MASTER THESIS SPEECH: GUIDELINES

Slide 1 - Introduction

Slide 2 - Thesis title

The title of the thesis is: fantasy football forecasts, an evolutionary algorithm.

In short, to give you a quick idea, the aim of my thesis is to try to predict the best footballers to line up in the game of fantasy football using an evolutionary algorithm.

Slide 3 - What is fantasy football?

But before starting I just want to give you a brief explanation on what is fantasy football because maybe not everyone knows its rules.

Fantasy football is a game based on a football championship, in our case the italian one, called, as you may know, "Serie A".
At the beginning of the championship the fantasy football players have to assemble a team of 25 footballers. To do so they have to virtually buy them using a fixed amount of fake money, so they have to plan all the purchases carefully in order to maximize the relation between the cost and the footballer performances.
To begin with during each football match, the fantasy football players must line up 11 footballers out of the 25 existing ones plus 7 reserves.
Then, after every match, these footballers will receive a grade which is added to all the previous matches grades. Finally, at the end of the championship, the player with more points will win the game.

So, it is easy to understand how important it is to forecast the footballers who would have better grades, so a player can line them up to get more points compared to his opponents.

Slide 4 - The aim of the thesis

Furthermore after this short overview about the game we can proceed with a more precise overview about the aim of the thesis.

The thesis has been written in collaboration with Teamies, a startup which has an online website which provides its customers with advice on how to line up the best footballers. They wanted to improve the back-end of the application and the forecast algorithm and I worked with them in order to achieve these goals.

So, the thesis has 2 main purposes:

- The first main purpose is to redesign and implement part of the application structure, that includes both:
    - The redesign and the implementation of a new database
    - The development of web scrapers to collect all the information needed from several websites
- The second main purpose is to rewrite and improve the forecast algorithm using an evolutionary approach.

Slide 5 - Statistics considered

All the forecasts are evaluated using several players and teams statistics. Here you can see some of them.

For example for the footballers it is important to know what their matches grades are, how many goals they have scored, or conceded if they are goalkeepers. Also their role is very important because the analysis are carried out dividing players by role as we will see later.

Whereas for the teams it is important to know their ranking, how many points they have, how many matches they have won, lost or tied.
Also the bets on the teams performances are very important because we can have a general idea on how the matches could end up.

Of course we also considered other statistics but they are not reported for space reasons.

Slide 6 - Evolutionary algorithm

Previously I said that the to improve the forecasts we decided to use an evolutionary algorithm, but what is it, actually?
In artificial intelligence evolutionary algorithms are algorithms that try to find the solution to problems taking inspiration from the biological evolution.

The general idea is quite simple. As you can see from the slide the algorithm starts generating an initial population of individuals. In other words it creates several random solutions for the problem.
Then the algorithm evaluates how good these solutions are. If a stopping criteria is reached, for example one of the solutions fulfills the goal we want to achieve, the algorithm stops. Otherwise using several decision rules it selects only some of the individuals and then it combines them in order to generate a new population.
Then this new generated population is evaluated and as mentioned before if a stopping condition is reached the algorithm stops otherwise it cycles over and over again combining all the new individuals to generate the new ones.

In our case we decided to configure the evolutionary algorithm using the parameters that you can see at the bottom of the slide.

- The initial population has been set to 700 individuals because we wanted to generate as many different solutions as possible at the beginning of the evolution. The reason is that the more different individuals you have at the beginning the more the probability to find better solutions increases because the initial solution space is deeply explored.
- Then, after the starting generation, the maximum size of the population is set to 200 which is a good compromise between computational time and solution space search.
- Variance and inertia can range between 0 and 1, in our case they are set to 0.9. The more the variance value is close to 1 the more the new individuals will be different from the old ones. While the more inertia is close to 1 the more slowly these changes will happen.
  Variance can be self-adapted if combined with inertia.
  In fact the idea of this configuration is to have a high variance at the beginning so it is possible to deeply explore the solution space while to slowly reduce this variation while time passes in order to converge to the final best result.

Slide 7 - Fitness function

In the previous slide i said that an evolutionary algorithm checks the quality of the solutions. But how does it do so? Well, To do so it uses a fitness function which checks how close the current generated solutions are to the final goals we want to achieve.

In this slide you can see the fitness function used in our case.
To function compares the software forecast of the footballer's performance with the real one he achieved. In doing so for every footballer we have an error relative to his forecast.
Then all the footballers errors are used to calculate the error mean and the standard deviation which will be used to evaluate the quality of the solution. Of course the closer these 2 values are to 0, the better.

Slide 8 - Application structure

As you can see here it is possible to have an overview on how the application is structured and how it works.

All the main application tasks, except from the evolutionary part, are controlled by Chron, which is a job scheduler for Unix systems. As you can see in the top part of the slide it is in charge of:

- starting the collection of the information from the websites running the web scrapers using the run time environment Node.js
- then it is charge of running the scripts to store the downloaded information in the database, also in this case using Node.js
- and finally it is in charge of running the C++ software which will generate the forecasts for the following matches using the footballers and teams statistics stored in the database. To do so the software combines these statistics weighing their importance using specific parameters. And these parameters are the ones that the the evolutionary algorithm will have to improve.

Whereas, for the evolutionary tasks, as you can see at the bottom of the slide, I used MicroGP which is an open source evolutionary algorithm.
MicroGP is launched by a bash script. Then MicroGp will generates the parameters to evaluate and then it will run a C++ software, which basically is the fitness function I discussed before.
This C++ calculates, using the generated parameters, the errors made on the footballers forecasts. These results are reported to MicroGp so that it can check if a stopping condition is reached and if not it repeats the process just discussed until that a satisfactory solution is found.

Slide 9 - Test introduction

Ok, now I think we have a good overview on how everything works so we can proceed and speak about the tests we carried out.
In this slide you can see an introduction on the dataset used for the tests and how these tests are organized.

The dataset we used is relative to the previous championship, the one played in 2017 and 2018.
Theoretically we should have 38 different daytimes but the startup Teamies for the previous year had only information starting from the 6° daytime. So, in practice, we had 33 daytimes to analyze.
For every daytime we have 20 teams and around 1000 footballers. Of course not all of these footballers had played because many of them were substitutions/reserves who did not enter the field.

Now, focusing on the tests guidelines, you can see that we decided to divide the forecasts analysis accordingly to the footballer's role. The reason is that, depending on the role, the parameters that the evolutionary algorithm have to find could largely vary. For example for a striker the parameters relative to how many goals he scored are more important compared to a defender.
Then we also decided to run tests on single daytimes and on all the championship daytimes.

Of course the analysis have been carried out splitting the dataset in two parts: one for training, around 75% of the dataset, and one for testing, around 25% of the dataset.

Slide 10 - All daytimes: error analysis

The idea of the startup Teamies was to find the most efficient single set of parameters to use for all the daytimes. So we decided to run tests on single daytimes to understand if this goal was achievable. In fact, in running these single daytime tests, we wanted to have a general idea on how the parameters would have changed across all the different daytimes and for the different footballers roles.

After running some tests we could observe that quite a lot of parameters ranged in specific intervals. This means that the goal of finding a possible single set of parameters was probably achievable so we started to run tests on all the daytimes together.

As you can see here, the chart shows, for the role strikers, the improvements in percentage relative to the error mean and the standard deviation.
At first sight, it is possible to observe that the results are quite good but…

Slide 11 - All daytimes: footballers improvements

...looking closely on how many footballers have a lower error compared to their original forecasts, it is possible to notice that they are not so many.
The reason is that in these tests we evaluated together all the footballers without considering their probability to play. One example on why this situation could give a great negative impact on the model is that some footballers with a probability to play of 45%, very unlikely to play, could actually enter in the field and perform quite well. In doing so the model would find a great difficulty to adapt because this situation is not easily predictable.

Slide 12 - All daytimes with filter: error analysis

Probability to play >= 70%

To solve this problem we decided to try to repeat the test filtering the dataset according to the footballers probability to play. In this case we considered the players with a probability to play equal to or bigger than 70%. The results were very good. In fact as it is possible to see in the chart, relative to the strikers’ role, both the error average and the standard deviation improved very well.

Slide 13 - All daytimes with filter: footballers improvements

Probability to play >= 70%

We can also see the improvements achieved considering how many footballers have a lower error compared to their original evaluations. The results are very good in fact from the chart it is possible to see that for all the roles the improvements are on average around the 75% and the 85%.

Slide 14 - All daytimes with filter: error analysis

Probability to play between 45% and 69%

Then we also performed tests for footballers with a probability to play between the 45% and 70%. In this case the results were not so good for the same reason explained before, or that these footballers have a very high uncertainty to play which is very complex to model.

We have good improvements for both the error mean and the standard deviation but if we consider how many footballers have a lower error compared to their original forecasts, the average improvements are only around 65-70%.

Slide 15 - Conclusions

After all the tests we came to several conclusions.

The first thing to say is that the goal of reaching an optimization in the forecasts has been achieved. In fact for footballers with a probability to play larger than or equal to 70% the results are extremely good. For footballers with lower probabilities results are acceptable but could have been improved.

But it is important to underline that the group of footballers with higher probability to play is the most important one for the forecasts.

In fact, usually, in this range of probability there are all the footballers that would enter the field and who would give the larger contribution to the points that a fantasy football player can get. So, having achieved positive results for them, is a clear sign that the optimization was successful.

So then, what comes next?

Well of course the forecast algorithm can be improved and to do so an obvious strategy could be to perform more tests tuning in different ways the parameters. For example changing the filtering values or changing the parameters used in the evolutionary search.

Another interesting strategy could be to create a new specific algorithm for footballers with lower probability to play. One idea could be to try to predict when a regular footballer could be substituted during a match and predict which other player could enter in the field in his place. To understand which regular footballer could be changed, a possible procedure could be to collect information about his physical state, for example if he has some small injury, in order to forecast his ability to play for all the length of the match and, if substituted, to provide a more accurate forecast for the players who will substitute him.

And then, of course, it would be interesting to perform the same analysis using a neural network. Neural networks are brain-inspired systems which are intended to replicate the way that humans learn.