ISyE 6748 Project
Sandia National Laboratories
Predicting UCL Injury in Baseball Pitchers Using Pitch-Level Data

Cale Williams
cwilliams461

Spring 2024

1 Problem Statement

In baseball, pitchers often need to undergo ulnar collateral ligament (UCL) reconstruction, or Tommy John surgery (TJS). The UCL is a ligament in the elbow and can be torn from overuse. Over $\frac{1}{3}$ of pitchers in Major League Baseball (MLB) in 2023 had TJS at some point in their career. Recovery takes roughly 18 months, meaning affected players miss at least a full season.

This analysis attempted to predict if a pitcher will need TJS in a season using machine learning techniques on pitch-level data.

2 Data

2.1 Sources & Collection

The data used in this analysis is from the 2015-2023 MLB regular seasons and comes via their Statcast[1] tracking technology. The pitch-level data was scraped using the *pybaseball* Python library. Other pitching-related data needed was retrieved and saved manually from Baseball Savant[2]. TJS information came from Jon Roegele's extensive Tommy John Surgery List[3]. This analysis could not have been done without his efforts.

[1]mlb.com/glossary/statcast
[2]baseballsavant.mlb.com
[3]TJS List

2.2 Processing

TJS cases were limited to surgeries performed 2015-2023 between the months of May and October. The month filter was implemented to avoid injuries due to offseason activities.

The starting dataset includes 1.3 million rows (pitches) across 9 seasons and 117 pitchers. The data was summarized by pitcher-season combinations, referred to as pitcher-seasons for the remainder of this work. There were 739 pitcher-season combinations in the initial set. For example, pitcher Bartolo Colon pitched in each season from 2015 to 2018, so he accounts for 4 of the 739 pitcher-seasons. Pitcher-seasons with fewer than 100 pitches thrown were removed from the dataset. A histogram of pitch count in pitcher-seasons is shown in Figure 1.

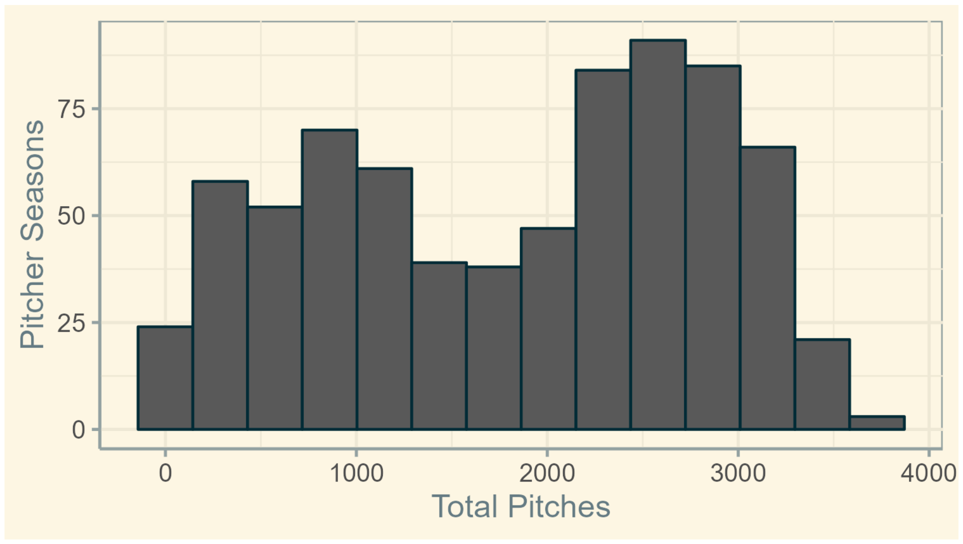

Figure 1: Pitch Count Histogram

The three most common pitch types were used in this analysis. These pitch types are classified by Baseball Savant and are based on pitch qualities like movement and velocity. The pitch types included were the four-seam fastball ($n = 442,924$), sinker ($n = 264,237$), and slider ($n = 191,693$), making up 66.2% of the full dataset. Pitcher-seasons in which each of these pitch types were not thrown at least 5 times were removed from the dataset.

Because not every pitcher throws each of these pitch types, some pitch sample sizes in a pitcher-season may be small and some pitcher-seasons may be excluded. For example, in 2015, R.A. Dickey's most used pitch was a knuckleball, followed by a four-seam fastball and an eephus pitch. He did not throw a single sinker or slider, so his 2015 season was not included in this analysis. Another relevant example is Zack Greinke's 2020 season in which he threw the following pitches:

- four-seam fastball ($n = 432$ pitches),
- changeup (232),
- slider (170),
- curveball (161),
- and sinker (1).

Although he threw the relevant pitch types, his low sinker count filters this pitcher-season out. Additionally, his changeup is more prominent than his slider and sinker, but would be ignored even if he threw enough sinkers to qualify. These kinds of omissions could be reevaluated in future analysis.

The intersection of the qualified pitcher-seasons for each pitch type yielded 345 pitcher seasons comprising of 494,062 pitches. The data was randomly split into training (70%) and test (30%) sets. The dependent variable is a binary variable dependent on whether the player underwent TJS, ending the respective pitcher-season. The data was imbalanced, with the training set containing 11 of 241 (4.6%) positive (ended in TJS) pitcher-seasons and the test set containing 8 of 104 (7.7%) TJS cases.

Pitch Type	**Pitches**	**Pitcher-Seasons**
Four-Seam Fastball	442,591	686
Sinker	264,104	577
Slider	191,565	521
Inner Join Dataset	494,062	345

Pitch data for pitches thrown in minor league games and practice/warm-up sessions is unavailable and thus, is not included in this analysis. This neglects some stress placed on the pitchers' arms, but the knowledge gap is unavoidable.

3 Analysis

3.1 Feature Engineering

The features available and examined for use in this analysis are shown below. Further information about these features can be found in the Statcast documentation[4]. The distributions for these features are shown in Figure 2.

Feature	**Type**	**Description**	**Unit**
release_speed	numeric	velocity	mph
pfx_x	numeric	horizontal movement	feet
pfx_z	numeric	vertical movement	feet
release_pos_x	numeric	horizontal release position*	feet
release_pos_y	numeric	distance release position*	feet
release_pos_z	numeric	vertical release position*	feet

*from catcher's perspective

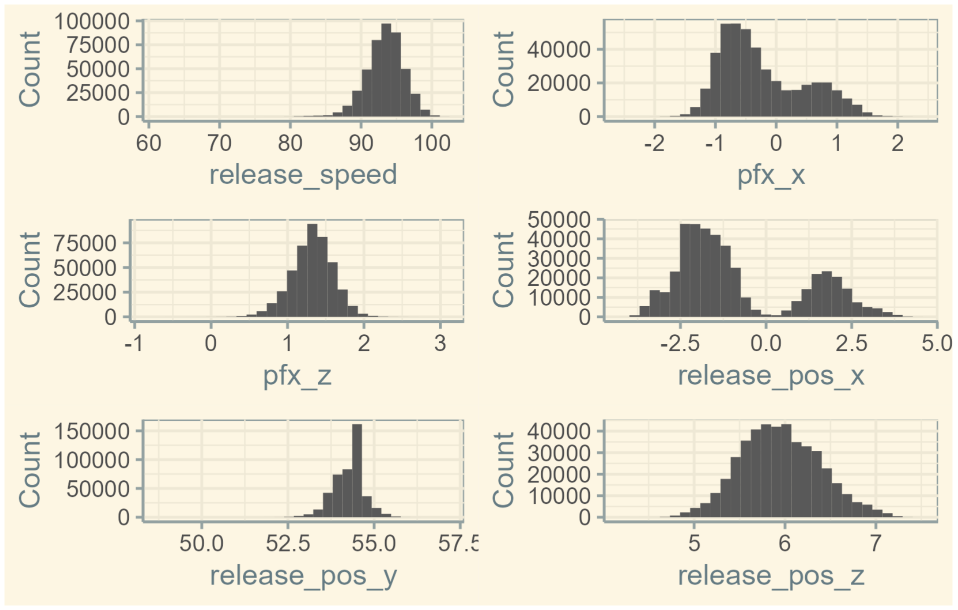

Figure 2: Feature Distributions

It is evident that the horizontal movement and release position distributions are bimodal. This is due to the dataset including both right and

[4]baseballsavant.mlb.com/csv-docs

left-handed pitchers.

Across each pitcher-season, the pitch metrics were summarized for each pitch type. See Figure 3 for an example of a pitch release speed summary method.

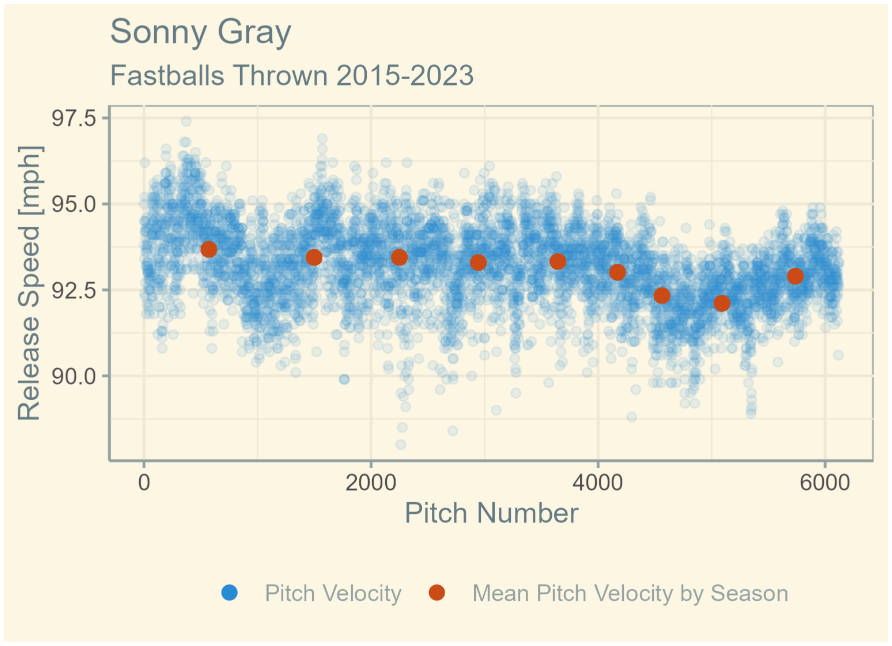

Figure 3: Pitch Metric Summary Example

Release speed was summarized for each pitcher-season via the mean function to create *mean.release_speed*, but other metrics used different functions as shown below. Additionally, some features were used multiple times to capture distribution characteristics that would be ignored by using solely the mean.

Feature	Summary Function
mean.release_speed	**mean**(*release_speed*)
q90.release_speed	**quantile**(*release_speed*, probs = 0.9)
sd.release_speed	**sd**(*release_speed*)
absmean.pfx_x	**abs**(**mean**(*pfx_x*))
sd.pfx_x	**sd**(*pfx_x*)
mean.pfx_z	**mean**(*pfx_z*)
sd.pfx_z	**sd**(*pfx_z*)
sd.release_pos_x	**sd**(*release_pos_x*)
sd.release_pos_y	**sd**(*release_pos_y*)
sd.release_pos_z	**sd**(*release_pos_z*)
prop	**n() / sum(n())**

Each of these features was created for each pitch type in each pitcher-season. For clarity, a description of each metric is given as:

- *mean.release_speed*, *mean.pfx_z*: average of respective feature,
- *q90.release_speed*: release speed 90th percentile value; a **max()** term could probably be used instead, if desired,
- *sd.release_speed*, *sd.pfx_x*, *sd.pfx_z*, *sd.release_pos_x*, *sd.release_pos_y*, *sd.release_pos_z*: standard deviations of respective metrics; an attempt to capture change or variance within a pitcher-season,
- *absmean.pfx_x*: absolute value of the average of horizontal movement; this controls for pitcher-handedness bimodality shown in Figure 2,
- *prop*: proportion of pitch type thrown.

A prefix was added for each pitch type, e.g., *si.sd.pfx_x* refers to the standard deviation in the horizontal movement for the sinker in a pitcher-season and *ff.prop* refers to the proportion of total pitches a pitcher threw a fastball in a season. A few rows and columns of the training dataset, including the dependent variable *tjs*, are shown below.

Pitcher ID	***Season***	***ff.prop***	***si.prop***	***sl.prop***	***ff.mean.release_speed***	***si.mean.release_speed***	***sl.mean.release_speed***	***tjs***
607536	2022	0.237	0.224	0.220	90.2	89.6	85.4	0
425844	2022	0.383	0.022	0.074	89.2	89.4	81.8	0
656427	2017	0.293	0.257	0.249	93.4	92.6	84.5	0
607200	2017	0.044	0.513	0.060	92.4	93.5	81.2	0
572070	2018	0.312	0.191	0.390	95.8	96.0	89.2	1

3.2 Model Building

Initially, ARIMA and CUSUM models were built in an attempt to model pitch metrics like release speed as a series or signal. The idea was that prior to injury, some trend may be common as found in a report by Mayo et al[5]. However, these did not yield fruitful results and efforts were shifted to classification models. Thus, the models evaluated include logistic regression, decision tree & random forest, and support vector machines.

3.2.1 Logistic Regression

To start, a logistic regression model was built on all of the predicting variables. At an α level of 0.1, only *si.sd.release_pos_x* was found to be statistically significant with a p-value = 0.078. The confusion matrix and relevant metrics are shown below.

		True	
		0	**1**
Predicted	**0**	94	7
	1	2	1

[5]Preventing Tommy John Surgery: The Identification of Trends in Pitch Selection, Velocity, and Spin Rate Before Ulnar Collateral Ligament Reconstruction in Major League Baseball Pitchers

AUC	Recall	F1 Score	Balanced Accuracy
0.503	0.125	0.182	0.552

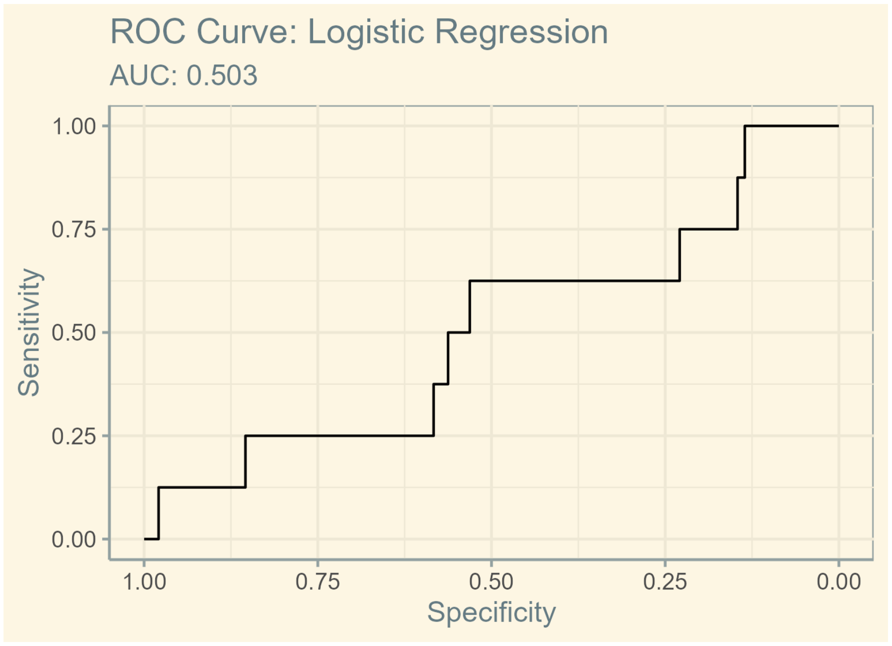

Figure 4: ROC Curve

To improve the model, L1 regularization was employed which forced all variable coefficients except *ff.sd.pfx_x* to 0. Thus, a new model was built with only this feature, resulting in $\beta_{ff.sd.pfx\_x} = -19.7$ with p-value $= 0.006$.

		True	
		0	**1**
Predicted	**0**	16	3
	1	80	5

AUC	Recall	F1 Score	Balanced Accuracy
0.542	0.625	0.108	0.396

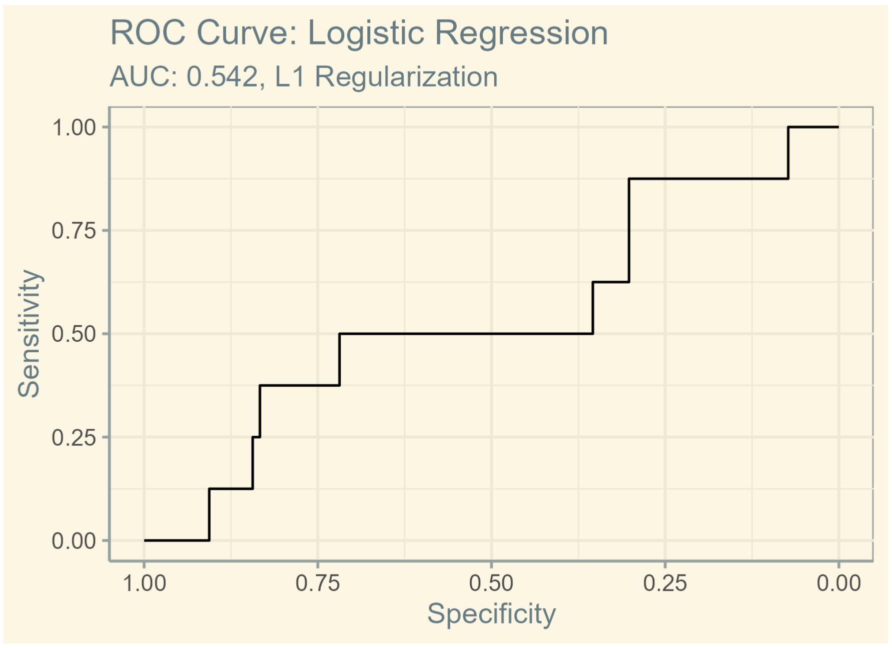

Figure 5: ROC Curve

Backward selection was also tested with the selected variables and the model summary is shown below. Most feature coefficients are significant at an α level of 0.1.

Feature	$\boldsymbol{\beta}$	**p-val**
ff.sd.pfx_x	-20.0	0.031
ff.mean.pfx_z	-3.8	0.069
si.absmean.pfx_x	-2.8	0.125
si.sd.release_pos_x	-17.9	0.046
si.sd.release_pos_y	11.4	0.041
sl.absmean.pfx_x	-2.8	0.089
sl.mean.pfx_z	2.7	0.106
sl.sd.release_pos_x	15.4	0.090
sl.sd.release_pos_z	-22.2	0.182

		True	
		0	**1**
Predicted	**0**	58	3
	1	38	5

AUC	Recall	F1 Score	Balanced Accuracy
0.521	0.625	0.196	0.615

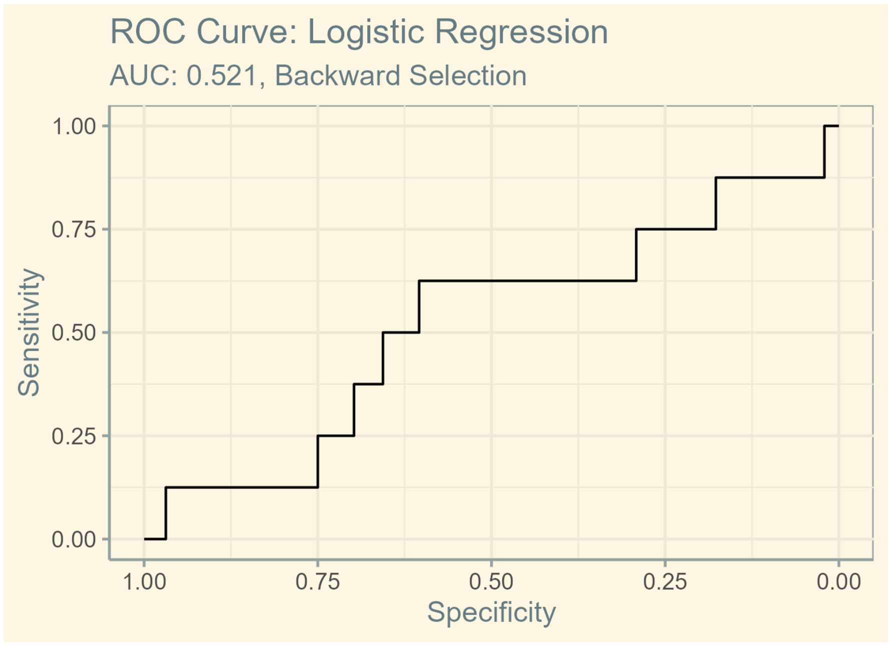

Figure 6: ROC Curve

3.2.2 Decision Tree

A decision tree was built with a minimum terminal leaf size set to 3 pitcher-seasons. The resulting tree is shown in Figure 7.

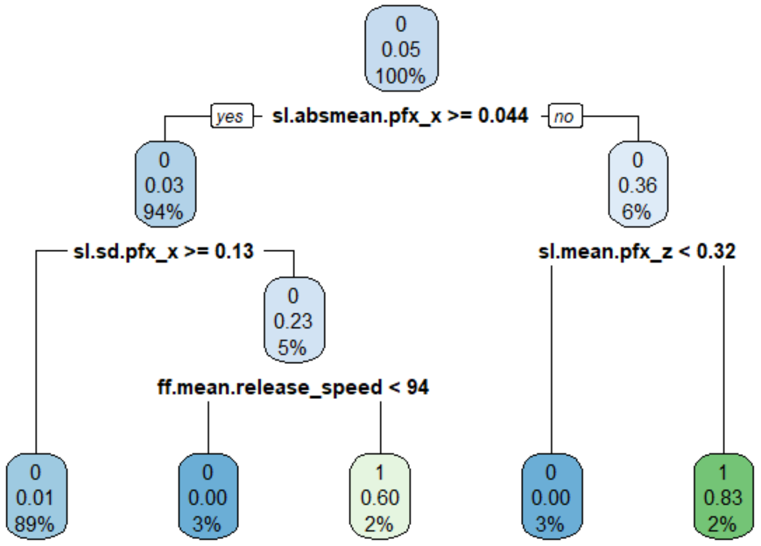

Figure 7: Decision Tree

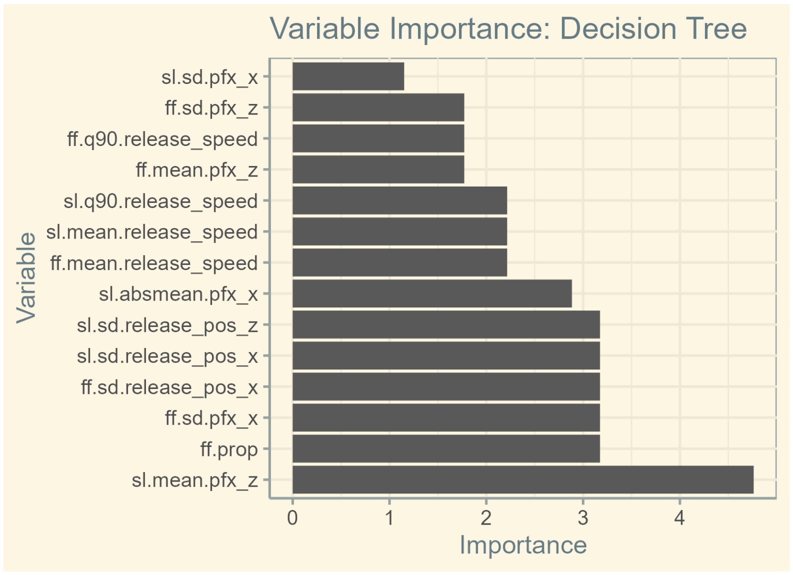

Figure 8: Decision Tree Variable Importance

		True	
		0	**1**
Predicted	**0**	1	0
	1	95	8

AUC	**Recall**	**F1 Score**	**Balanced Accuracy**
0.448	1.000	0.144	0.505

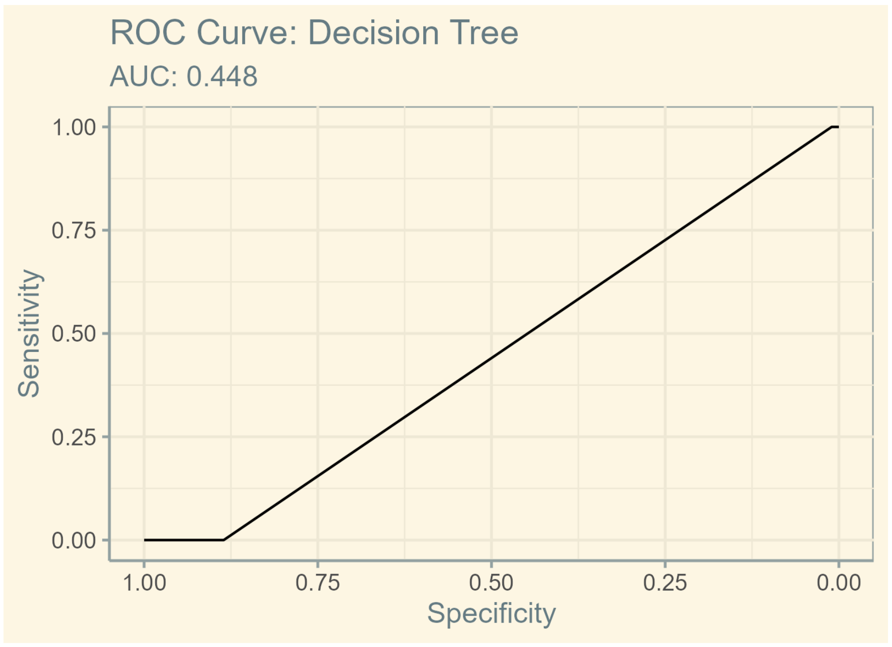

Figure 9: ROC Curve

3.2.3 Random Forest

Next, a random forest model was built. Results are shown below. The variable importance metric used was mean decrease in accuracy, or how the model declines when the feature is removed.

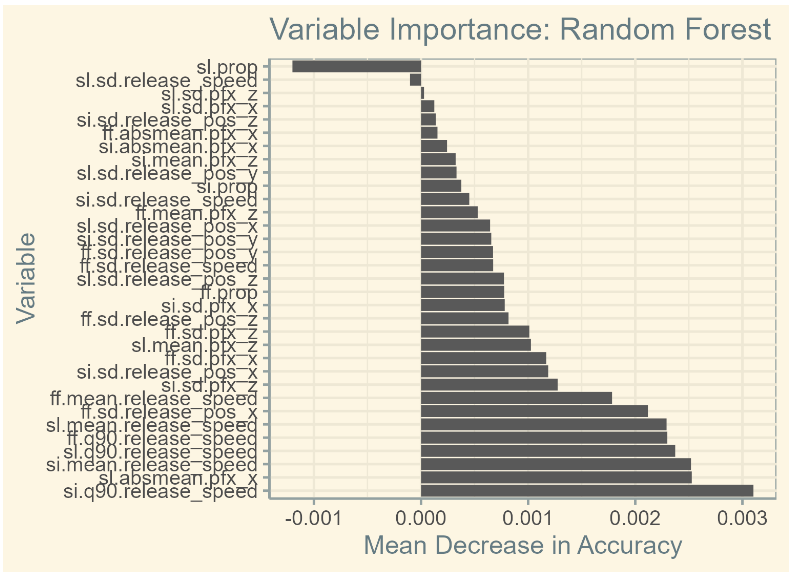

Figure 10: Random Forest Variable Importance

		True	
		0	**1**
Predicted	**0**	47	6
	1	49	2

AUC	**Recall**	**F1 Score**	**Balanced Accuracy**
0.577	0.250	0.068	0.370

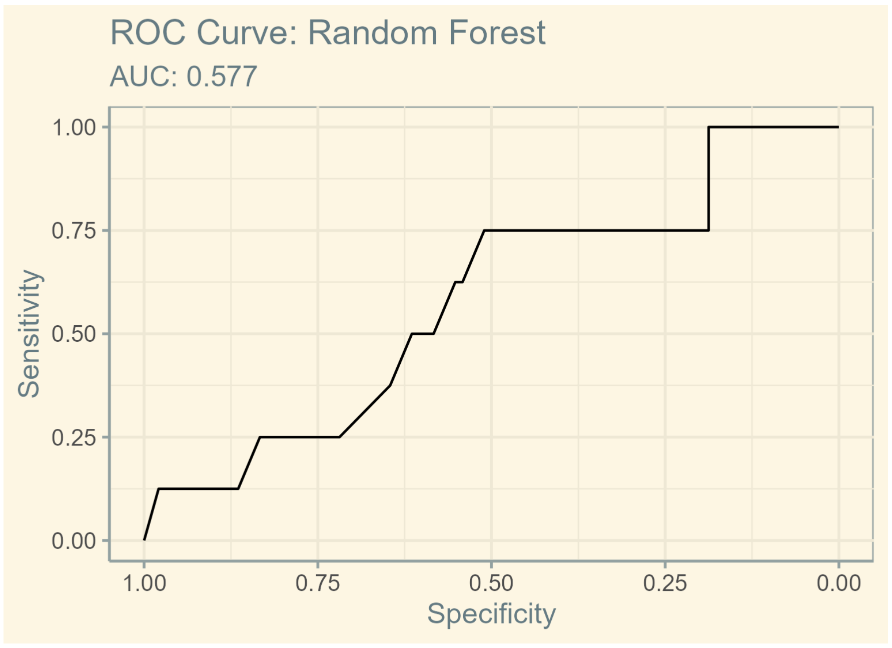

Figure 11: ROC Curve

3.2.4 Linear SVM

The linear SVM model results are shown below.

		True	
		0	**1**
Predicted	**0**	7	2
	1	89	6

AUC	**Recall**	**F1 Score**	**Balanced Accuracy**
0.595	0.750	0.117	0.411

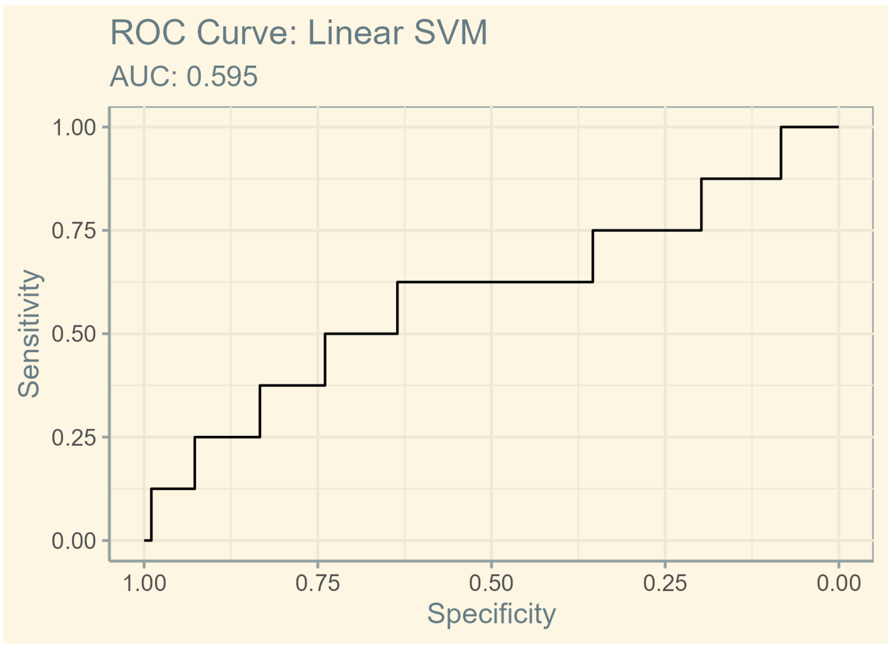

Figure 12: ROC Curve

3.2.5 Non-linear SVM

Finally, a non-linear SVM was built.

		True	
		0	**1**
Predicted	**0**	52	3
	1	44	5

AUC	**Recall**	**F1 Score**	**Balanced Accuracy**
0.505	0.625	0.175	0.583

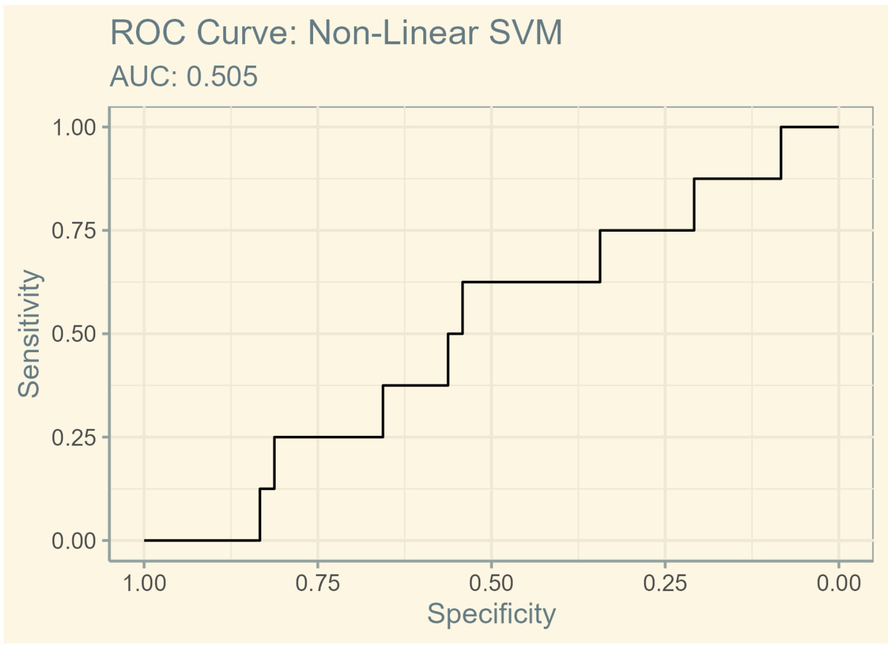

Figure 13: ROC Curve

3.3 Model Evaluation

Model summaries are below. To judge these models, AUC, recall, F1 score, balanced accuracy, and the confusion matrices were investigated. The imbalanced data does not allow for simpler metrics like accuracy to be used meaningfully. Because the consequences of a false negative are high, recall instead of precision was factored in. A false negative could lead to avoidable injury and lost games and money. A false positive, on the other hand, would presumably lead to unnecessary inspection or rest.

For the logistic regression model, the backward selection method outperformed the model with all of the features and the LASSO model. The decision tree predicted all but 1 pitcher-season would end in TJS, thus its perfect recall score should be ignored. To build the decision tree, tuning on the terminal leaf and leaf split sizes was performed, but it evidently was not effective. The random forest model only predicted 2 of 8 TJS cases correctly. The linear SVM was similar to the decision tree, in that there were so many false positive cases that it is essentially useless. However, the non-linear SVM had a reasonable balance of classifications and correctly classified 5 of 8 TJS cases. Overall, the logistic regression model with backward selection and the non-linear SVM performed superior to the other models evaluated.

Model	**AUC**	**Recall**	**F1 Score**	**Balanced Accuracy**
Logistic Regression	0.521	0.625	0.196	0.615
Decision Tree	0.448	1.000	0.144	0.505
Random Forest	0.577	0.250	0.068	0.370
Linear SVM	0.595	0.750	0.117	0.411
Non-linear SVM	0.505	0.625	0.175	0.583

From the logistic regression model coefficients, we learn that pitch release position and horizontal and vertical pitch break seem to be the best explainers of TJS odds.

4 Conclusion

Although the best models built are still not great at classifying seasons in which Tommy John surgery will occur, they are useful. Results are also in line with other analyses like Woodrum's[6] and Rendar & Ma's[7].

Future work could include improving on the preliminary ARIMA and exponential smoothing models, further feature engineering, and more data collection.

[6]Predicting Tommy John Surgeries

[7]Predicting Ulnar Collateral Ligament Injury in Rookie Major League Baseball Pitchers