MALIS Project Final Deliverable: ML for ATP Tennis Matches Prediction

Lorenzo Cascioli, Raphael Toumi

Fall 2020

1 Main idea

The basic idea for this work was to study some supervised Machine Learning approaches in order to predict ATP tennis match outcomes using historical players' performances across a wide variety of statistics. It follows that the outline is a binary classification problem on samples where the data of the two players (or, more precisely, the differences between them) are contained: class 0 means that player 1 wins, and viceversa for class 1. The general workflow involves the raw data manipulation to get the precise statistics needed to describe a match, and then the experimentation and fine-tuning of different ML models to see the one achieving the best results, aiming of course at having a positive ROI (Return Of Investment) on the long-term, and in any case improving the result with respect to some trivial betting strategies (e.g. betting on the player favoured by the odds or on the higher ranked player). For those that do not know how a tennis game develops, [1] and many other references online explain it very clearly.

2 Data managing

The raw data are taken from two open-source repositories: [2] and [3], the first being very useful for the comprehensive statistics, the latter because it has all the bookmakers odds on the matches. These two datasets joint are believed to have all the needed information for the project.

The creation of ad-hoc data for each match was not an easy task at all, and the work done by M.Sipko [1] was of great help in the matter. The available data are simply punctual statistics registered for each played match, while the data needed to build the models should reflect the historical performances of the players up to the match they are about to play. Before presenting the adopted features, an important specification: it has been already said how the problem is seen as a simple binary classification task, with label 0 indicating player 1 wins, and label 1 indicating player 2 wins. Having retrieved the features for each player, then, the easiest way to put the two together is to represent a feature with the simple difference between its value for player 1 and for player 2. In this way, a symmetric model is built, ensuring that there are no differences in the weight assigned to the same feature for the different players (as it would be possible in case they were represented separately). Thus the feature **rank**, for example, will actually represent the rank difference $rank_{p1}$ - $rank_{p2}$. That said, three types of features have been built: certain, historical and combined features. A summary of each of these features is reported in Appendix A.

The work, however, needed consistent refinements. First of all, not all the past matches are equally representative to describe a player's form: most recent matches will surely be more informative. Secondly, tennis is played on a variety of different surfaces, and many players perform very differently across the various surfaces. To take this into account, the average done on past matches did not use the same weight for each past match of a player: a **time discount** was adopted, making the weight of a match decrease as its temporal distance from the match about to be played increases, and a **surface weighting** was also applied, greatly favouring past matches played on the same surface as the one of interest. Specific values for the surface weighting have been retrieved by evaluating a correlation matrix in performances across the different surfaces (hard, clay, grass) analyzing all the available training matches. This approach is believed to be more solid with respect to more traditional 'splitting by surface', as it tries to keep and use all the past matches with different weights (instead, using only past matches on the same surface might reduce too much the available data).

Moreover, to have an approximately constant number of past matches examined to construct each match's feature vector, it has been chosen to limit the analysis to the 5 past years of ATP activity. But this alone is not enough to solve a relevant problem: clearly, for some players lots of data will be available, while for others very few. Also, for players that do not have a long history of matches in the circuit, the few present data might be misleading in the definition of their actual quality. Hence, two further tricks were introduced.

1. A **common opponent** model was implemented, similar to the one presented in [4]; in practise, wanting to ensure that a fair comparison is made between the two players, only matches against common opponents in the circuit are considered, provided that a minimum number of common opponents exists for the two players. This of course reduces the amount of available data for each match, but it should greatly improve the fairness of the comparison. Basically, the average statistics are evaluated for each player with respect to every opponent that has also played against his adversary, and then these

statistics are averaged through all the common opponents to get a final measure for the considered player.

2. Furthermore, a measure of **uncertainty** was appended to each match, describing "how certain" the feature vector is - basically, this measure is evaluated as the inverse of the sum of the weights of past matches available in the preparation of the match's features. This should, later on, be of help in identifying which are the matches whose feature extraction has been more reliable, and those which are instead likely to bring only noise.

Having put into practice all these methods, through some Python scripts the feature vectors for all the 2010 to 2020 matches were built. For each match, the vector contains all the described features, the names of the two players, the uncertainty measure, the winner (label to be predicted), the odds taken from Bet365 and some further information (tournament id and year of the match) which may be useful for the data splitting.

3 ML models

Having produced the final csv file with all the samples and proceeded with some data cleansing and normalization, several models were tried for the binary classification task which is the core of the project. The idea is that each ML model is trained with the data coming from all the ATP matches from 2010 to 2016 for which the features are available (i.e. with enough past information on the two players), using as validation and test set the following seasons' data, to see if predictions on new matches are sufficiently accurate.

This last part involves, after having obtained from the model the winning probability of each player, the implementation of a betting strategy on all the test matches. Having chosen how to bet, a simulation covering all the test matches is run and the final outcome for the ROI is considered as performance metric.

3.1 First experiments

Several experiments with different ML techniques were made to see how the various models behave. Unfortunately (but not that surprisingly) using all the training matches and all the validation matches, the ROI is always negative. Logistic Regression, Random Forest, SVM, and MLP were used (directly from *Python sklearn* library) with the obvious parameter tuning, but it seems impossible to get favorable results even with clever betting strategies like the Kelly Criterion (more on this later). Indeed, while still performing better than other trivial betting strategies like simply betting on the predicted winner (from ML probabilities or from the odds), it is not sufficient, as the ROI constantly stays in the interval [-10%,-5%], without significant differences between the different classification models that are adopted. It seems though, as also the literature of this field says, that Logistic Regression is the classifier which behaves better for this sort of problem, so some further optimizations were thought starting from this model.

3.2 Big Tournaments + Big Players Approach

The problems encountered in the first large-scale experiments can be surely attributed to several different aspects.

Surely, a considerable amount of training data which have high uncertainty is bringing noise into the classifiers: indeed, taking into account only the matches with lowest uncertainty value (up to a certain %) might make the models stronger. Performing some detailed analysis on Logistic Regression, it turned out that the best results were achieved when only considering the 25% less uncertain training matches in the definition of the model. Furthermore, it was pointed out that if it makes sense to select the 'best' matches for the training, it is also reasonable to do it also for the final betting: it was hence decided (to get a reasonable trade-off between having a consistent amount of bets and avoiding bets on very uncertain matches) to keep only the most certain 50% of the validation matches for the betting simulation. This makes total sense, since (as a bettor knows) there are many matches with too few past data on the players where betting is extremely hard.

It must also be recalled that the data which are being used are limited and some key characteristics of a player's game style cannot be deduced from them (e.g. having the average numbers of winners and unforced errors per game might be very useful to better describe a player). Hence, it is not expected that the

descriptive power of these models is exceptional, also considering the very high variability which in intrinsic in sport events.

Nevertheless, the knowledge of some characteristics of the tennis world can help in developing the algorithm so that it works definitely better. Indeed, it is commonly known among the experts that the big tournaments (Grand Slams and ATP Master 1000), on which the top players attention is very focused upon, tend to have much more predictable results. Following this perspective, a decision was made with the general goal of reducing the number of placed bets for the sake of a higher success rate on the placed ones. This approach finally produced a new setup which was positively tested on the 2017-2020 data.

The general idea consists of two main points:

1. bet only on major tournaments matches (4 Grand Slams + 9 ATP Masters 1000 [1]);
2. bet only on matches where a top player is involved - this because of two reasons:
   (a) top players usually win, so an higher overall accuracy can be expected;
   (b) if a top player loses and a bet was placed upon his opponent, the reward will be big.

   The definition of a top player is clearly of difficult definition. In a pretty straightforward way, a list of the top players for the validation set was built by considering players which entered the Top 6 in the ATP rankings at least once during such time span. This is somehow a little distortion, as the actual idea would be to bet, at each tournament, on the players which are occupying the top seeds in the ranking; retrieving all such temporal data for the three years is though very difficult, so this simplification needed to be adopted. Nevertheless, the test set was built on 2020 data with this pure interpretation of the top players as the current Top 8 of the ATP rankings. Results for both sets will follow, and their exact composition is reported in Appendix B.

Having prepared in such way the subset of validation/test matches on which bets are going to be placed, it must be recalled that the model is still the original and simple Logistic Regression implemented before, since it is believed that it should have enough descriptive power to be sufficiently good with the new, restricted test data. Only one little change has however been introduced: after some feature selection was performed (with results often too foggy to allow the drawing of general conclusions), the 'rank' feature was deleted from the feature set, as removing it seemed to always improve the final ROI.

So this is the final structure for the Logistic Regression model and its validation/test data. Some further interest was also put into selecting a betting strategy appropriate to this setup.

3.3 Betting strategy

Many different possibilities exist to shape a solid betting strategy. They range from very simple ones (i.e. betting on the player whose winning probability is higher) to very complex ones, like the Kelly Criterion. The Kelly Criterion [5] is indicated by several sources in the literature as the most favorable strategy on the long-term, as it adapts the bet amount as part of a fixed maximum size looking at how big is the edge that the bettor believes he has over the bookmaker. Also intermediate strategies, however, can be built with rather satisfying results, as it will be shown.

The betting strategy adopted in the end is simpler than the Kelly Criterion, because it proved to give good results in any case, even being a bit less conservative than with such approach. Basically, following Kelly's approach, a bet is placed only on matches where the estimated winning probability for a player is higher than the one implied by the odds: this means that a bet is *not* placed on each validation/test match, but only on a subset of them where it is believed that the algorithm has outperformed the bookmaker. Differently from Kelly, which adapts the bet size according to the difference between these two probabilities, here it was chosen to bet always 1€ on each match; this was done for simplicity but proved to work sufficiently well in the end.

The described method, therefore, involves betting *at most* on one of the players for each match; however, as said, if the odds seem better than the estimated probabilities, the match is not considered. Such bet,

[1] ATP Finals were skipped due to problems in retrieving the exact ordering of Round Robin matches

anyway, can be placed either on the favoured player (and this frequently happens) either on the underdog, if it seems that the odds he has been assigned are too high for him: in these cases, betting on the underdog might be worth the risk, as if he ends up winning the reward for the bettor will be usually consistent. While all of this is true, it seemed too unrealistic to waste money on opponents of a top player which realistically (according to the model) have an extremely low winning probability. Hence, a threshold was set (and fine-tuned) when the algorithm suggest to bet on the underdog: the bet is allowed only if the model assigns him a winning probability of at least 30%, otherwise the match is skipped.

With this redefined setup and an ad-hoc betting strategy, some new tests were run on the 2017-2020 data.

4 Results

As reported before, the model is a Logistic Regression classifier (built with cross-validation) that uses all the available features but the 'rank' and is trained on the 25% less uncertain matches from the period 2010-2016 (the resulting amount of used training samples is slightly less than 4000 matches).

A remark must be made at this point: from the beginning, the training set period has always been the same, with matches from 2010 to 2016. The original idea was to use a 7-2-2 split, using 2017-2018 data as validation set and 2019-2020 data as test set. However, an unfortunate change had to be introduced: since on the test set the aim was to use the original definition of top players (hence the true Top 8 at each tournament), it was too time-consuming to get such temporal data for two full seasons. Therefore, it was chosen with some regret to limit the test set to the sole 2020 season, moving 2019 into the validation set. This surely means that some further analysis should be conducted on new test data, either considering future matches in 2021 or scraping the ATP website to obtain precise data also for 2019 on the evolution of the Top 8 rankings. 2020 was way easier as, being the season shorter due to Covid-19, the rankings remained pretty static and could be easily replicated in the betting streaks. This might of course be a problem, because on the other side a worryingly limited number of matches is considered as test set.

Nevertheless, it can be asserted that some rather interesting results were obtained working on the validation set. With the described model, validation set structure and betting strategy, the betting simulation over the three years 2017-2018-2019 ended indeed with a positive ROI. Table 1 shows the results of the simulation on the validation set compared to two quite trivial baselines: a betting strategy which always bets on the player favoured by the odds, and another one which always bets on the player favoured by the model outcomes. To have an idea of how a 100€ budget evolved over the validation years, see Appendix C.

Betting strategy	**Success Rate**	**Investment**	**Return**	**Net Profit**	**ROI**
Favoured by the odds	75%	440	420.05	-19.95	-4.53%
Favoured by the model	75%	440	433.48	-6.52	-1.48%
Predicted winner at better odds	59%	241	262.01	21.01	8.72%

Figure 1: Betting results in € for different strategies on *Validation Set*

From these results, a couple of conclusions can be drawn at first:

- the model, while still failing to achieve acceptable results when applied on all the matches, seems to have in any case enough descriptive power to be useful (it has 75% accuracy on properly selected validation matches): in this setup, indeed, a consistent improvement is obtained following the model's probabilities with respect to the odds;
- it is true that the model brings some improvement, but it needs to be paired with a clever betting strategy for the algorithm to become really effective: the adopted one seems to fulfill the need, achieving a positive result quite in line with the initial expectations.

A remarkable difference can be immediately spotted in the betting amount: as explained before, the final setup does not bet on every match, but only when it is believed to have an edge over the bookmaker (and the considered player is not too much of an underdog): the idea of avoid betting on matches where we cannot outperform the bookmaker has apparently had some success, and is key for a good functioning of this

structure.

It is not only that, though, that allows a more profitable betting: the last method risks betting on the underdog when it believes that he has been assigned by the bookmaker a winning probability that is too low, and he seems to actually have discrete chances of victory. This reflects in a lower overall betting accuracy, but in higher ROI, as the cases where a bet is placed on a winning underdog are very well rewarded. This a very peculiar aspect of this type of problem: a maximization of the accuracy alone is not enough to end up with a satisfying result: the accuracy gives an idea of the descriptive power of the classifier, but the values of the output probabilities and their relation with the bookmaker's odds are what in the end defines the functioning of the system.

The same model was then applied on a fresh test set, which for the previously explained reasons is limited to the 2020 matches. For these, it was possible to place bets only on the Top 8 players of the year, precisely adopting the original idea that wants to bet only on the currently Top 8 seeded players.
When passing to the test data, as expected the accuracy of the classifier decreases, since several hyper-parameters (e.g. the % of best training matches considered to build the model) have of course been fine-tuned on validation data. Indeed, the accuracy on the selected test matches is slightly below 70%. This, however, does not directly imply that the results will worsen accordingly, as discussed right above. And looking at table 2 it can be seen how the final result still seems to work out well.

Betting strategy	**Investment**	**Return**	**Net Profit**	**ROI**
Favoured by the odds	60	55.73	-4.27	-7.12%
Favoured by the model	60	59.34	-0.66	-1.10%
Predicted winner at better odds	38	41.55	3.55	9.34%

Figure 2: Betting results in € for different strategies on *Test Set*

It is very well known that these results must be taken with many precautions, as they are derived from the matches of a single ATP season which was even shorter than normal due to the Covid-19 pandemic. Nevertheless, it is favourably impressing that the results for the final ROI, even if obtained after a limited amount of bets and therefore not necessarily enough general, closely follow the ones obtained on the 3-year long validation set, allowing to think that a certain coherence might exist in the elaborated strategy.

5 Conclusions

Summing up, for the application of ML models to extremely large-scale betting strategies, where it is required to place bets on all the ATP tournaments of the year and for all the players, the task still look really hard, due to the intrinsic beauty of sport that is often dominated by high uncertainty. It might though be of help to have access to more complete data sources (like [6]) and to dig deeper into the construction of ad-hoc neural networks. That said, Logistic Regression, according to the experiments and to the literature, was identified as the most suited classifier for this task, and some further analysis produced a valid alternative approach based on that.

Indeed, using Machine Learning together with additional knowledge of the world of tennis allowed to build a new, more constrained algorithm: relying only on a precise subset of matches, considered to be more predictable, and with a betting strategy shaped ad-hoc, the new approach proved to behave pretty well on the three years of validation data, and the idea seems to be consistent also on new test data. Surely, some more experiments need to be run on new matches to see if this promising trend continues to hold, but the developed work allows to be optimistic.

Hoping that this project will be further improved in the future, it has already been useful, in any case, to explore a very interesting connection between sports betting and Machine Learning, and furthermore to demonstrate how ML can become really effective when paired with human knowledge of the specific application field, affirming once again how ML usually needs a strong human component in order to be fully exploited and perfected.

Contributions

- **Data managing**: L. Cascioli
- **General ML models and experiments**: L. Cascioli, R. Toumi
- **Big Tournaments + Big Players Approach**: L. Cascioli

References

[1] M. Sipko. *Machine Learning for the Prediction of Professional Tennis Matches*. URL: https://www.doc.ic.ac.uk/teaching/distinguished-projects/2015/m.sipko.pdf.

[2] Jeff Sackmann. *ATP Tennis Rankings, Results, and Stats*. URL: https://github.com/JeffSackmann/tennis_atp.

[3] URL: http://www.tennis-data.co.uk/alldata.php.

[4] A.M. Madurska W.J. Knottenbelt D. Spanias. "A common-opponent stochastic model for predicting the outcome of professional tennis matches". In: *Elsevier* 64.12 (2012), pp. 3820–3827. DOI: https://doi.org/10.1016/j.camwa.2012.03.005.

[5] *Kelly criterion*. URL: https://en.wikipedia.org/wiki/Kelly_criterion.

[6] *OnCourt tennis dataset*. URL: https://www.oncourt.info/index.html.

Source Code

The code for the generation of the feature vectors, for the ML models and for the betting simulations is available at this repository.

Appendix A Features

1. **Certain features**:
   - **rank**: the ATP ranking of the players.
2. **Historical features**: built performing the historical averaging of the performances of the player. They are:
   - **Fs**: average % of first serves in play.
   - **W1sp**: average % of points won on first serve.
   - **W2sp**: average % of points won on second serve.
   - **Wsp**: overall winning % on service points.
   - **Wrp**: overall winning % on return points.
   - **Tpw**: % of all points won.
   - **Tmw**: % of all matches won.
   - **Aces**: average number of aces per game.
   - **Df**: average number of double faults per game.
   - **Bpc**: average number of break points conceded per service game.
   - **Bps**: % of break points saved.
   - **Bpo**: average number of break points obtained per return game.
   - **Bpw**: % of break points won .
3. **Combined features**: they come from the combination of the historical statistics which may describe meaningful patterns of the players' game styles. They are:

- **Complete**: it is an attempt at measuring the completeness of a player by seeing how well he serves and returns. Hence, COMPLETE(p) = WSP(p) * WRP(p).
- **Serveadv**: this measure tries to measure how well a player serves with respect to its adversary return capabilities. It is the first feature, thus, that is computed from the beginning taking into account the adversary: SERVEADV(p1) = WSP(p1) - WRP(p2).
- **Fatigue**: this feature counts the number of games played since the beginning of the tournament by each player and gives a % difference between the two.
- **H2H**: the head-to-head feature is frequently used in tennis, where the past matches results between the two players might be very informative. Therefore, a feature was built so that H2H(p1,p2) measures the % of previous matches between the two won by player 1.

It must be recalled that each of these features is measured for both the players and then the difference is made between player 1 and player 2 value.

Appendix B Top Players Lists

B.1 Validation set (2017-2019)

For these years, due to difficulties in retrieving the temporal evolution of the rankings, a list was manually selected. It contains players which shined at some time during the three year time window, entering at least once in the Top 6 of the ATP rankings.
Such list is composed by: Roger Federer, Rafael Nadal, Novak Djokovic, Dominic Thiem, Daniil Medvedev, Alexander Zverev, Juan Martin Del Potro, Kevin Anderson, Marin Cilic, Grigor Dimitrov, Kei Nishikori.
Of course, this implies that a simplification is being adopted, as all the macthes of these players are selected, not those played when they were at the top of their form. The difference might not be important for some players, who constantly remained at the top, but it might be for players which have achieved a peak of their form for some months and have performed much worse in other moments.

B.2 Test set (2020)

For 2020, it was possible to fully retrieve the players which occupied practically constantly the Top 8 rankings positions, so the betting simulation that was developed should be more adherent to the original idea.
Such players are: Rafael Nadal, Novak Djokovic, Roger Federer, Dominic Thiem, Daniil Medvedev, Alexander Zverev, Stefanos Tsitsipas, Andrey Rublev.

Appendix C Budget Evolution

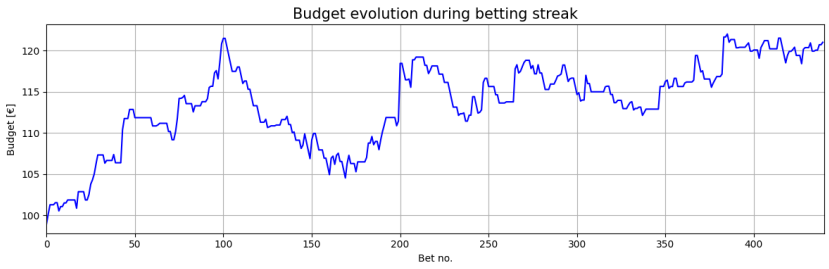

Figure 3: 100€ budget evolution during validation set simulation

The following graph simply shows how a 100€ budget evolved during the betting simulation of the three years taken as validation set. Despite the obvious oscillations in some short time windows that should to be attributed to the natural variability of the data, the evolution seems to depict a general growing trend in the amount at disposal of the bettor.

The result on 2020 test set is very similar, but was not reported because the time window is too small to allow drawing general conclusions.