CSCE 822 Course Project Proposal

Kevon Smith
University of South Carolina
Kevon@email.sc.edu

Abstract

In this paper, we would like to explore the applications of ML to sports general management by focusing on the margin of victory and outcomes of NFL games. We will use Neural Networks as well as recurrent models for this task, and manage to achieve results that would best fit the model. Training data would comprise game-by-gate data from 2007 to the latest season in 2021, containing a variety of offensive and defensive statistics.

1. *Introduction*

Sports are an integral part of today's mainstream media and society, especially nowadays with the pandemic still going on. They are a universal language that breaks many barriers. Whether they are your job, way of life, or just favorite form of entertainment, sports can always bring different people together. The technological advancements that have been made within the sports world is amazing. From wearable trackers in the balls to motion cameras for better accuracy within the field of play, sports data are now being rapidly generated and shared.

In this project, we will explore the applications of Machine Learning in the field of general management within sports, using a case study of the National Football League. More specifically, we wish to design sets of models that predict the outcomes and margins of victory for NFL matches based on in-game statistics. This model could also be used in predicting betting odds based on in-game performance, but that is for another project. Ideally, we would be able to come up with an estimate for this indicator that is more accurate than that of some betting platforms, and use this to our advantage to place bets.

2. Dataset

Our dataset was classified into three groups: weekly data, seasonal data, and play-by-play data, which consists of both data describing team performance and player performance.

For game data, we retrieved NFL data sourced nflfastR, nfldata, dynastyprocess, and Draft Scoutfrom using cooperdff's Python NFL library [1]. We made minor edits to the scraper in order to account for formatting inconsistencies in the data. This library provided a strong base to start collecting data on every team for every game since the 1999 NFL season. The data collected for every game included statistics such as weekly data, seasonal data, win totals, combined results, etc for all teams.

3. Algorithm

We will be using 3 main methods/steps for predicting the outcomes for individual games, and the season as a whole. Python libraries such as Scikit-learn [2], Surprise [3], and PyTorch [4] will be used for Random-Forest, SVD, and LSTM w/ also Neural Net, respectively.

We would also like to minimize the MSE when training our models. Would only train our models on a total of 5 seasons (2007, 2019, 2012, 2015, 2016), use the 2019 season as our validation set, and test our model on the 2021 season. (Not training any data for the 2020 season because of the COVID-19 pandemic).

3.1 Random Forest

Would train this as a baseline model before moving onto the more complex methods

3.2 SVD

We want to use SVD, or Singular Value Decomposition [5], to build a model similar to collaborative filtering, but in this we would use it to represent home and away teams where the algorithm captures information only about the home and away teams, and then predict the total points scored during the week and season to determine future season records. Collaborative filtering is unsupervised learning where the predictions are based on the data supplied from other people.

3.3 Neural Network

The Neural Network [6] captures information in the same manner as the SVDs, but it goes one step further by taking features of both teams involved as inputs. So, it can get the outcomes of previous training examples, but also the team ratings (offensive and defensive) of each of the teams involved over an x amount of previous games, etc.

3.4 LSTM

Would use LSTM, or Long Short Term Memory Network [7], as a way to process x amount of previous games one by one instead of them all in one lump sum. LSTM is a recurrent model where a hidden state is repeatedly computed by processing a new timestep. A new fully connected layer at the end to output the number of points scored by the two teams weekly and yearly is what we would like to achieve here.

4. Evaluation

The goal of our experiment is to evaluate the performance of our models primarily through the MSE between the offensive and defensive ranking values that would be used to predict the outcomes of games and seasons. In addition, we can also calculate the percentage of the time that our model would have correctly predicted that the total offensive rankings point was either over or under the Average team stats number provided by the NFL

References

[1] Cooperdff nfl-data-py library for interacting w/ NFL data https://github.com/cooperdff/nfl_data_py/

[2] Scikit-learn. https://scikit-learn.org/stable/

[3] Surprise. http://surpriselib.com/.

[4] Pytorch. https://pytorch.org/.

[5] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. Numerische mathematik, 14(5):403–420, 1970.

[6] M. C. Purucker, "Neural network quarterbacking," in IEEE Potentials, vol. 15, no. 3, pp. 9-15, Aug.-Sept. 1996.

[7] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. ¨ Neural computation, 9:1735–80, 12 1997.

5. Appendix

This project extends the result of neural networking recurrent models on NFL data to predict weekly scoring outcomes and seasonal outcomes.