Tennis Prediction Model

Final Report

Shayla Grymaloski

Ricky Huang

Amaan Makhani

John Schriemer

Bryan Travers

1.0 Introduction

The global sports market reached a value of nearly $488.5 Billion in 2018 and has been growing steadily [1]. This growth is attributed to the increase in sports analytics utilizing statistics and, more recently, machine learning. Using machine learning to analyze and predict sports results has proved helpful and lucrative for data scientists, coaches, and bookkeepers alike. In particular, prediction models using machine learning have been utilized by those in the growing sports betting market, which is predicted to grow to have an estimated market value of USD 155.49 billion by 2024 [2]. The demand for accurate predictive models in these markets has prompted a new problem space for machine learning, with many challenges yet to be solved.

Tennis poses some unique challenges concerning machine learning. With three different playing surfaces, varying climates and conditions, and diverse playing styles and strategies, the potential for noisy data is nearly guaranteed. Also, it has been difficult to predict a match outcome when any tennis match's momentum can swing in the matter of a few successfully placed strokes. David Foster Wallace has well-described tennis' unpredictability as "Just one single shot in one exchange in one point of a high-level match is a nightmare of mechanical variables". The large number of variables defining a match has made the problem of using machine learning to predict match outcomes intriguing and worthwhile for our team.

2.0 Problem

Our team used a well-established ATP tour dataset from Kaggle and trained the selected models described in **Section 4** to predict match and tournament outcomes [3]. In particular we predicted matches from 2000-2017 and then predicted the 2019 Australian Open. We then compared the results with the actual outcome of these matches and tournaments. The features we chose to use are detailed in ***Figure 1 and 2*** below. The features of each match are detailed in ***Figure 1***. The whole dataset was then used to create player specific features that used the columns from the original dataset and produced the player's averages over the timeframe. These calculated features are described in ***Figure 2***. The columns that were removed from the initial dataset and the cleaning process are described in **Section 2.1**.

We chose this dataset because there was a significant degree of data spanning several years and many matches. This dataset has also existed for multiple years and is stable. The hope for using this dataset was to discover factors that continuously impact tennis matches and to train our models to pick up on these trends.

Feature(s) name	Description
surface	The surface in which the match is played. The surface can be hard, grass, clay, or carpet.
best_of	The maximum number of sets played.
players_rank	The players rank.
name	The players name. Only used for comparison of results for not provided to the model.
player_id	Used to retrieve the player's statistics from the calculated player features table.
players_age	The players age.
players_ht	The players height.

Figure 1: Pre-match features.

Feature(s) name	Description
Avg 1st In	Average first serve in percentage.
Avg 1st won	Average first serve winning percentage.
Avg 2nd won	Average second serve winning percentage.
Avg SvGms	Average number of games played on serve.
Avg aces	Average number of aces.
Breakpt_ratio	The ratio of the total break points lost and saved. A higher value indicates a higher probability of holding their service. (Breakpoints saved / Breakpoints faced)
Avg df	Average number of double faults.
Avg svpt	Average serving percentage.
Score_ratio	The average ratio of the total number of games won by the loser and the total number of games won by the winner. A higher value indicates a closer match. (Loser games won / winner games won)
Avg minutes in win	The average duration of the matches won in minutes
Hand	The players hand, right (1) or left (0).

Figure 2: Player Features.

Initially, we decided to focus on the variation of platforms. This goal was then changed to varying models rather than varying platforms. We used Sklearn with a variety of models, as described in this report, to generate this final report and the presentation. We chose to move in this direction because we felt it would be more useful to spend time on creating, training, and evaluating our models rather than spending time to gain proficiency in TensorFlow. As previously mentioned, we modified our plan by using the dataset to gather player statistics and then removing

those columns in the original dataset. This is because the model could not process rows with multiple player's data within it.

2.1 Data Cleaning

The ATP Matches dataset contains details about ATP matches played since 1968; however, detailed match statistics are only available for matches after 1990. Since our group intended to explore what statistics dictate tennis match results in recent years, we only focused on the matches played from 2000 to 2017. Any columns of missing data were either filled with the appropriate values (0, median, or mode) or the row dropped. One of the challenges our group faced was dealing with the match scores. Tennis has a unique scoring system, playing sets up to 6 or 7, and playing best out of 3 or 5 sets. To combat these inconsistent scores, we created a new column, named score_ratio, which takes the loser's number of games and divides them into the winner's games. The higher the ratio, the closer the match. We have also used the original dataset to produce average statistics about players over this time frame. These stats were created through excel and using columns from the original dataset. The new calculated player features are described in **section 1.** The next challenge we faced was dealing with both the loser and winner data in each row. We added both player's calculated statistics in each match and randomized the target by rotating the data to avoid the models picking up on a consistent target value.

3.0 Background Info

When our group discussed the problem of predictive sports analysis, we found a few different reports and papers from others attempting to solve the same problem. Many of the papers encountered had an objective to predict the winner more accurately than professional bookmakers [1][2]. The papers found had a range of accuracies reported, from the mid-50s [3], up to 75 [5] and 85 percent [4] with thorough models completed by both professors and researchers. As such, our group aimed to have a model that could predict an ATP match's outcome with an accuracy above 65%.

Figure 4 displays a correlation heat map for all columns of the dataset. The four pockets of higher correlations are serving, returning, and match statistics. This can be attributed to the fact that a higher first-serve percentage and ace count correlates to more won serving games. On the other hand, areas with low correlations are player-specific items like name or country.

Our group has split the tasks up explicitly after we completed our progress report. Amaan was in charge of experimenting with an ANN model with non-linearity and creating the player dataset. Similarly, Shayla examined a linear ANN performance. In addition to cleaning the dataset, John implemented a random forest. Ricky was responsible for investigating the accuracy of a Naive Bayes predictive model. Bryan explored the logistic regression and SVMs models. Upon completion of the necessary experiments, the report was written, and all members created the presentation.

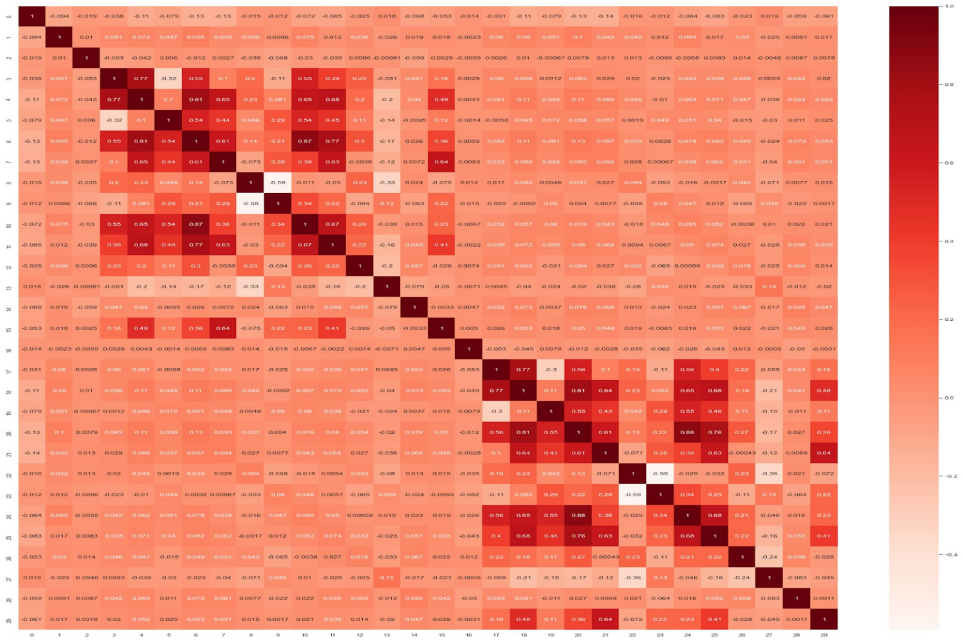


Figure 3: Correlation Heatmap for Modified ATP Dataset

4.0 Experimental Results

As mentioned above, each member of our team was responsible for developing their supervised learning model. After completing parameter tuning, most of the models reported accuracies near 65% when predicting the winner of a given tennis match. This section outlines the initial steps taken to implement the models and compares them against one another.

4.1 Logistic Regression

The first model we examined was a logistic regression model using an L2 regularization penalty. Experiments were run for a variety of training and test sizes, with test sizes ranging from 10-30% of the total data set. The regularization constant was varied using logarithmic spacing over a 100 model sample. **Figure 5** depicts a subset of the 100 models created where the most variability in reported error occurred. As the figure shows, the highest accuracy obtained by logistic regression was 66.18%. This result occurs as C approached .00065.

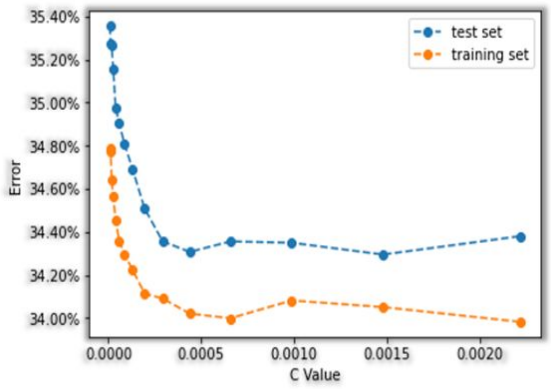


Figure 5: Varying C vs. Error

4.2 Random Forest

The next model used to predict match winners was Scikit Learn's random forest classifier. A decision tree was used to visualize the features deemed most important to the model, as seen in **figure 6**. The decision tree was checked for overfitting and pruned using the cost complexity

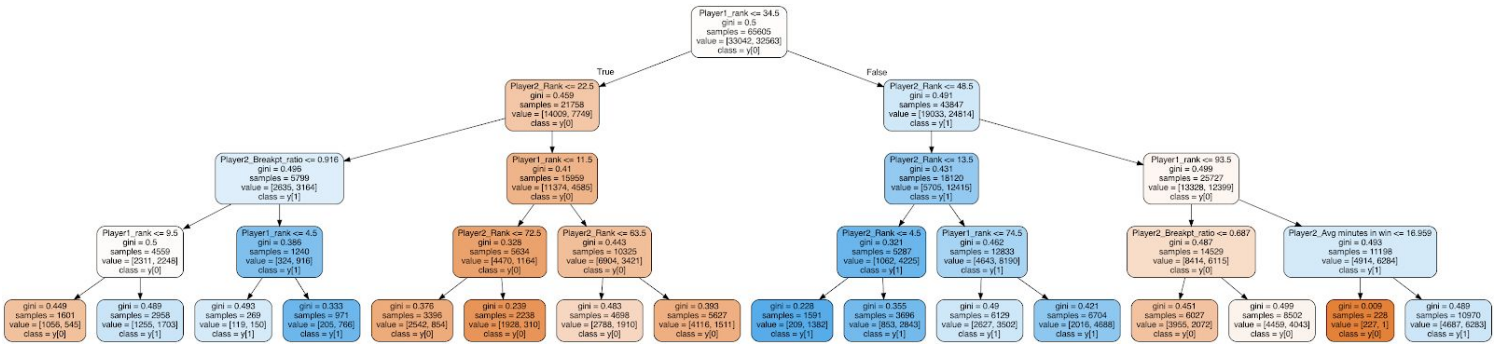


Figure 6: Decision Tree Visualized

Parameter, alpha, against the accuracy of both the testing and training sets as shown in **figure 7**. After this analysis, the optimal depth was determined to be 10 nodes. The next step was comparing the forest sizes that produce the least true error. After running some tests, the optimal forest size was determined to be 82 trees. The combination of tree depth, alpha value, and forest size produced a final match prediction accuracy of 67.87%.

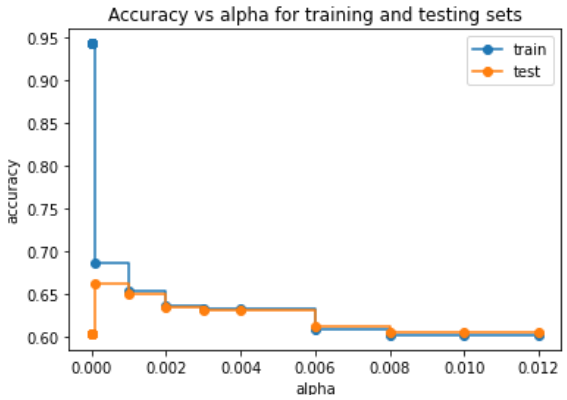


Figure 7: Cost Complexity Parameter vs Accuracy

4.3 Naive Bayes

The third model we used was scikit-learn's Naive Bayes classifier. Using Gaussian Naive Bayes, **Figure 8** shows the classification report generated from 4 experiments. The experiments conducted were changing the test size to 0.1, 0.2, and 0.3. All the experiments used no randomization.

```
Total number of mislabelled data points from 90 test samples is 31
The test/train size: 0.1/0.9
              precision    recall  f1-score   support

         0.0       0.67      0.66      0.67        47
         1.0       0.64      0.65      0.64        43

    accuracy                           0.66        90
   macro avg       0.66      0.66      0.66        90
weighted avg       0.66      0.66      0.66        90

Total number of mislabelled data points from 180 test samples is 73
The test/train size: 0.2/0.8
              precision    recall  f1-score   support

         0.0       0.50      0.62      0.55        73
         1.0       0.69      0.58      0.63       107

    accuracy                           0.59       180
   macro avg       0.59      0.60      0.59       180
weighted avg       0.61      0.59      0.60       180

Total number of mislabelled data points from 270 test samples is 109
The test/train size: 0.3/0.7
              precision    recall  f1-score   support

         0.0       0.57      0.62      0.59       129
         1.0       0.62      0.57      0.60       141

    accuracy                           0.60       270
   macro avg       0.60      0.60      0.60       270
weighted avg       0.60      0.60      0.60       270
```

Figure 8: Classification report

From **Figure 8**, we can see the accuracy is around 60% to 66%. After running the experiment multiple times, the best accuracy is around 64%. **Figure 9** is the graph of the learning curve produced using Naive Bayes with varying training sizes.

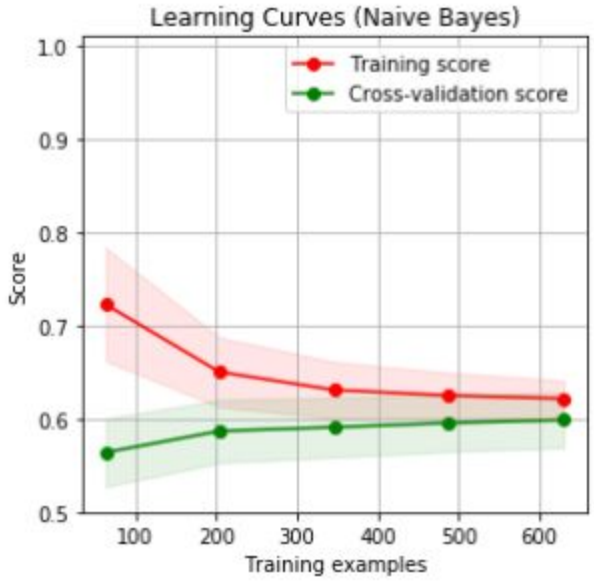


Figure 9: Training Example vs. Accuracy of Naive Bayes

As seen in **Figure 9** the training accuracy and cross validation accuracy converge together within the range of 60% to 66% with an average accuracy is around 64 percent.

4.4 Support Vector Machine

The support vector machine performed the worst out of the models explored in this report. **Figure 10** summarizes the results of a randomized search over possible hyperparameter values with the gaussian kernel. Similar searches were performed over linear and polynomial kernels. The kernels in order of decreasing performance were polynomial, linear, then gaussian. The polynomial kernel yielded the best results of the three kernels when the degree parameter was selected as 61. Accuracy for this model reached 60.1%. A linear kernel came second with an accuracy rating of 57.8%. The gaussian kernel performed the worst on the dataset reporting an accuracy of 55.7%.

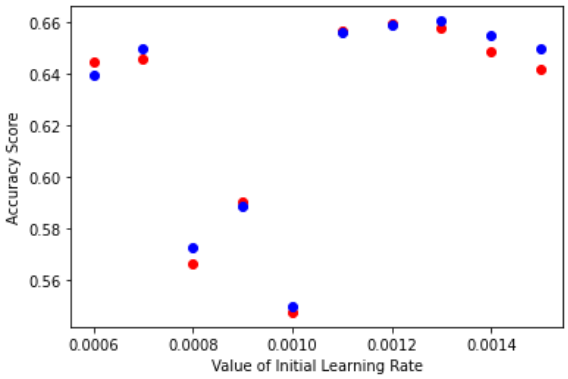
In the case of the gaussian kernel, the 5.7% accuracy performance better than chance was likely due to correct predictions associated with extreme values in the dataset, which quite reliably result in a win, such as when a very highly rated player faced a low ranked player. Although the SVM's performance was generally poor, there was enough variance in the results to conclude that the polynomial kernel yielded better performance than the other two explored models. This result can be explained by the fact that non-linear kernels are more robust to noisy data than linear kernels. Since the data set included many varying weight features, it is unlikely that there was a linear hyperplane that could split the data. Although an optimal non-linear hyperplane was not found, this method still predicted match outcomes better than a linear separator.

```
Model with rank: 1
Mean validation score: 0.557 (std: 0.037)
Parameters: {'C': 598.8312818901775, 'class_weight': 'balanced', 'gamma': 3.507
974882059411e-05, 'kernel': 'rbf'}

Model with rank: 2
Mean validation score: 0.542 (std: 0.065)
Parameters: {'C': 67.80223445968146, 'class_weight': None, 'gamma': 8.991991404
994507e-05, 'kernel': 'rbf'}

Model with rank: 3
Mean validation score: 0.540 (std: 0.011)
Parameters: {'C': 1.0766893427414026, 'class_weight': None, 'gamma': 0.00189580
1568335958, 'kernel': 'rbf'}
```

Figure 10: Top Randomized Searches Over Hyperparameters for gaussian kernel.

4.5 Neural Network without non-linearity

For this model, the dataset was trained using Neural Networks with a perceptron. Hyperparameter tuning was performed on the variables alpha, a comparison of adaptive vs invscaling learning rate, and starting learning rate value. This was done using k-fold cross-validation with a k value of 5. From running experiments by changing the parameters, we were able to find optimal parameters for the best training and test accuracy. The highest accuracy that was produced for the test data was 66.039%. Below are the results from two experiments that were run testing parameters for changing the value of alpha and for changing the initial learning rate with adaptive learning chosen. The red data plots represent the test data result and the blue data plots represent the train data results.



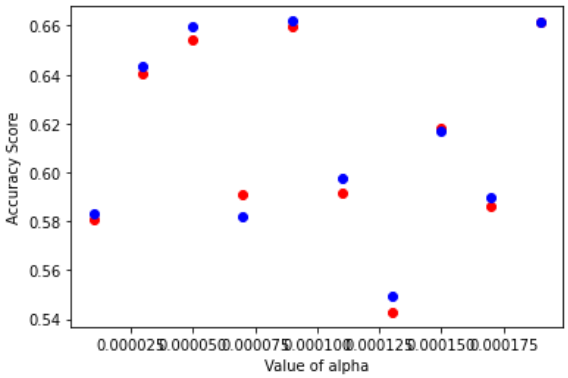


Figure 11: Initial Learning Rate with adaptive learning ***Figure 12: Varying alpha***

The most optimal parameter found were as follows:

Learning rate = ‘adaptive’

Initial Learning Rate = 11/10000

Alpha = 11 *(10**-5)

4.6 Neural Network with non-linearity

The last model explored was an artificial neural network with a non-linear structure. The main experiments run with this model were varying the number of hidden layers, number of neurons in the hidden layers, the activation function, the solver, and the learning rate method.

It was seen through experiments that the number of iterations did not affect the accuracy of the model. It was also seen through experiments that the ideal number of neurons could be selected from multiple ranges in the graph below. After k cross-validating multiple values in this range, 20 neurons was the selected value, this was also guided by past the need for increased model complexity which is required with this dataset.

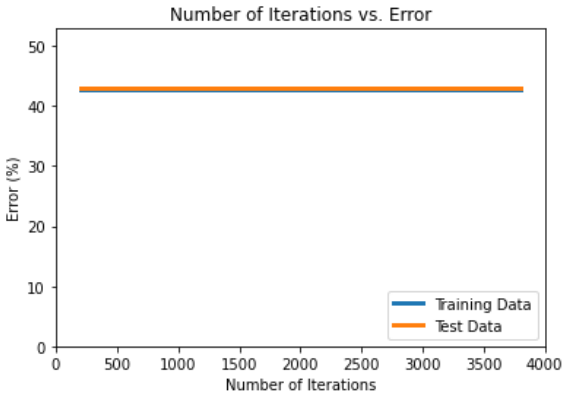


Figure 13: Varying Number of iterations

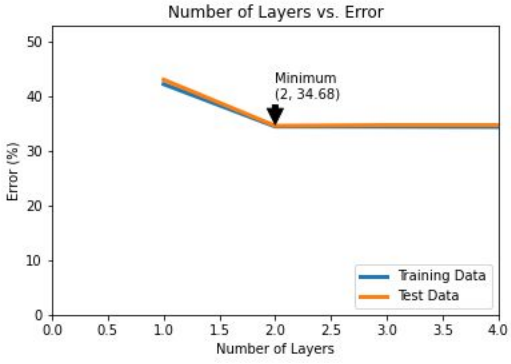


Figure 14: Varying Number of Layer vs error

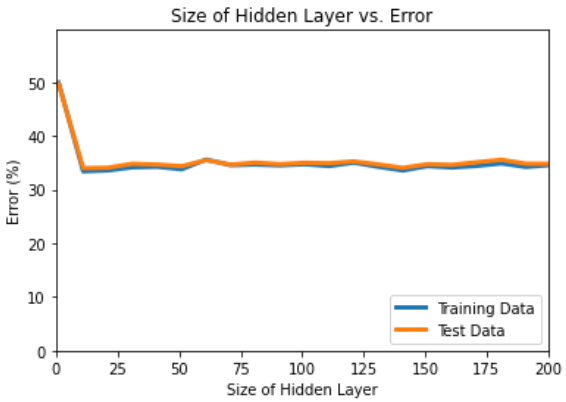


Figure 15: Varying Size of Hidden Layer vs Error

The logistic activation function was compared against the tangent activation function and performed better by 2.11%. The adaptive and inverse scaling learning rate methods were compared which resulted in the adaptive learning method to perform better by 1.02%.

The best parameters after using 5-fold cross-validation were two layers with 20 neurons in each layer, adaptive learning, and the use of a logistic activation function. The best 5 fold cross-validation accuracy was 65.13% after the parameters were hypertuned.

4.7 Tournament Predictions

Each model was tuned using 5 fold cross validation to allow for evaluation and comparison. After cross validating the models they were used to predict the Australian Open 2019 tournament starting from the round of 128. The accuracy of each predicted round and the tournament average is below in **Table 1**. The overall tournament accuracies were ranked to provide an idea of the best predicting model.

Model	Accuracy(%)								Rank
	R128	R64	R32	R16	Q	SF	F	Overall	
Logistic Regression	65.63	28.13	16.67	37.5	25	50	100	47.24	**6**
Random Forest	84.38	37.5	27.77	12.5	25	50	100	59.05	**2**
Naive Bayes	68.75	40.62	27.77	37.5	50	100	100	55.11	**4**
SVM	70.31	40.63	22.22	25	25	50	100	52.75	**5**
Linear Neural Network	75.00	50	50	50	50	100	100	64.57	**1**

Non-Linear Neural Network	71.88	43.75	33.33	37.5	50	100	100	58.28	**3**

Table 1: Results for each round of 2019 Australian Open

In the above table, we can see that the best tournament model prediction model was 64.57% using the linear ANN. Below are figures comparing the prediction accuracy of all the models for game and tournament prediction. The best match predictor was the random forest model with an accuracy of 67.87%. An observation made between **figures 16 and 18** is that logistic regression was an effective match predictor but a poor tournament predictor. After analyzing the results produced we came to the realization many evenly matched matches were predicted incorrectly. The predicted brackets of each model are shown in **Appendix A**.

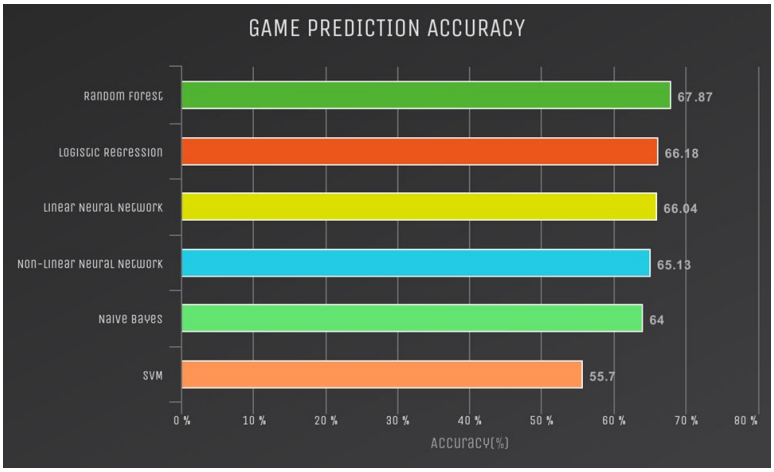


Figure 16: Overall results for match predictions

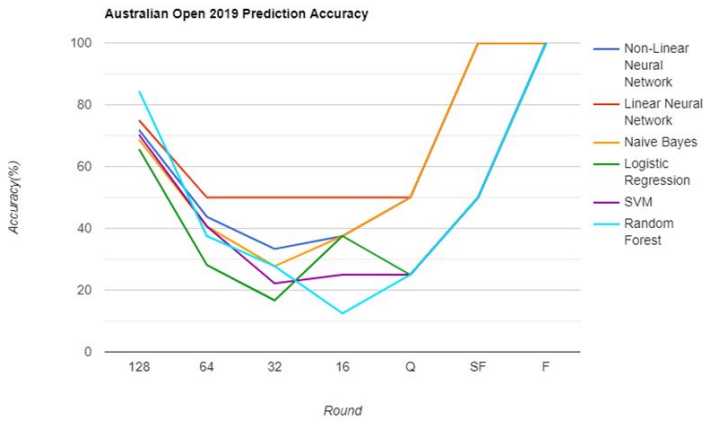


Figure 17: plotted results for each round of 2019 Australian Open

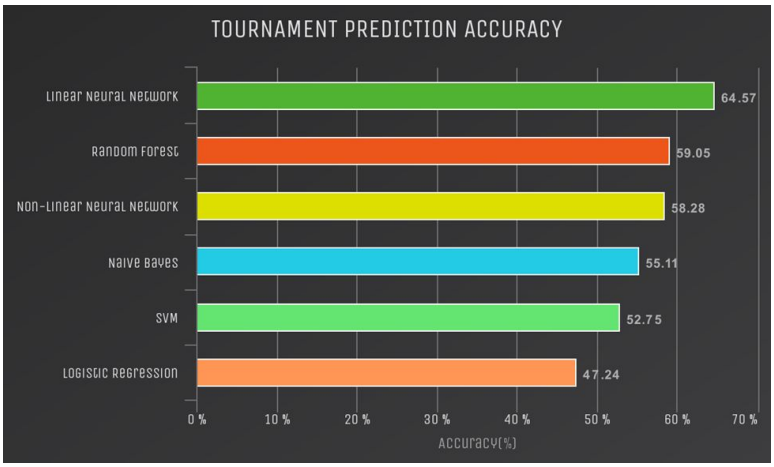


Figure 18: Overall Accuracy of Predictions for the 2019 Australian Open

4.8 Model Comparison and Future Work

Using the six models to predict a tournament outcome and match outcomes, we can see the results in **section 4.7.** For tournaments, the linear neural network had the highest accuracy and logistic regression had the lowest accuracy, while the other four models fell within the range between 52 to 59%

However, the prediction accuracy for matches was best predicted by a random forest obtaining the highest accuracy of 67.87% while the SVM had the lowest accuracy of 55.7%. Overall, we achieved a game prediction accuracy of around 65% from the results. This met our goal set from the time of our proposal.

For future experiments, we would be interested in analyzing and experimenting with more dimension reduction to obtain a better prediction accuracy, possibly. This also includes investigating ways to improve match predictions for evenly matched players, probability of a dark horse in a tournament and predictions on tiebreakers. We would also look into more specific factors that could affect the game's flow like weather, player fitness, and player's head-to-head records. As for tournament prediction, instead of doing only one tournament, we would be interested in analyzing all grand slams within a given year.

5.0 Conclusion

From the experiments, we have analyzed and obtained an average of 65% prediction accuracy using the models we experimented on and can predict a tournament outcome with an average of 55% accuracy. Our goal for future experiments would be achieving an accuracy of 70% or better since the best prediction accuracy we obtain was 67.87% from the random forest model.

As discussed in **section 4.8**, there is still room for future experimentation to obtain better accuracy, and including other factors to be able to predict more complex situations that happen in a real ATP tennis tournament.

6.0 References

[1] ltd, R., 2020. Sports Global Market Opportunities And Strategies To 2022. [online] Researchandmarkets.com. Available at: <https://www.researchandmarkets.com/reports/4770417/sports-global-market-opportunities-and-strategies?utm_source=BW&utm_medium=PressRelease&utm_code=ctvc8g&utm_campaign=1244426+-+Sports+-+%24614+Billion+Global+Market+Opportunities+%26+Strategies+to+2022&utm_exec=joca220prd> [Accessed 20 June 2020].

[2] R. ltd, "Sports Betting Market by Platform, by Type, and by Sports Type: Global Industry Perspective, Comprehensive Analysis, and Forecast, 2017-2024", Researchandmarkets.com, 2020. [Online]. Available: https://www.researchandmarkets.com/reports/4853933/sports-betting-market-by-platform-by-type-and. [Accessed: 20- Jun- 2020].

[3] S. VM, "ATP matches, details of the ATP matches since 1968", *Kaggle.com*, 2020. [Online]. Available: https://www.kaggle.com/sijovm/atpdata. [Accessed: 08- Jul- 2020].

[4] Cornman, A., Spellman, G. and Wright, D., 2020. *Machine Learning For Professional Tennis Match Prediction And Betting*. [online] Stanford University. Available at: http://cs229.stanford.edu/proj2017/final-reports/5242116.pdf

[5] Sipko, M. and Knottenbelt, W., 2015. *Machine Learning For The Prediction Of Professional Tennis Matches*. [online] Imperial College London. Available at: https://www.doc.ic.ac.uk/teaching/distinguished-projects/2015/m.sipko.pdf

[6] Kaggle., 2016. *Match Outcome Prediction in Football* [online] Kaggle.com. Available at: https://www.kaggle.com/airback/match-outcome-prediction-in-football

[7] Gu, Wei & Saaty, Thomas. (2019). Predicting the Outcome of a Tennis Tournament: Based on Both Data and Judgments. Journal of Systems Science and Systems Engineering. 28. 317-343. 10.1007/s11518-018-5395-3.

[8] A. Somboonphokkaphan, S. Phimoltares, and C. Lursinsap. Tennis Winner Prediction based on Time-Series History with Neural Modeling. IMECS 2009: International Multi-Conference of Engineers and Computer Scientists, Vols I and II, I:127–132, 2009

Appendix A- Tournament Brackets

Tournament Prediction Results Using a Neural Network with Linearity

Tournament Prediction Results using a Random Forest