Injury Prediction for Distance Running Athletes

Selina (Xinyue) Wang
Brown University Data Science Initiative
Final report
Dec. 07 2022

https://github.com/selinawaang/Data1030-Project

Introduction:

The goal of this project is to predict whether an athlete will get injured on a given day based on their recent training history. The motivation behind this project is that sports teams want to prevent injury of their athletes, so it will be useful to know which factors contribute to injury, and signal athletes to take precautions to prevent injury.

The dataset used for this project is the replication data for the research paper "Replication Data for: Injury Prediction in Competitive Runners with Machine Learning" by Lovdal et al. from the Bernoulli Institute of the University of Groningen. It includes data from 74 distance runners recorded over a period of 7 years. In total it has 72 features and 42,766 rows. The feature variables consist of the training log of the past 7 days leading up to the event day. This is a classification problem with the target variable being a binary variable indicating injury or no injury.

Because each athlete's data will theoretically have a different distribution, to simplify this project, I only used data from one athlete to train a model that would predict injury for that single athlete, this gives a total of 730 datapoints.

One study that uses this dataset is the original study by Lovdal et al. They used a bagged XGBoost model and achieved an average value of 0.724 for the area under the ROC curve.

Exploratory Data Analysis:

Choosing an athlete:

Athlete 29 was chosen because they have the highest fraction of injury out of events attended, meaning that they are the athlete most likely to get injured. The scatter plot below shows the relationship between the fraction of injury and the number of events attended by each athlete.

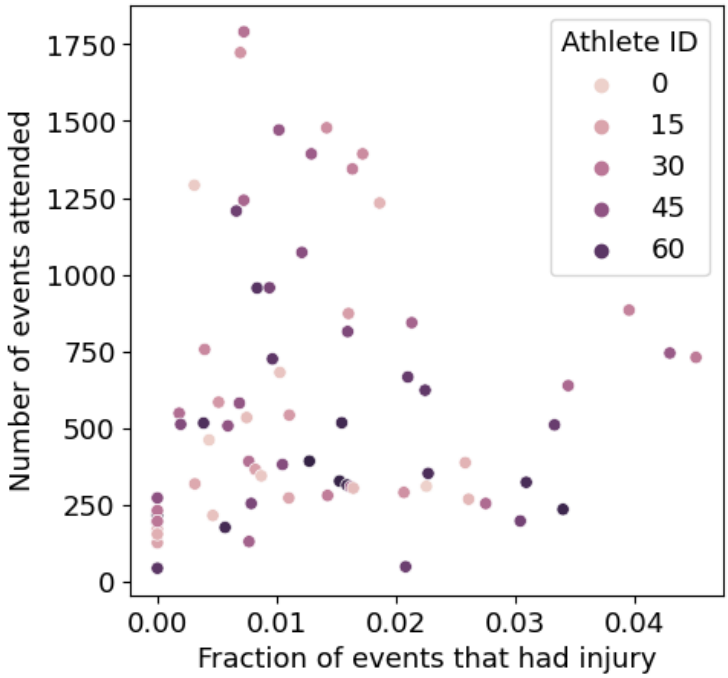

Figure 1 injury and events attended for each athlete

Target Variable:

The balance of the injury class is only around 5%, making this an imbalanced dataset.

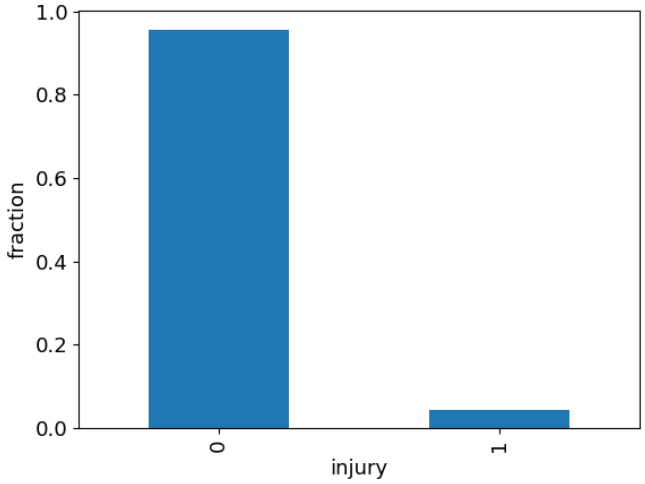

Figure 2 fraction of datapoints in each class

Feature Variables:

All of the features are numerical. The features measuring qualitative values (perception) are bounded between 0 and 1. The quantitative features are bounded below by zero but do not have a clear upper bound and have a long tail.

The figure below is the histogram for the number of kilometers ran on the day right before the event. It shows two important characteristics of the distribution of this variable: there is a large number of zeros, and there is a long tail. The long tail suggests that a standard scaler may be useful in the preprocessing step. A similar distribution is observed in almost all the variables related to training distance.

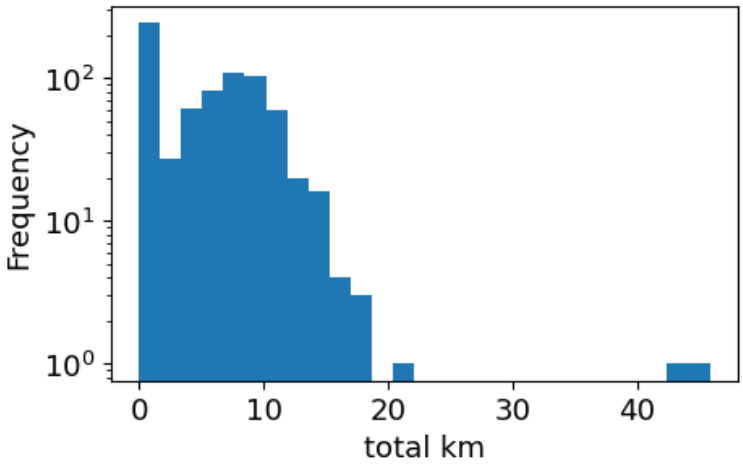

Figure 3 distribution of total km ran in training the day before event date

Box plots were used to visualize the relationship between quantitative features and injury, but not much insight was gained from these. However, the athlete's own perception seems to be

more strongly correlated with injury. The figure below shows that an athlete's perception of stronger exertion is positively related to injury.

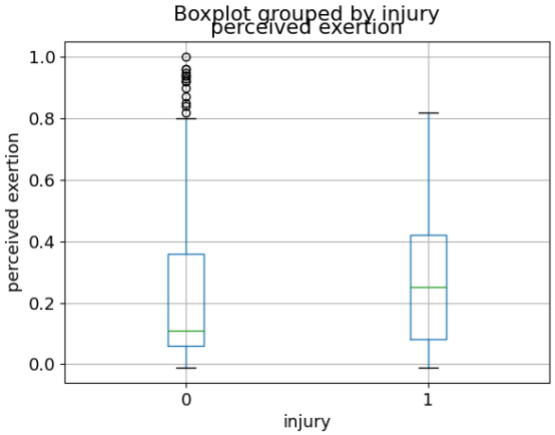

Figure 4 perceived exertion the day before event day for each class

Methods:

Feature engineering:
The features contain training data for the past 7 days. I added the average and maximum values for the past 7 days as additional features to the dataset. Minimum values over the last 7 days are 0 for all features, so those were not included in the dataset. I also added a lag feature of the target variable, this feature indicates whether the athlete was injured on the previous event day.

Splitting:
The events in my dataset happen in chronological order, so my training, validation, and test sets were also kept in chronological order to avoid information leakage. Because most of the ML algorithms I tried were deterministic, and time-series splitting is also deterministic, I decided to split my data into 6 folds so I can test the model performance on different test sets to account for uncertainty.

For the first iteration, I set the earliest fold as the training set, the second fold as the validation set, and the third fold as the testing set. And then for the next iteration, the training set becomes a superset of the previous training and validation sets, the validation set is the previous test set, and the test set is the forth fold, and so on. Since my dataset is imbalanced, and the class 1 datapoints are not evenly distributed over time, I designed the folds so that each fold contains approximately 5 class 1 datapoints. However, this also means that the folds are not equal in size. The figure below illustrates my splitting method.

Using this splitting approach, each algorithm will be trained 4 times on different training and validation sets, and will give 4 different models and 4 different test scores.

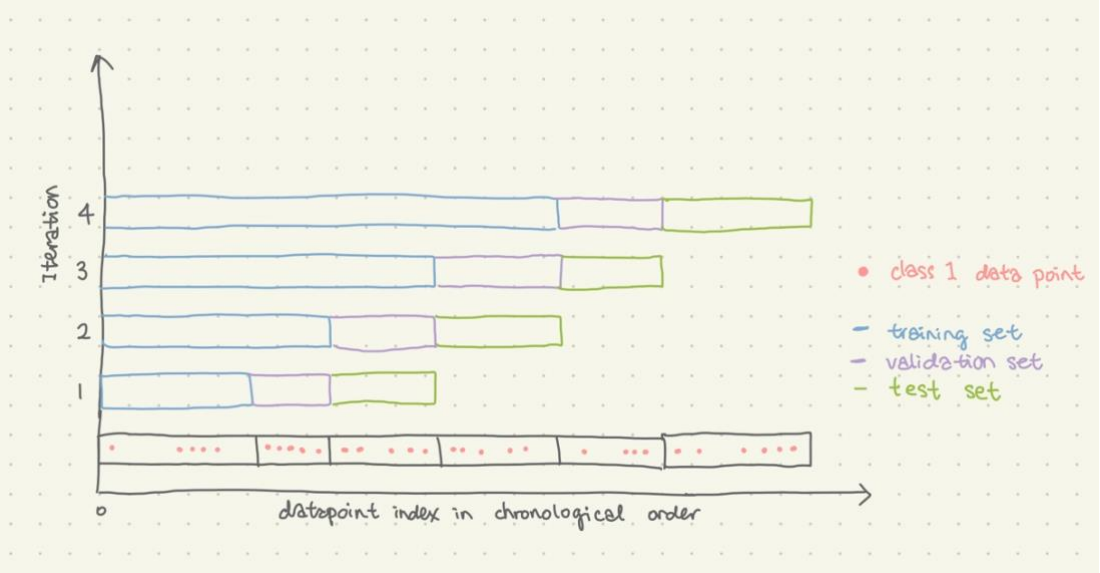

Figure 5 splitting strategy

Preprocessing:
For preprocessing, I used the standard scalar for all the variables because they can all be treated as continuous variables, and most of them do not have a clear upper bound and have a long tail. The lag of the target variable was also transformed using the standard scalar to make it easier to interpret the coefficients of my linear model.

Evaluation metric:
I chose the f_2 score as my evaluation metric for two reasons. First, since my dataset is highly imbalanced, the f-score is appropriate as it does not involve True Negatives in its calculation. Second, in this scenario there is not much harm in having a high False Positive value (falsely classifying an event as having injury) because that would only lead to athletes taking better precautions. This means I would like to capture a large proportion of the conditional positive samples, so I chose a high beta value of 2 which would put more weight on recall.

ML models and parameters:
The four models I tried are logistic regression, SVC, random forest classifier and k nearest neighbors classifier.

The table below summarizes the parameters I tuned for each model and the values I chose.

Model:	**Parameters:**
Logistic Regression	'C': [0.0001,0.001,0.1,1] 'penalty':['l1','l2']
SVC	'gamma': [1e-3, 1e-1, 1e1, 1e3, 1e5], 'C': [1e-2, 1e-1, 1e0, 1e1]
Random Forest Classifier	'max_depth': [1, 3, 5, 10, 20] 'max_features': [0.25, 0.5,0.75,1.0] 'max_samples':[0.25,0.5,0.75,1.0]
KNN	'n_neighbors': [1,3,5,10,15,20] 'weights': ['uniform','distance']

Figure 6 Parameter tuning

For the first three models, I set the class weight be “balanced” because I have a very imbalanced dataset and setting this parameter significantly improved the performance of all three models. To reduce overfitting, I included both l1 and l2 regularization in the logistic regression, and applied smaller values for the maximum depth of the random forest classifier.

Results:

Model performance:

The baseline f_2 score was calculated by setting all predictions to class 1. Results from the four models are summarized in the table and figure below.

Model	Test score mean	Test score standard deviation
Logistic Regression	0.506	0.284
Random Forest Classifier	0.469	0.070
SVC	0.460	0.239
KNN	0.262	0.170
Baseline	0.402	0.200

Figure 7 Model performance results

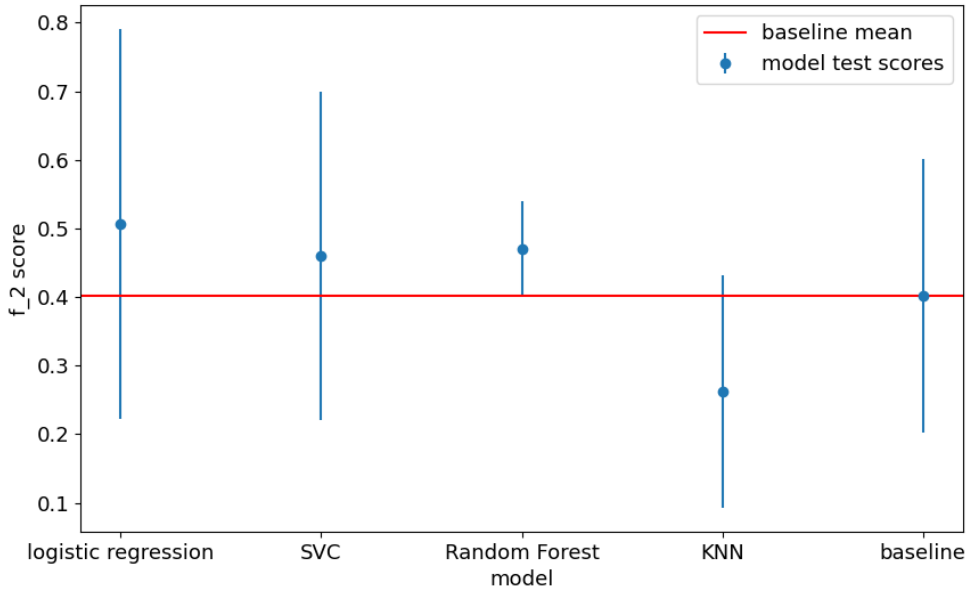

Figure 8 Model performances compared to baseline

The logistic regression model has the highest mean test score which is around 0.5 standard deviations, however it has a very large variance. In contrast, the random forest model has a slightly lower test score but significantly lower variance. So, I compared the test scores of these two models for each of the 4 iterations, and found that the logistic regression model consistently performed better than or at baseline, while the random forest model performed below baseline in the second iteration.

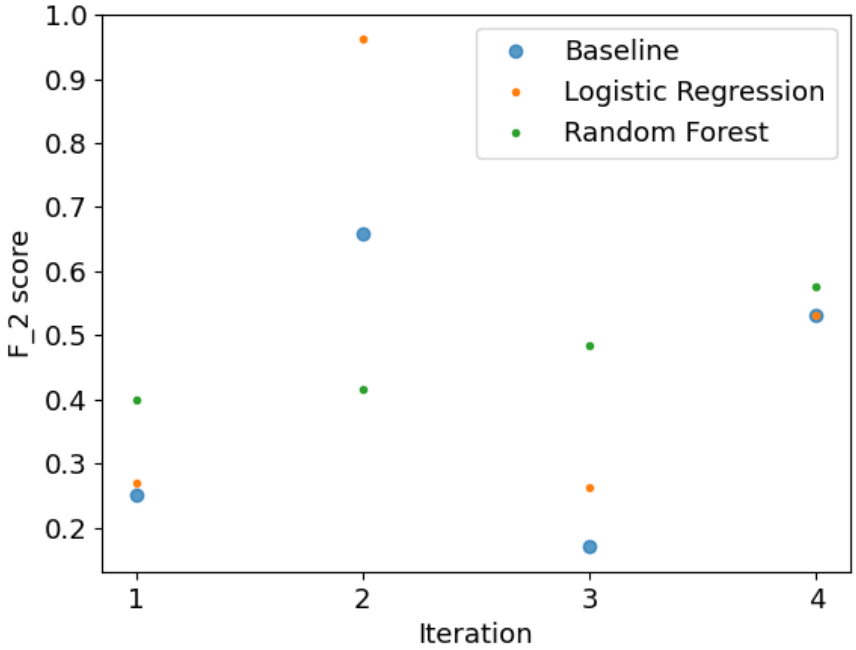

Figure 9 test scores comparison for logistic regression and random forest

Based on these findings, I decided to choose logistic regression as the best model.

I believe one of the reasons why logistic regression outperformed the other non-linear models is that because this dataset is very small, splitting the data introduces quite a bit of uncertainty so the distribution of datapoints in the test and training sets can be quite different. The more complex models all suffered from overfitting, which was hard to control even with parameter tuning, while a simple linear model performed better because it is less prone to overfitting. The unexpected result was that KNN performed worse than baseline, likely because I did not adjust the weight of the model.

Global feature Importance:

Out of the 4 logistic regression models generated, I picked the one with the largest f_2 score to examine its feature importance.

Global feature importance was calculated in three ways: permutation importance, size of the coefficients, and SHAP global importance. In each of these calculations, the lag feature of the target variable was consistently ranked as the most important feature, and its importance is significantly larger than other features, meaning that the model relied most heavily on whether the athlete was injured during the previous event day to predict whether the athlete is injured on a given day. This makes sense as many of the class 1 (injury class) points in the dataset occur on consecutive event days.

Other features such as ‘total km’, ‘maximum total km over the past 7 days’, and ‘maximum total km of sprinting over the past 7 days’ were all ranked as important features, which make intuitive sense. The one feature that was surprising was ‘perceived recovery’ which has high importance but a positive coefficient, which means that the model is more likely to predict injury if the athlete’s own perception of recovery is high. This suggests that perhaps the coach should not use the athlete’s own judgement of recovery as an indication of whether they will be injured.

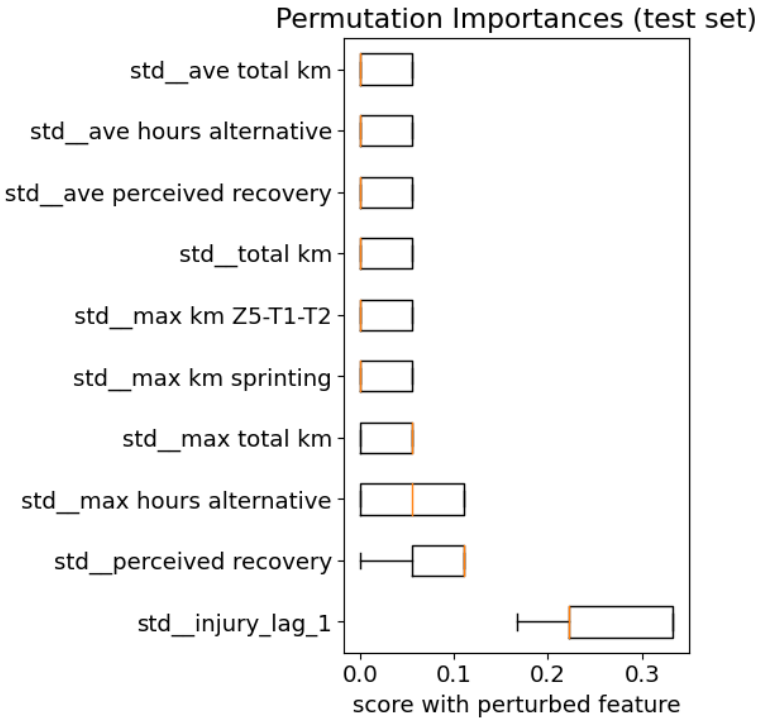

Figure 10 Permutation importances

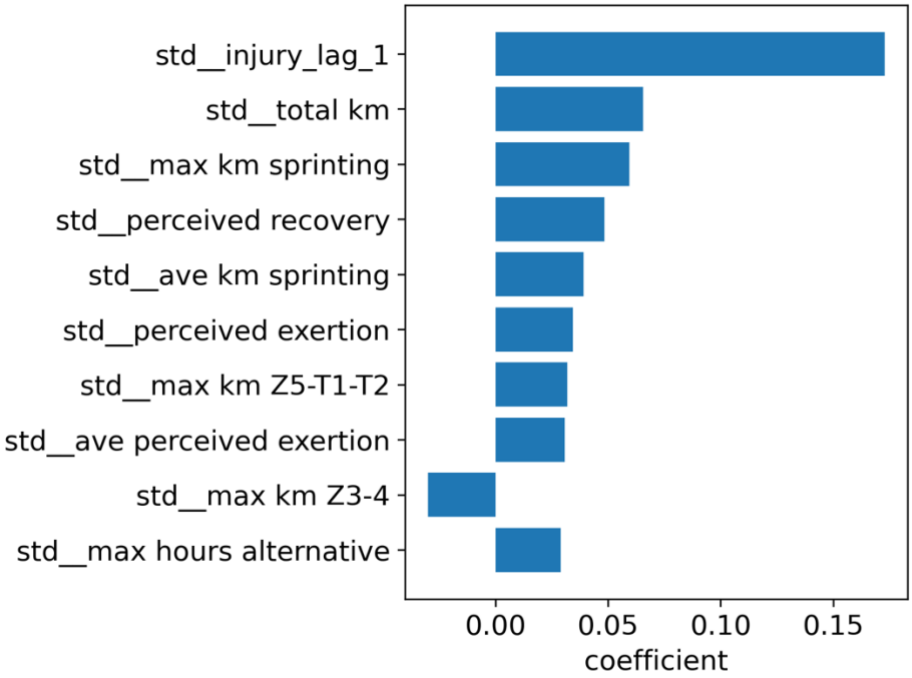

Figure 11 Coefficient size

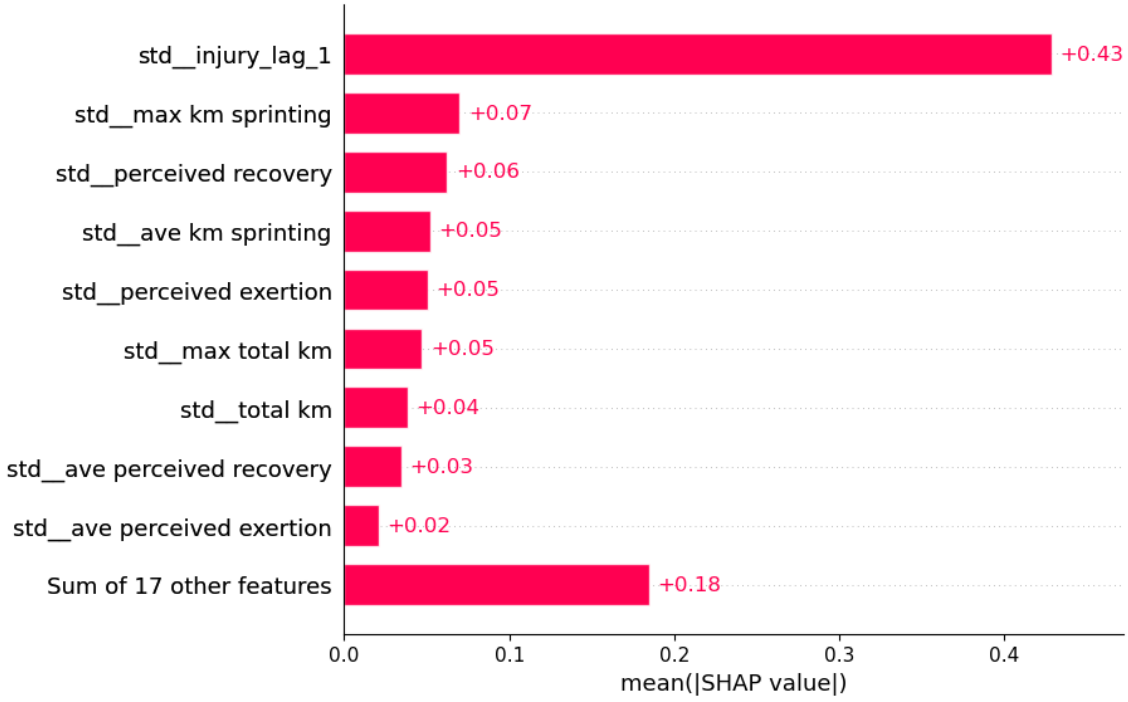

Figure 12 SHAP global feature importance

<u>Local feature importance:</u>

I examined 4 different datapoints (indexed 0, 1, 8 and 14) in the test set for local feature importance using SHAP values.

Observation 0 is a true positive and an injury was present on the previous event day, so not surprisingly the lag feature of the target variable that is pushing this datapoint towards being in class 1.

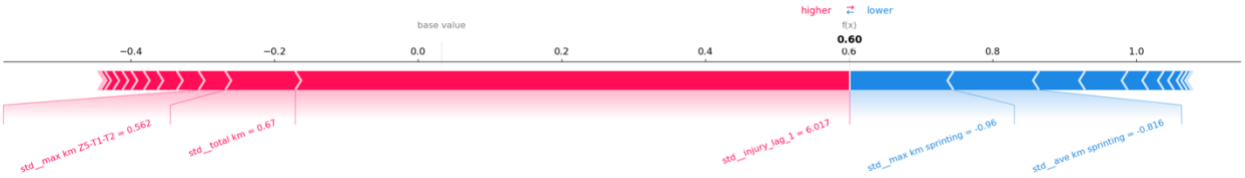

Figure 13 SHAP force plot for observation 0

Observation 14 is also a true positive, but there was no injury on the previous event day, so the model used other features like 'perceived recovery' in classifying this datapoint into class 1.

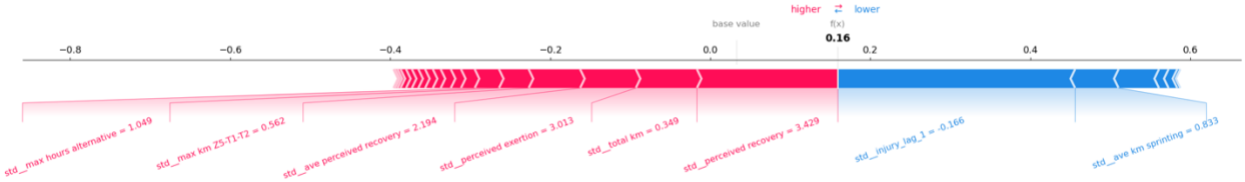

Figure 14 SHAP force plot for observation 14

Observation 8 was a true negative, and the lag feature of the target variable also played a most significant role in predicting the point as class 0.

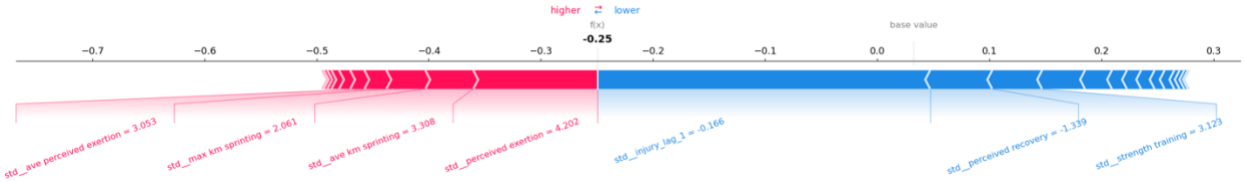

Figure 15 SHAP force plot for observation 8

Observation 1 was a false positive, and the prediction for this point was also largely due to the lag of the feature variable. From this observation we can see that the model seems to rely too much on the lag feature, which leads to misclassifying points into class 1 when they are preceded by an event day with injury.

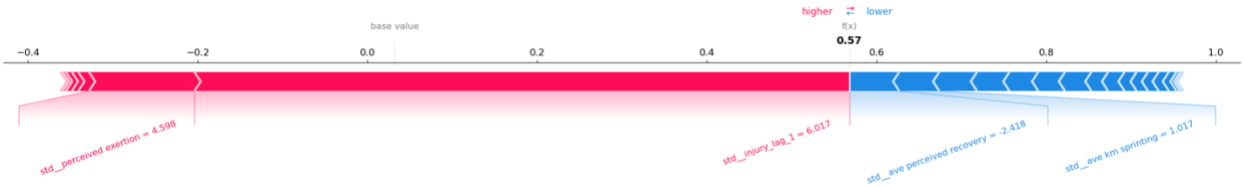

Figure 16 SHAP force plot for observation 1

Outlook:

The biggest drawback for this project is the small size of the dataset, so a natural next step is to experiment if using the entire dataset with 74 athletes can help my models capture more patterns and improve predictions for an individual athlete. It would also be worthwhile to train models for different athletes and compare model performance, perhaps diving deeper into the distribution of data for different athletes and why certain athletes are easier to predict than others.

Another improvement is to try out different algorithms, such as XGBoost, and experiment with other ways to deal with the imbalanced nature of the data set, such as adjusting the weight of the KNN algorithm.

References:

Lovdal, Sofie; den Hartigh, Ruud; Azzopardi, George, 2021, "Replication Data for: Injury Prediction In Competitive Runners With Machine Learning", https://doi.org/10.34894/UWU9PV, DataverseNL, V1

Lovdal, S., den Hartigh, R., & Azzopardi, G. (2021). Injury Prediction in Competitive Runners with Machine Learning. *International journal of sports physiology and performance*, *16*(10), 1522–1531. https://doi.org/10.1123/ijspp.2020-0518