1. As a member of the Research and Development team, how would you decide whether a Minor League pitcher prospect is ready to be promoted to the Majors?

As a member of the Research and Development team, I would consider sabermetrics on the pitcher such as Adjusted Earned Run Average, Innings per Start, Strikeout-to-Walk ratio, and more to decide if a pitcher was ready for the show. Adjusted Earned Run Average normalizes a player’s earned run average across the entirety of the league. This statistic accounts for external factors like ballpark and opposing team. Once adjusted, a pitcher is compared to the rest of the league with a score of 100 being the league average. Being able to compare and rank a pitcher against his peers on a level playing field is an exciting concept and one with a lot of value. Statistics like Innings per Start and Pitches per Start help quantify the durability and consistency of a pitcher, two essential characteristics of a quality starter. The higher the Innings per Start and Pitches per Start means that the pitcher is durable and makes it deep into games, a valuable asset for a 162-game season. Raw strikeout and walk numbers can be misleading because a power pitcher can record many strikeouts while being wild and ineffective and a finesse pitcher can be successful while recording few strikeouts. The Strikeout-to-Walk ratio is a valuable replacement since it tells a more complete story. Sabermetrics alone are powerful tools but factors including work ethic, character, and injury-history are also pivotal to a player’s major league success. A conversation with the prospects minor-league coaches could provide additional insight about the player.

Source: Advanced Stats | Glossary.” *Major League Baseball*, m.mlb.com/glossary/advanced-stats.

1. What's the worst piece of open source software you've used, and why?

Open source software is a computer software that allows users to modify its source code and change the software to fix bugs, make improvements or adapt the software for the user’s own needs. Open source software or OSS, encourages collaborative efforts which has both advantages and disadvantages. Since becoming a data science major about a year ago, my experiences with open-source software aren’t overly expansive however I have used OSS’s such as R, Python, MySQL, Atom, and Redis in varying degrees. Of the open source software that I have been exposed to, the worst is Redis for a variety of reasons. The first being that Redis requires three Masters each with two slaves to set up the architecture. Each Master has hash-slots assigned to them which shard the data. An issue with this is that when a Master who is holding slots fails, the data which was to be written to those slots will be lost. Additionally, your dataset must fit comfortably in the memory provided. The whole dataset is stored in RAM which can be costly for the client as well. Yet another issue I have with Redis is that it supports no query language or joins which makes key management much more difficult than it has to be. New to the tech community, the fact that Redis uses a command line interface was off-putting to me. There are no user-friendly shell’s or applications to help work with the data as is the case with many other software including Python and R.

While Redis itself has some issues, there are also issues with open source software as a whole. OSS is meant to provide flexibility, freedom and cutting-edge technology to its users through collaborative work however this isn’t always the case. An issue many users have encountered is that their opinion or their feedback doesn’t actually matter. Reported bugs or suggested solutions are all too often ignored. The attitude seems to be if you can’t create the fix yourself the issue won’t be fixed. Even in situations where a functional patch was provided, users are often belittled and their work is disparaged leaving many users to stop contributing altogether.

Although OSS has issues, I find them more useful than harmful. Programming languages like Python and R which are often used for computer and data science applications and are both open source. These software are popular because they are free, generally simple to install, and have a plethora of documentation and support. In fact, R is my preferred software because of the sheer number of packages that are readily available. Packages are updated and added frequently giving the user access to many powerful tools. This is only possible because of the software’s large and active community. Open source software isn’t perfect. I could nitpick about every nuance I find slightly inconvenient but the fact of the matter is, with active community members all working collaboratively towards a common goal, amazing things can be accomplished for the benefit of the people.

Source: “Why the Open Source Software Model Is Fundamentally Broken.” *BetaNews*, 20 Apr. 2015, betanews.com/2015/04/20/why-the-open-source-software-model-is-fundamentally-broken/.

<https://github.com/Nick-Cianci/Phillies-Internship-Task>

Citation for webscraping:

<https://rpubs.com/Radcliffe/superbowl>