Match Outcome Prediction in Singles Tennis

Setup

Singles tennis is a sport in which two opposing players compete against each other, and a binary outcome is achieved, with the player winning the best of three or best of five sets being awarded with the win for the match. As with any sport, betting is available for almost all professional tennis matches, where pundits and spectators alike make predictions on the winner of the match. The outcome of a correct prediction is then tied to the value of the spectator's wager multiplied by the odds of the player winning.

The objective of our study is to determine the feasibility of creating a classification model on predicting the outcome of tennis matches, such that predictions made are financially profitable with betting odds in the long run, assuming a $1 wager is placed for each tennis match. Specifically, our predictions will be focused on whether the player who lost the first set, is able to recover and end up winning the match. A prediction of win, will therefore indicate that the player whom lost the first set is predicted to win the match, and subsequently a prediction of lose, equates to the player continuing to lose the remaining sets and hence lose the match.

The dataset used for this study is obtained from Tennis-Data.co.uk (Data Files: All Competitions n.d.), which provides us with historic match data, dating back to 2001, and containing features such as the names of the players playing, their respective ranks at the time of match, the current round within the tournament, court surface, the betting odds before the match, amongst many others. It's important to note that the 2020 dataset will *not be included* in the training of the model due to potential confounding variables being present, such as the lack of crowd presence within tournaments due to COVID-19, whom may have otherwise allowed a certain subset of players the ability to tap into the energy of the crowd to improve performance (Futterman 2020).

The success of this study will be based on the ability to develop a classification model that is financially profitable with the predictions made on the outcome of the match, as previously outlined above. This is quantified through a custom loss function such that incorrect predictions result in a loss of $1, and correct predictions are multiplied with the adjusted pre-match betting odds. Due to the lack of available data on the actual betting odds after the first set has been played, a conservative approach will be applied to calculate estimated profit/loss of our model. As such, correct predictions that the player will continue to lose will only result in the addition of $1, which is a net profit of $0; taking into account that $1 is also the cost of the wager. The full matrix is illustrated in Appendix, Table 1.

In addition, the f1 score of the positive class will be used to quantify performance between classification models, due to the nature of our imbalanced dataset in conjunction with betting odds placing a greater financial value towards the minority class; where the minority class referred to, are the instances where the player has lost the first set but ends up winning the match. As such, *greater focus is placed on the ability to predict situations in which the player who lost the first set*, is able to win the match.

Approach

Feature Engineering

In conducting this study, cleansing and transformation functions have been performed to ensure that our dataset is structured and contains the right set of features for our classification models. Specifically, as outlined in Assignment 1, the original dataset is structured based on the winner and loser of the match, where columns are provided as "Winner" and "Loser" with corresponding columns prefixed with "W", "L" denoting the respective stats of the winner and loser players. The transformant function applied ensues that the data structure is agnostic of the match outcome, where we instead create "Player" and "Opponent"

feature sets, with subsequent columns prefixed with "P" and "O". The primary difference is that the person in "Player" column will always be the person who lost the first set of the match. This therefore allows us to create an attribute for the dependent variable called "Has_Won" indicating the match outcome from the perspective of the person in "Player" column, with 1 indicating a win and 0 as a loss.

Dimension reduction is also critical in ensuring the quality of the data used for the models. One of the ways we achieve this is the aggregation of related features within our dataset. Specifically, we have player's rankings and the number of games won in the set which is originally described in two columns each, one for each player. However, we can treat this data as being relative to each other, where the player is either n ranks higher or lower than their opponent, and the same can be said for the difference in games won during the first set as well as the momentum/form between the players. Therefore, we aggregate these columns to create the following differential columns: Rank_Diff, Set1_Diff and Momentum10_Diff.

The second way we achieve dimension reduction is through feature selection. Through data exploration, we can determine features such as the player names, location and name of the tournament, and the date amongst others as being insignificant for our classification model. These features will therefore be dropped since it will cause our dataset to be sparse during the creation of dummy variables for these of these nominal features. Moreover, due to the absence of the average betting odds for the dataset prior to 2010, we will instead calculate the average betting odds ourselves into a feature, and dropping the betting odds sourced from each individual source/company. The result of this is the aggregation of multiple betting odd columns into two columns containing the mean odds for the player and their opponent.

One thing to note is the lack of ATP points data for the 2004 dataset and earlier. As a result, a significant portion of our dataset is missing the ATP points feature which would render these entries unusable for our models if left within the dataset. However, since rankings are derived directly from the player's ATP points (ATP n.d.) feature, then it is considered safe to remove the points data as long as we retain rankings, since it captures the same information. After feature selection, we are able to reduce the number of dimensions from 54 to 11, prior to the creation of dummy variables which would increase our attribute count to 23. At this stage, we have achieved a state of our dataset in which we've removed all unnecessary features and aggregated relative data, in which we're ready to use this as the base dataset for our experiments.

Experiment Setup

The classification models considered for this study include logistic regression, random forest, and XGBoost. Logistic regression is chosen amongst the models due to its simplicity in setting a decision boundary to determine its decision in binary classification problems. The use of tree ensemble algorithms is then considered such as random forest and XGBoost, due to their use of bootstrapping within each split, reducing the effects of class imbalance. Furthermore, all three algorithms provide the option to balance class weightings, by placing a greater penalty on the misclassification of the minority class, and therefore better handle our imbalanced dataset.

Prior to comparing the performance of each algorithm, the best approach to the following must first be considered:

1. Data pre-processing: No pre-processing, Standardisation, Standardisation + PCA
2. Further measures to address class-imbalance: None, SMOTE, SMOTE + majority class under-sampling

In order to determine the best configuration possible, based on the options for data pre-processing and class-imbalance measures outlined above, we will first determine the best approach to data pre-processing. Once the best approach has been determined for data pre-processing, the winning selection will then be

included within our pipeline for determining the best approach to addressing class-imbalance. The way we conduct the tests will be as follows:

1. The dataset will be split into training and testing datasets, with a ratio of 0.75/0.25 respectively.
2. Hyperparameter optimisation will be performed for all three classifiers through GridSearchCV with 3-fold cross-validation to ensure that we have the best possible model for each of the data pre-processing approaches above. This means that GridSearchCV will be performed 9 times in total, three per classifier.
3. Once the hyperparameters have been determined for each classifier, the mean average f1 score for the positive class will then be calculated on a 5-fold cross-validation set. The pre-processing configuration that resulted in the highest score for each classifier will then be included within the pipeline when determining the best approach to addressing class-imbalance
4. Hyperparameter optimisation will again be performed for all three classifiers through GridSearchCV, except this time, it will be including the best data pre-processing method from step 2, as well as the use of additional measures for class-imbalance. GridSearchCV will then be performed six times, twice for each classifier; one with SMOTE, the second with SMOTE + under-sampling.
5. Same as step 2, the mean average f1 score for the positive class will be calculated on a 5-fold cross-validation set to determine if additional measures to address class-imbalance is required for any of the classification algorithms.
6. The model with the highest score from step 4 will then be considered as the best classification model for predicting the match outcome, and will finally be tested against our test dataset, which the model would not have seen yet.

Within the configurations mentioned for data pre-processing above, the use of standardisation is included to allow us to centre the dataset to a mean of 0, and compress the features to a standard deviation of 1 (Liu 2020). Additionally, the inclusion of PCA is with the expectation that the reduction of features fed into the model will result in greater accuracy, as PCA produces linear combinations of the features such that the first n components is capable of explaining 95% percent of variance within the dataset, where $n < N$ features.

Furthermore, there has been research done in the generation of synthetic data samples to address class-imbalances, commonly known as Synthetic Minority Oversampling Technique (SMOTE), with positive f1-score when combined with under-sampling of the majority class (Chen, Liaw and Breiman n.d.). For the purposes of this study, we will also consider the use of SMOTE alone, and see how it compares with the recommendation provided in the article above of using SMOTE with under-sampling. The baseline for this step will be the scores achieved by the classifiers without the use of SMOTE and/or under-sampling, where only class weighting was only applied to provide greater penalty for the misclassification of the minority class.

Within the classification algorithms, key parameters will have to either be selected based on its relevance within the experiment or tuned through hyperparameter optimisation. One of the key design decisions include the use of balanced class weightings for all three algorithms. As previously discussed, this allows the model to provide a greater penalty for misclassification of the minority class, as our dataset is imbalanced. This is also considered our base approach to addressing the imbalance. The use of bootstrapping and subsampling is also enabled for both random forest and XGBoost, to prevent overfitting the model to our training data. This means that during each node split when the trees are being built, the data in which the model is fitting against will not always be the same. Another important consideration for our tree ensembles is the way in which we control the depth of the trees. Within XGBoost, we have decided to use hyperparameter optimisation to tune the max depth of the tree, while random forest does not have a max depth, instead relying on the parameter min_samples_split to determine if it can further split the internal node.

Results

The outcome for the first stage of the experiment has been conclusive. With our selection of classification models, the ability to predict whether a player whom lost the first set will end up winning the match, as measured by the f1-score of the positive class, shows that performing standardisation and/or PCA does not improve performance. As demonstrated in Appendix Table 2, the score achieved by logistic regression and XGBoost are roughly the same across all three pre-processing methods, with no pre-processing applied still prevailing. Random forest did achieve a better result with standardisation applied, however the score is only marginal to its score without any pre-processing. As such, to keep the experiment simple, *all three classification models will have no pre-processing applied for the next stage of the experiment*.

During the second and final stage of model selection, we then tested for any improvements that performing additional handling for imbalanced datasets may have. As shown in Appendix Table 3, implementing SMOTE and SMOTE + under-sampling did not appear to have any material impact on the performance of our model. The f1-score achieved for SMOTE + under-sampling is just marginally under the results obtained with the base model, which only has class weightings applied. In saying that, random forest was able to achieve a materially better score, increasing from its previous best of 0.39 to 0.43. Moreover, previous research done on performance improvements when SMOTE is performed with under-sampling compared to just SMOTE alone (Chawla, et al. 2002) does appear to have some weight. Scores for SMOTE with under-sampling being higher than scores with just SMOTE performed, albeit the difference being materially insignificant.

Overall, we can also determine from Table 3 the best classification model that we have built for this study. Whilst not the clear winner, *XGBoost does appear to be the best classifier with no pre-processing applied*, and Logistic Regression also with no data pre-processing applied coming in a close second place. Random forest whilst placing in last place still performed at a level similar to the other two classifiers, particularly when SMOTE and under-sampling is applied within the dataset. Otherwise, random forest with class weightings alone does not seem to handle imbalanced datasets well from observations within this study.

Finally, with XGBoost as the classification model, with no data pre-processing and no additional use of SMOTE nor under-sampling, we are able to conduct a prediction on the test dataset in which the model has not seen before. Shown in Appendix - Table 4, we can see that the model performed in-line with results seen during the 5-fold cross-validation test. When it comes to the positive class, the model is only 34% right, when it predicts that the player will win the match. However, at least it does have a recall score of 65%, meaning that it is at least been able to correctly capture 65% of all the cases where the player did end up winning. This means that the model tends to overpredict that the player will win the match, as shown by its low precision but high recall.

Conclusion

Whilst we have been able to develop a classification model that is better suited to this problem over the other classifiers, the best classification model that we've developed still is not able to generate a profit over the long run. However, it is important to consider that projections for profit and loss of this model is based on a conservative approach where the only way to generate a profit is with the correct prediction of the positive class, where the player whom lost the first set is able to win the match. As shown in Table 1, all other outcomes either result in break-even or a loss in order to be conservative with results.

Therefore, future work on this model should make use of actual betting odds *after the first set has finished, instead of pre-match betting odds*. Furthermore, availability of first set data such as number of points won on the opponent's serve or number of breakpoints created by the player is likely to improve this model, potentially making it more viable. Moreover, Figure 1 in Appendix shows that our model actually results in a greater loss than simply betting on the favourite player to win. At present, this model is not viable.

Appendix

Table 1: Profit and Loss Matrix for $1 Wager in a Match

		Reward	Profit/(Loss)
Incorrect Prediction	Match outcome is opposite of the prediction	$0	**(-$1)**
Correct Prediction	Prediction is player lost. Match outcome is player lost	$1	$1 - $1 = **$0**
	Prediction is player win. Match outcome is player win	$1 * Pre-match odds	**Pre-match odds - $1**

Table 2: Positive Class F1-Score on 5-fold Cross-Validation Results for the Tuned Classification Algorithms with Different Pre-processing Methods Applied

	None (Base)	Standardisation	Standardisation + PCA
Logistic Regression	**0.44623**	0.44469	0.43579
Random Forest	0.39496	**0.39641**	0.34496
XGBoost	**0.44672**	0.44672	0.43344

Table 3: Positive Class F1-score on 5-fold Cross-Validation Results for the Tuned Classification Algorithms with Different Methods of Handling Class Imbalance Applied

	None (Base)	SMOTE	SMOTE with Under-sampling
Logistic Regression: No Data Pre-processing	**0.44623**	0.44352	0.44438
Random Forest: No Data Pre-processing	0.39496	0.32021	**0.43563**
XGBoost: No Data Pre-processing	**0.44672**	0.44372	0.44489

Table 4: Classification report on the predictions of the best classification model (XGBoost)

Prediction	Precision	Recall	F1-score
Lose	0.89	0.70	0.79
Win	0.34	0.65	0.45

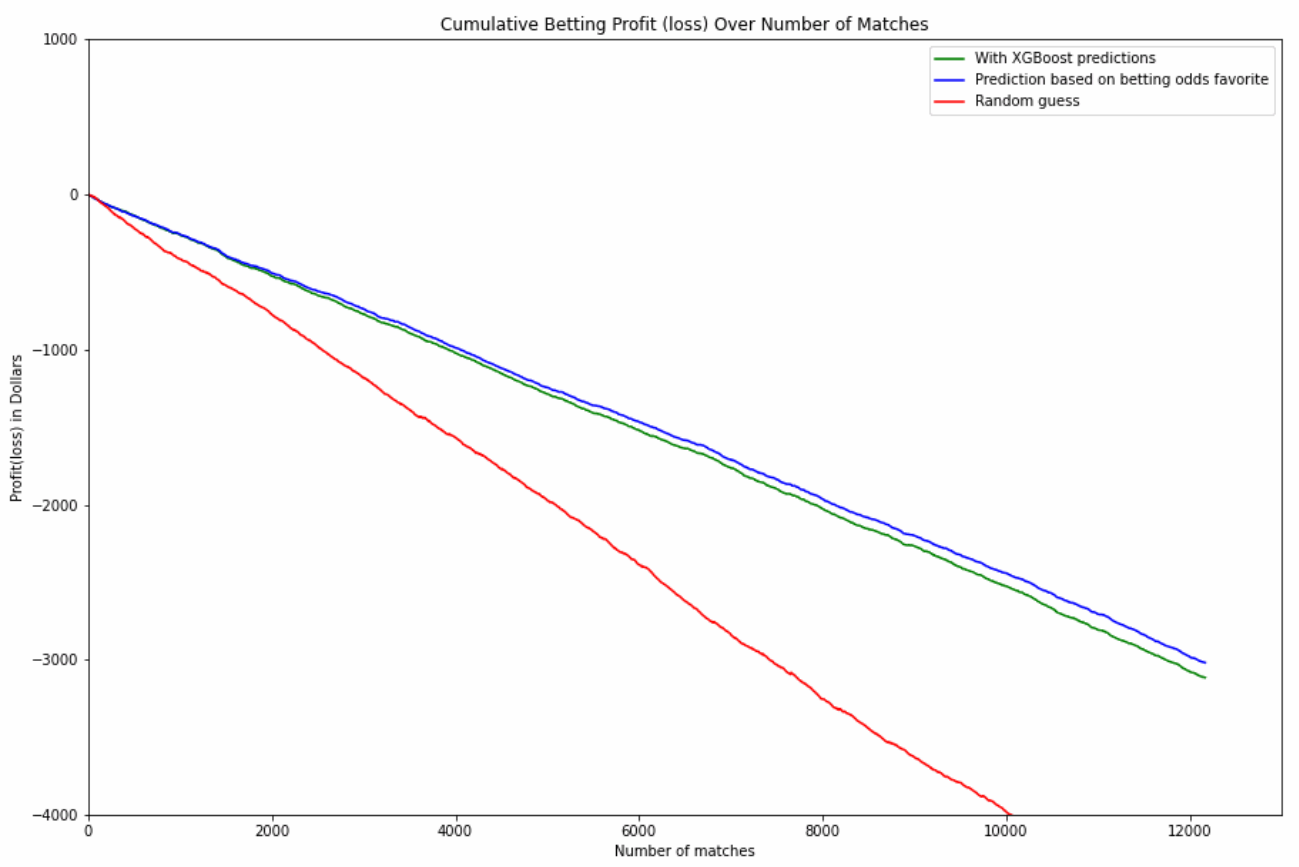

Figure 1: Cumulative Betting Profit (loss) Over Number of Matches, Assuming $1 Wager per Match

References

ATP. n.d. *Rankings FAQ.* Accessed November 12, 2020. https://www.atptour.com/en/rankings/rankings-faq.

Chawla, Nitesh V, Kevin W Bowyer, Lawrence O Hall, and Philip W Kegelmeyer. 2002. "SMOTE: Synthetic Minority Over-sampling Technique." *Journal of Artificial Intelligence Research* 32.

Chen, Chao, Andy Liaw, and Leo Breiman. n.d. *Using Random Forest to Learn Imbalanced Data.* Accessed 20 11, 2020. https://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf.

n.d. *Data Files: All Competitions.* Accessed November 11, 2020. http://www.tennis-data.co.uk/alldata.php.

Futterman, Matthew. 2020. *At the U.S. Open, Silence Is a Sweet Sound for the Underdogs.* Sep 09. Accessed 11 21, 2020. https://www.nytimes.com/2020/09/09/sports/us-open-crowd.html.

Liu, Clare. 2020. *Data Transformation: Standardization vs Normalization.* Apr. Accessed Nov 21, 2020. https://www.kdnuggets.com/2020/04/data-transformation-standardization-normalization.html.