**New York Yankees Statistical Analysis Questionnaire**

**Nolan McCafferty**

1. Suppose you are asked to build a model that predicts salaries for arbitration-eligible players going through the arbitration process. You are given a list of possibly-relevant variables. Walk through your steps for this research. How would you test your model?

My first step to answer this question would be to do some initial research in the field of salary arbitration. After reading a few scholarly articles or blog posts I should have a good idea of a starting point for my research. I could then formulize an initial hypothesis in my head about what good salary predictors could be (age, WAR, etc.). At this point I would focus on the inputs and outputs of my model. I would go through the given dataset and examine each of the variables. The cleaning of the data could include removing null values or researching and finding the correct values for those variables. The next step would be to determine the desired output of our model. In this case, the output would be a predicted salary for the given player.

In terms of the specific model for this research question, one possibility could be utilizing a neural network. However, without focusing too much on the model the input data will be split into training and testing subsets in order to validate the model predictability. Each variable in the dataset (age, position…) would be an input node in the network and after a few hidden layers would produce an output. The model would be trained on the training data, adjusting the neurons every iteration after comparing the model result to the actual result. Once the model is trained it will be tested on the test data and the accuracy can be observed. The model can then be adjusted and hyperparameters tuned depending on the predictive power of the model. Once/if the predictions of the model are satisfactory, it can be used on current arbitration-eligible player data that does not have salary results, in order to predict salary outcome.

1. What questions in baseball do you want to investigate?

Here are some questions that I would be interested in researching:

* How realistic would it be to implement an opener for an entire season, or just a playoff series? The basic idea behind the opener makes sense: teams score the most runs in the first inning because it is the only time that the top of the order is guaranteed to come to bat. Therefore, using your high-leverage reliever in the first and then bringing in the “starter” to face the bottom of the order in the second should decrease runs allowed in theory. Even more promising is the notion that you can play match-ups in the first inning. When playing teams like the Astros and Angels whose first several hitters are all righties, a righty-dominating opener could be used in the first. Although the opener is clearly suited to teams with poor starting rotations (Orioles, Rays minus Snell) and/or strong bullpens (**Yankees**, A’s, Brewers), the idea of maximizing the utility of bullpens is one worth exploring.
* More specifically, could using an opener in the National League until the pitcher’s spot in the order comes up, pinch hitting for him, and replacing him with the traditional “starter” add offensive value to a lineup by limiting pitcher at bats? This seems like an even more promising strategy when on the road because you would only have to throw the opener for an inning (if you can get through 7/8 guys in 2 innings at the plate). To complete this project, you would need to figure out the average number of innings that a pitcher would have to throw before his spot in the order comes up. Also, you would need to calculate the exact amount of offensive value added by removing a pitcher at bat early in the game. This could lead to valuable research in limiting pitcher ABs in general.
* Is the shift bringing an end to the age of the coveted pull-heavy left-handed batter? It seems like people have been saying “if only I was a lefty I would’ve gotten drafted” or “sorry, we’re looking for middle-guys that swing lefty” forever, but is advanced scouting and the shift finally eliminating the advantage that lefties have always enjoyed? I grew up a Mariners fan in Seattle and watched us give Kyle Seager a huge contract, only to watch him decline into a mediocre at best third basemen this last year. Given his hitting style, I can’t help but hypothesize that the recent league-wide adoption of serious shifts has contributed to that decline. It would also be very interesting to create a metric that measures how much a player is affected by the shift.

1. Describe a project you worked on (baseball or non-baseball, academic or personal, solo or in a group) that demonstrates the skills you would bring to the Yankees. Feel free to attach a pre-existing write-up or work sample if applicable.

This past summer I worked as a research assistant for an economics professor here at Pomona College named Manisha Goel. The goal of the project was to analyze hundreds of millions of job postings to create interpretable numeric vector representations for them that could then be used for various economics research projects. Manisha is an economics expert who is interested in machine learning, so she hired me to do all the coding and computational work. Before we acquired the massive dataset from Burning Glass Technologies, one of the world’s largest online job markets, I initially scraped raw text job postings from Indeed.com to begin building the models in preparation for the big data. I built the following machine learning models: word2vec, doc2vec, and LDA. LDA is a topic modeling algorithm that we used to create more interpretable vectors for analysis. Once the Burning Glass dataset came in, I began adapting the models I had made for the Indeed postings to the much larger corpus. The hundreds of millions of documents proved to be too much for the topic modeling algorithm we had been using, so the current focus of the project is adapting the LDA code to run on GPUs in order to speed up the computation.

All of the coding for this project was done in Python. The initial scraping utilized the Beatiful Soup library and various other libraries including nltk, gensim, and scikit-learn were used to create the machine learning models. The models are presented in several Jupyter notebooks available on my GitHub. The most recent evolution of the project has been to adapt the models to run on GPUs using the CUDA platform. This project gave me a good introduction to machine learning and independent research that I cannot wait to apply to baseball, an area that I am much more interested in.

1. Assume these fictional players are of equal stature: everyday players, playing the same position and possessing like handedness, having identical defensive skills, with similar age and experience in the big leagues, as well as comparable contract statuses. Which player (A, B, or C) would you argue had the best season and why? Do you think this player will provide the most offensive value moving forward or would you prefer one of the other players and why?

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **G** | **PA** | **AB** | **R** | **H** | **2B** | **3B** | **HR** | **RBI** | **BB** | **SO** | **SB** | **CS** | **AVG** | **OBP** | **SLG** | **OPS** | **wOBA** |
| **A** | 151 | 640 | 581 | 99 | 159 | 29 | 0 | 42 | 127 | 59 | 139 | 2 | 0 | .274 | .341 | .540 | .881 | .374 |
| **B** | 143 | 602 | 485 | 108 | 140 | 37 | 3 | 21 | 87 | 117 | 151 | 17 | 5 | .289 | .427 | .507 | .934 | .405 |
| **C** | 159 | 677 | 597 | 87 | 160 | 31 | 0 | 30 | 102 | 80 | 76 | 28 | 2 | .268 | .355 | .470 | .825 | .357 |

In choosing which player had the best season, I calculated a few more columns on the end of the table (without including HBP, IBB, and SF which should not have added much of any difference). Although it is not immediately obvious from the original table, after the addition of OPS and wOBA (calculated using the 2018 MLB constants) that player B had the best season, followed by player A then player C. Player B has a much higher OPS and wOBA than both the other players, mostly due to his ability to get on base. His walk rate is **much** higher than the other two players. He was also a decent base stealer (nowhere near player C) but more valuable than player A in that regard. The only real negative to player B is his relatively high strikeout rate, but that definitely did not stop him from getting on base. Thus, I think that player B will continue to provide the most offensive value moving forward and would prefer him over the other two players.

1. The accompanying CSV file, matchupdata.csv, contains the results of 10,000 total at-bats.  There are three columns:

- "batterID": identity of the batter (1-100)

-"pitcherID" identity of the pitcher (101-200)

-"result": value of the result of the at-bat on a continuous scale

Next season, each batter in the dataset (1-100) will get exactly 300 at-bats. Each individual at-bat will be against a randomly selected pitcher from all the pitchers in the dataset (101-200), not just those that the given batter faced in the dataset. Project each batter’s average result next season. Along with your projections, please provide a written explanation of your methodology and attach all code that you used to arrive at your answer.

My projections can be found in the file projections.csv and the code can be found in matchup\_code.R. I used the dplyr package in R. To complete the projections, I initially took the average result for each batter in the initial 10,000 at bats. However, I realized that simply the average would not necessarily be representative of the true skill of the batters if, for example, a batter happened to face the best pitcher in the group 100 times out of 150 at bats. Thus, I decided to incorporate the skill of the pitcher’s faced into my projections as well. To better evaluate the pitchers I calculated the z-score of the averages for each batter and subtracted each result by that score, so that the if the batter was “good” the result would go down. I then used these “scaled” results to calculate the averages for each pitcher. Finally, I used these “scaled” pitcher averages to similarly compute the “scaled” averages for the hitters. Therefore, my projections should be slightly more robust to who the batters initially faced.

1. The accompanying CSV files ‘pitchclassificationtrain.csv’ and ‘pitchclassificationtest.csv’ contain information from over 20,000 pitches from six different pitchers over three years. There are 12 columns:

- "pitchid": a unique identifier for each pitch.

- "pitcherid": identity of the pitcher (1-6). The identities are the same in both datasets. Pitcher 3 in the training set is the same pitcher as Pitcher 3 in the test set.

- "yearid": year in which the pitch occurred (1-3).

- "height": height in inches of the pitcher.

- "initspeed": initial speed of the pitch as it leaves the pitcher's hand, reported in MPH

- "breakx": horizontal distance in inches between where a pitch crossed the plate and where a hypothetical spinless pitch would have, where negative is inside to a right-handed hitter.

- "breakz": vertical distance in inches between where a pitch crossed the plate and where a hypothetical spinless pitch would have, where negative is closer to the ground.

- "initposx": horizontal position of the release point of the pitch. The position is measured in feet from the center of the rubber when the pitch is released, where negative is towards the third-base side of the rubber.

- "initposz": vertical position of the release point of the pitch. The position is measured in feet above the ground.

- "extension": distance in feet in front of the pitching rubber from which the pitcher releases the ball.

- "spinrate": how fast the ball is spinning as it leaves the pitcher's hand, reported in RPM

-"type": type of pitch that was thrown (will only appear in the training dataset).

Your goal is to give the most likely pitch type for all of the pitches in the test dataset (year 3) using information from the training dataset (years 1-2). Note that the pitchers in the datasets do not correspond with any specific real pitchers but are meant to be representative of real data. Please include the following with your final submission:

1. CSV with two columns: the pitchid and the corresponding predicted pitch type
2. write-up of your method and results, including any tables or figures that help communicate your findings
3. all code used to solve the problem

My results can be found in the Jupyter notebook pitchclassification.html and/or pitchclassification.ipynb, either can be opened in a browser and the predictions are in pitch\_predictions.csv.