Project Report: Get Rich from Betting

Lu Si Hong (slu335) **Aryani Paramita** (aparamita3) **Jonathan Pradipta** (jpradipta3) **Kristof Meszaros** (kmeszaros3) **Jadug Parusa** (jparusa3)

Abstract

Sports betting sites provide certain odds-on certain events on which individuals can bet, e.g., who is going to win a tennis match? If we had a model that could predict with 100% accuracy which person will win, we could always just bet on that player and have extraordinary returns. With the availability of numerous public detailed datasets on tennis matches and their players one might wonder if it is possible to build a reliable model to predict the winner.

1 Introduction

With the advent of new and improved technologies, more extensive calculation and storage capability, "machine learning" has become more and more an integrated aspect of our daily lives. Its applications can be found anywhere from the financial sector to the biochemistry and medical sectors. As such it is not surprising that it can be found as well in the sport industries, in particular on the use of machine learning models for sports prediction either for economical benefits or rather just as an interesting research topic.

Sports events and the prediction of their results through scientific and stastical analysis has been going on for a long time, pre-dating the use of machine learning models, where simple qualitative modesl that uses easily computed stastics measure such as batting average, are used in an attempt to predict the winning odds. However with the advancement in computer science and the significantly improved accessibility to powerful computers, the used of maching learning has become more prevalent in the sport industries.

2 Literature Review

There have been quite a few research papers conducted on the use of maching learning models to predict sport events ranging from the simple use of machine learning algorithms such as the logistic regression used by Wilkens [5] to train the raw match records to predict the winner of the matches or to more complex betting strategy such as the one used by Hubacek [4] where they combine a neural network model with a betting strategy that is designed to optimize profit expectation while balancing the profit variance in an attempt to exploit the sports betting market.

Wilkens uses multiple machine learning models and algorithms such as logistic regression, neural network and random forest in an attempt to predict the outcome of a match by analyzing approximately 39,000 record of matches. He uses dataset combining the match records, player records and the betting market data. It builds on top of previous established stasitical and machine learning techniques toinvestigate the information connection between the betting odds and historical player and the match data to predict future match outcomes. He found out that the most important variable in predicting a future match outcome is the player rankings and the odds provided by the betting bookmaker, while variables such as age difference, home vs away advantage are hardly significant and does not provide additional prediction power.

An interesting and different approach was tried by Forrest & Mc Hale [3], instead of using models and algorithms to predict the outcomes of matches, they instead choose to focus on developing a new econometric approach using simple statistics measures such as standard deviationa and skew to estimate the relationship between the odds and returns of a tennis betting market. Based on this new approach they found out that there are positive biases through the odds in the wagering or betting market for tennis.

3 Objective

For our project, we have choosen tennis as our sport activites since it has receive less attention

than other activities such as soccer or baseball. Our main purpose in this project is to build a reliable statistical model to predict the probability of the matches' results, with the help of a few research questions

1. Out of each statistical model, what parameters are statistically significant in determining the result of a match – would player attributes matter more than match attributes?

2. Are we able to assign a score to whether player on each end is a stronger player (Safer Winners/Rising Underdogs) based on the past match results and fit into the statistical model

3. Can we break down players into a vector of statistics? E.g. overall good serve%, performance on different tracks etc... so that basically we would clash stats against each other, not players.

If our model can predict results above expectations, we can leverage this to either create sports betting sites, or even have a betting hedge fund with superb risk-adjusted return.

If we would like to have our own sports betting site, we would want to make sure we provide odds which makes it profitable for us to operate. This requires us to use our model to provide some estimates about which team/person is going to win. Of course, we can use other betting sites' odds as benchmarks, but if we could finetune them with our models, it would increase our profitability and give us an edge. Or alternatively we could create a hedge fund for sports betting, and simply use the investors' money to make bets, and take a 30% cut from profits for example.

4 Data Colllection & Transformation

4.1 Data Collection

Our team are using 3 data sources to assist in building the models

- Tennis match records for the last 10 years (2011 - 2021)
- WTA player information
- WTA latest player ranking

these datasets are sourced from

- Kaggle Women's Tennis Association Matches [2]
- Jeff Sackmann's Tennis WTA Data (3 data sources) [1]

The first data set contains the records of 29021 tennis matches with 49 features (columns) for each match. For our analysis we are focusing only on the below 17 features since the remainng feature are dropped due to significant amount of missing or invalid data.

- **Tournament:** Tourney_ID, Tourney_name, Surface, Tourney_Level, Tourney_Date
- **Winner:** Winner_name, Winner_hand, Winner_ioc, Winner_age, Winner_rank, Winner_rank_points
- **Loser:** Loser_name, Loser_hand, Loser_ioc, Loser_age, Loser_rank, Loser_rank_points

The second data sets contain the personal information for 60,653 player, it has the following fields

- Player first name
- Player last name
- Player ID
- Player IOC
- Hand
- Height
- Date of birth

The final data set contain the ranking of the top 1425 tennis player that was taken as of 12 September 2022, it has the following fields

- Player ID
- Player Rank
- Player Points
- Player # World Tour

4.2 Data Transformation

Before building the model, it is first necessary to clean and transform the data to ensure that it is in the best format before feeding it to our models. We cleaned the data by removing any rows with invalid data or "NA"s and further re-format the data by replacing the winner and the loser columns with player 1 and player 2 respectively, where we will use our models to predict the chance of player 1 winning (player 2 losing).

Following is the transformation logic for player 1 and player2 for e.g. id, age, name, rank_points:

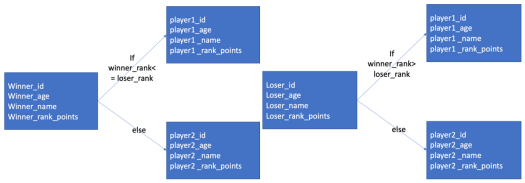

Figure 1: Data Transformation Logic

Match_outcome value will be indicated as:

- 0, when player 1 wins the game
- 1, when player 2 wins the game

Surface for each Match records has 3 values: Hard, Clay, Grass. As each of the variable is not superior to one another, surface require variable transformation to indicator variable. Similarly, for each of the player's hand data, we will convert notation 'L' (Left Handed) and 'R' (Right Handed) into 1 and 0. Tourney levels will also be one-hot encoded, it has the possible values of: I P F G O D.

For our initial analysis of the model, we decided to first only take the year 2011 to train the data, as such the remaining data can be used to test the accuracy of our model as time progresses. And as the per the usual model training convention we further split the 2011 data into training data (70%) and test data (30%).

5 Approaches

As part of our initial approach we have designed a few logistic model to predict the result of the match. For our initial approach our model variables are selected based on a subjective qualitiative approach. The variables are picked rationally based on whether we believe they would have a significant impact, for these excerise we picked the following variables from the first dataset:

- Player 1 Rank
- Player 2 Rank
- Player 1 Score
- Player 2 Score
- Player 1 Age
- Player 2 Age
- Surface Type
- Tournament Level

and created a few models based on a combination of those variables. Below are the models that we designed.

Model 1 (Base Model)

$$\text{Winner} = \beta_0 + \beta_1 + \text{ Player 1 Rank} + \beta_2 \text{ Player 2 Rank}$$

Model 2

$$\begin{aligned}\text{Winner} = \beta_0 &+ \beta_1 \text{ Player 1 Rank} \\ &+ \beta_2 \text{ Player 2 Rank} + \beta_3 \text{ Player 1 Score} \\ &+ \beta_4 \text{ Player 2 Score} + \beta_5 \text{ Player 1 Age} \\ &+ \beta_6 \text{ Player 2 Age}\end{aligned}$$

Model 3

$$\begin{aligned}\text{Winner} = \beta_0 &+ \beta_1 \text{ Player 1 Rank} \\ &+ \beta_2 \text{ Player 2 Rank} + \beta_3 \text{ Player 1 Age} \\ &+ \beta_4 \text{ Player 2 Age}\end{aligned}$$

Model 4

$$\begin{aligned}\text{Winner} = \beta_0 &+ \beta_1 \text{ Player 1 Rank} + \beta_2 \text{ Player 2 Rank} \\ &+ \beta_3 \text{ Player 1 Score} + \beta_4 \text{ Player 2 Score}\end{aligned}$$

Model 5

$$\begin{aligned}\text{Winner} = \beta_0 &+ \beta_1 \text{ Player 1 Rank} + \beta_2 \text{ Player 2 Rank} \\ &+ \beta_3 \text{ Player 1 Score} + \beta_4 \text{ Player 2 Score} \\ &+ \beta_5 \text{ Surface Type} + \beta_6 \text{ Tournament Level}\end{aligned}$$

where "Winner" denotes whether player 1 wins or lose.

6 Model Results

Table 1 below summarizes the model accuracy (in %) for each of the models mentioned above when we we set the thresold of $p = 0.5$ to determine if player 1 wins.

Model 1	Accuracy (in %)
Model 1	64.97%
Model 2	64.22%
Model 3	64.97%
Model 4	64.10%
Model 5	65.09%

Table 1: Model Accuracy Result

as we can observed model 5 has the highest accuracy of 65.09%, however there are no significant difference in the model accuracy across the 5 models. Since our accuracy are based on the threshold assumption of $p = 0.5$ we are interested to find out the change in the model accuracy if we change the threshold.

For this execerise we used model 5 prediction with different p values. Figure 2 graphs out the accuracy of the model across different p values, we can clearly observe that as the p value decrease so does the accuracy model and that between the higher p values (0.5, 0.6) there are no significant differences in the model accuracy.

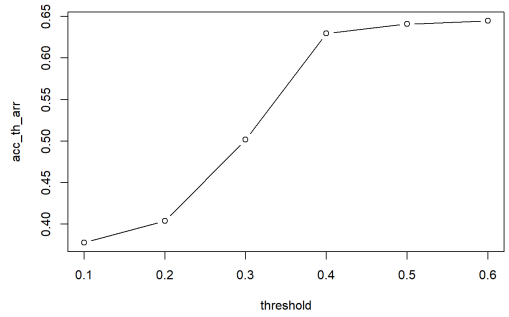

Figure 2: Model Accuracy with Different p threshold

To obtain a better insight on the prediction result breakdowns we created the confusion matrix for each of the model as tabled in Table 2. A common occurence across the confusion matrix is the ability of our models to close to perfect predict a player losing, however it severely lack the capability to correctly predict the chance of a winner. In such a case increasing the p value may balance out the predictions between lossing or winning.
All of this implies that our models may not be a good fit of the dataset, to check this we produce the

Model	Prediction	Reference 0	Reference 1
1	0	505	267
	1	15	18
2	0	515	283
	1	5	2
3	0	510	272
	1	10	13
4	0	515	284
	1	5	1
5	0	507	268
	1	13	17

Table 2: Confusion Matrix

diagnostic plots for each of our models. Figures 3, 4, 5 & 6 display the diagnostic plots for model 5 which has the highest model accuracy. The diagnostic plot confirms that the models are not a very good fit to the data, based on the residuals, and that further improvement to the models or a new model can better suit our needs.

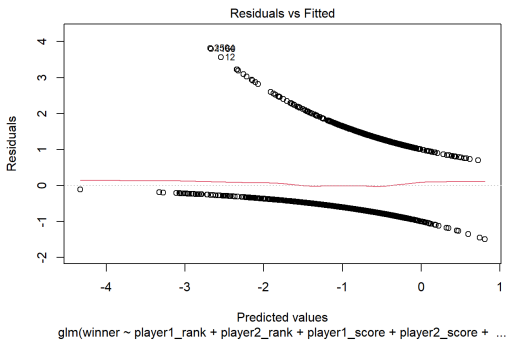

Figure 3: Residual vs Fitted

As a final test of our logistic regression model, we will would like to predict the matches outcome for the subsequent years by using the first 2 years data to train the model. From this we aim to understand if the model effectiveness is reduced as time goes

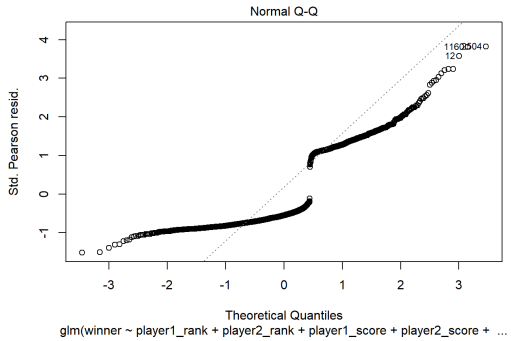

Figure 4: Residual QQ Plot

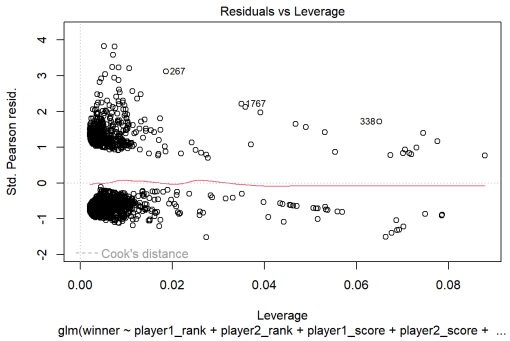

Figure 6: Residual vs Leverage

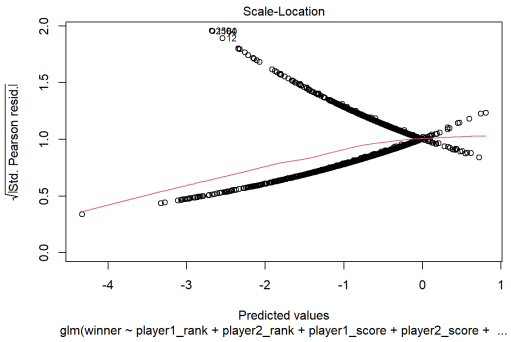

Figure 5: Scale Location

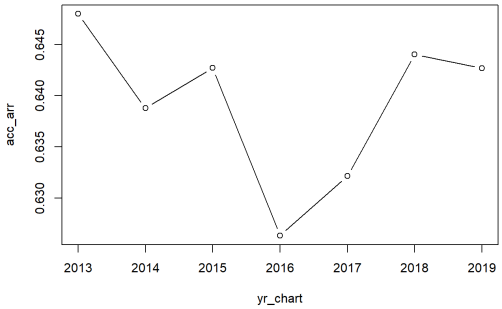

Figure 7: Model Accuracy across years

by and tennis player changes and ranking changes. Figure 7, present the model accuracy across the subsequent years.

Based on Figure 7 it can be observed that the accuracy of the model across the subsequent years are more or less remains the same. This implies that model is neither time dependent nor player dependent, since it relies mostly on the player rankings to predict the outcome of the match. It assumes that most of the uncertainty regarding the player performances are already captured in the player ranks itself.

7 Advanced Models

Based on the results presented in the previous section, our logistic regression models do not provide a satisfactory result in it's prediction power, mainly on it's capability to accurately predict a winner. As a consequent we fitted out more advanced model with hyper-tuning the parameter of the mdeol as well, below are the list of more advanced models that we trained.
Models that were fitted and their performance evaluated based on hyperparmeter tuning or variable selection

- K-nearest neighbors (KNN)
- Logistic regression
- Random Forest

Models that were fitted without hyperparmeter tuning

- Linear Discriminant Analysis (LDA)
- Quadratic Discriminant Analysis (QDA)
- Naive Bayes

On top of the new models, new variables were added as well in the model. These variables were added

- Hand (Left or Right Handed)
- Player's Height
- Differences Between Player 1 and Player 2
  - Rank Difference
  - Age Difference
  - Height Difference
  - Score Difference

For KNN, LDA, Naive Bayes only the numerical columns were used (we dropped hand, surface & tourney level). For QDA only the 4 differences columns were used as predictors because this model requires no collinearity between variables.

These are 6 models used for classification with different underlying statistician and mathematical assumptions, hence giving different results. With the testing error, we could easily check which would perform the best on new data

Further, based on our previous simple logistic regresion models, we observed that the accuracy of the model more or less remains the same across the years, we decided that for the advance models to utilize the whole dataset allowing us to train the models on more data points.

7.1 KNN

For KKN model training we used Monte Carlo corss-validation 5 times

1. Split further the training set into 80% training and 20% for validation set
2. Fit the KNN model with $K = 1, 3, 5, ..., 71$
3. Recording the misclassification rate on the validation set

After getting the 100 results, we took the mean for all k's, and the results as seen in Figure 8 , K=71 produced the lowest validation error of 0.356. Based on the best K value we re-train the whole

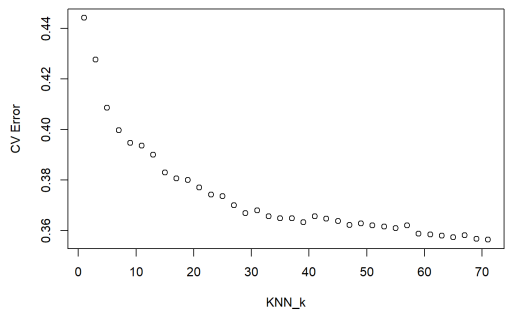

Figure 8: KNN CV Error

training dataset based on the selected K value and predict on the test data. The results of the prediction is tabulated in Table 3

Pred\ Ref	0	1
0	2867	1496
1	216	162

Table 3: KNN Confusion Matrix

7.2 Logistic Regression

For logistic regresion we attempt to improve our previous models by implementing the backward selection algorithm that will remove variables from the full model based on the AIC value. We first implemented the full model with all the variables

$$\begin{aligned}\text{Winner} = \beta_0 &+ \beta_1 \text{ Player 1 Rank} \\ &+ \beta_2 \text{ Player 2 Rank} + \beta_3 \text{ Player 1 Score} \\ &+ \beta_4 \text{ Player 2 Score} + \beta_5 \text{ Player 1 Age} \\ &+ \beta_6 \text{ Player 2 Age} + \beta_7 \text{ Player 1 Hand} \\ &+ \beta_8 \text{ Player 2 Hand} + \beta_9 \text{ Surface} \\ &+ \beta_{10} \text{ Tournament Level}\end{aligned}$$

Based on this model, we implement the backward selection to select the most suited variables in the model. The results of the model is provided in Table 4.

Pred\ Ref	0	1
0	3003	1573
1	80	85

Table 4: Logistic Regression Confusion Matrix

7.3 Random Forest

For random forest we used the out-of-bag error estimator to fine tune 2 hyperparmeters on the training set

- Number of variables sampeld as candidates for each split = 1,2,...,16
- The minimum size of terminal nodes = 1,3,5

For each iteration, 400 trees were created. We found out that the lowest error model was the model with node size set to 3 and the number of predictors is 1 with an out-of-bag error of 0.3579. Further based on the variable importance, the most important terms in the model, based on the Gini index, are the players score, rank and age.

		Confusion Matrix			
Model	Test Error	True Neg	False Pos	False Neg	True Pos
KNN	36.1%	2854	1483	229	175
Logistic Regression	34.9%	3003	1573	80	85
Random Forest	35.0%	3057	1631	26	27
LDA	35.3%	3029	1619	54	39
QDA	47.6%	1226	400	1857	1258
Naive Bayes	43.5%	1590	583	1494	1075

Table 5: Confusion Matrix All Models

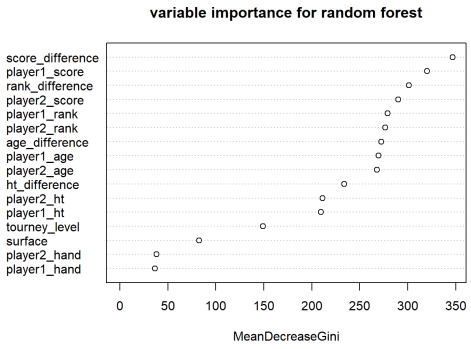

Figure 9: RF Variable Importance

Based on the tune model, the predicted test data result is provided in Table 6

Pred\ Ref	0	1
0	3044	1632
1	39	26

Table 6: Random Forest Confusion Matrix

7.4 Bayes, LDA & QDA

As previously mentioned for these models, we do not hypertune the parameters, instead we only used the numerical columns in the dataset to train the model.
The results of the models are tabulated together with the other models in Table 5 .

7.5 Model Results

The Table 5 provides a consolidated results for each of the above advance models. Generally speaking in the testing set in 35.0% of the time the lower ranked (rank 1 is better than rank 2 etc.) won, so even with just always assuming that the lower ranked will win will yield similar results to these models. This means that rank and score (which are very strongly negatively correlated, as can be seen in the correlation plot provide in Figure 10) explain most of the variance basically among all the variables in the dataset already.

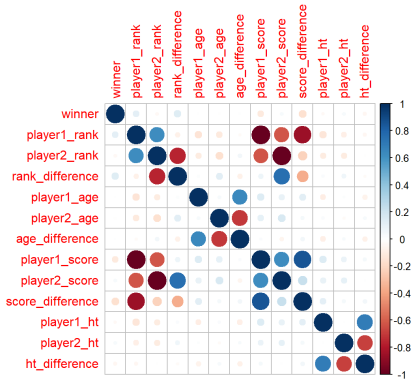

Figure 10: Correlation Plot

Since none was able to outperform simply choosing the lower ranked player, it seems that for example height, the hand (left or right-handed), and age is not as important variables as we have previously thought they are.

8 Conclusion

A noticeable observation of the confusion matrix provided in Table 6 is the different profiles of the confusion matrix for different models. For example the QDA and Naive Bayes model provides a significantly higher true positive compare to the rest of the models, however this comes with the cost of a lower true negative and a higher false negative.

Nevertheless this profile brings up an interesting future research topic, instead of using one of the models as a predictor, why don't we choose a combination of the models. To achieve it is worth exploring ensemble methods since it might provide enhanced results. Such example will be stacking, where the weight of each model are estimated via cross-validation, not simply using majority-vote. This method will put a lower amount of weights on model that have lower cross validation accuracy, hereby balancing out between the advantage and disadvantage of each model.
Future research ensemble method

9 References

References

[1] Jeffsackmann github: tennis_wta. https://github.com/JeffSackmann/tennis_wta. Accessed: 2022-09-20.

[2] Women's tennis association matches. https://www.kaggle.com/datasets/gmadevs/wta-matches. Accessed: 2022-09-20.

[3] David Forrest and I.G. Mchale. Anyone for tennis (betting)? *European Journal of Finance*, 13:751–768, 12 2007.

[4] Ondřej Hubáček, Gustav Sourek, and Filip Železný. Exploiting sports-betting market using machine learning. *International Journal of Forecasting*, 35, 02 2019.

[5] Sascha Wilkens. Sports prediction and betting models in the machine learning age: The case of tennis. *journal-of-sports-analytics/jsa200463*, 2001.