Capstone Project 1 NBA Player Analytics Data Wrangling

Pre-Data Wrangling Steps

For this project a number pages from basketball-reference.com were scraped to attain the URL's of NBA centers who started their career during or after the 1980 season (the 3-point shot was introduced that year, so this year was chosen to allow for fair comparison). In total there were 647 such players, and each player's data was scraped from their respective URL. Given the goal of the project and scikit-learn's required format for a pandas dataframe, all player data was appended to one dataframe and exported to a .csv file. During the export, each NaN value was replaced by -1.

Data Wrangling Steps and Justification

The goal of this project is to compare and evaluate the strengths/weaknesses of NBA centers originating from the US and EU. This is mostly an evaluation of player performance, however since player performance is determined, to some degree, by factors such as upbringing, basketball culture and so on, this is in part an analysis of the quality of basketball development in the US and the EU- though not directly. We therefore perform our data wrangling steps with this in consideration.

Identifying NaNs and Changing Variable Types

Firstly, we convert all those -1 values to NaNs. We then drop an irrelevant *'Unnamed: 0'* column and provide the correct variable types for each column (columns converted to int and categorical types). This has the benefit of saving space and speeding up certain operations. One of the most important columns in the dataframe is the *'US or EU'* column that we created in the scraping file so we need to make sure that all US or EU labels are correctly labelled, and deal with instances where we don't have a label (denoted by 'Neither', or NaN if for some reason the US or EU label could not be assigned).

Ensuring Data Quality

The dataframe has 5122 rows and 547 of them have the 'Neither' label (a row is defined as an NBA season played by a player, with the columns denoting per game statistics such as points per page). The 'Neither' label was generated by 32 unique countries, however 2 of them are mislabelled (Macedonia and Bosnia and Herzegovina are in Europe), so we correct this by providing 'EU' labels for the rows where the 'US or EU' column is 'Neither' and the country of origin is one of the 2 countries named above. Also, in that list is Canada and the US Virgin Islands. In the case of Virgin Islands, all those rows correspond to Tim Duncan, who was American, so it makes sense to give those rows a label of 'US' for the 'US or EU' column. There are 78 rows with Canada as the 'Country', which makes up a sizeable chunk of those 547 'Neither' rows. Given this, the proximity of Canada to the US, and the existence of the Toronto Raptors (the only Canadian and non-US team in the NBA) it makes sense to label Canadian born players as 'US'.

These are the only changes that we can make with reasonable justification, other rows are correctly labelled as 'Neither'. It does not make sense to include them in the analysis as 'the other group' given the small proportion of data compared to the other 2 categories, and the fact that players from roughly 28 countries make up that data, thus we couldn't arrive at any useful insight. Furthermore, after correcting the labels, the 'Neither' label accounts 461/5122 rows, so we aren't dropping a substantial amount of data. What we lose in terms of rows, we gain in clarity of data and lack of assumptions made (if we tried to provide 'US' or 'EU' labels for those rows for example).

Dealing with NaNs

There are various reasons for NaNs. The most noticeable examples are where almost entire rows are NaNs and they correspond to instances when a player did not play during that season, hence it makes sense that there is no data there. The reasons are for this varied, ranging from playing overseas, to having an injury. There are 229 such rows. Given the small proportion, it would make most sense to get rid of the data rather than try to infer what it would have been if they had played that year. Trying to infer something like this adds further assumptions and would negatively affect our data quality, especially so since the task of this project, in part, is to predict those values. We therefore drop those rows. There were also certain rows where almost all values were 0.0. Since they carry little to no information we drop them also.

In the case of 3P% (3P made/3P attempted) for no attempted 3 pointers, the 3P% value is NaN (0/0) mathematically this is referred to as indefinite form. It does not make sense to remove entire rows because of this value, and it doesn't make sense to make those entries 0% (as that would either mean that you shot many and missed all or shot 1 random one and missed). Those 2 instances are very different, but in the case of a model the circumstance doesn't matter. Hence, I removed the column from the dataframe, and if we need to use it we can calculate it from the 3P columns, and if not, we have dealt with 'getting rid' of a lot of NaNs. The same logic was applied to the FT% column.

Working with NaNs for the 'Draft Placing' column is a difficult issue. Firstly, 471 rows have NaN values, which is quite a large proportion. Secondly, this is not a data error, some players in the NBA were un-drafted and this is quite normal. Thirdly, these rows are valid data rows. The simple case would be to drop the column, however this could be a very important feature in the predictive model so this is not an option. The other option would be to infer the values somehow, but as before, this would add further assumptions to our data and it wouldn't really make sense to reverse engineer draft placing from other statistical measures. This issue will be touched upon during the stages of EDA and feature selection.

The case of the 'GS' column (Games Started) is very similar. Only 41 rows have NaNs in the 'GS column' and after closer inspection we see that all those rows correspond to seasons 1979-1980 or 1980-1981, so perhaps there just simply wasn't 'GS' data for those players. As before, we could not infer values for this column, so we will keep it in unless we must get rid of it, since removing 41 rows of good data because of a null value in a potentially insignificant column doesn't make sense.

Conclusion

In the end we are left with a dataframe with 4396 rows of clean data (disregarding the beforementioned cases of the 'GS' and 'Draft Placing' columns). We export this dataframe as 'Player Data Clean.csv'.