Capstone #1 Project - Data Wrangling Report

Pawandeep Jandir

Introduction: This analysis aims to use NFL Scouting Combine (Combine) data of drills and tasks to predict which round a player might get drafted in the NFL Draft. Three sources are used. The first is from the NFL Savant website and has data from 1999 to 2015. This is not the only website with this information but it also has provides measurements for player hands and arms. In order to check for player and drill/task completeness, Sports Reference is used. The third source, DraftHistory, is needed for NFL Draft result completeness. I know that not all players who participate in the Combine get drafted. Conversely, not all players who get drafted participate in the Combine. Further, not all players who are at the Combine will participate in all drills and tasks. So it is expected that there are null values throughout the Combine datasets. This must be kept in mind throughout the entire process.

Acquisition: After identification of data sources, the next step is to obtain it somehow. The list below gives the details for each data source.

1. NFL Savant : This is by far the easiest of the trio. The site offers a csv file of the Combine data. This can then be read into a pandas dataframe very easily.

2. Sports Reference : The data stored by Sports Reference is in html tables. Thus, the site has to be scraped using requests and BeautifulSoup. After getting the full contents of the html table using those packages, I construct a function which can create a dataframe from it. After this dataframe is created, a single column consisting of draft team, round, pick, and year is split up into their own separate columns with str.split(). Each year is on a separate page, so we loop through all the webpages one at a time while appending each created data frame. At the end, the dataframes are concatenated to a single dataframe. To avoid making repeated calls to the Sports Reference website, this dataframe is saved (via pickle) locally. This should allow for better reproducibility as well.

3. DraftHistory : Similar to Sports Reference, Draft History also has their data available on their website in html tables. A very similar approach was used since much of the framework to scrape was already in place. The approach is identical up to the html table to pandas dataframe conversion. After that, there are differences in how the resultant dataframe should be formatted. The draft round column is littered with "\xa0" strings so that needs to be removed. Also the draft round column needs to be forward filled since the website html tables only displays the draft round number when there was a new round. Looping over the relevant years, this dataframe is also saved to a local pickle file for later analysis.

Even though correctly parsing the html tables took time, it provided very useful since I was able to reuse code.

Cleaning and Munging: I spent by far the most time cleaning and ensuring data quality. I worked on the three datasets in order. Immediately there was an issue when I read in the NFL Savant csv file. Two rows had elements which had commas in the field. Obviously this is a problem for csv files. However, the issue was confined only to two rows, so instead of trying to find a general method to deal with such edge cases, I went and manually changed the csv file to remove the offending commas. This way, I could load up the data and continue.

I spent considerable time looking at what data was missing and how best to fill it (if at all). One way already mentioned was using additional datasets. I loaded up the Sports Reference dataframe and ensured the columns are the correct type. However, it is obvious this dataset is also not complete. I use pd.merge()

to full outer join the two dataframes and include an indicator variable for manual inspection of the joined dataframe. I wrote a small function which compared and combined the values of two columns (one from each dataset) to see the percentage of null values. This way there is a quantitative way to determine how combing the two datasets (for a particular column) can recover missing data. On average, there is approximately a 10% recovery. This works about how I expect. Unnecessary columns are dropped.

At this point it is worth cleaning up the player college values. Because I joined two different data sources not every college is listed with the same name. For instance, the University of Southern California can be referred to as Southern California or just USC. This needs be consistent across whatever datasources I use. After this step, I turn my focus on the draft round and pick data which seems to be missing more values than expected. A closer look reveals the 2014 year is totally missing these values. Even worse, both data sources mislabel draft round completely. In both they refer to the pick within a round instead of the round itself, while the pick refers to the overall pick and not the pick within a round. This necessities inclusion of the third datasource, DraftHistory, to get the correct draft round and pick data.

This time I use a left join to merge the two dataframes. This is because not all drafted players participate in the Combine. Any players missing in the DraftHistory dataset mean they were not drafted, and thus have legitimate null values in their draft pick data. This third dataset is also used to fill the draft team column which seemed to be missing some values. After combining columns as before, a more complete dataframe is now forming.

The last step is to fill in the null data throughout the dataset. There are various ways to do this and I explore the two basic ways: mean and median imputation per column. A quick look at the outliers and statistics in each column reveal no particularly significant differences between the two. However, there are more outliers than expected, so to be safe, I impute the null values with the relevant column median and not the mean. The outliers, perceived or otherwise, will not be changed at this time since these are real measurements. The last missing values to change are the draft round and pick data. For now, those values are set to -1, though this may change later.

Among the many columns in the datasets I have combined, I only added one. I grouped similar player positions together into a positiongroups column. This is because positions can be fluid going into the NFL so it is very useful to view college players at this coarser level. Additional columns may also be added relating to a player's college stats and a player's college itself in the future.

Conclusion: Some changes may have to be made later on in the analysis process, but for now this is the final dataframe. This fully cleaned dataframe is also saved locally, again as a pickle file. The accompanying wrangling jupyter notebook can be viewed for the full technical details outlined in this document.