

Predicting Running Injuries with Machine Learning Models

Elgin Vuong

October 14, 2022

Abstract

Is it possible to predict running injuries with only a dataset and machine learning models? This paper explores this question by using classification models, including the Logistic Regression model and the Random Forest Classifier model. In the dataset used, ten features were taken into account when predicting running injuries. With slight modifications, the Weighted Logistic Regression and over and down-sampling Random Forest Classifier models were used to mitigate the imbalance in the dataset. The results suggested that the best model was Weighted Logistic Regression and that the best score metric to take into account was the F beta score.

1. Introduction

Injuries in sports are a significant deterrent to an athlete's success. In running, a single injury can sometimes be career-ending. In order to prevent injuries or minimize the amount of injuries that a runner has, we have attempted to solve the issue by using machine learning models to predict on a running injury dataset. Many factors go into determining an injury. These factors include the amount of running, different types of running workouts, the time spent running, type of shoe, terrain, stretching, and many more. Running injuries can be hard to predict because anything possible can happen that can cause an injury and sometimes these events are unforeseeable. This paper leverages a dataset of runners' workouts and associated injuries to learn a classifier that can be useful to predict running injuries.

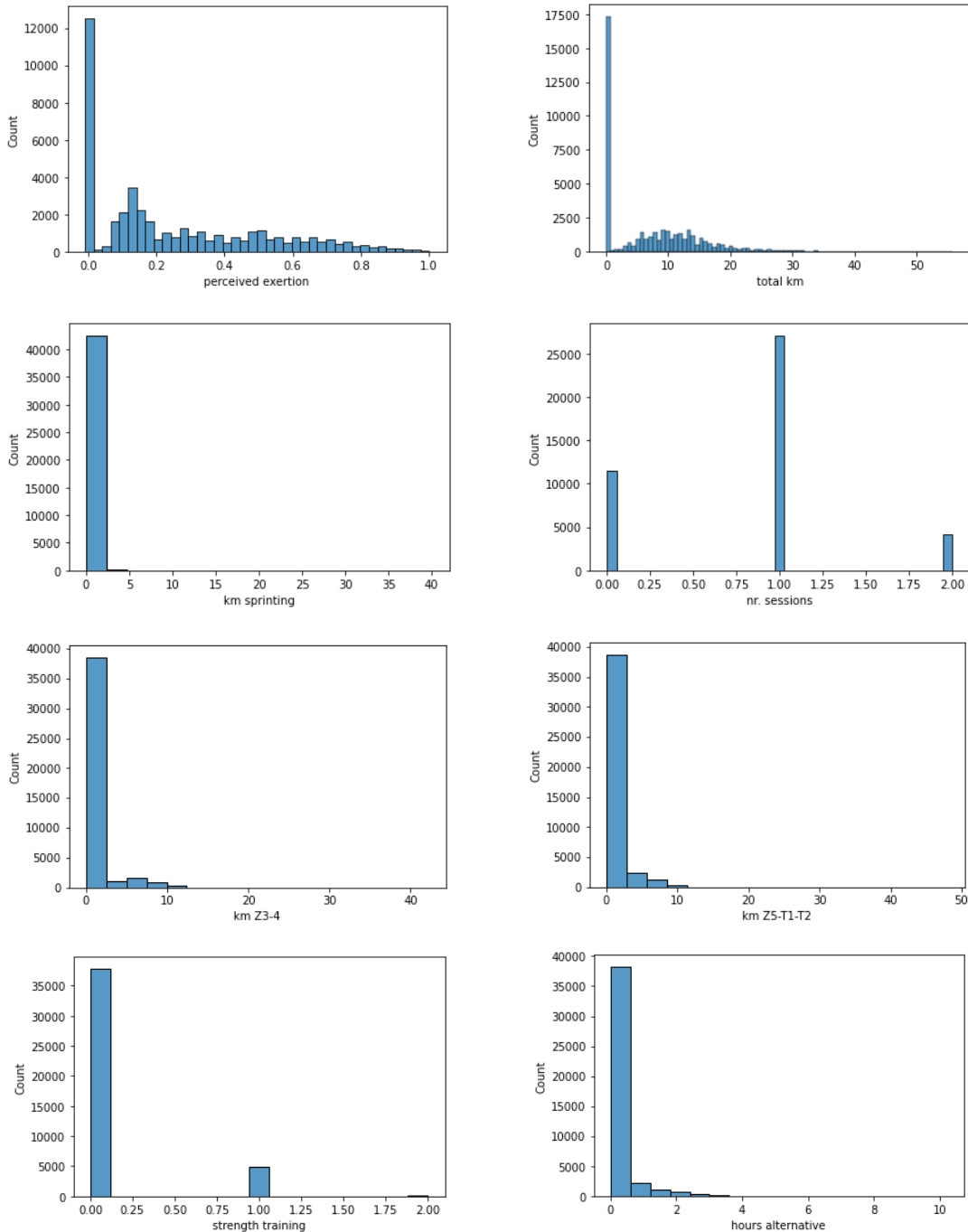
2. Background

As this is a sports-related machine learning model paper, specifically on running injuries, there have been few studies done on this topic. However, the paper from which the dataset was from, started some studies using machine learning models. They used another machine learning model, the XGBoost Classifier model. Their study was more extensive as they used two datasets, both weekly injury datasets, and daily injury datasets. The study yielded AUC scores that were decent but not yet 100%. They concluded that their model performed better on the daily injury dataset. They also indicated future research conducted on this dataset should try to improve the performance of the model.

3. Dataset

In this project, the data that we used was a running injury dataset from Kaggle, but was originally from the paper *Injury Prediction In Competitive Runners With Machine Learning*. In the paper, two datasets were used—a weekly injury dataset and a daily injury dataset. We decided to use the daily injury dataset because it was smaller and easier to work with. The dataset had 42766 samples and 73 columns. We cut down the number of columns to 71 because the extra columns did not affect the injury prediction. Of those columns, 70 are features and the last column is the injury prediction. The injury prediction is denoted with 1's and 0's; 1 represents an injury and 0 represents no injury. There are 70 features because every 10 features represent one

day of the week. We plotted the features in histograms to visualize the data (pictured below). With the dataset being very large, we ran into the problem of it being imbalanced. Over 98% of the dataset was for non-injury samples. After loading the data into a Google Colab notebook, we split the data into training and testing sets using the `train_test_split` function. Approximately two-thirds of the data was used for the training set and one-third was used for the testing set.



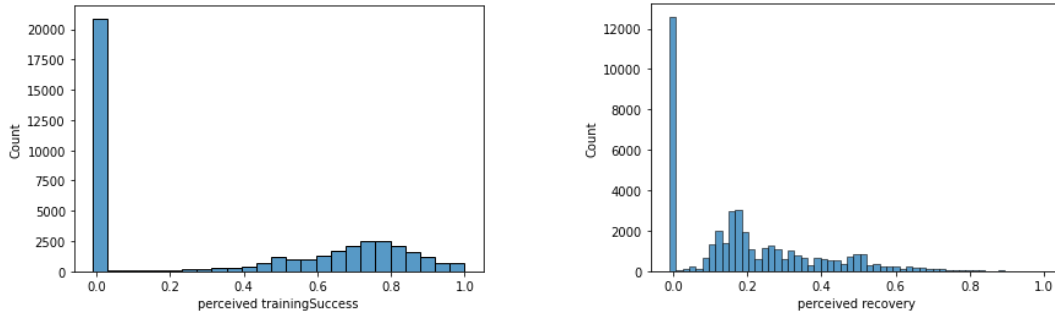


Figure 1: Features of the dataset

4. Methodology / Models

Determining whether or not a runner is injured or not is a classification problem so we used the Logistic Regression and the Random Forest Classifier baseline models. First, we fitted the model on the training set so that we could eventually compute against the testing set. Then to test the effectiveness of the models, we evaluated seven metrics (accuracy score, precision score, confusion matrix, area under the curve, recall score, F1 score, and F beta score). We decided that the most important metric would be the F beta score because it would be evaluated taking both the precision and recall scores into account. For this application, we decided false negatives were more costly than false positives, and thus set the beta value to 1.2. This metric generated a numerical value that helped visualize how accurate the testing set was at predicting injuries and the effectiveness of the classification model. However, due to the dataset being so big and imbalanced, we decided to research other models that were more effective at handling imbalanced datasets. One possible solution was to use a weighted logistic regression. With the weighted Logistic Regression model, we set weights for 1's and 0's that were predicted. Because there were way more 0's predicted (no injury), we gave more weight to the 1's in hopes of combatting the imbalance of the dataset. We then started hyperparameter tuning and toyed around with the ratio of weights to find the most optimal weight for this dataset that would generate the greatest F beta score. Another possible solution that we tried was to artificially augment our data by using the `RandomUnderSampler()` and `RandomOverSampler()` functions to oversample and undersample the data. Oversampling added data samples to the minority class in order to help balance the imbalanced dataset. Undersampling removed data samples in the majority class to help restore balance. We trained our training data on both the `RandomUnderSampler()` and `RandomOverSample()` functions. Then, we proceeded to implement the oversampled data and undersampled data into our Random Forest Classifier model. Once again, we evaluated the seven metrics to find which model would be the best to determine running injuries. Below are the figures for the models we tested.

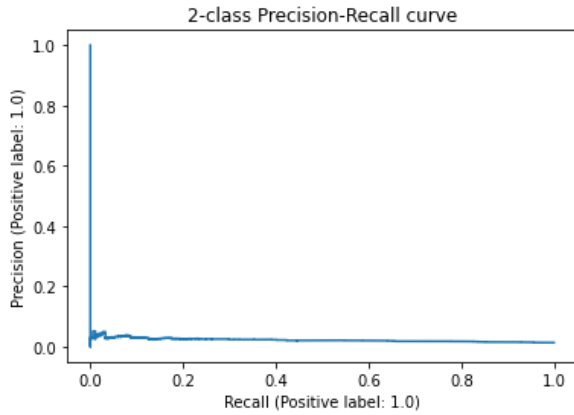


Figure 2: Baseline Logistic Regression Precision-Recall Curve

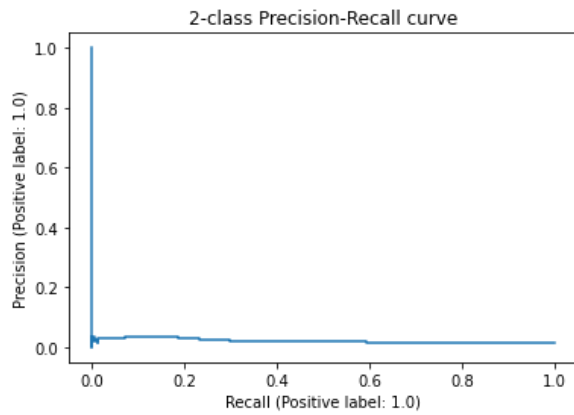


Figure 3: Baseline Random Forest Classifier Precision-Recall Curve

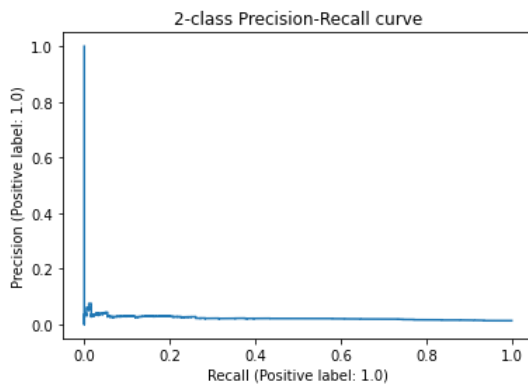


Figure 4: Weighted Logistic Regression Precision-Recall Curve

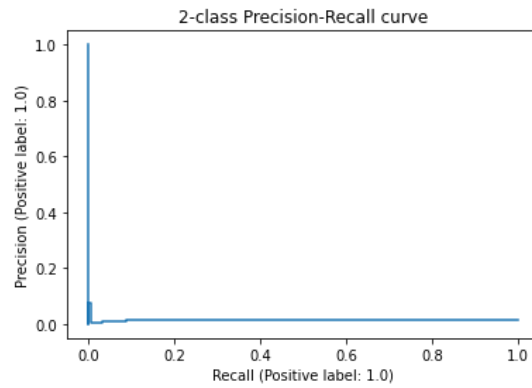


Figure 5: Oversampled Random Forest Classifier Precision-Recall Curve

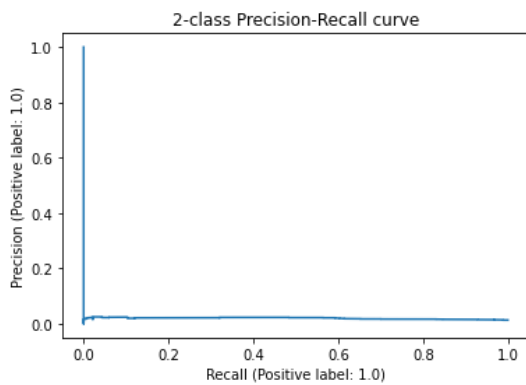


Figure 6: Undersampled Random Forest Precision-Recall Curve

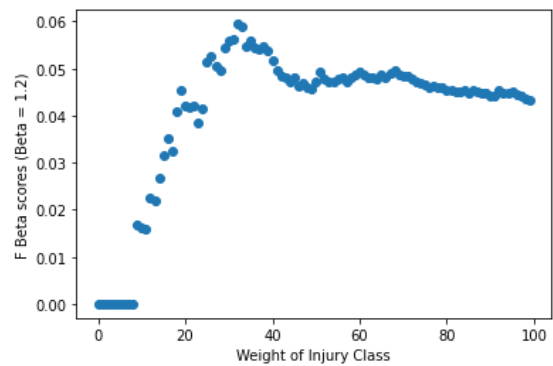


Figure 7: Hyperparameter Tuning: Weighted Logistic Regression Precision-Recall Curve

5. Results and Discussion

The results were relatively the same as the original baseline model, although they were slightly better. After doing a few hyperparameter tuning by changing the weight of 1's, we found that the ratio of 0's to 1's was best at 1:32 because it yielded the highest F beta score. However, we noticed that the precision and recall scores for each model we used were very low. Below is a table of precision, recall, and F beta scores for the models we tested.

Scores			
Model	Precision	Recall	F beta (F_β)
Baseline Log Reg	0	0	0
Weighted Log Reg	0.0297	0.1956	0.0594
Baseline RFC	0	0	0
Oversampled RFC	0	0	0
Undersampled RFC	0.0191	0.6141	0.0446

Table 1: Scores for each classification model

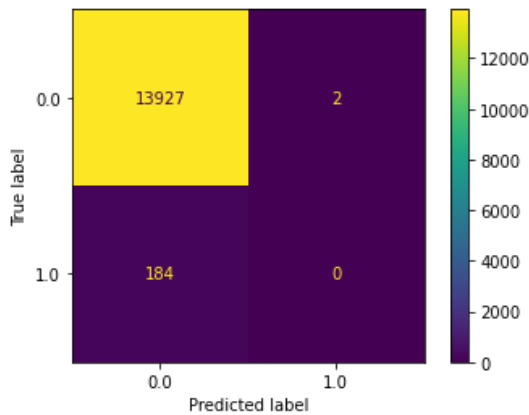


Figure 7: Baseline Logistic Regression Confusion Matrix

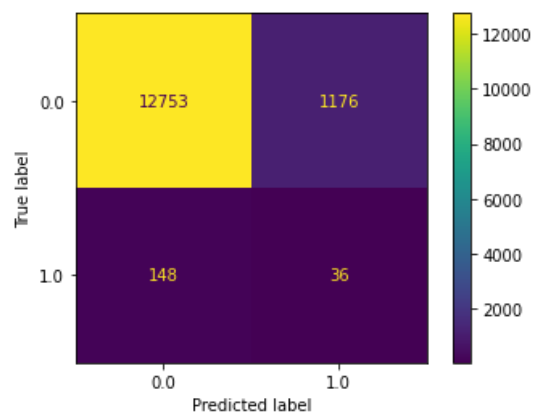


Figure 8: Weighted Logistic Regression Confusion Matrix

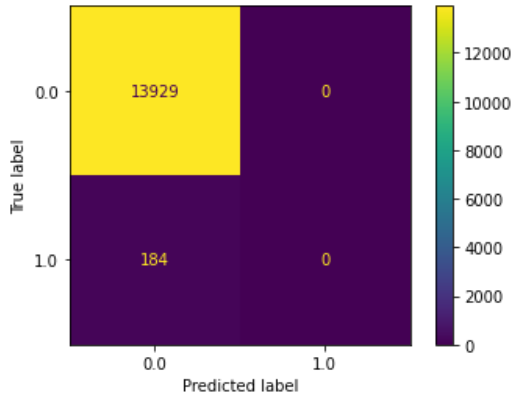


Figure 9: Baseline Random Forest Classifier Confusion Matrix

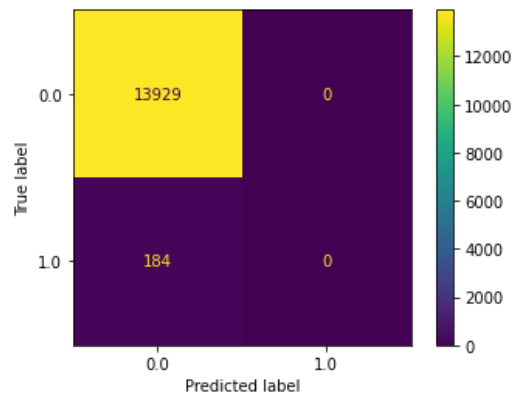


Figure 10: Oversampled Random Forest Classifier Confusion Matrix

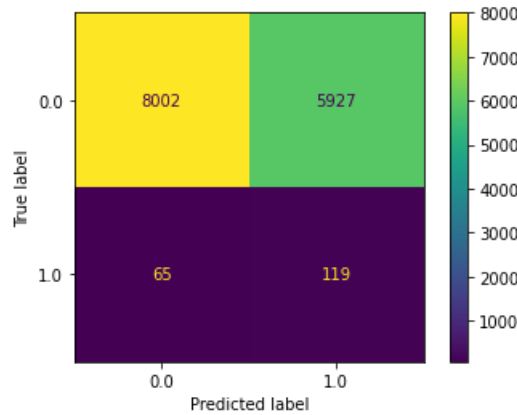


Figure 11: Undersampled Random Forest Classifier Confusion Matrix

Based on the table above, the best model was the weighted logistic regression model because it had the highest F beta score. The F beta score was the most important metric because we wanted to find how we could best strike a balance between false negatives (predicting someone with an injury as no injury) and false positives (predicting someone with no injury has an injury). Our choice of F beta score for the model metric also led to a ranking of model performance that we thought made sense based on the confusion matrices shown above. After these considerations, we found that the best model for this data is the Weighted Logistic Regression model.

6. Conclusions

This project has attempted to predict running injuries on a dataset that has been used before in a research paper. By using many different classification models on the dataset, we have furthered the study of predicting running injuries. Although, this dataset was very unique because it had many troublesome characteristics. For instance, it was a very large dataset with a large imbalance. As a result, we were only able to do so much work with this dataset. Our results

prove that there is more work that can be done on this running injury dataset to mitigate the imbalance in the dataset to predict the injuries more precisely. Our research has set a small milestone in attempting to predict injuries on this imbalanced dataset. Future researchers can build upon this paper by finding even more efficient machine-learning models to predict running injuries. In the future, I would like to continue finding better models for this dataset as well as work with other sports injury datasets that are more balanced to identify the similarities and differences with this research.

Acknowledgments

I would like to thank Joe Vincent for his help and guidance on my research journey.

References

- Chmait, N., & Westerbeek, H. (2021). Artificial Intelligence and Machine Learning in Sport Research: An Introduction for Non-data Scientists. *Frontiers in Sports and Active Living*, 3. <https://www.frontiersin.org/articles/10.3389/fspor.2021.682287>
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>
- F-beta score*. (n.d.). Hasty.Ai. Retrieved September 29, 2022, from <https://hasty.ai/docs/mp-wiki/metrics/f-beta-score>
- guest_blog. (2020, July 23). Imbalanced Classification | Handling Imbalanced Data using Python. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/>
- Iyer, S. R., & Sharda, R. (2009). Prediction of athletes performance using neural networks: An application in cricket team selection. *Expert Systems with Applications*, 36(3, Part 1), 5510–5522. <https://doi.org/10.1016/j.eswa.2008.06.088>
- 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/ijsspp.2020-0518>
- Maalouf, M., & Siddiqi, M. (2014). Weighted logistic regression for large-scale imbalanced and rare events data. *Knowledge-Based Systems*, 59, 142–148. <https://doi.org/10.1016/j.knosys.2014.01.012>
- Narkhede, S. (2021, June 15). *Understanding Confusion Matrix*. Medium. <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>