FLEX Project Guidelines

Git hub repository has been created and named FLEX, can be found by searching

**terror12 / FLEX**

Mission Statement:

Build a predictive engine that will output projected best lineups for weekly entries into FanDuel.

Methodology (rough thoughts, will improve based on research and trials):

FanDuel creates a price value for each player. We will be judging a players value as 2-1/2 times their price. This value will be compared with numerous different player ranking hosts (yahoo, espn, nfl.com etc.) The standard deviation will be found between every possibility of each host.

Ex.) FanDuel player = $6000 ….. Yahoo weekly projection player =$6800

I will most likely be looking at the lowest positive standard deviation.

-The idea will be to use the projections of these experts to come up with a system to predict the best starting lineup using 100% of the salary given.

The players with the highest positive standard deviation will be considered a must start. There will be a program in place that will produce starters by using the players considered must start while using all of the salary given.

Looking into Home and Away matchups could be an important parameter.

Future (Post build):

* Break down engines model by position not just by host
* To update Google Sheets to hold most recent data, if set up correctly, we will just need to open sheets when hosts put out new data and the data will be called automatically into sheets.

Data:

* Player names will need to be called in with rest of stats (possibly convert player names to id numbers)
  + FanDuel:
  + Rotoguru1.com

- Holds archives of all past FanDuel player salaries. Dating back to 2011.

- This will be used to try and identify patterns with our standard deviation model.

* + Google Sheet

- Google sheets will be used to pull data from Rotoguru1.com to start building our database.

- We will set up Google sheets to automatically update our database as new information comes out.

Coding:

* R:
  + Will be using R to perform statistical analysis, and to call in all data from google sheets.

TO DO LIST:

* Collect all data from Rotoguru1.com and enter into google sheets
* Find data archives for all past predictions of yahoo, espn, nfl.com etc.
* Create R code to call in data from google sheets and perform data analytics on it
  + Use capstone project as baseline for R code.
    - Insert all data
    - Group all data into data groups
    - Find standard deviation for all players based on their price value compared with other engines predictions
    - Build lineup
      * + Need ability to remove injured players by user input
      * Need to rank players from smallest positive STD to largest
      * Then need to come up with formula to enter in the players starting from top to bottom using all of the salary given.
    - Analyze all previous perfect lineups and find what the STD of the projections is
      * Should be used to analyze which platforms prediction seem to be more succesful
    - Create format of 1QB, 2RB, 3WR, 1TE, 1K, 1DEF as template for code to enter players into.

RESEARCH:

* Other projects that are trying to do similar things
  + Find other FanDuel or drafts kings models and learn from them.
* Google Sheets automation
* Find the standard deviation statistics based on all past perfect lineups
* Create visual model to show how engines will be built
  + One model will be Fanduel value compared with yahoo player rankings to find the players with the smallest positive standard deviation
  + Next model will be Fanduel value compared with espn player rankings to find the players with the smallest positive standard deviation
  + Another model will be all three of the combined to find that standard deviation and build a line up based on all threes projections.
* Look into players with questionable tags from the data archives.

Big Data Analytics:

* WEKA
  + Look into WEKA algorithms when models are created.

Debugging:

* Possible future bugs to be aware of
  + Did host update/edit their website
    - Within website check the html table number