Scouting Reports, Classified:
Harnessing the Solid-Average Raw Power of Natural Language Processing

Jonathan Lewyckyj

Abstract

Using tf-idf vectorization, logistic regression, and SVM, this project uses text classification models to predict a FV grade for a minor league baseball player given a scouting report about him.

1 Introduction

Every Major League Baseball team employs a department of scouts that travel around the country evaluating players, often minor league, college, and high school players, by mostly qualitative methods. The scouts watch how the players bat, pitch, play the field, and interact with teammates and opponents. Since these are often young players who haven't reached the highest level, the scouts' goal is to assess how a player will progress as he gets older and more experienced and reaches the highest level. After seeing a player of interest, the scout will prepare a report on the player, including an overall grade (20 to 80 scale, in increments of 5 generally, with 80 being the best), grades for specific skills (same scale, for skills like speed, fielding ability, power), and a written report, often 1-2 paragraphs. As more quantitative, empirical data has become more widespread in the game, the usefulness of the reports have dwindled; when assessing thousands of players, it's much easier to use numerical data than to read thousands of reports. Rather than ignore the written report, my project will endeavor to use natural language processing to find common word- and phrase-level tokens in the written reports, and the associated probabilities of those tokens appearing in a high-grade report and a low-grade report. At a minimum, this project could help scouts be more consistent in their language (i.e., if you think a player is only a 40-grade prospect, don't use tokens that usually appear in 70-grade reports). Ideally, this project could uncover insights about players that might otherwise be looked over. Essentially, this project is a text classification problem, with ordinal labels, designed to rate baseball players.

2 Related Work

There are not many related works to my particular problem in a sports scouting context, and furthermore, the works that I could find could not be described as academic papers. Thus, I will briefly discuss some articles I could find using text analysis on scouting reports, across a number of sports.

One article$_1$ used sentiment analysis on tweets by known NHL scouts and media members about NHL draft prospects. This article is different from my project in a number of ways: 1) There are many authors, rather than 1 author, so the language they use could differ in important ways, 2) The data being analyzed are tweets, rather than longer-form reports, 3) It is not a traditional text classification problem with labels; rather, this article uses an existing dictionary of positive and negative words, and uses a simple algorithm (Breen's approach) that adds +1 for every positive word and -1 for every negative word, 4) There is an uneven number of observations about each player, rather than 1 report per player). Thus, it is hard to draw any insight from this article on how it might apply to my own use case.

Another article$_2$ used scouting reports, in conjunction with other information, namely numerical scouting grades, NFL combine stats,

and players' measurable (height, weight, etc.), to predict a player's earnings. The textual reports were broken down into overview, strengths, and weaknesses, which helped the author identify which n-grams are associated with positive or negative values. The author pulls n-grams from these three sections containing textual reports. Full results are not shown, just two examples, though it seems like the author is using n-gram features to fill in gaps that the relationship between numerical scouting grades and salary is not capturing.

Another article[3] looked into MLB scouting reports, a historical database of over 73,000 reports from 1991-2003 from the Cincinnati Reds organization. This article mostly focuses on the numerical scouting grades (which are being used as labels in my project): How the grades correlate to actual statistical performance, inherent conservatism in assigning grades, and how grades differ from one evaluator to another. For text analysis, the authors do pull out some words or phrases, and take the average value of players that had those n-grams in their report, but there is not much else involving text. This article's information is useful context for me, and helped serve as the inspiration for my project, but offers little in terms of natural language processing.

3 Dataset

My dataset comes from the Fangraphs (a popular baseball analytics website) Prospect Board, which includes information on mid-to-high-level minor league baseball players. The most relevant fields for this project are 'FV', which is an all-encompassing Future Value grade, and will serve as the labels for this project, and 'Report', which is the written report on each player. FV is on a 20-to-80 scale, with 50 being average, and goes by increments of 5. However, there are some important details to note. There are also grades such as 40+, which can generally be seen as between 40 and 45. Also, while 50 is considered average, that means average for a Major League player; most minor leaguers will fall short of even reaching the Majors, so a 50 FV grade is very strong for a prospect. Thus, we are left with a highly-skewed distribution.

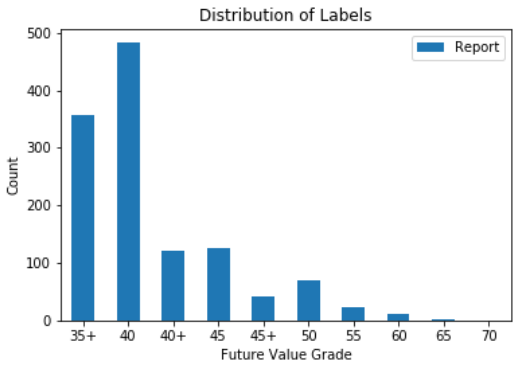

Figure 1: Original distribution of labels

There are too many categories for a proper classification algorithm to work, so I will reduce to 3 categories. However, it would not fit the use case to balance the classes through this recoding. As mentioned, a 50 FV prospect would project to be an average MLB regular, so one group (3) will be 50 or higher grades. Another group (2) will be 40+, 45 and 45+ FV prospects, or players that project as below-average starters or good role players. The final group (1) will be 35+ and 40 FV prospects, including players projected to not make the Majors or be only marginal role players. There is still imbalance in these classes, though the distribution should be able to be worked with.

Finally, pitchers and position players are evaluated and written about very differently, and almost certainly need to be separated. I will run separate models to project pitchers and position players.

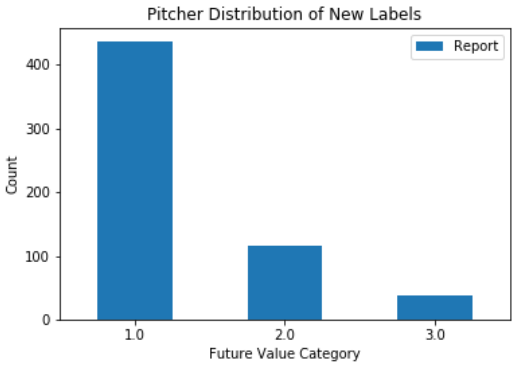

Figure 2: Recoded distribution of labels for pitchers

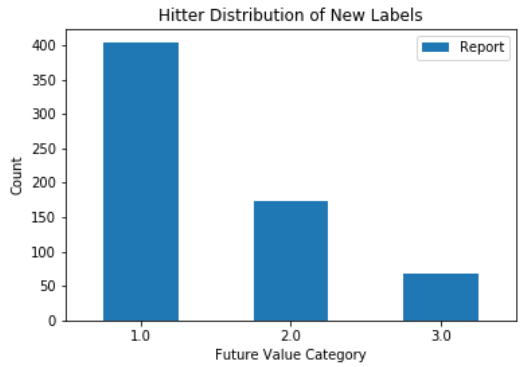

Figure 3: Recoded distribution of labels for hitters

There are 589 pitcher reports, and 647 hitter reports, broken down as shown above into the new recoded categories. The average length of a pitcher report is 112 words, and its 127 words for a hitter report.

4 Method

The two main methodologies I used were logistic regression and SVM, using tokenized text and tf-idf vectorization to create features of 1grams, 2grams, and 3grams. I also tried using fasttext, but even with trying upsampling and downsampling, the imbalanced classes led to unproductive results.

I began with text pre-processing. Converting to lower case and removing punctuation were easy decisions. I also removed numbers, though possibly with the loss of information. A phrase such as ‘fastball that sits 94-97 and touches 98’ would be valuable information for a scouting report for a pitcher, but there are too many variations of how a phrase like this could be used to provide a meaningful ngram.

Next, I created dataframes of features using various tf-idf vectorizers. I used 3 different vectorizers for both hitters and pitchers (1gram, 1+2gram, 1+2+3gram). For computational ease, I set a limit of 1000 features.

For logistic regression, I attempted 6 different models for both hitters and pitchers. The 6 different models differed in using 1gram vs. 1+2gram vs. 1+2+3gram, and also by lbfgs vs. saga solver (6 different combinations of those 2 parameters). All models used the l2 penalty, balanced class weights, and multi_class = ‘multinomial’.

For linear SVC, I attempted 3 different models for both hitters and pitchers, with just the ngrams differing. For each model, I tried many different combinations of learning tolerance and the regularization parameter to find the best macro-averaged F1-score. All models used the l2 penalty, hinge loss, balanced class weights, and multi_class = ‘ovr’.

All models for both logistic regression and linear SVC used a 75% training / 25% test split, with the same split across all models.

5 Results

In general, the models were better at classifying Category 1 prospects (35+ or 40 FV). This is to be expected, given the imbalance in classes, which the tuning parameter could only do so much to correct. As a result, I did not focus on correct classification rate as an evaluation metric, but rather F1-Score for Category 2 and 3 prospects.

The best performing model for pitchers was a logistic regression using 1+2grams and lbfgs solver. It had a 0.88 precision, 0.90 recall, and 0.89 F1-Score for Cat1 players, a 0.48 precision, a 0.52 recall, and 0.50 F1-Score for Cat2 players, and a 0.80 precision, 0.40 recall, and 0.53 F1-Score for Cat3 players, resulting in a 0.64 macro-averaged F1-Score. Not accounted for in the metrics is that it classified 3/10 of Cat3 players as Cat1, and 3/10 of Cat3 players as Cat2. These misses are considered the same in recall, but since the labels are ordinal, a misclassification of Cat3 as Cat2 is less wrong than of Cat3 as Cat1. Another model, 1+2grams and saga solver, was slightly better at classifying Cat2 players, and identical at classifying Cat3 players, but a bit worse at classifying Cat1 players.

```
Accuracy: 0.811
[[104  10   1]
 [ 11  12   0]
 [  3   3   4]]
```

	precision	recall	f1-score	support
1.0	0.88	0.90	0.89	115
2.0	0.48	0.52	0.50	23
3.0	0.80	0.40	0.53	10
accuracy			0.81	148
macro avg	0.72	0.61	0.64	148
weighted avg	0.81	0.81	0.81	148

Table 1: Output summary of best pitcher model

The best performing model for hitters was an SVM model using 1+2+3grams. It had a 0.78 precision, 0.77 recall, and 0.77 F1-Score for Cat1 players, a 0.51 precision, 0.54 recall, and 0.53 F1-Score for Cat2 players, and a 0.27 precision, 0.24 recall, and a 0.25 F1-Score for Cat3 players, resulting in a 0.52 macro-averaged F1-Score. As mentioned for pitchers, the recall for Cat3 players is superficially low, since most of the misclassifications were of Cat3 as Cat2, rather than Cat3 as Cat1.

```
Accuracy: 0.648
[[76 16  7]
 [17 25  4]
 [ 5  8  4]]
              precision    recall  f1-score   support

         1.0       0.78      0.77      0.77        99
         2.0       0.51      0.54      0.53        46
         3.0       0.27      0.24      0.25        17

    accuracy                           0.65       162
   macro avg       0.52      0.52      0.52       162
weighted avg       0.65      0.65      0.65       162
```

Table 2: Output summary of best hitter model

6 Discussion

In some ways, this project is limited in scope, and should only be seen as a prototype for using natural language processing in sports. I'll go over some ways I think a larger or different scope could provide better results, and some pitfalls that may still remain.

The first issue is the small size of the dataset (589 pitchers, 647 hitters). This contributes to the imbalance of classes, not that the issue would go away with more sample. A remedy for this could be to use scouting reports over a long period of time, versus just analyzing top pro prospects at current time, or incorporating reports from multiple sources, though that could raise the issue of multiple authors writing about prospects in different ways.

Another issue is that I limited the tf-idf vectorizers to 1000 features for computational purposes, and didn't go beyond 3grams. Using more features, using larger phrases, or even moving to a deep learning architecture could improve predictive performance.

I was not surprised that the pitcher models were more accurate than the hitter models, with the main reason being that pitchers serve in one of two roles (starting pitcher or relief pitcher, with relief pitcher inherently being less valuable), while position players can fall under 9 different positions (including DH). As a result, pitchers are written about more similarly: scouts write about their fastball, each of their offspeed pitches, and their mechanics, and how those mechanics might affect command and durability. There still can be some variability. A scout might see a 70-grade fastball, 60-grade slider, and poor command that might lead to a relief role and a 40+ FV, whereas plus command might lead to a starter role and a 60 FV. A model might overemphasize the similarities in fastball and slider grades, and misclassify as a result.

However, for position players, there is even greater variability. A plus-hit, plus-power (and the ngrams associated with those tools) first baseman might be given a lower FV, since he has little else going for him, while a plus-hit, plus-power shortstop would be an elite prospect. The bar is just so different for different positions, and doing separate models for different positions would be unfeasible given that the dataset is already quite small. Defense is also written about so differently for different positions that good qualities for a shortstop and good qualities for an outfielder almost have no overlap, confusing the model further.

The best solution to this is how scouting reports are formatted. Instead of a free-form paragraph, where a scout might write about a position player's hit tool, power, glove, arm, and speed in the same paragraph, along with makeup, injury history, and other information, a better setup would involve separate fields where the scout could write about each tool individually. Then, an NLP model could be used to predict each tool given the specific information about the tool. For equivalent language in a Yelp review, the reviewer would rate and write about the food, service, and atmosphere of a restaurant separately instead of altogether. This would likely lead to more accurate predictions.

Finally, the ultimate goal of using NLP in sports is not to predict the scouting grade, which can often have a lot of variability in assessing future production, but to predict the future production

itself. That goes beyond the scope of this project, and would involve a lot more data over a long period of time, though hopefully this project can serve as a kickstart to that process.

7 References

1. https://model284.com/2019-nhl-draft-prospect-sentiment/
2. https://deepfootball.com/scouting-reports/
3. https://www.theringer.com/mlb/2019/3/4/18249155/cincinnati-reds-scouting-report-series-part-1-data-findings

8 Code

https://github.com/JLewyckyj/NLP_Project/tree/master