Predicting Pitcher Injuries Using Sabermetrics and Machine Learning

Arjun Nichani

Background

- Roughly one third of the pitchers in the MLB (Major League Baseball) have experienced a UCL Injury.
- The UCL (Ulnar Collateral Ligament) is a primary stabilizer in the arm and plays an important role in all throwing sports.
- A UCL tear may require Tommy John surgery, preventing the player from playing for up to 15 months.
- For all the progress made in science and mathematics, in different areas of baseball, the prediction and prevention is a frustrating problem.
- In a game where everything is analyzed in great detail, not even sabermetricians have been able to reduce the rate at which pitchers get hurt.

Task

- A system that can predict UCL injuries in pitchers before they occur, so they can be shutdown before completely tearing the ligament, may be extremely beneficial to MLB teams and trainers.

Methods

- In this project I used various machine learning techniques to create a binary classifier that predicts imminent Injuries.
- In addition, I attempted to determine statistical significance of the various features to determine what factors play relevant roles in UCL tears.

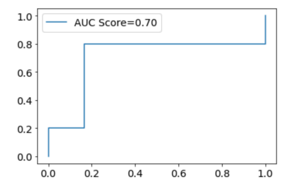
Dataset and Feature Selection

- The dataset was composed of 50 different MLB pitchers.
  - 30 of the pitchers had not torn their UCL.
  - 20 of the pitchers tore their UCL.
- Data was collected from http://www.brooksbaseball.net and http://www.baseballprospectus.com.
- Pitchers needed to pitch minimum 5 games.
- 17 different features were collected.
- All of the data is taken from starting pitchers.
- Data was randomly split into training and development sets.
  - 80% of the data was in the training set, 20% was in the development set.

Feature	Description
GS	The number of games started by the pitcher that season (if played more than 5 games) or the season before (if played less than 5 games current season)
Tot NP	The total number of pitches that were thrown that season
Max NP	The maximum number of pitches thrown in 1 game that season
Avg NP	The average number of pitches per game for that season
Tot PAP	The total number of pitcher abuse points that season
Max PAP	The maximum number of pitcher abuse points per game
Avg PAP	The average number of pitcher abuse points per game
Cat 1	The number of games in which the pitcher threw under 100 pitches
Cat 2	The number of games in which the pitcher threw from 101-109 pitches
Cat 3	The number of games in which the pitcher threw from 110-121 pitches
Cat 4	The number of games in which the pitcher threw from 122-133 pitches
Cat 5	The number of games in which the pitcher threw over 133 pitches
Stress	The amount of stress the pitcher endured over the season
Tot BB	The total number of breaking balls thrown (slider or curveball)
Avg BB Velo	The difference in the average velocity of a breaking ball thrown in the 1st month and the last month of the season (slider or curveball)
FB Speed Drop	The difference in the average velocity of a fastball thrown in the first month and the last month of the season (4 seam, 2 seam, or cutter)
BB Speed Drop	The drop of speed of breaking balls from the 1st month of the season to the last (slider or curveball)
Tot BB /Tot NP	The number of breaking balls thrown per pitch during the season (slider or curveball)

Logistic Regression

- This was the first, most basic, model I attempted to use.
- Used as a baseline model.
- Multiple trials of the algorithm were used.
  - Each trial used a unique seed value.
  - This changes the train/test split for each trial.
- After running 100 trials of the logistic regression model, the average AUC was **0.724**.



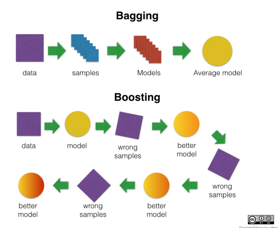
An Example of the ROC curve of one trial of the Logistic regression model. For this specific seed, the model produced an AUC of 0.70 on the test data

Logistic Regression with Cross Validation

- In addition to the traditional logistic regression model, I employed a logistic regression model that implemented cross validation.
- Cross validation is a method that makes the most of a small data set.
  - First the dataset is split into n partitions.
  - It holds out one partition of the train set to use as a validation set.
  - It then repeats the process n times, holding out a different partition each time.
- I implemented a 5-fold cross validation technique.
- After running the logistic regression, with cross. validation, the average AUC measure on the entire dataset was **0.849**.

Boosting

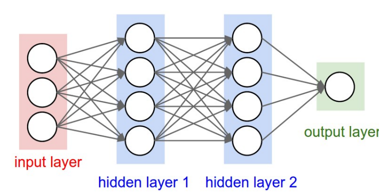
- The next method that I attempted to use was called boosting.
- I used XGBoost in order to create the new model.
  - XGBoost implements a technique called gradient boosting.
- Gradient boosting in an ensemble method that seeks to produce a strong classifier based on weak classifiers.
- Models are added on top of each other iteratively.
- The errors of the previous model are corrected by the next predictor.
- In gradient boosting, a new model is fit to the residuals of the previous prediction.
- After running 100 trials (with different seeds) of the XGBoost model, the average AUC was **0.806**.



https://bradzzz.gitbooks.io/ga-dsi-seattle/dsi/dsi_06_trees_methods/3.1-lesson/readme.html

Single Hidden Layer (Shallow) Neural Network

- I implemented a neural network in order to increase the complexity of the model.
- The neural network is a superior model due to fact that it explores a larger family of mathematical functions between input and output.
- The nonlinear activation function used in within the nodes of the hidden layers are ReLU while the node in the output layer uses a sigmoid.
- I began with a neural network that contained only one hidden layer.
  - This was done to get a feel of the performance of the neural network.

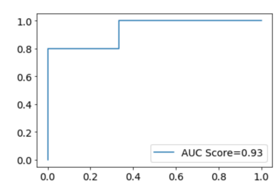


https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/

- There was some fear that the neural network would overfit due to the small data set
- The optimal size of the hidden layer was 10 nodes.
- Unfortunately, the single hidden layer performed quite poorly
- After running 100 trials (with different seeds) of the single layered neural network, the average AUC was **0.735**.

Multilayered (Deep) Neural Network

- After the single layer neural network did not produce good results, it was important to check a multilayered network.
- The goal of the multilayered network was to increase the complexity of the model to see if it performed better.
- I was able to determine that optimal size of the neural network contained 4 hidden layers.
  - I did this by adjusting the number of layers and the size of each of the layers.
  - The first hidden layer contained 30 nodes.
  - The second hidden layer contained 20 nodes.
  - The third hidden layer contained 10 nodes.
  - The fourth hidden layer contained 5 nodes.

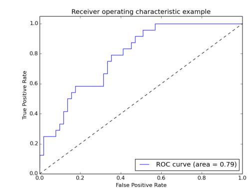


An Example of the ROC curve of one trail of the deep neural network. For this specific seed, the model produced an AUC of 0.73 on the test data

- After running 100 trials (with different seeds) of the multilayered neural network, the average AUC was **0.821**.
- This was the best model.

The Evaluation Metric

- It is best to optimize one evaluation metric when doing machine learning.
- The metric that I selected was the AUC (Area under the curve) of an ROC (Receiver Operating Characteristic) curve.
- The ROC curve is a curve that graphs the relationship between the true positive rate (how many injuries you predict as injuries) and false positive rate (how many healthy pitchers you predict as injured)
- The area under this curve is a good measure of the strength of the model.
- The closer the AUC is to one, the better the model is.



https://scikit-learn.org/0.15/auto_examples/plot_roc.html

- A completely random guess should produce an AUC of 0.5 (the the dotted line).
- This measurement is better than the traditional accuracy measurement as it includes sensitivity and specificity of the model. It is also not dependent on the threshold.
- Coaches or trainers may have a different preference to what type of error they would rather make.
- By using AUC, we know that the actual performance of the model, which can be modified to meet the coach's or trainer's need.

Feature Significance

- There is much debated in the field of machine learning and medicine about the purpose/use of "black box" algorithms
- While system that perfectly predicts UCL tears would be extremely helpful, learning the reason behind the injury could prove to be more beneficial.
- This is reasoning behind feature significance in this project.
- I did the statistical analysis on the logistic regression model.
- This model assumes linearity of the various features
- No one feature was statistically significant enough to be below the 0.05 threshold.
  - The 0.05 p-value threshold is generally accepted in statistics to be the largest p-value that you can have and still reject the null hypothesis.
  - The p value is the probability that a value is equal to or more extreme than the value you obtaining, assuming the null hypothesis is true.
- The most statistically significant features were maximum number of pitches, Stress, and Average Fastball speed drop.
- The maximum number of pitches was inversely proportional to the likelihood of UCL tears. The lower your maximum, the more likely you are to injure your UCL.
  - One explanation of this could be a lingering minor injury which affects stamina
  - As the pitcher pitches with their minor injury, they are more likely to injure their UCL
- Stress is a statistic that was calculated by baseballprospectus.com. Its goal was to determine the amount of stress that a pitcher underwent during that specific season.
- Fastball speed drop is a feature that I manually calculated. It is the difference between the pitcher's average fastball speed in the last month of their season and their average speed in the first month of the season.
  - The more their velocity dropped, the more likely they were to be injured.
  - This, again, could show fatigue or even a lingering injury for the pitcher.
- Since these features are not statistically significant we can not conclude that they are relevant to predicting injuries but they are features to look into if this is replicated with a larger data set.

Conclusion

- The results were quite promising for the small amount of data that was collected.
- As expected, the neural network was the best model, followed by the random forest, and then the logistic regression.
- No individual features were statistically significant enough in building the model to predict UCL tears.

Applications/Extensions

- Gather more data.
  - The project was limited due to the small amount of data that was collected. If more data was collected, the multilayered neural network should outperform the logistic regression and random forest by a lot more than it did.
- Increase the number of features
  - It would be great to expand the list of features
- Attempt to gather biometric data
  - There is a lot of speculation that mechanics play a large role in injury
- Try other machine learning techniques.
- Use other evaluation metrics.
- Analyze feature significance within the random forest and neural network models.

Mentor: Professor Jelena Bradic

Arjun Nichani: anichani@ucsd.edu