National College of Ireland

Configuration Manual

MSc Research Project
MSCDATOP

Andrew Baumann
Student ID: 20156138

School of Computing
National College of Ireland

Supervisor: Jorge Basilio

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Andrew Baumann
Student ID:	20156138
Programme:	MSCDATOP **Year:** 2022
Module:	MSc Research Project
Supervisor:	Jorge Basilio
Submission Due Date:	Monday 19th September 2022
Project Title:	A Multi-Stage Clustering Algorithm to Re-Evaluate Basketball Positions and Performance Analysis
Word Count:	2884 **Page Count:** 10

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

Signature: .....

Date: 13th August 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

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

Configuration Manual

Andrew Baumann
20156138
MSCDATOP

---

1 Introduction

This configuration manual aims to provide information on the software tools used in the implementation of this project. The environment and libraries are highlighted and the choice of each explained thoroughly. Finally, a step by step running order is explained before expected outcomes and examples are presented.

This manual does not outline or explain the installation of standard software or tools, but rather acts as a guide for replicating the project analysis or running of the code provided with the assumption of knowledge for the broader aspects of installation of environments and libraries. With that being said, the environments, interpreters, libraries, and all other needs will be outlined in a clear and concise manner.

2 Hardware Specification

Much of the running throughout this project has been separated into separate scripts, with some being more intensive and time consuming than others. One such example would be the various web scraping scripts, which will be discussed in sections to follow. As well as the scraping, the final master dataset formed is rather extensive, so an under-powered computer may struggle at times.

Whilst relatively user friendly given the splitting of the scripts, due to this somewhat heavy-duty nature of some of these files, I have listed the hardware specifications of the device used throughout this research and analysis in Figure 1, which as seen, are relatively standard and not out of reach for most devices. I would suggest any replications made be done so on a device with similar or better specifications. It is also worth noting that this project was carried out using Microsoft Windows 10 operating system but could be replicated on other systems that support Python environments if preferred.

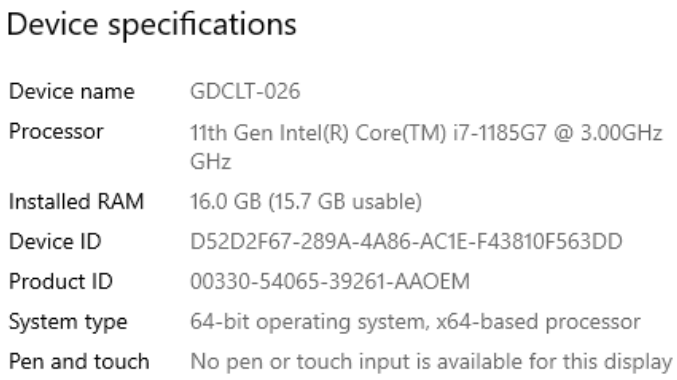

Figure 1: Hardware Details

3 Required Software

To reiterate, the instillation of standard software and platforms will not be discussed in this manual as stipulated in the handbook, but it is worth noting the languages, environments, and libraries used as well as how the data was sourced.

3.1 Python

Python was the data processing programming language throughout this analysis for a number of reasons. It possesses exceptional versatility in being able to connect to various data sources, as well as converting data in multiple formats, and excellent visual capabilities. Due to its better handle of web scraping libraries, the PyCharm environment was used for this first step, whilst the Notebook style of Jupyter was preferred for the second stage of the data analysis and model building.

3.1.1 PyCharm IDE

PyCharm is a dedicated integrated development environment (IDE) tool with complete set of tools for Python development. An IDE allows the building of applications in combining common developer tools into a single graphical user interface (GUI). PyCharm operates under a run all in script feature and possess better assistance and debugging capabilities than most common IDEs. PyCharm is targeted at developers seeking to write complex coding, often with one primary objective in mind. It is also more compatible with the web scraping libraries used both in terms of running and updates.

For the reasons listed above, and simply from personal experience in web scraping libraries and their suitability in certain other IDEs, PyCharm was selected as the environment for the first stage of the project in web scraping and data gathering. Given the files for this section are Python files (.py), they could be run in other IDEs, but deviations from the suggested and trialled environments should only be done with familiarity with the processes.

3.1.2 Jupyter Notebook

Jupyter Notebook is an open-source IDE used to create Jupyter documents that can be created and shared with live codes, including Python. The primary difference in this analysis is that is runs these Python Notebook files (.ipynb) rather than the standard Python files. Where it benefits over other IDEs such as PyCharm is two-fold; its allowance for both inline code execution using blocks and for single line graphing support.

The two main advantages listed above allow for individual running of sections of code within Jupyter, and thus make it the ideal for projects requiring data exploration and visualisation, as well as testing model development in a step-by-step manner seeing output for each section as it comes. This proved invaluable and was particularly useful for areas such as parameter assessment, as well as developing and building upon various models throughout this process.

3.1.3 Libraries Required

There are a number of libraries which allow for exemplary processing, transformation, mining, model generation, and evaluation, including but not limited to, Pandas, Scikit-learn, and Matplotlib. As discussed, there were two separate environments used, one for the web scraping and one for all that followed that. Figure 2 shows the libraries required for the scraping part, as well as the start and end year entry fields at the top which simply need to be changed to obtain a file for this range. As seen, the primary libraries for this scrape are BeautifulSoup and Selenium, the nature of each will be discussed in Section 3.4.

```
start_year = 2013
end_year = 2022

import time
from pathlib import Path

import pandas as pd
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.by import By

import chromedriver_autoinstaller


file_path = str(Path.home() / 'Downloads')
pd.set_option('display.max_rows', None, 'display.max_columns', None, 'display.width', None)
```

Figure 2: Python libraries for web scraping scripts

Given the need for web scraping to access the data, as will be discussed following this, chromedriver-autoinstaller is also a noteworthy import for the scraping side of this code. This ensures the latest ChromeDriver instance is automatically downloaded before the scrape and is supported on both Windows and Mac operating systems. This is significant given the

formation of the dataset by the automatic iterating through years in web browsers to form a master dataset, as has been set up in the coding for this project. Whilst headless browser options were also explored for the scraping stage, the running of the chrome browser invisibly in the background in this manner caused inconsistencies as is often seen in web scraping. As such and given the size of the data being scraped, for any replications of this, enabling this feature would not be recommended.

The libraries required for stage two, including every step from forming the master dataframe with the scraped data all the way to the machine learning model development and assessment are seen in Figure 3. Due to the fact that three separate files were scraped in the first section of this, one each for player statistics, advanced statistics, and shooting outputs, Pandas and NumPy proved valuable in the forming of a master dataframe from these various scraped files. For visualisations, Seaborn and Matplotlib were vital, and for the model developing, testing, and assessing, SciPy and Scikit-Learn, or Sklearn, and numerous modules were availed of heavily.

```
In [1]:
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import scipy.cluster.hierarchy as sch
import seaborn as sns
import sklearn
import sklearn.cluster as cluster
import sklearn.metrics as metrics

from sklearn import metrics
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import cohen_kappa_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import silhouette_samples, silhouette_score
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import scale
from sklearn.svm import SVC
```

Figure 3: Python libraries for dataframe forming, model development, and analysis

3.2 Data Source

The analysis was centred around basketball for this research, specifically the National Basketball Association (NBA). Statistics and availability of data in the NBA online was a major advantage in researching within this field through the NBA themselves or through partnered statistical resource websites such as Basketball Reference, which was the source of data for this analysis.

Basketball Reference is widely considered to be the best place for data relating to the NBA and is even used by those working within and reporting on the league itself. There is a vast amount of data available dating back decades and as such, proving to be the ideal environment for data sources. The only downfall is the downloading of this data. While it is available to view in tables on their website, there is no download link, and even if there was, data is only shown one year at a time. As such web scraping was a necessity for obtaining the data in readable CSV files.

3.3 Fair Use of Data

Section 9 of the NBA's terms of use specifies the requirements in using NBA statistics. These are primarily centred around providing a prominent attribution to the NBA for the use and not using this information for gambling or fantasy sports purposes. Given the scope of this research is neither for gambling nor fantasy sports, and that ample reference and acknowledgement has been given to the NBA, the use of such data in the accompanying report is acceptable under their terms of use.

3.4 Web Scraping

The scraping process begins with date range seen in Figure 2. After this a loop is created in which a web browser instance is opened directed to the appropriate site for the correct year before the process is repeated with the next year. Selenium is availed of in this manner in order to set up the simulating of clicking and navigating in this automated browser instance. As mentioned, there is no download link even after directing to the appropriate table, and as such, BeautifulSoup is used to fetch the data from the html and append it to a dataframe in Python, before the process begins again for the next year. This is repeated until the data from all years in the parameters set at the top have been collected and concatenated into one dataframe for all years. The process contains three separate scripts for the three separate tables, which are compiled into one master dataframe after this scraping.

The nature and difficulty in simply downloading the data as a CSV meant for this combination of web scraping libraries used. Selenium can used to program the script to act as a normal user and navigate to the right page and show the appropriate table in its entirety. BeautifulSoup on the other hand is used to read this data and bring it into Python in a readable Pandas dataframe format. The scripts created allow for a seamless collaboration between the two libraries in this manner, with the only cause for concern being to set the appropriate date range before a CSV is outputted with all relevant data. Due to the relative high complexity and computational power of Selenium in particular over BeautifulSoup, run times can be a bit long and strenuous on the device for broader date ranges, and as such any device should be similar in specification to that shown to be used previously in this manual.

4 Order of Runs

The aforementioned split of the web scraping done in PyCharm, and analysis carried out in Jupyter means there are two separate groups of code to discuss as well as the order in which to run them. The scraping and gathering of data will naturally have to come before any analysis, and so starting in PyCharm with the scraping scripts is necessary. As mentioned, there are three dataframes being scraped from Basketball Reference and have been divided into one script each. By simply entering the date range, seen in Figure 2, and running the script, data is scraped for the time specified and a CSV file is downloaded after the run. All three scripts are seen in Figure 4, as well as the latest ChromeDriver instance from the auto-installer. Given one CSV is outputted per Python file and all will be run after to create a master dataframe, it does not matter what order these three are run, but only that they are run before the next stages to be discussed.

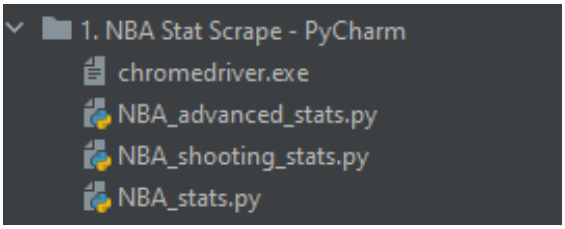

Figure 4: Web scraping scripts, as seen in PyCharm IDE

Moving on to the second phase of the code runs, all scripts are numbered for ease of understanding and running, as seen in Figure 5, as well as given a name describing their purpose. The following section will describe each fully as show some outcome examples.

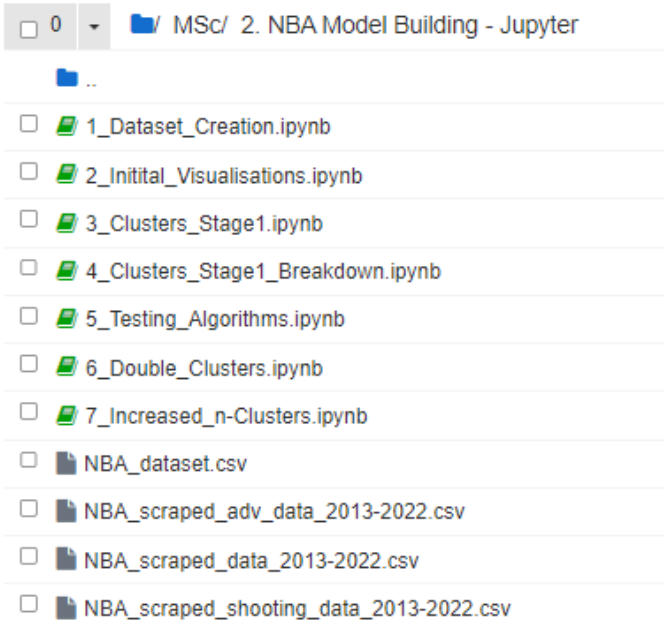

Figure 5: List of all scripts after data has been scraped, as seen in Jupyter Notebooks

5 Expected Outcome

5.1 Data Scraping - PyCharm

Each of the three scraping scripts work by iterating through each year in the set range and opening a Chrome browser for the year in question before collecting the data shown from Basketball Reference and writing it to a Pandas dataframe. Figure 6 shows an example of this ChromeDriver executable in action, while Figure 7 shows the run output in Python with progress as each year is collected. After iterating through each year and gathering the data, each file in the scraping stage of the scripts will output one dataframe relating to the information being pulled, as shown in Figure 8. These three will be fed into the next stage of scrips to form a master dataframe and file for analysis and model development.

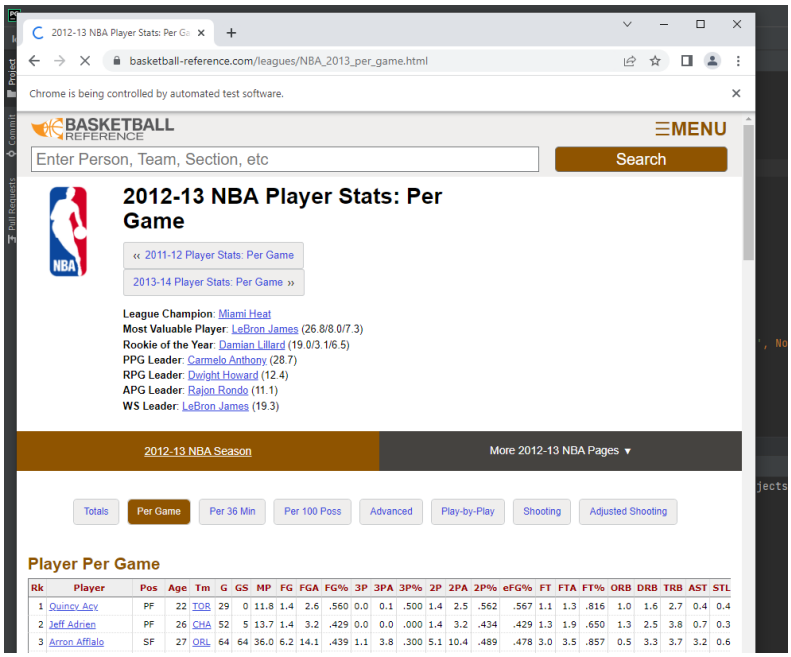

Figure 6: Automated chrome window from ChromeDriver in Python which iterates through each year allowing the data to be pulled and written to Python

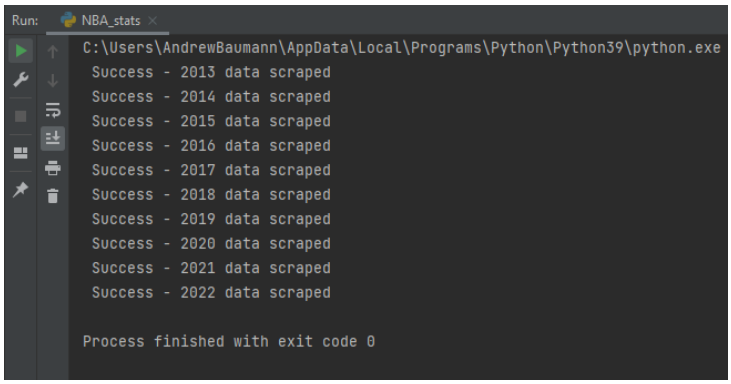

Figure 7: Python run output showing progress of each year's data scraped

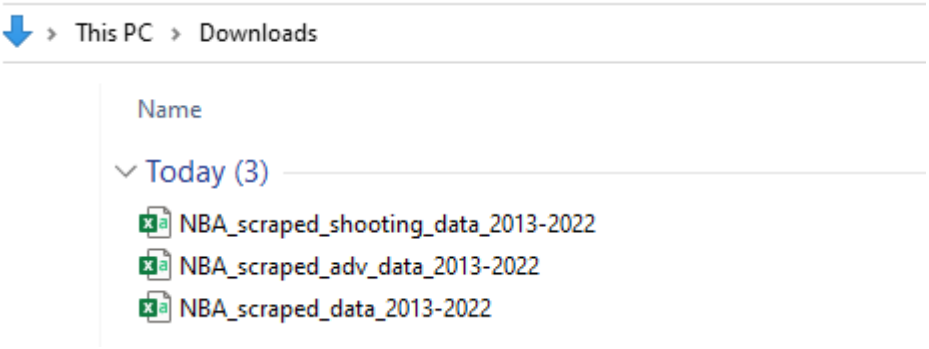

Figure 8: Output of three scraped dataframes for the specified date range

It is worth noting, if there are any difficulties in running the scraping scripts, it is almost certainly related to updates in the ChromeDriver file seen in Figure 4, as these update often and require the latest version to be included for a successful run. Necessary versions can be downloaded at https://chromedriver.chromium.org/downloads and simply pasted to the script location to replace the current exe file if required.

5.2 Analysis and Model Development - Jupyter

The first file in the analysis stage, labelled Dataset Creation takes the three scraped CSV files, seen at the bottom of Figure 5, and formats them into one readable master CSV file, labelled as NBA_dataset.csv, for the analysis to follow this step. Such formatting includes addressing blank values, converting column data types into readable formats for Python and Pandas, cleaning of redundant data, forming new columns based on data present such as a mid-range shooting statistic rather than five separate values for this, and overall merging of the data in an accurate and tidy manner. In the interest of consistency, any athletes who had played ten or fewer games in a year were removed, leaving the scope of the final dataframe just shy of five thousand rows and almost fifty columns.

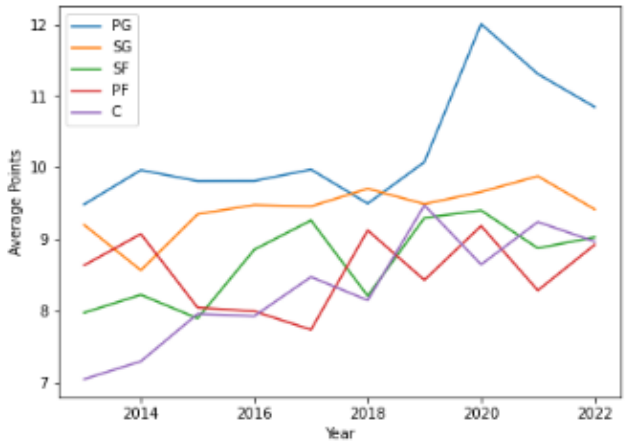

Figure 9: Example of the initial EDA showing average point per position from 2_Initital_Visualisations

The second script of Initial Visualisations is largely the exploratory data analysis (EDA) of this master dataframe before any model development. This largely consists of graphs and charts assessing the positional breakdown of the players as well as developing an overall understanding of the data. Next, Cluster Stage 1 sees the building of functions to cluster and assess this data, as well as the development of the optimal parameters for the model building such as number of components and clusters for this first stage of clustering. This third script also delves into the analysis of this initial clustering and shows the benefit, and frankly the need, to implement a second instance of clustering on these initial groupings.

The fourth script focuses on the first clustering results and explores the breakdown of these new groupings by analysing the type of players in each cluster and how they compare to the conventional positions in basketball that have been proposed as outdated here. The fifth script of Testing Algorithms shows the exploration of trying different methods for the second clustering instance, based on one of the initial cluster outputs which showed a striking resemblance to the All-Star players, or the top performing players in a given year in the NBA. By assessing other methods such as Hierarchical Clustering or by rescaling the data between clustering stages, it was shown that the method of K-Means performed best by comparison to this All-Star grading though confusion matrix scores as well as sensitivity, specificity, and accuracy scores. This was also supported by the academic research on algorithm choices.

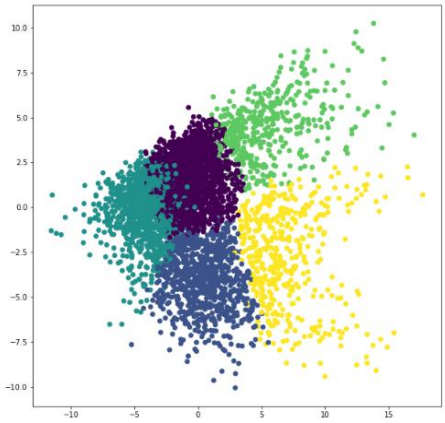

Figure 8: Scatterplot showing first stage of clustering

With the methodology of the first and second clustering established, the sixth script of Double Clustering develops this model fully from start to finish, as could be run in isolation from scripts two to five if the purpose is to just see the final model development. The first five clusters are again developed before each group is taken in isolation and clustered again leaving the final output of ten clusters, which are examined in terms of prominent statistics and players in each before Support Vector Machine and Gaussian Naive Bayes functions are created to produce F1, accuracy, and kappa scores for each as well as a ROC curve. This is included as a means of justification for the second instance acting as a talent and skill separator, as well as an evaluation means given clustering results alone can often be subjective or left open to interpretation. The full reasoning for this inclusion in the methodology can be read in the report itself.

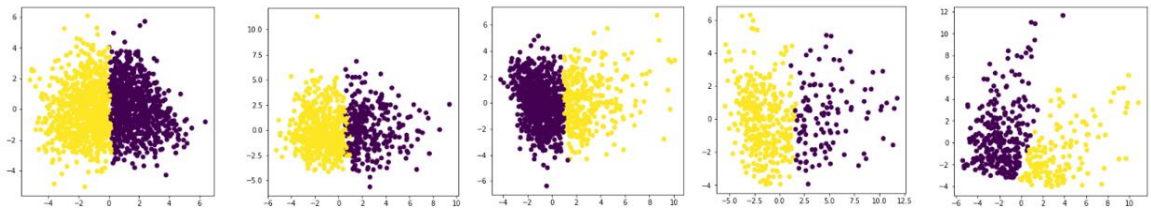

Figure 9: Scatterplots for second instance of clustering

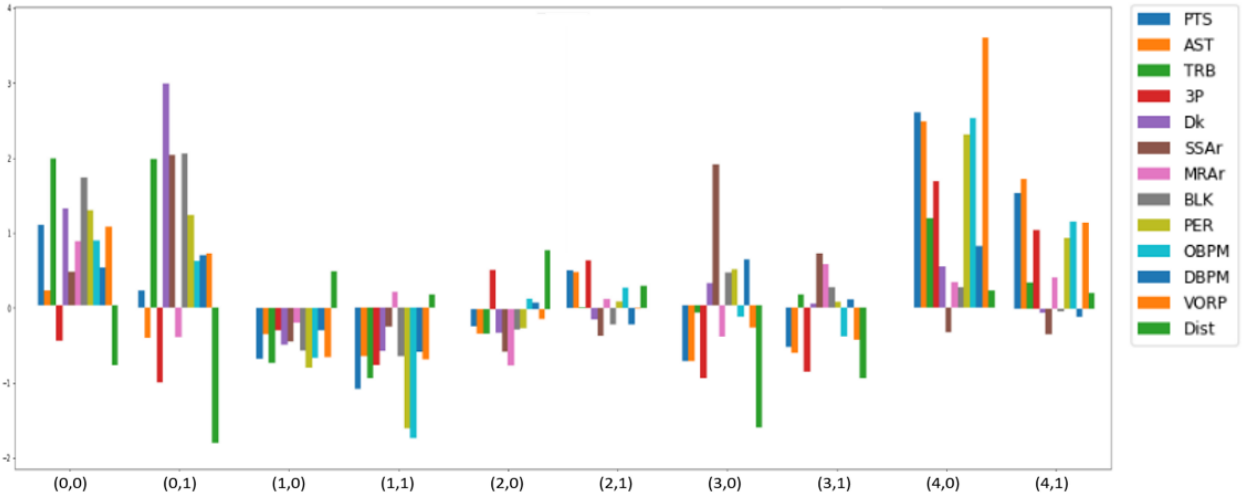

Figure 10: Graphical output after second level of clustering applied

Finally, the seventh script of Increased N-Clusters explores the possibility of a further talent separation in the second clustering instance by using clusters of three rather than two in this second stage, giving fifteen groupings rather than ten. Whilst there is certainly scope for development and further research into this, it was omitted from the final report and is shown here only as a means of future exploration and methodology development potential.

6 Further Contact

The nature of the scripts submitted and the running of them should prove to be a straightforward process with the numbering provided, but should any issues arise in recreating the run process or if there are any lingering questions on the process, please do not hesitate to contact me at x20156138@student.ncirl.ie .