MSiA 423 Mid-Project Review

Alex Burzinski

MLB Predictor App

Highlights

- Learning more about APIs
- Learning more about AWS
- Seeing some interesting results from my initial models

Learning More About APIs

- For my project, I looked into several different sports statistics related APIs
- Some did not have the data I needed
- Some had the data, but required payment
- The one I settled on is both free, and has the data I need
  - However I did run into a bit of an issue as the API can only be called for one player and one season at a time
  - This proved difficult as I needed many years of historical data for many players
  - I was able to get the data I needed using some loops and other things, but it does take over an hour to generate my historical files
- If I were doing this project for a company, I would consider using one of the faster APIs that has a price tag, as getting the data more quickly might outweigh the cost

Learning More About AWS

- I really enjoyed learning more about how to use AWS and about how all the pieces fit together in the context of our project
- To me, it is kind of amazing that a developer can create an RDS instance and start reading and writing data to it within minutes, without changing any code (using SQL Alchemy)
- It is also very cool that a developer can spin up a new EC2 instance, install required software packages, and start running an app on that instance within minutes
- In the past I have really only written one-off python scripts, so I really enjoy getting to develop an app from end to end, seeing the lifecycle of the app, and hosting it in AWS

Seeing Initial Model Results

- My project is trying to predict which Major League Baseball players will win annual awards this season
- I really enjoyed playing with my initial models, specifically looking at historical data and predictions
  - It was interesting to see players in past season that won awards when my model predicted they would not win, and to see players that did not win that my model predicted would win
- I will continue to try other types of models and will continue to tune these models. It will be fun to see how close my final model is at predicting which players will win which awards when this seasons awards are given out in November

Sprint Progress

- While I did not finalize the model my app will be using during this sprint, I did make very good progress in other parts of the app
  - I created an initial model that I will continue to improve upon
  - I downloaded and cleaned the historical data used to train my model
  - I developed a process to update current MLB stats on a daily basis
  - I created a private S3 bucket in AWS and wrote scripts to upload both historical data and daily data to the private bucket
  - I created a public S3 bucket where I put my historical data. Since the process to create my historical dataset from the API is very long running, my QA partner and anyone else who wants to run my app will be able to download the data from here
  - I created an RDS MySQL instance and developed scripts to create tables and write data to the tables in either SQLite or MySQL
  - I provisioned an EC2 instance and made sure my project scripts would run on it
  - I developed test scripts and documentation for many of functions, modules, and scripts I wrote

Demo

- Since my model isn't totally finalized and I haven't started on the Flask app, I don't have any cool visuals yet
- I do have some command line outputs
  - This the is output from my player data being inserted in RDS

```
2019-05-15 22:52:54,784 root         INFO     Added 10498 players to database successfully
```

  - And this is the output of my team data being inserted into RDS

```
2019-05-15 22:52:55,034 root         INFO     Added 30 teams to database successfully
2019-05-15 22:52:55,034 __main__     INFO     Data ingestion conpleted
```

Lessons Learned

- It can be difficult to find data, especially some sports data, from an API without paying for the data. Before beginning a project that relies on data from an API make sure that API has data you want and can access easily
- I had quite a bit of trouble getting python scripts I wrote to import other modules I wrote. After much practice during this project, I know where/when I need to add __init__.py files and where/when I need to use sys.path.append()
- I learned a great deal about AWS, specifically about how to change the environment of an EC2 instance to get my code to run easily on that instance
- I learned to add documentation and logging to my project and code as I develop, because it can make it much easier to debug and fix issues when code from multiple modules is being run in one script

Next Sprint

- During this next sprint, I have two major items to work on:
  - Finalizing my model
    - The initial model I have been working with is a logistic regression model, however, I want to some other models (boosted tree, random forest, etc)
    - I am still not totally certain which features I would like to include in the model. I am thinking of trying to limit the number of features, so it is easier for users to understand where/how I am coming up with predictions I the app
  - Creating my flask app
    - For my app, I am envisioning 4 different web pages I need to develop (a 5th is in the ice box)
    - Each of these will be reading from RDS, and none will be writing any data anywhere
    - I intend to spend some time on the UX of these pages to get them to look nice
  - I also need to continue with the testing and documentation of my project (I feel that I have been doing a good job of this so far)