NFL Injury Risk Analysis

Veronica Figueroa

Brown University

https://github.com/veronicafigueroa/NFL-Injury-Risk-Analysis

Introduction

The National Football League is the most popular sporting organization in the United States of America [1]. As the sport's status increases, however, so do its negative effects. One study indicates that, when standardizing for hours of participation, football produced the highest injury rates of any American sport. [2]. Economically, this poses a large problem. In 2019, the NFL spent over half a billion dollars on injured players [3]. More importantly, however, this high incidence rate jeopardizes player safety [4, 5]. Football injuries need to be reduced, but there exists much discourse regarding the root causes behind them. Using two cross-referencing NFL-provided Kaggle datasets, I attempted to construct a machine learning algorithm to answer the following question: By looking at a range of potential significant factors, can we accurately predict the incidence of NFL player injuries?

These datasets chronicle 267,033 individual plays over a two-season stretch and cross-reference each other with three indexing fields (PlayerID, PlayID, and GameID). There are 11 features that describe environmental and player conditions during the play (Field Type, Player Position, etc.), and each data point (play) is labeled as injury or non-injury, making this a classification problem. Furthermore, the datasets are extraordinarily imbalanced, with 99% of the data points in Class 0 (non-injury). There exist few missing values, with the majority of the features missing less than 1% of their values, but they are widespread between rows, resulting in 13.1% of the total data points having missing values.

Exploratory Data Analysis (EDA)

For ease of coding, I merged the two datasets based on those index fields. For EDA, I explored various initial correlations between injury occurrence and feature fields. When analyzing injuries by roster position and field type, I was intrigued to see that the occurrence of

injuries on natural and synthetic grass amongst the player positions was relatively equal; this was interesting because NFL players often complain about synthetic fields predisposing them to injury [5].

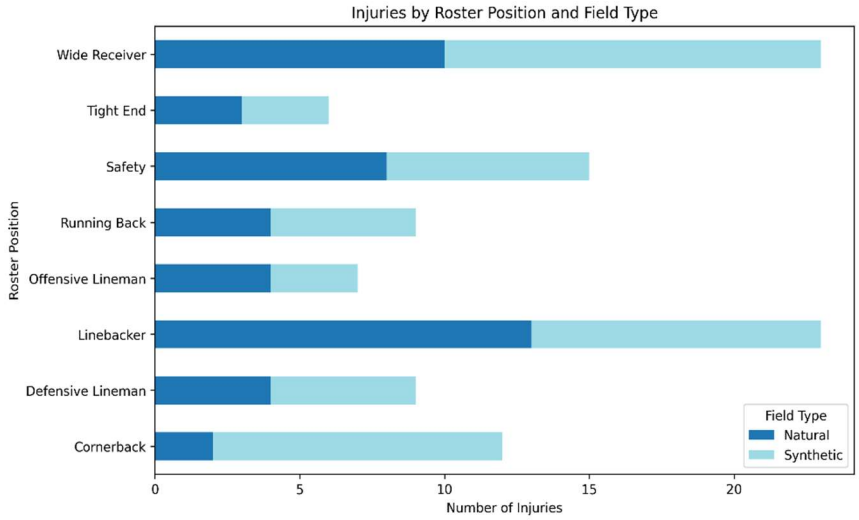

Figure 1. Stacked bar plot displaying injury distributions among NFL roster positions and field types.

I also explored potential relationships between injury classification and temperature. I found it interesting that injury plays seemed to occur in middling temperature ranges, while non-injury plays occur during both those and extreme temperatures. One would assume that extreme temperatures would make the body function less optimally and predispose athletes to injury; however, this can likely be attributed to mere dataset imbalance, with non-injury plays more likely to cover a wider range of temperatures.

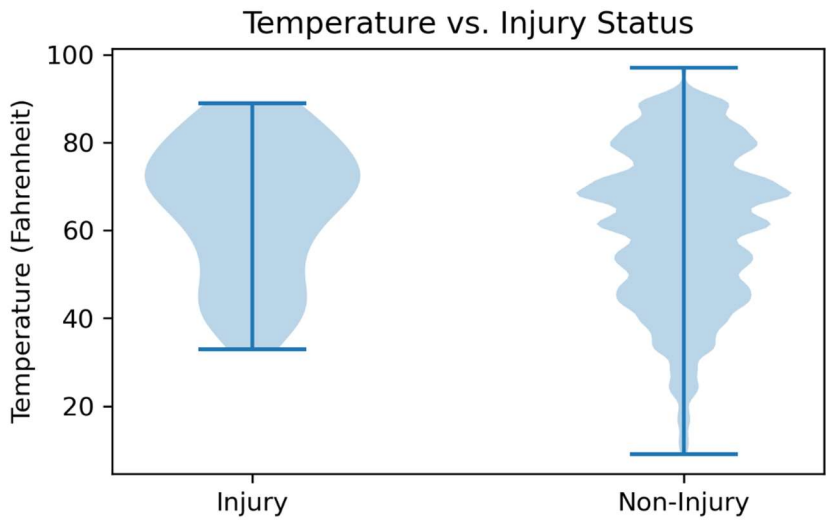

Figure 2. Violin plot displaying Injury distribution among various temperature ranges.

I also examined injury distribution among all athletes' game-specific play number. This represents the current play number that a specific athlete has participated in during one game. There was a clear correlation here, with higher injuries occurring during earlier game plays; perhaps this could represent the "warming up effect," the sudden muscle shift from static to dynamic motion early in the game.

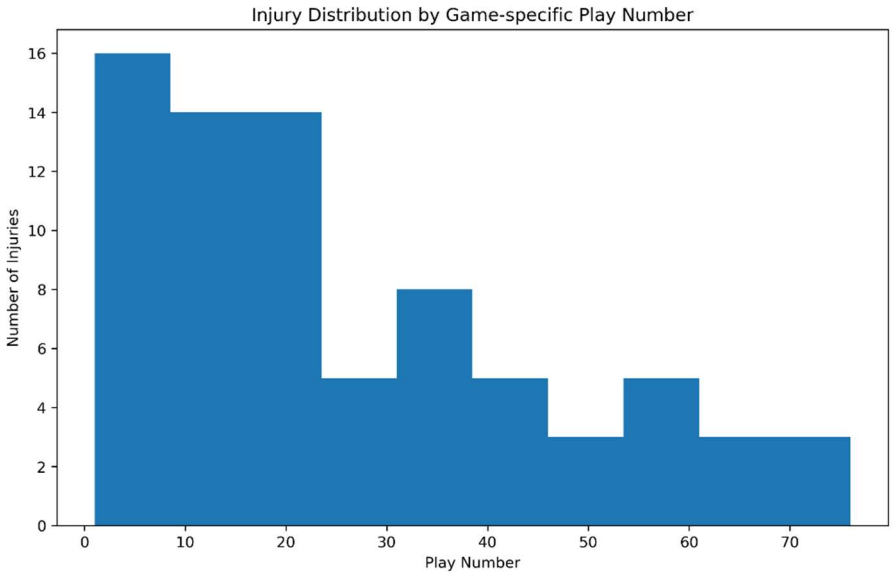

Figure 3. Histogram displaying injury distribution among athletes' game-specific play number

This EDA showcased several interesting trends that can be explored further and validated/discredited with algorithm results.

Methods

Because this dataset represents hundreds of thousands of plays among a mere 250 players, this data is clearly *not* identically independently distributed (i.i.d.). There are multiple plays from the same player(s) throughout, which introduces dependencies between the data points. Because of this, I used **GroupShuffleSplit** to split the data based on PlayerID, ensuring that all of the plays from one player remain in one of the three sets: train, validation, and test.

I then standardized features by conglomerating synonyms of the same category of each feature. For example, were many redundant names to represent the same thing across many categorical features ("Closed Dome," "Dome," "Dome, closed," etc.). Therefore, I combined all like synonyms for future algorithm efficiency and correctness.

Because there is a mixture of numerical and categorical features, I used a **StandardScaler** and **OneHotEncoder**, respectively, which resulted in 66 features. I also used a **SimpleImputer** to fill missing values. Imputation can have its issues, but there were few missing values across many rows, so I decided that this circumstance warranted it. I then fit and transformed the training set and transformed the validation and test sets.

When choosing an evaluation metric, I examined my dataset's nuances and algorithm purposes. Because my dataset is extremely imbalanced, there will be many True Negatives (TNs); therefore, I do not want a metric that relies on TN, such as accuracy. My goal is injury prevention at all costs, even at the risk of being overly cautious. This means that I consider False Negatives (FNs) worse than False Positives (FPs), so I chose **recall** as the appropriate metric.

The imbalance in my dataset demanded algorithms that have **class_weight** or equivalent parameter to handle it. I used logistic regression, random forest classification, support vector machine (SVM) classification, and XGBoost. I developed three helper functions to use within all four algorithms: model evaluation and confusion matrix display, cross-validation hyperparameter printing, and pickle dump model saving.

For each model, my pipeline consisted of model training and 3- or 5-fold hyperparameter tuning, depending on complexity, runtime constraints, and model-specific number of hyperparameters to tune. This was done via **GridSearchCV** to loop through all hyperparameter combinations and collect the results. The best model was deemed the one with the specific hyperparameters that contributed to the best validation mean recall score. I then took that model and applied it to the test set to measure generalization error, the model's performance (test recall) on previously unseen data.

For each model, I used the **class_weight = balanced** model parameter in attempt to handle some dataset imbalance, as well as **StratifiedKFold**. I appropriately tuned parameters by expanding and/or shrinking value ranges depending on the previous run's best parameters; when a model's best hyperparameter value was near the end of the provided range, I expanded it in that direction.

For logistic regression, I then tuned the "C" parameter on [0.01, 0.1, 1, 10, 100] to cover a wide range of possible parameters with L2 regularization and the "lbfgs" solver. For random forest, I tuned "n_estimators", "max_depth", "min_samples_split", "min_samples_leaf", and "class_weight' on values [5, 10, 20], [1, 2, 5], [1, 2], and ['balanced', {0: 1, 1: 5}, {0: 1, 1: 10}], respectively. For SVM Classification I tuned "gamma" on [1e-3, 1e-2, 1e-1, 1e0, 1e1] and "C" on [1e-2, 1e-1, 1e0, 1e1, 1e2]. Additionally, for this model I had to subsample 10% of the data

because anything more took well over a day to run, with limited result improvement. Finally, for XGBoost, I tuned “max_depth”, “learning_rate”, and “n_estimators” on [3, 5, 10], [0.1, 0.3], and [50, 100, 200], respectively. Furthermore, this model used a slightly difference **class_weight**-esque parameter to handle dataset imbalance: I used **scale_pos_weight** to balance the loss function and prevent overwhelming influence of Class 0, tuned to the proportion of Class 0 to Class 1 in the training set.

In addition to properly tuning hyperparameters, I addressed uncertainty due to splitting variability by calculation of recall mean and standard deviation across cross-validation loops, ensuring that multiple splits would be considered. This adds to the robustness of the final score. Furthermore, I addressed uncertainties due to non-deterministic models, such as Random Forest and XGBoost, via random seed fixing and **StratifiedKFold** usage, ensuring consistent evaluation across multiple folds.

Results

I constructed a DummyClassifier to naively predict all points to be of the majority class, Class 0, which resulted in a baseline recall of 0. This is extremely low, but because of the dataset’s extreme imbalance, I knew that model performance, even when testing for recall, would be dismal at best. This shows that even with recall values 0.7000 and below, some algorithms can be better than baseline. The test, mean, and standard deviation recalls of all machine learning algorithms are displayed below.

Table 1. Machine Learning Algorithm Results

	Test Recall	**Mean Recall**	**St. Dev. Recall**
Logistic Regression	0.3500	0.2246	0.0494
Random Forest	0.3500	0.2836	0.0461
SVM Classifier	0.0000	0.5000	0.4082
XGBoost	0.7000	0.4958	0.1424

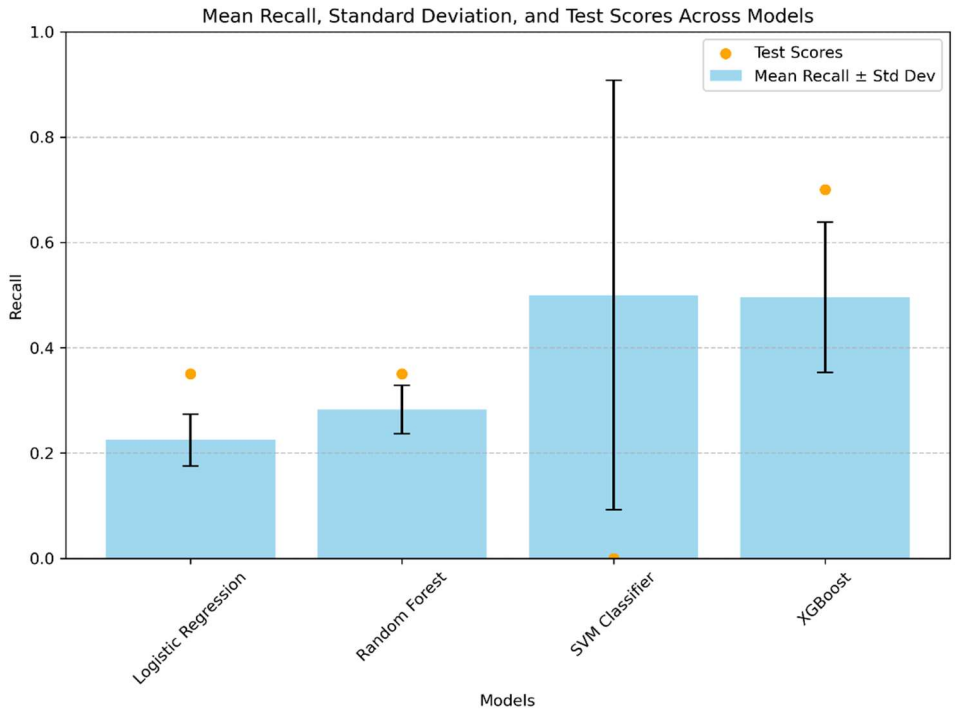

Figure 4. Mean and standard deviation recalls compared to test recall for four machine learning algorithms.

Logistic regression and random forest classification had similar results, with test recalls of 0.3500, both about 4.5 standard deviations above zero. SVM Classifier performance was extremely poor, with a test recall of 0.0000, the baseline recall; however, it is interesting to see that mean recall and standard deviation recall are both roughly 0.5000; this indicates that

throughout all cross-validation folds, this classifier predicted either all injuries correctly or all incorrectly, indicating the struggle the model had when maximizing the margin.

The best model was XGBoost, with a test recall of 0.7000, nearly 5 standard deviations above recall. This makes sense because this model has more advanced built-in handing of imbalanced datasets than other models, with **scale_pos_weight**, large dataset efficiency with parallelization, and is robust to overfitting with **max_depth** and **early_stopping** parameters. It is important to note, however, that prioritizing this algorithm for recall breeds a high incidence of FPs, which is poor performance for anything outside of recall.

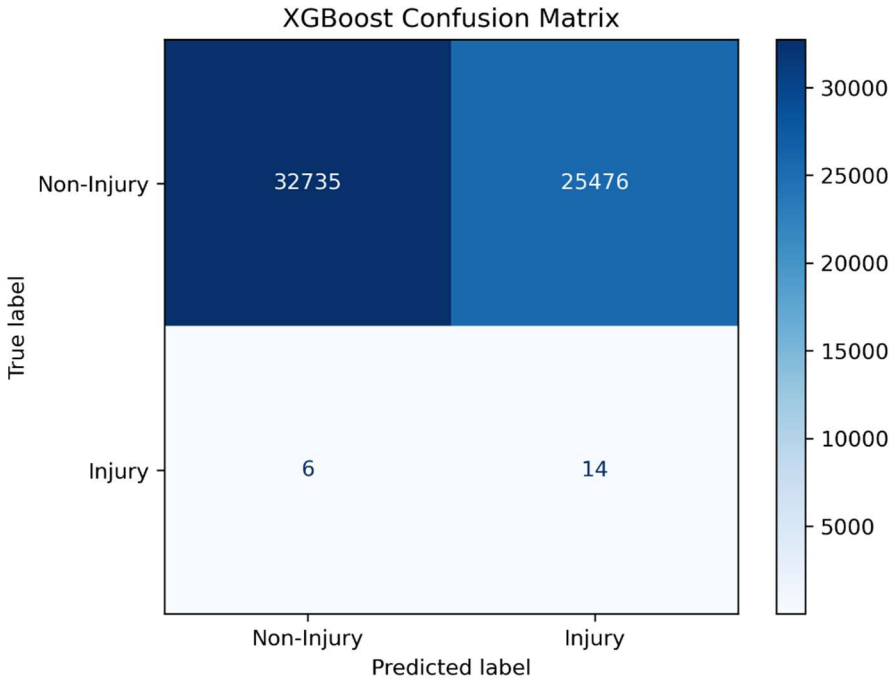

Figure 5. Confusion matrix for XGBoost model performance.

Because of the dataset's extreme imbalance, I knew that model performance, even recall, would be dismal at best. Even though my values proved to be above baseline, they were not great

compared to general industry deployed model standards. While specific test recall values might not be promising, feature importance could be.

I created a SHAP global bar plot and global summary plot to analyze global feature importances.

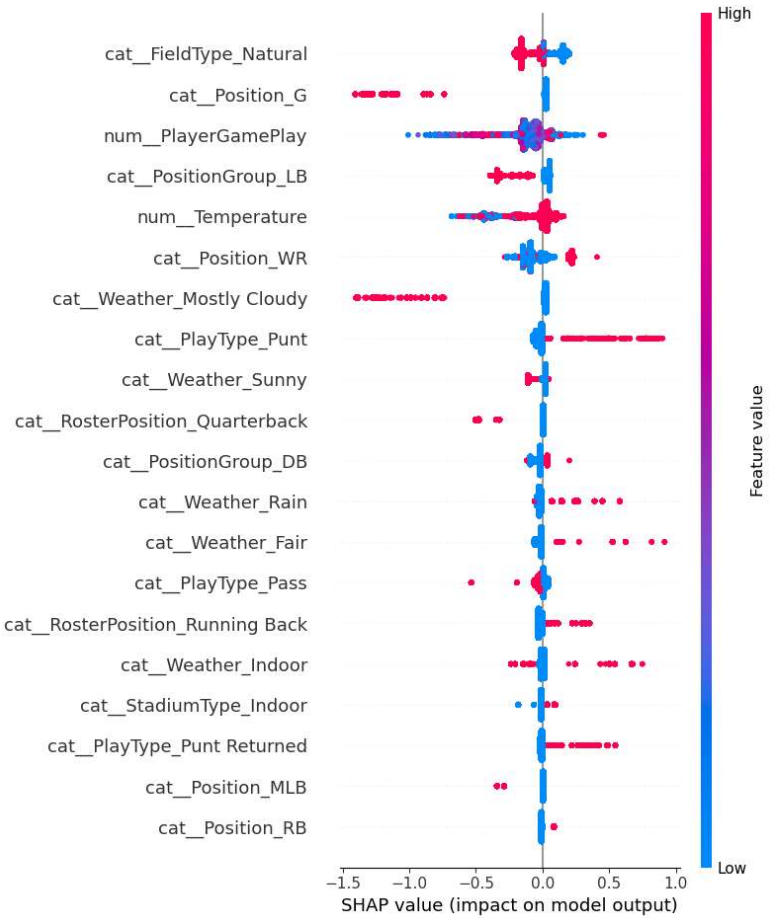

Figure 6. SHAP global summary plot

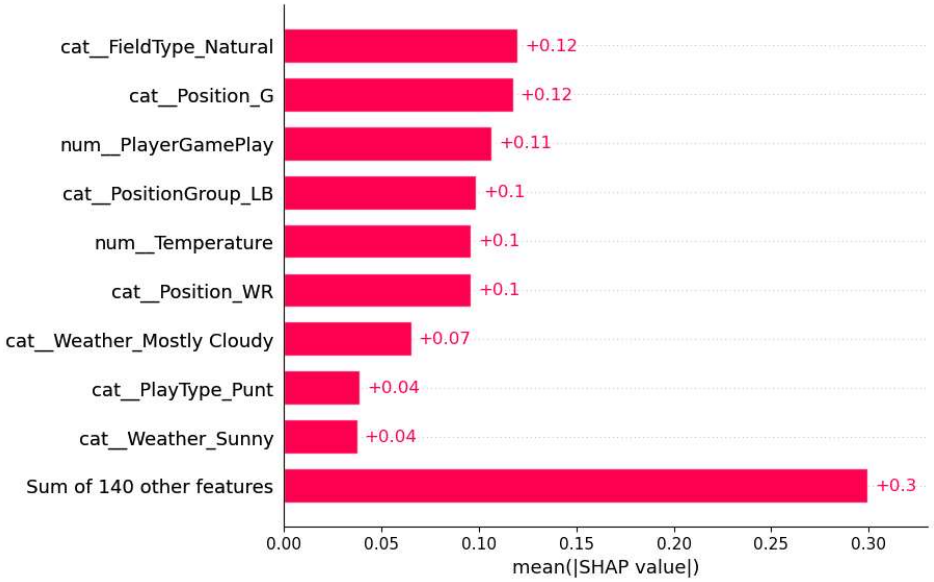

Figure 7. SHAP global bar plot

Interesting features include categorical **FieldType_Natural** with a value of +0.12, the most important feature. When viewing the summary plot, low values (category = 0) indicate non-natural (i.e., artificial) grass as a predisposition to injury, while high values (category = 1) indicate natural grass as a predisposition to non-injury. This echoes player sentiments of artificial turf catching their cleats and increasing impact stress [5], which could feasibly increase injury risk.

Another interesting feature is categorical **PlayType_Punt** with a value of +0.04. The summary plot shows that high values (category = 1) indicate punt plays as a predisposition to injury. Contextually, this makes sense because football punts are high impact plays, with a transition of offensive power that involves all players on both teams running straight into each other.

A third interesting feature is continuous **PlayerGamePlay** with a value of +0.11. While the summary plot is a bit jumbled due to the amount of data points for this feature, there appears to be more low values that predispose athletes to injury. This reiterates the "warming up effect" explored in EDA, which feasibly makes sense: early game plays involve a sudden shift from static to dynamic movement, which could result in improperly warmed-up muscles being exposed to sudden high loads they cannot handle, in contrast to later game plays where muscles have been continuously used and are properly warmed.

Local feature importance reiterates these trends, and shows more specifically which features have a positive or negative effect on injury predisposition.

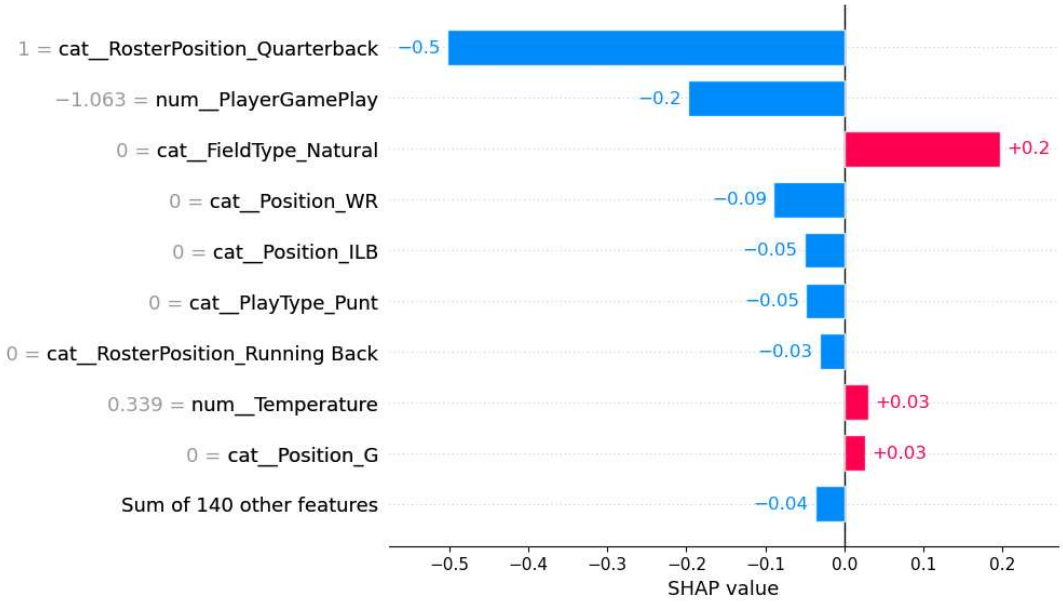

Figure 8. SHAP local feature important (feature = 7) bar plot.

Outlook

If I had more time and computing power, I would explore how SVM Classifier results change (i.e., if I could get the results above baseline) with more computing power and time to run the full dataset, not only 10% subsampling. Furthermore, though over- and under-sampling

can have their issues, it would be interesting to see how a combination of SMOTE and/or ADASYN and/or majority class under-sampling would affect the results. While the existing models have some innate ability to handle class imbalanced, this dataset was too extreme; perhaps more forceful methods such as SMOTE would be warranted to achieve better analyses.

Overall, while my choice of extremely large and wildly imbalanced dataset all but guaranteed poor test results, many algorithms were above baseline recall, and feature importances have biological and football-specific plausibility regarding injury predisposition.

References

1. Jones, F.M. (2024, Feb 7). Football Retains Dominant Position as Favorite U.S. Sport. *Gallup*. https://news.gallup.com/poll/610046/football-retains-dominant-position-favorite-sport.aspx
2. Carter EA, Westerman BJ, Hunting KL. Risk of injury in basketball, football, and soccer players, ages 15 years and older, 2003-2007. *J Athl Train*. 2011 Sep-Oct;46(5):484-8. doi: 10.4085/1062-6050-46.5.484. PMID: 22488135; PMCID: PMC3418954.
3. Walker, T.M., Fenn, L. (2020, Jan 28). AP analysis: NFL teams lost over $500M to injuries in 2019. *AP Newss.* https://apnews.com/article/9c8cac02c23272f4c147e49e758b1b98
4. Prescod, P. (n.d.). NFL Players' Injuries Aren't the Only Terrible Aspect of Their Working Conditions. *Jacobin.* https://jacobin.com/2023/03/nfl-players-union-report-conditions-owners-wealth
5. Wells, A. (2022, Nov 12). Cooper Kupp, More NFL Players Advocate for Grass Fields over Turf amid Safety Debate. *Bleacher Report.* https://bleacherreport.com/articles/10055511-cooper-kupp-more-nfl-players-advocate-for-grass-fields-over-turf-amid-safety-debate