal itself is well responded to but perhaps the fee would not be in general.

10.0 References

Bibliography

Costaglio, G., Fuccella, V., Giordano, M. & Polese, G., 2009. Monitoring Online Tests through Data Visualization. *Monitoring Online Tests through Data Visualization,* 21(1), pp. 773-784.

Deloitte, 2022. *Deloitte Football Money League 2022.* [Online]
[Accessed 20 March 2022].

Lange, D., 2021. *Average annual first-team player salary in the English Premier League in the United Kingdom (UK) in 2019/2020, by football club.* [Online]
Available at: https://www.statista.com/statistics/547090/average-annual-first-team-player-salary-in-football-clubs-english-premier-league-uk/
[Accessed 20 April 2022].

Lange, D., 2021. *Average annual player salary in Ligue 1 in 2019/2020, by team.* [Online]
Available at: https://www.statista.com/statistics/675528/average-ligue-1-salary-by-team/
[Accessed 20 April 2022].

Lange, D., 2022. *Top-20 European soccer clubs by total revenue 2020/21 season.* [Online]
Available at: https://www.statista.com/statistics/271581/revenue-of-soccer-clubs-worldwide/
[Accessed 1 May 2022].

Lopez, A., 2022. *Average annual salary paid to male soccer players in Spain's highest professional league in 2020/2021, by club.* [Online]
Available at: https://www.statista.com/statistics/675461/average-la-liga-salary-by-team/
[Accessed 25 April 2022].

MacInnes, P., 2021. *French football crisis deepens as TV rights offers fail to reach reserve price.* [Online]
Available at: https://www.theguardian.com/football/2021/feb/02/french-football-crisis-deepens-tv-rights-offers-fail-to-reach-reserve-price-ligue-1-2
[Accessed 22 April 2022].

Soccerment Research, 2020. *The Growing Importance of Football Analytics.* [Online]
Available at: https://soccerment.com/the-importance-of-football-analytics/
[Accessed 20 January 2022].

Taylor, L. & Conn, D., 2020. *Premier League clubs set for £500m collective loss due to coronavirus.* [Online]
Available at: https://www.theguardian.com/football/2020/jun/11/premier-league-clubs-set-for-500m-collective-loss-due-to-coronavirus
[Accessed 11 April 2022].

Transfermarkt, 2010. *www.transfermarkt.co.uk.* [Online]
Available at: https://www.transfermarkt.co.uk/
[Accessed 21 January 2022].

11.0 Appendices

This section contains information that is supplementary to the main body of the report such as the project proposal and monthly journals.

11.1.0 Project Proposal

National College of Ireland

Project Proposal

Analysing the impact of the COVID 19 Pandemic on Association Football's Top 5 Leagues and its finances.

07/11/2021

Computing

Data Analytics

2021/2022

Andrew Kelly

X18212158

X18212158@student.ncirl.ie

Contents

11.1.1 Objectives

In this project I am going to analyse the impact that the COVID 19 pandemic had on Association Football. I will analyse how the likes of revenues of football clubs changed during the pandemic such as how they were affected by having either zero fans in attendance or when they were operating on limited capacity. I will look how transfers and wages of clubs were affected during the pandemic due to all matches being televised opposed to only limited amounts games being televised during a normal season. I will investigate how this change in income affected the transfer strategies of football and the structure of how of the deals were made by comparing to how this was done pre pandemic and I will look at all major European leagues and how they got on compared to the English Premier League. I will also look at how clubs financed their new contracts for players and see if this was any different to how it was done previously. As we are now coming out of the pandemic and stadiums are now seeing the return of fans, I will look at how football clubs' finances have recovered in the wake of all this and try to analyse where football transfers may go in years to come. I will attempt to see trends in where expenditures of clubs are going and will Premier League continue to dominate all finances in football. I will also aim to see what public sentiment is in relation to transfer fees in football.

11.1.2 Background

One of the main drivers behind me picking this topic for my project was due to my love of football and my fascination with the sheer amount of money that is in the modern game. Through the last year and a half, football fans have heard and seen so much about clubs have struggled during the pandemic and this evident with the layoffs to staff made by some clubs, the temporary cut in wages to playing and non-playing staff at others, and some clubs even ceasing to exist due to the toll it took on them. A large part of the narrative then became about how clubs would be lacking the funds to make many major transfer signings during the summer if any at all with some saying they would have to sell players in order to buy new ones. As summer came along, we saw that a lot of English clubs could seemingly still pay big transfer fees for players but what was also seen was the fact that European clubs weren't able to offer such fees and were even forced to pursue loan deals with options to buy. All of this was of great interest to me, and I want to prove or disprove some of the narratives that were pushed. I will meet the objectives by collecting and data and display it in such a way that we will be able to definitively prove whether such claims were true and get a true picture of how football was impacted financially by COVID.

11.1.3 State of the Art

Deloitte carry out annual reviews of football finances, but their reviews will only show the finances of a given year while not comparing to others. In my project I will be looking at pre covid, during covid and post covid finances while also looking to predict where football finances are going over the next few years. Deloittes report only looks at revenues of football clubs while mine will aim to analyse the likes of transfer fees, wage structure while also showing how some leagues were far less affected than others. This report will paint a very clear picture of finances pre, during and post covid and how each of the major European leagues were affected. The Deloitte report also only carries out financial analysis by stating figures while my report will carry out the likes of regression models, sentiment analysis and LSTM.

11.1.4 Data

The type data that will be required for this project are the financial details of football clubs such as matchday revenues, tv deal revenue, sponsorship revenue, income from transfer fees and expenditure on transfers. I will require attendance figures in stadiums for when they are at full capacity and when they had to run at limited capacity. We will gather the average price per ticket and even try to find some information on the spending habits of fans at the stadium pre and post pandemic. We will gather share prices of publicly trading clubs in order to compare. Some places where I will gather my data from are from the Swiss Ramble blog and their Twitter feed. The UEFA club licensing benchmarking report and can also take some financial figures from Deloittes financial report. Other sources will be from news outlets who sometimes report on the finances of a specific club or league. Clubs by law are required to declare their finances in order to comply with the likes of financial fair play regulations so these figures will widely available and are posted on a yearly basis. For the share prices of clubs, I can check stock exchanges to see what the share price was of any given club at any given date so I can gather the price from numerous periods before during and after the pandemic to show how they have changed.

11.1.5 Methodology & Analysis

I plan to complete this project in sprint cycles. This is a very common approach to completing projects in companies and is an approach that I am familiar with from my 6-month internship. During my entire time there I worked on a team which operated in spring cycles. I found this to be a very affective and manageable to allocate your tasks and get them done in the most efficient manner. To do this I will break down each individual task that needs to be done in order to complete the project. I will then create a Kanban board on a site like Trello and input all the tasks. I will try to work out how long each task should take and create a series of sprints to put each task in to in order to best complete my project. A good basis for different milestones will be when I look at different deadlines that are due. I can look at these deadlines and make judgements on how much work should be done by this time and then set the sprints accordingly. A good way to find out how long each task will take will be to will be to carry out some sample tasks that will require a similar type of work and then use that as reference while adjusting for now knowing how to carry out such a task. Working in a sprint cycle will allow for better focus on each individual task, should improve overall quality of each task, improve productivity and due to the adaptability of a spring it will mean that any changes that need to be made will have less of a negative impact on the project and my ability to complete it.

11.1.6 Technical Details

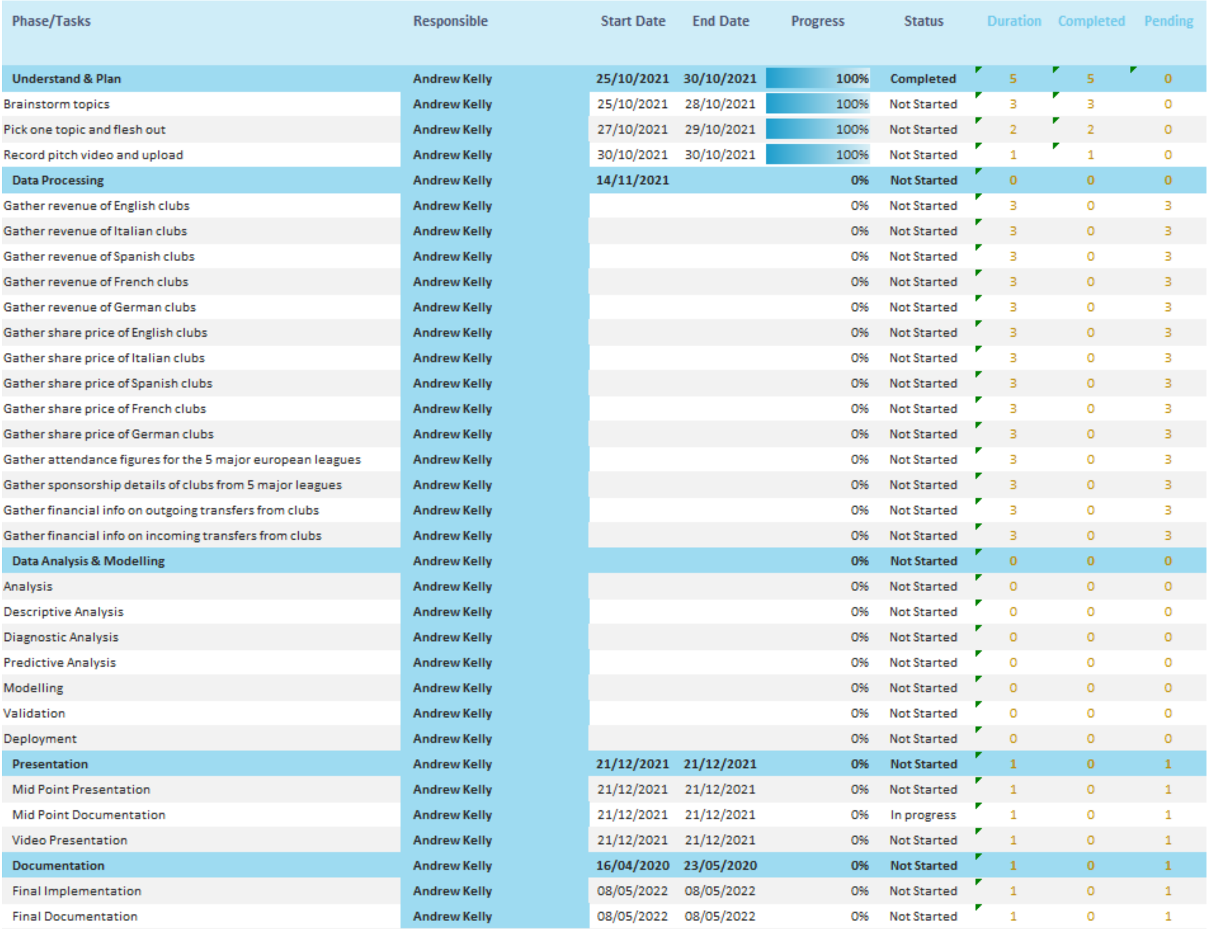
Regression Models – For this project I will implement regression models such as simple linear regression and multi linear regression. These are two very commonly used models for prediction to them having a dependent variable with predictor values know as independent variables. While multi linear is considered better due to an increased number of variables I am interested to see how both models work when implemented. These models will allow me to see trends in where transfers are going but also attempt to make predictions on transfer prices.

Sentiment Analysis – This approach will be used in order to gain an idea of public sentiment in regard to major transfers and the fee involved. This will be achieved by collecting Tweet's and running a sentiment analysis model on them through Python. This will be a challenge to teach myself, but I look forward to it and think that as a model it will provide great value to the project to see what the public really think of transfer fees.

LSTM – This is a prediction model that is commonly used in predicting stock prices. It allows a model to learn what values are more important when training a model and storing them long term while also forgetting the less relevant values to a model which also helps accuracy. This will be interesting model to teach myself and is regarded as a very good method of price prediction so will be useful in our attempt to predict player price of players at a specific age.

All analysis will be run in Python with libraries such as Beautifulsoup for web scraping, keras for LSTM and sklearn for my regression models. Tweets will be collected through Twitter's developer platform and analysed with libraries in Python such as NLTK and Vader lexicon. All visualisations will be created programmatically through Python with libraries such as Matplotlib and Plotly.

11.1.7 Project Plan



Phase/Tasks	Responsible	Start Date	End Date	Progress	Status	Duration	Completed	Pending
Understand & Plan	**Andrew Kelly**	**25/10/2021**	**30/10/2021**	**100%**	**Completed**	5	5	0
Brainstorm topics	**Andrew Kelly**	25/10/2021	28/10/2021	100%	Not Started	3	3	0
Pick one topic and flesh out	**Andrew Kelly**	27/10/2021	29/10/2021	100%	Not Started	2	2	0
Record pitch video and upload	**Andrew Kelly**	30/10/2021	30/10/2021	100%	Not Started	1	1	0
Data Processing	**Andrew Kelly**	**14/11/2021**		**0%**	**Not Started**	0	0	0
Gather revenue of English clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Gather revenue of Italian clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Gather revenue of Spanish clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Gather revenue of French clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Gather revenue of German clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Gather share price of English clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Gather share price of Italian clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Gather share price of Spanish clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Gather share price of French clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Gather share price of German clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Gather attendance figures for the 5 major european leagues	**Andrew Kelly**			0%	Not Started	3	0	3
Gather sponsorship details of clubs from 5 major leagues	**Andrew Kelly**			0%	Not Started	3	0	3
Gather financial info on outgoing transfers from clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Gather financial info on incoming transfers from clubs	**Andrew Kelly**			0%	Not Started	3	0	3
Data Analysis & Modelling	**Andrew Kelly**			**0%**	**Not Started**	0	0	0
Analysis	**Andrew Kelly**			0%	Not Started	0	0	0
Descriptive Analysis	**Andrew Kelly**			0%	Not Started	0	0	0
Diagnostic Analysis	**Andrew Kelly**			0%	Not Started	0	0	0
Predictive Analysis	**Andrew Kelly**			0%	Not Started	0	0	0
Modelling	**Andrew Kelly**			0%	Not Started	0	0	0
Validation	**Andrew Kelly**			0%	Not Started	0	0	0
Deployment	**Andrew Kelly**			0%	Not Started	0	0	0
Presentation	**Andrew Kelly**	**21/12/2021**	**21/12/2021**	**0%**	**Not Started**	1	0	1
Mid Point Presentation	**Andrew Kelly**	21/12/2021	21/12/2021	0%	Not Started	1	0	1
Mid Point Documentation	**Andrew Kelly**	21/12/2021	21/12/2021	0%	In progress	1	0	1
Video Presentation	**Andrew Kelly**	21/12/2021	21/12/2021	0%	Not Started	1	0	1
Documentation	**Andrew Kelly**	**16/04/2020**	**23/05/2020**	**0%**	**Not Started**	1	0	1
Final Implementation	**Andrew Kelly**	08/05/2022	08/05/2022	0%	Not Started	1	0	1
Final Documentation	**Andrew Kelly**	08/05/2022	08/05/2022	0%	Not Started	1	0	1

11.2 Monthly Journals

Andrew Kelly – X18212158 – Journal 1

So far for my software I have not been able to make too much progress due to how early we are in the process. We also have yet to receive confirmation of whether our chosen topic is suitable to do so it is hard to make to do too much when this is the case.

We have received confirmation of who our project supervisor is so that is a big step on the path to starting our project as I can soon communicate with them about my project idea and how to go about it.

My plan is to investigate the effects of covid on the financial side of football so I have started to look for some resources to find this financial information along with looking into the languages and technologies that I will use for the project. At the moment the language I am leaning towards most is Python, followed by R. For the technologies I will they will most likely include the likes of SPSS and an IDE like R Studio if I use R and the likes of Pydev, Pycharm or Sublime Text should I use Python.

I had wanted to do my project on something that I had a strong interest in to keep me even more engaged in it and through that drive me to do even better in it. Football has always been a great passion of mine and I've been interested about the financial side through reading football finance blog posts from a great twitter page that goes by Swiss Ramble and through the writings of author Simon Kuper who has penned books such as Soccernomics and Football Against the Enemy. Though I am focusing on football statistics, they are what initially got me interested in data analytics and over the past few years I have followed many great bloggers and their work on football analytics and how it used to measure how well teams and players perform and then also how football clubs are now making analytics a big part of their recruitment process of players and the best example of this in recent would be a football club like Brentford who have had great success in recent years and almost solely rely on analytics for their recruitment. The past year we have heard so much about the financial impact that COVID has had on football, and I would like to be able to quantify and visualise this to see how true it is, the extent of it, and does it also apply to the beast that is the Premier League.

This month due to the volume of work we had due with assignments I did not get to complete as much as I would have liked for my software project. I did however find some websites where I can scrape some of the data that I need like transfermarket.com where I have been able to scrape transfer fees of football players which will be an important dataset for this project. In order to do this, I used Pycharm as my IDE and I used Python to scrape the data from the site. I have also found an annual review by Deloitte in to finances in football where I can scrape data from for a wider range of financial figures.

Andrew Kelly – X18212158 – Journal 2

This month due to the volume of work we had due with assignments I did not get to complete as much as I would have liked for my software project. I did however find some websites where I can scrape some of the data that I need like transfermarket.com where I have been able to scrape transfer fees of football players which will be an important dataset for this project. In order to do this, I used Pycharm as my IDE and I used Python to scrape the data from the site. I have also found an annual review by Deloitte in to finances in football where I can scrape data from for a wider range of financial figures.

Andrew Kelly - X18212158 – December Journal

This was another remarkably busy month in terms of assignments, but I was successfully able to scrape lots of data that I need from transfermarkt.com while also finding some other sources of data for my analysis. The key part though was to get the transfer data as this is a big basis of the report. This was also some code that I expected to be the trickiest as I am still learning to work with Python, but I am happy with the data I have scraped and how it looks in my csv files. I was also able to start testing plots that I will use for the report in both Python and through the use of tools such as Tableau which I find to be a very nice tool to work with and displays the data in appealing and easy to understand ways, but I hope to create my own visuals of a similar level through the Python language. The next steps for me will be to get all the transfer data I need while also starting to collect data on a range of football finances such as their revenues, ticket sales etc., Once all this data is collected and plotted, I will then look to do some predictive analysis to see where the future of transfers is going. I am hoping that with the less classes we have next semester along with the improvement of my programming skills and my understanding of what needs to be done that I can make significant progress on this project.

January/February Journal

This was a period where I was able to make good progress with my project as I was learning about models such as sentiment analysis and LSTM which will be useful to the project. There were learnt by myself in my free time and tested with in order to learn how to implement them to my project. These were challenging but I thoroughly enjoyed learning about them and look forward to implementing them. Fortunately, at this time I had no assignments due for my classes so I could focus on learning these models.

March/April Journal

This period was very busy due to having numerous CA’s and TABA’s due for both my classes with some even overlapping each other which made time a premium. These were tricky assignments that I needed to perform well in so I couldn’t dedicate the ideal amount of time to the project, but I endeavoured and managed to implement my regression models, sentiment analysis, and LSTM. There was also a group project in this period which could often take up more time than expected but I am happy with the progress made in this period.

May Journal

This was the last month in the project, and I am very happy with the progress made thus far as all the models were implemented and conclusions were ready to be drawn from them. The majority of late April and May were spent working on the report which I knew would be a long process. As of now I am nearing the deadline and have finished all sections of the report with just the project showcase poster left to submit. This project has been a tremendous challenge and one I am delighted to have overcome and am overall very happy with my project. I am immensely grateful to my project supervisor for their support throughout the project along with all the staff who helped us along the way no matter how big or small the matter.