Predicting Running Injuries Capstone

**Problem Statement and Subject Area**

The goal of this project is to create a machine learning model to predict if a runner is due for an injury, based on their prior training. This would be of value seeing as there are 500million runners in the United States alone, and of those 500million, 50% face an injury each year that pulls them away from running for a period (1, 2). Running injuries are clearly a fairly common problem that even the likes of professionals struggle to predict and avoid. Athletes often gauge whether to keep pushing based on their subjective judgements on how they feel. These judgements can be easily blurred by a hunger to win, but utilizing machine learning would remove these biases and help keep athletes safe. This machine learning model could be applicable to coaches and athletes from novice all the way up to professional levels, for monitor their likely hood of injury and knowing when to take the foot off the gas.

**Data acquisition, quality, and completeness**

The data was created for a study at the University of Groningen, in the Netherlands. It followed a team 74 of semi-professional track and field athletes running middle- long distances (800m up to the marathon), for 7 years, with the hopes of also creating a machine learning model to predict injury as well.

Overall, the data set was impressive in that we initially doubted the possibility of doing such a project because so many features would have to be included to attempt to predict injuries. However, this data set came with 73 features and had over 42,000 rows. The features showed 7 repeats of 10 features, representing metrics taken from the 7 days leading up to an injury or not. While this rolling count made it possible to factor the training from the 7 days prior into the injury prediction, it impossible to work with the data as if it were a true time series. The only other issue with the data was the occurrence of -0.01, which will be discussed in the *Preprocessing Exploration and Analysis* section. The only thing we would potentially add to the data set would be non running metrics, i.e. sleep or measures of activation and stretching prior and post exercise.

**Preprocessing Exploration and Analysis**

To prepare this data for analysis, we first checked for any null values or duplicates, of which we found none. Then all features were looked at in terms of distribution, unique features, descriptive statistics, and in comparison to the target (injury), to look for any preliminary trends. During this process we found the most predominant value in all the *perceived-* columns (exertion, success, recovery; for all 7 days), was -0.01. These features were meant to be on a 0- 1 scale (as was noted in the initial study), which added to the confusion. After further investigation the -0.01s were converted to 0 considering that it maintained a distribution similar to the majority of features, (having 0 as the predominant value).

During the data exploration a better understanding of the date feature was also established. While it was understood that it was a count of consecutive days for each athletes, it was confusing when the columns distribution showed periods with nearly no athletes, but an increase afterwards. It was realized that when an athlete was injured and took time off the count kept running which would explain why a continuous value like the Date could have a dip in it. This would explain how athletes could reach a Date over 2500, but would have no date record at days around 600. It also explains how we can have record of 74 athletes but have dates where no more than 35 are checked in at a time (see figure 1).

**Figure 1**: Date distribution

Chart, scatter chart

Description automatically generated

It was interesting to note that despite the mean injuries per athlete being 7, there were several athletes that remained un injured as well as athletes that encountered up to 35 injuries (see figure 2). This stemmed interest in creating a model that allowed inspection into which features influenced the odds of injury.

**Figure 2**: Injury vs Athlete

Chart

Description automatically generated

**Modeling**

Initially a logistic regression was selected seeing as the prediction of injury was a binary classification problem, and the use of a logistic regression would allow for the inspection of coefficients to better understand which features were having the greatest impact on the likelihood of being injured. However our results were not promising.

The initial untuned model had an accuracy score of 98% but it was revealed that this was due to the highly imbalanced data set (98% non-injured), the model was simply classifying all inputs as resulting in jury. The f1-score ended up being our scoring of choice, in that it utilized precision and recall. The confusion matrix of the initial logistic regression can be seen bellow in figure 3, where we can see the model labeling 0 data points as 1 (injured), ignoring the 100 injuries present in our training dataset.

**Chart

Description automatically generatedFigure 3**: Confusion matrix of LogReg 1

We went on to optimize the logistic regressions hyper parameters, up sample the injury class (several times) , and down sample the non-injured class, (several times), however the best f1score achieved was .04. Said score was achieved in our model with optimized hyper parameters and non-injured down sampled to only 2x the size of the injured class (732 non-injured, 366 injured). The confusion matrix from said model can be seen bellow in figure 4.

Chart, treemap chart

Description automatically generated**Figures 4**: Confusion matrix from LogReg with optimized hyper parameters and noninjuries down sampled from 27,000 to just 732.

Throughout all the previously mentioned logistic regressions we only saw marginal improvements in recall and hardly any improvement in the f1 score. It was concluded that the logistic regression may not be well suited enough for such an imbalanced data set, and we instead decided to utilize a different machine learning model.

The next model selected was the XGBoost model, as it is an ensemble method that fits each sub model (regression tress) off the residuals of the previous model in order to correct for misclassifications. Additionally, it has an array of hyperparameters to help compensate for class imbalances (i.e. subsample, and scale\_pos\_weight).

An initial baseline XGBoost model was ran which also defaulted to classifying everything as non-injured, and an f1 score of 0. Afterwards hyperparameters were manually tuned to establish initial ranges, then a grid search was run to find the optimal hyper parameters with the established ranges. Unfortunately, issues with re-running cells led to the deletion of prior attempts using gridsearches and up sampling to optimize our XGBoost model. Admittedly these attempts were fairly unsuccessful with the f1-scores remaining around 0.1, so it was no major loss.

Ultimately for the sake of time it was decided to utilize Randomized Search cross validation to optimize our hyperparameters. Our initial gird search with n\_iter set to tis default of 10, yielded our most accurate model yet, with an f1 score of .71, the model only misclassified 83 false positive injuries, but had 0 false negatives, which in the case of predicting injury would be acceptable as it is always better to air on the side of caution, (see figure 5 for the confusion matrix , parameters and classification report of said model.

Figure 5: confusion matrix and Classification Report of RandomCV optimized XGBoost

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedChart, treemap chart

Description automatically generated

**Findings and Conclusions**

Despite short comings on our test set, we temporarily had a model that could predict with a recall of 100% and precision of 55%, whether an athlete would get injured based on their previous 7 days of training.

It is how ever unfortunate that the logistic regression did not perform better, in that being able to inspect the coefficients would provide valuable insight on the features predominantly leading to injury.

Given more time we would further optimize the model using a grid search on a narrower range of values and would attempt to see if down/up sampling improved these scores further.

Future steps would be to apply a clustering algorithm to the data. During the data investigation, while comparing athletes to injury, it was noted that while the mean number of injury was 7 there were athletes that fell far above with up to 35 injuries, and athletes that fell far below, with no reported injuries (see figure 2). Despite lacking time to explore this option, it would be interesting to apply a clustering algorithm to potentially extract the groups of athletes based on injury occurrence and examine the traits that make up each. This would also be practical in the future for classifying new athletes as being injury prone or not prone.

Once complete, this model could also be expanded to include not only training metrics, but sleep metrics as well (based on something as simple as a whoop bad), or even nutritional information (i.e. something as simple as a column with values ranging from 0-1 and athletes score if they met expected servings for particular food groups), to more accurately narrow down the many variables that result in a runner’s injury.

Despite some technical setbacks, given more time, as well as a data set with more injuries, we believe we could optimize a machine learning model to accurately predict injuries and save millions of athletes the pain of season ending.

References

1. <https://www.statista.com/statistics/190303/running-participants-in-the-us-since-2006/#:~:text=Around%2049%20million%20people%20in,at%20least%20once%20in%202021>.
2. <https://www.livestrong.com/article/13730338-running-statistics/>