Capstone Project Proposal

Jordan Purser
July 27th, 2017
Udacity Machine Learning Engineer Nanodegree
Capstone Proposal

Fantasy Football Captain Prediction Model

Domain Background

Fantasy sports, where sports fans compete by selecting fictional teams of their favorite players and score points based on the ingame statistics of those players, has proved to be a successful recipe the world over. From Football to baseball, cricket and basketball, most popular sports have an arena where friends, family, colleagues and alike can duke it out for bragging rites and prizes.

In Australia, one of, if not the biggest fantasy sports is football (Australian Rules). Over 100,000 actively participate in the competition each year and many (myself included) take it very seriously. With multiple competition cash and non-cash prizes, betting markets, and locally organized prize pools/bragging rites, its not hard to understand why people are willing to go that extra mile to gain an edge on their competition. I personally have been playing for around 10 years and despite multiple strong finishes, have never been in earshot of winning the grand prize.

The basic premise of the game is to select a team of players who will score points based on the statistics they accumulate in-game. The goal is to generate as many points as possible. Coaches also have to make decisions on who to field on a given week (benched player's points do not contribute to your score) and who to pick as captain (captain's points are doubled).

As you can imagine, there is no shortage of statistics, expert opinions and peer recommendations on which to base your decisions. Academic research tends to be light on the topics of prediction and strategy, perhaps due to the informal nature of fantasy sports or maybe because of its competitive nature. A rare article on the topic by Roman Lutz (Lutz 2015) discusses the use of Support Vector Regression and Neural Networks for predicting quarterback scores in the NFL. There are however, vibrant communities such as the one at DT talk (http://dreamteamtalk.com/) where people share research and opinions. The people that run DT talk are also in charge of the official AFL website's fantasy news, and along with the folks at champion data (the AFL's official statisticians, https://www.championdata.com/), represent the authority when it comes to AFL fantasy. For the purposes of my study, I will consult and compare with the most appropriate research produced by these organizations.

For the purposes of this analysis, 'AFL' will be used to refer to the Australian Football League, 'fantasy coach' to the individual playing fantasy football, 'player' to the actual footballers, 'round' to a playing fixture round, 'season' to a playing fixture season.

Problem Statement

One of the most important decisions in AFL fantasy is that of the weekly captain choice. The significance of the captain choice comes from the doubling of the captain's score. This in-game mechanic combines a coach’s skill and luck and has a significant impact on team score, ranking and whether or not you win a prize.

The goal of the captain choice is to pick the highest scoring player possible in order to maximize the points doubled. We must however do this ahead of time. This leads to the question I will attempt to answer with my analysis. Can we create a accurate and reliable prediction of a player's score ahead of time for the purpose of ordering captain choices?
Stated mathematically, the goal is:

y' = model(x1, x2.....xN)

where y' represents the predicted score of player for the coming round and x represents the input features (discussed below) for that player. 'Model' is a machine learning algorithm that takes inputs from an arbitrary player and uses them to make a prediction of that players current round score.

Datasets and Inputs

For the purpose answering the above question, I propose the following features to feed into my model:
prev round player score
prev 3 round avg player score
prev 5 round avg player score
season avg player score
prev player score against opponent
prev player score at venue
prev 3 round avg team points for
prev 3 round avg opponent team points against
prev team score against opponent team
prev team score at venue

Each player, each round is a datapoint. You can think of the above features as a snapshot of a player's/team's historical performance as it stands immediately prior to the beginning of a round. For example, prior to the beginning of round 19, 'Fremantle' player 'Lachie Neale' had:

a prior round (round 18) score of: 95
a prev 3 round avg score of: 105
a prev 5 round avg score of: 106.4
a season avg score of: 102
a prev score against round 19 opponent (GWS) of: 169
a prev score at venue (Spotless Stadium) of: 169
a prev 3 round avg team (Fremantle) points for: 1528.67
a prev 3 round avg opponent team (GWS) points against: 1479.33
a prev team (Fremantle) score against opponent (GWS) team: 1590
a prev team score at venue (Spotless Stadium): 1590

The player 'Lachie Neale' will record a row like this for every game he plays as his data will change each round depending on what he scores, who his opponent is, what the venue is, how his team scores and how recent teams have performed against the upcoming opponent. This same data will be recorded for each player across each round.

The above features can be constructed (albeit with some decent feature engineering) from historical scoring data collected from the official AFL fantasy website (https://fantasy.afl.com.au) and fixture data collected from Wikipedia (https://en.wikipedia.org/wiki/2017_AFL_season). Available data consists of 87 rounds. Data will be gathered on every player playing in a given round. If data is not available for any of the above features, that player will be excluded from predictions for that round. This is to ensure that the model runs smoothly and is made on the premise that players without historical data are too risky to be chosen as captains. Approx No datapoints: 87 rounds x 18 teams x 22 players = 34452

Solution Statement

In order to solve the captain choice problem, I will build a supervised learning regression model that takes in an arbitrary player's data for arbitrary round and attempts to predict that player's score for the next chronological round. After creating predictions for all players in a given round, I will then order these predictions to identify those players predicted to score highest. These players will represent the best captain choices for the round.

I will create models using Naive Bayes, Gradient Boosted Trees, Support Vector Machines, and Neural Networks and evaluate the results using the R2 metric on validation data in order to select the highest performing model as my predictor. The above models have been chosen due to their breadth of approaches and historical success when applied to supervised learning problems.

Written mathematically:

sort(y') for all players in round

where y' represents the predicted player's score in a given round.

Please note that this approach does not use separate models for each player. For the purpose of the model, there is no distinction between players. Each datapoint is simply some arbitrary historical data used to generate a prediction.

Benchmark Model

Currently there are a number of different heuristics and methods for choosing a captain employed by AFL fantasy coaches. 3 of the most popular are:

1. Choosing a high averaging player (as determined by season average score)
2. Choosing a in-form player (as determined by rolling 3 week average score)

3. Choosing a captain mentioned in the 'Calvin's Captains' article, a discretionary approach to captain selection produced by DT talk that considers many of the above features when making a suggestion (as determined by article player rank)

For the purposes of my analysis all three will serve as benchmarks for evaluating my model, as each is sufficiently unique and reflective of popular current approaches of fantasy coaches. I am also genuinely interested to see if my model can outperform them.

Explicitly, each model will consist of an average of the top 5 player's scores as measured by each metric (highest avg players, most in-form etc). For example, the 'Calvin's Captains' article usually makes 5 player recommendations for captain each week. The scores of the 5 players in that round will be averaged to represent the benchmark value for that round. In the case of bye rounds* the top 3 will be used and if players are injured or suspended the next highest ranked player will take their place.

The reasons I have decided to go with an average for my benchmark models include 1. It is reflective of the average performance of a fantasy coach using the method. 2. Considering multiple players suggested by each method gives a more robust measure of the performance of that method, as on any given week there are outlier scores from players that are unlikely to be repeated.

*Bye rounds consist of many teams having a bye, therefore leaving a vastly reduced player pool to select from.

Evaluation Metrics

The primary evaluation metric will be the average actual score of the top 5 ranked players (as predicted by our model) over all rounds. This metric will provide a measure of the average performance of the captain picks of our model over the testing period. The units will be in points scored, so the metric will be directly interpretable and comparable to the benchmarks. Since the model and the benchmarks scores are directly comparable, we can simply take the difference to measure relative under/overperformance. The model will be deemed a success if it is able to achieve a higher average score over the testing period when compared to the benchmark models.

For the purpose of evaluating the various machine learning models to be trialled as predictors, I will be using the R2 metric generated on the validation dataset. This will give a measure as to the fit of the model to the data.

Other metrics that will be evaluated include, outperformance count (simple count of number of outperformance rounds in testing set when model is compared to benchmarks), number of times actual top round scorer was predicted with method (within its top 5), and the average performance of top pick as determined by each method.

Project Design

Workflow

1. Gather data
- scoring and fixture data to be scraped from relevant websites in raw messy form. Will probably be easiest to simply copy and paste into spreadsheet as data is tabular and only spread over a few pages
- data to then be lightly cleaned by removing unneeded data and saved in a structured format. Probably easiest do do this in spreadsheet and/or using pandas dataframe, then to write to CSV
- perform EDA on data to identify potential problems and missing data. Pay particular attention to missing values, wrong data types, outliers. Use pandas and matplotlib
- gather missing data and fix erroneous data. Will probably be a manual process if errors are small and/or infrequent
- iterate until data is satisfactory

2. Engineer features
- write scripts to generate above features from raw, cleaned data. Will probably need to use a combination of raw python and pandas
- write scripts to construct benchmark models specified above as they consist of simple rules and do not require ML
- sense check output features and models using EDA. Once again pay particular attention to missing data and outliers. Manually calculate a couple of examples of each feature to check script. Use pandas and matplotlib
- save engineered features and benchmark models. Write to CSV since data will be tabular

3. ML model preprocessing
- normalize input features. Use either numpy or built in preprocessing from sklearn
- create cross-validation sets using sklearn. Keep 60% of data for training, 25% for validation and 15% for testing. Data cannot be randomized when constructing sets as it has time series elements to it

4. ML model training and tuning
- import relevant models from sklearn (Naive Bayes, Gradient Boosting, SVM)
- construct neural network in tensorflow
- determine parameters for tuning for each respective model
- train and tune models, using gridsearch where appropriate. Iterate until satisfied with parameters
- decide on final model based on validation set performance. Use R2 score to evaluate models
- calculate predictions on testing set using winning mode
- break predictions into common rounds
- order predictions for each round
- identify top 5 players each round for calculation of model performance

5. Evaluate model against benchmarks
- calculate top 5 predicted average actual scores for each method
- calculate evaluation metrics for model and benchmarks
- visualize and interpret results using pandas and matplotlib

6. Reflect on analysis
- discuss questions such as:

What was the result of the analysis?
Why was that the result?
What problems did were encountered during the analysis?
What would you do differently next time?
Suggestions for improving the results?

Resources

Software:
python 3.5
numpy 1.12.1
pandas 0.19.2
matplotlib
seaborn 0.7.1
scikit-learn 0.18.1
tensorflow 1.0.0
jupyter 1.0.0

Websites:

http://dreamteamtalk.com/	- DT talk
http://www.afl.com.au/	- AFL
https://www.championdata.com/	- AFL statisticians
https://en.wikipedia.org/	- Wikipedia
https://fantasy.afl.com.au/	- AFL fantasy

Journal Articles:
Lutz, Roman 2015, 'Fantasy Football Prediction', ARXIV, viewed 28 July 2017, <https://arxiv.org/abs/1505.06918># Machine Learning Engineer Nanodegree