Modeling Baseball Player Salaries and Team Wins

The Problem

Every off-season, sports fans watch the movement of players from team to team with great interest. When highly-sought players sign huge, multi-year contracts, it can inspire disbelief, disgust or excitement in fans.

In Major League Baseball, players without a team -- due to contract expiration, early release, or being undrafted -- are free agents. Prominent free agents typically have several years of playing statistics, so their value should be well-understood by all parties.

Every year, many of these free agents sign new contracts, often for record-breaking terms.

Fans and journalists want to know where these players will end up and how much they will be paid. Players' agents and general managers want to understand the market value for a player. Agents want to maximize salary, while GM's seek to maximize team wins and minimize spending.

Stratospheric salaries can appear to be irrational, but baseball and other professional sports are businesses. In theory, players are paid rationally according to their contribution to the revenues of the team. Runs, walks, strikeouts: these are the products of a baseball team, and the more they are produced, the more likely the team is to win. Increasing wins should result in more revenues from television and ticket sales.

If wins are crucial to the bottom line, then which stats are the best predictors of wins? Do the models for player salaries and team wins agree on the most important measures of on-field production?

Getting and Cleaning the Data

Initially, salaries and player data were acquired from several online sources, including USA Today, and MLB.com. In addition, team payrolls were extracted from http://www.stevetheump.com/Payrolls.htm.

Data was scraped using Python, the Scrapy library and PhantomJS, which provides a scriptable, headless web browser.

A collection of Python scripts takes the raw data and converts it into clean tables by removing strange characters, aggregating player-year level row data, converting currencies into numeric values, etc. Features such as career stats and averages were derived. For input to the predictive model, a script was written to combine yearly salaries with prior-year performance

data, including aggregate (career) stats for important performance metrics. Each row is a vector of the following dimensions:

```
Annual Salary
Salary Year
Team
Adjusted Team Payroll
Log Adjusted Salary
Statistic.1.Year-1,..., Statistic.n.Year-1
Career.Statistic.1,.... Career.Statistic.n
Team Position.1, …. Team Position.n
Num All Star Appearances
Num Post Season Appearances
```

Year-1 refers to the prior year.

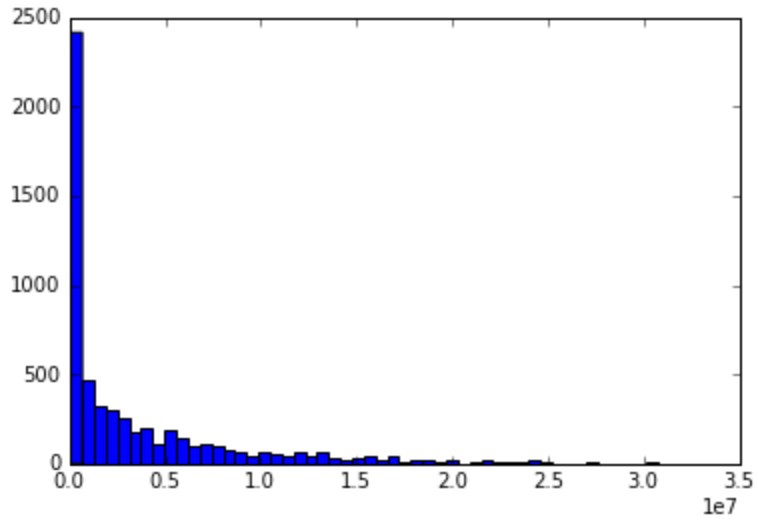
Each player statistic (Statistic1… Statistic-n) is a single dimension and measures fielding, pitching or batting performance. A single year of statistics for the prior season is included with each observation, although the generate_observations.py script can combine multiple years into a single row (Year - 1, Year - 2, etc.).

For many of the player statistics, a career statistic is calculated.

At a later stage in the project, I decided to find a more reliable source of cleaned and normalized data. I used the Lahman online baseball database, available as a collection of CSV files and SQL.

Exploratory Analysis

I looked at the distribution of player salaries for 2011 through 2015:



We see that the distribution is not normal. The data is skewed heavily towards lower salaries, with a long tail and few very high salaries.

I took note of this and later tried regressing on log(Adjusted Annual Salary), described in the section on modeling.

Next, I considered various statistics and their correlations to salaries. These variables were plotted against salaries. Also, corr() was used to see the correlation matrix and guide feature selection for the subsequent modeling step.

Significant correlations were found between salaries and the following variables:

- Batting: RBI (Runs Batted In), TB (Total Bases), HR (Home Runs), PSN (Power Speed Number), G (Games), Number of Seasons, H (Hits), 2B (Second Base), 3B (Third Base)
- Pitching: SO (Strike Outs), W (Wins), L (Losses), IP (Innings Pitched), GS (Games Started), ER (Earned Runs)
- Fielding: Num Seasons, G, PO (Put-outs), E (Errors), A (Assists)

Regression Model

Using scikit-learn, a linear regression model was trained using the “Adjusted Salary” as the response variable. Since baseball salaries grow over time, the “Adjusted Salary” (Annual Salary / Average Annual Salary) allows for an apples-to-apples comparison of salaries in two different years.

Team Payroll data was merged with this data frame. The “Adjusted Team Payroll” is the aggregate team payroll divided by the average team payroll per year.

Since players can play multiple fielding positions, the variable “Position” indicates either the position played in the most recent prior year of play or “MULTIPLE,” if multiple positions were played.

Based on the exploratory analysis described in the previous section, the data was subsetted using relevant variables and 5-fold cross-validation was used to test the results. R^2 was 0.65 +/- 0.08.

Decision tree regression (0.61 +/- 0.12) was tried with no significant improvement in accuracy.

Classification Model

Salaries were binned into quartiles and two different classification models were trained on the data. KNN and SVC (Linear kernel, C=10, tuned using Grid Search) were trained and tested, with SVC performing slightly better (0.65 vs. 0.62). As one might expect, increasing the number of bins worsened the score.

Improving the Regression Model

Several improvements over simple linear regression were tried in order to improve accuracy and reduce the number of features

Regressing on log(y), where y = the Adjusted Annual Salary, performed worse than regressing on y. This is described in further detail in the “LogLinearRegression” iPython notebook.

I experimented with RFE (Recursive Feature Elimination) for variable selection. RFE, by default, chooses half of the variables. It yields comparable accuracy to our original regression model, with R^2 = 0.64 +- .09. At this point, I was unsure about the magnitude or statistical significance of the remaining variables. It seemed that additional variables could be eliminated, since the number selected was completely arbitrary.

Using the statsmodels package, I fit an OLS regression model and inspected the resulting coefficients and their p-values and confidence intervals. Through trial-and-error, I selected a few statistically significant variables that yielded a reasonably accurate model. This simplified linear regression model is as follows:

Adjusted Salary = 0.15 + 3.29 * Batting_Career_TB - 0.32 * Pitching_Career_IP + 6.3 * Pitching_Career_SO + 3.22 * Num_All_Star_Appearances - 0.34 * NO_POSITION + 0.2 * FIRST_BASE + 0.4 * SECOND_BASE

Statsmodel gives us an R^2 of 0.64, and only 7 of the 38 original variables were used.

Regularization

GridSearchCV was used to find the optimal parameters for Ridge Regression, Lasso and ElasticNet. Then the coefficients were examined from the best-fitting estimators.

With Ridge Regression, as expected, all 38 predictor variables were selected, and R^2 was 0.62. LASSO selected 25 variables, and R^2 was 0.62. ElasticNet selected 28 variables, and yielded a score of 0.62.

I was curious how well the regularization approaches compared to manual feature selection, so I selected the variables from the model we constructed using p-values and performed 5-fold cross-validation with linear regression. I also tested ElasticNet regression using 5-fold cross-validation and the parameters we found earlier. The average accuracy for both approaches was 0.62 (+/- 0.19).

For both ElasticNet and for the simplfiied model we constructed from 7 variables, the variance in scores was higher than for regression on all 38 variables.

Observations

Salaries are highly correlated with total bases (TB), which makes sense. This statistic counts bases earned by the batter through hits. It does not count walks or other bases. OBP (On-Base-Percentage), which includes walks and “hit by pitch,” has little effect. OBP was eliminated by LASSO, ElasticNet and RFE.

Interestingly, runs in combination with hits has little effect. While runs are important at the team level (as we shall see in “Modeling Team Wins”), this suggests that teams primarily pay players to get on base by hitting. “R” (Runs) was eliminated by LASSO, ElasticNet and RFE. Given that OBP was also thrown out, it seems that hitting is valued much more than walks.

Innings Pitched and Strikeouts are also highly correlated with salaries. When I performed correlation analysis, I found Strikeouts to be more highly correlated with salaries than other common measures of pitching performance like Earned Runs. However, different variables were selected depending on the regression approach used.

Players with All Star appearances are paid more, proportionally, to their number of appearances in the All-Star game versus players with no All-Star appearances.

Pitchers appear to be paid less, on average, than other positions. “MULTIPLE” was thrown out, as it had insignificant effect. This makes sense, since it is likely to be correlated with other positions.. First and second base players are paid more, on average, which is supported by more than one regression approach.

Other Thoughts

An R^2 of 0.64 leaves a lot of room for improvement. This means a lot of variance in salaries is not explained by the variables we studied. Either more data needs to be used, or there are other factors at work.

Behavioral finance explains irrational behavior in stock markets, and might also explain why salaries are only moderately correlated with performance on the field. Biases and errors in judgement could be responsible for the unexplained variance. It's also likely that players are paid for potential -- rather than past -- performance. This would explain the existence of highly-compensated players with little to no MLB playing experience.

Modeling Team Wins

General managers care about winning. If they can predict wins from player performance, then they should be able to allocate team payrolls more effectively.

I constructed a "Team Wins" model uses linear regression to predict the team winning percentage from several key stats, including runs, strikeouts, earned runs, stolen bases, etc. This data also contains so-called "3-Year Park Adjustment" factors that rate the home stadium for pitchers and batters. Some home fields favor batters and some fields favor pitchers.

The resulting R^2 was 0.90, and the variables and their coefficients are shown below:

Winning Percentage = 0.45 + 0.79 * R - 0.36 * AB - 0.02 * THIRD_BASE_HITS - 0.05 * HR - 0.04 * BB - 0.04 * SB - 0.01 * SF - 0.63 * RA + 0.36 * ER - 0.32 * ERA + 0.07 * CG + 0.03 * SHO + 0.11 * SV + 0.06 * E + 0.18 * FP + 0.01 * BPF - 0.01 * PPF

Recommendations

"Runs" is the variable most strongly correlated to winning. Therefore, teams should maximize runs. There is controversy in baseball about which player attributes contribute the most to runs, though. Is it hitting power? Stolen bases? Ability to get on base (including walks)? In general, the ability to hit appears to be overvalued, while players with high OBP (on-base percentage) are less valued. By using players in the latter category, an effective team could be built while minimizing the total outlay on player salaries.

Some questions for further investigation:

- Are great hitters overpaid relative to players with good OBP? This seems to be the case, but it would be interesting to quantify this.
- Does base-stealing factor into pay and/or wins?
- Are there particular attributes of pitchers or of fielders that are overvalued relative to what is required to win?

For fans, journalists, and GM's, the Salaries model described above can be used to get a "ballpark" figure for what a player might command in the open market. It also explains what drives player compensation and informs recruiting and negotiation efforts.