**Feedback:**

1. We should design the model in a way where a baked in cap isn’t necessary

* We agree with this; ideally, the model shouldn’t assume that lifting progress continues in a linear fashion indefinitely.
* For linear regression, this might be more of an issue. In response, we are experimenting with adding log features to try and capture the logarithmic nature of lifting progress. We are also looking into outliers in our data sets that could contribute to unreasonable results (some lifters made unreasonable progress that may have had to do with PEDs or errors in data entry).
* We are leaning towards a KNN because of this issue. A KNN will likely consider the context around lifters and their progressions better than linear regression, which might be more susceptible to outliers in data.

1. We should look at how much a given feature such as PEDs effects the result

* As we continue refining our models, we will experiment with different features to see how much each feature influences predictions. As a stretch goal, we would also like to train models on untested lifters and allow the user to compare predictions of both models.

1. The snapshot nature of our data might capture lifters at different stages of progression

* This is one of our bigger concerns. Like with the previous feedback, we are learning towards a KNN to counter this issue. By considering a large set of similar lifters, we hope to capture many lifters at many different stages of progression, contributing to a more balanced prediction. One possible stretch goal could be to capture this variance in experience and give a range of predictions (or percentiles).

**Work done since pitch:**

* Data Cleaning
  + We cleaned the data in a Jupyter notebook using a pandas dataframe for sufficient data visualization and data alteration functions. Raw data consisted of 3,290,035 entries with 42 features, where each entry was an instance of a powerlifting performance for a person at a certain competition. Cleaned the data to remove useless features like ‘State', 'Federation', ‘BestSquat, ’and 'ParentFederation' while keeping crucial features ‘Name’, ‘Sex’, ‘Age’, ‘Date’, ‘BestBench’, ‘BodyweightKg’, ‘WeightclassKg’, and ‘Date’. Only kept entries that were in a tested, sanctioned organization (to disallow PED’s swaying result in initial models) and limited equipment (kept raw and wraps for lifts, no slingshot as could drastically affect bench numbers). Further, keep only people that had multiple entries (> 1), since there would be no gauge for progression for people who only competed once. After all data analysis and cleaning, the dataset had 726,426 entries with 7 features.
  + Note: additional data manipulation will be needed to get data in proper form for each model used (Scaling, Normalization, OHE, Group-by names, etc,)
* KNN
  + Model: Built a KNN regression model that trains on data where each input is a unique individual with a sex label, starting bench number, starting body weight number, starting age, a list of benches, and a list of dates corresponding to each bench. The bench list and date list are used to fit simple linear regression models saved in a list, where each index corresponds to the matching entry. Then, the sex labels are one-hot encoded withOneHotEncoder() and the starting bench number, starting body weight number, and starting age are all transformed using StandardScalar(). The encoder and scalars are saved for transforming our test data into predictions. After all data is scaled and the linear regression models are fit, the transformed training data is entered into a ball tree. For prediction we are trying to predict bench increase given a number of days. So, when predicting the testing data is transformed in the same manner as the training data, and the k closest points in the ball tree. The indices are retrieved for the k closest points and the linear regression estimate for the bench based on the input number of days to be calculated. The average of these bench estimates is returned for our test point bench increase prediction.
  + Training/Results: The model was trained on 129,224 data entries and tested on 32306.2. Three types of error were calculated, Root Mean Squared Error (RMSE) Mean Absolute Percentage Error (MAPE), and Absolute Error. The results of these errors were calculated for different k values, in which the best results were achieved with a k of 20 where RMSE is 30.758840176270468, MAPE is 8.841328572408262 and Absolute Error is 8.589341736628077. The errors are decent, and a solid start. In essence, given estimates were roughly 8kg off on average for the true progression.
  + Issues: Works much better with a smaller number of days, less than 365. Likely due to the nature of simple linear regression estimates, and the fact most entries don’t have data past 365 days since start. Potential bodyweight changes not accounted for. Snapshot aspect of data not accounted for as well.
* Regression
  + Model: We built Linear Regression, Ridge Regression, Lasso Regression, and Elastic Net models to predict the increase in bench press weight after a given number of days. The model takes Age, Bodyweight, DaysSinceFirstMeet, and InitialBench as parameters (we are experimenting with taking the log of some of these features). In this case, DaysSinceFirstMeet is a proxy for the number of days into the future you want to predict.
  + Data cleaning: For these regression models, we did some additional data cleaning. Rather than one-hot-encoding sex as a feature, we split data into separate files for each sex, then made predictions separately (we also plan to experiment with using the original data set and weighting sex features). We also added the following features:
    - InitialBench: The bench recorded at lifter’s first meet.
    - BenchIncrease: The increase in bench from the current meet to the lifter’s first meet.
    - DaysSinceFirstMeet: The number of days since the lifter’s first meet.
  + We did initial cleaning with these features, such as removing lifters with only 1 meet and removing instances where BenchIncrease was negative. We are looking into removing suspiciously high BenchIncrease instances that could skew data, and might make the required number of meets higher.
  + Training/Testing (requiring more than 1 meet and cutting off BenchIncreases of 100 or more (subject to change))
    - Male data: The training set consisted of 217632 samples while the testing set consisted of 54408 samples. The linear regression model had a mean absolute error of 8.65 kg, Ridge Regression had 8.65, Lasso had 10.78, and Elastic Net had 9.82.
    - Female data: The training set consisted of 105684 samples while the testing set had 26421 samples. The linear regression model had a mean absolute error of 5.464 kg, Ridge Regression had 5.465, Lasso had 6.786, and Elastic Net had 6.096.
  + Issues: While the MAE doesn’t seem too high, the data seems pretty skewed when graphed. The model seems to predict smaller increases well, but underestimates higher increases in the long term. One possible solution could be experimenting with different data cleaning, such as requiring a higher number of meets or analyzing data for long term lifters.
* Neural Network
  + Model: We implemented a neural network to predict the bench press increase over time, using the cleaned dataset with key features including age, body weight, days since the first meet, and initial bench press value. The neural net was designed to be relatively simple, with three hidden layers of sizes 128, 64, and 32 neurons. We used the ReLU activation function for all hidden layers, which helps the model capture non-linear relationships in the data. The output layer used a linear activation function, for regression, and the model was evaluated using Mean Squared Error (MSE) as the loss function and with Mean Absolute Error (MAE) as an additional evaluation metric. Using a learning rate of 0.001
  + Training/Results: The model was trained on 101,104 data entries and tested on 25,276 entries. Over 50 epochs, the neural network achieved a final training loss (MSE) of **104.52** and a MAE of **7.263**. These results suggest that the model’s predictions deviate, on average, by around 7.3 kg from the actual bench increase values. The validation loss (MSE) indicates that the model is generalizing reasonably well without significant overfitting.
  + Issues: One challenge with the neural network is its sensitivity to outliers and noise in the data. For instance, lifters with less standard progressions could influence the predictions. Additionally, the neural network does not account for the diminishing returns typically observed in strength progression, which might be better captured by a logarithmic transformation or feature engineering.
  + Planned Updates: Future iterations of the model will experiment with logarithmic transformations on features like DaysSinceFirstMeet to better capture the nonlinear nature of lifting progress.

**Planned updates**

* KNN
  + Will see how good we can get KNN by achieving the lowest MSE, MAPE, and Absolute Error. Thinking of weighting the one-hot encoding of the Sex feature to see how different progress between men and women can affect prediction. Further, I want to take into account if there is a change in body weight. Can do this by adding body weight as a feature to each learning regression estimate. Finally, I intend to try and fix the issue of working with large numbers of days. Could try by creating a custom linear regression model with caps or might try out how Lasso and Ridge regression models work as well.
* Regression
  + We will continue to experiment with further data cleaning, such as increasing the number of required meets and identifying outliers in the data. We also plan to experiment with using the original data set and weighting sex features higher, then comparing to the current models.
  + We also plan to continue experimenting with tunable parameters for Ridge, Lasso, and Elastic Net.
* Clustering
  + To enhance prediction accuracy, I’ll explore a hybrid approach using K-Means clustering to cluster lifters by similar characteristics (such as age, body weight, days since first meet, and initial bench) and then applying different models within each cluster. By grouping lifters based on progression patterns, this method should allow each model to capture specific trends for each lifter, such as fast progressors versus those who plateau. Once clusters are defined, I’ll train separate neural networks or KNN models for each group.

**Group Feedback**

Our group is working well together. We discussed our project before and after each class session, and each worked on a different approach to the problem (KNN, regression, and neural networks). We also have a group chat to discuss the project outside of class.