**A place to store notes related to this project**

* Only a place to store quick text based notes, everything related to code and whatnot should be used in the playground.ipynb file

Pybaseball documentation: <https://github.com/jldbc/pybaseball/tree/master>

* Check the docs folder of the repository for specific functions

General Data Science Project Flow

1. Get the data
2. Clean the data
3. EDA of data
4. Feature Engineering
5. Modeling
6. Model Evaluation
7. Done?

So I got the data from fangraphs and was doing a bunch of EDA on it until I realized that I kinda needed to do more research into what factors actually are a good predictor of future baseball performance, rather than just looking at what fangraphs has.

* I am going to restart the EDA using data also from statcast

I am going to separate this project into several parts, the first part is going to be an EDA heavy part using things like PCA.

However, since I don’t have a lot of time, I am going to take a subset of the dataset, and use that to train a neural network using pytorch,

* In the future, I would like to use things like the PCA results to train a more fully fledged machine learning model, but I don’t know if I have the time right now and want to prioritize learning things like pytorch.

Template neural network using pytorch:

import torch.nn as nn

class BaseballModel(nn.Module):

def \_\_init\_\_(self, input\_size):

super().\_\_init\_\_()

self.fc = nn.Sequential(

nn.Linear(input\_size, 64), # Input layer

nn.ReLU(), # Activation function

nn.Linear(64, 32), # Hidden layer

nn.ReLU(),

nn.Linear(32, 1) # Output layer (1 neuron for regression)

)

def forward(self, x):

return self.fc(x)

self.fc defines a **feedforward neural network** using nn.Sequential, which stacks layers together.

* Simplest type of artificial neural network where information flows **only in one direction**, from the input layer to the output layer
* Input, hidden, output layers
* Other neural networks include, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Transformer Networks, etc

nn.Sequential basically tells python to go step by step through the code in the ().

I don’t know what I was expecting but I am completely stuck.

* Creating a model was much more tedious then I was expecting
* Like tuning the hyper parameters and what not, that’s extremely boring
  + I don’t even know how neural networks work anyway
* I have to spend hours learning pytorch if I want to get anywhere
  + That is going to take forever, and I don’t have the time for that
* I needed to plan this out better
  + I shouldn’t have spent hours trying to figure out which parameters to use and which were going to be dropped
  + I should’ve started with this project from the start of break instead of doing the other project
  + Like I didn’t need to get the statcast stuff.
* Now I don’t have a project at all