Factors in Tennis

John Soderstrom

May 10, 2020

1 Introduction

Clean, Final Notebook.
Original Notebook.

Data from the Association of Tennis Professionals replaced the original choice of Ocean Wave data off the coast of Australia. The new choice provided a relative surfeit of information in both the number of records and number of inputs for the model. It further benefits from an increased personal interest and a more concrete understanding of a question to be answered.

Which player will win?

Given information about two players in a match and information that applies to both, how accurately may we predict the winner? How early may we predict the winner? Once a match begins, new data with every point can add to the model's accuracy. Unfortunately, the value of the prediction often drops as the match progresses, and a close game will likely stop the prediction from improving as quickly.

The first model created attempts to predict the winner of a tennis match using all types of inputs, including what can only be known after a match completes and the prediction has the least value. The second model attempts to predict the winner before the match begins, with less information and a lower success rate. Each model can predict with greater accuracy than simply picking one side each time, thanks in part to a near perfect balance in outputs. I would like to model data from games in progress to test the accuracy when partial data is known, but I do not have any such training data.

2 Data Preparation

After removing unnecessary columns and rows with missing values, over 40000 rows remained with 36 input columns. There are three general categories of inputs the dataset provides: general information about the match, player specific information known before a match, and player specific information known fully after a match. 6 more inputs were removed from the original models after testing for feature importance; the inputs were selected from the first category, including the year and month. They were previously kept in case a trend existed when combined with other inputs, but they were originally expected among the weakest features.

While not covered in previous reports, another model will attempt to predict the winner without the information from the last category. The number of inputs reduce from 30 to 12 and the accuracy with a basic model drops immensely, despite remaining stronger than a model that always predicts one option or the other.

The original dataset always set the winning player's information first and the output used for this model did not exist. Before creating a model, half of the rows swapped the winning and losing player's information and added an output that marked if the first or second player in the row won. As a result, the outputs were balanced perfectly, 50% being 0 and 50% being 1. Randomly splitting the data in multiple cases nearly perfectly maintained that balance in both training and validation

sets. The worst observed split contained approximately 52% of one output and 48% of the other.

General Information

The weakest category, general information contains data common to both players. These inputs provide minimal value on their own, but may have an impact when combined with others. All 6 inputs removed from the original models come from this category (year, month, number of players, round number, length of match (in minutes), length of match (in sets)). The size of this group dropped dramatically as a result.

- Court Surface
- Number of service games

Player Specific (Before Match)

The most valuable inputs from this section are each player's rank and rank points. Both provide a similar measure of a player's skill, but while rank provides a constant difference from one rank to the next, rank points allow the gap between each rank to change. The other inputs are weaker, but still provide more useful information on their own than the previous category.

The following inputs exist once for each player.

- Dominant Hand
- Height (cm)
- Age
- Rank
- Rank Points

Player Specific (After Match)

While these inputs tend to provide the strongest effects in correctly classifying and predicting results, they are not known until after a match ends. The accuracy of a basic model that includes these inputs is far stronger than one without them, but it is more limited. If data about a match were known but the victor was not, these inputs would be useful. However, inputs from a match in progress could still provide a stronger prediction of which player will win. They will remain used for the model as previous reports, but will be ignored while building a second model that is limited to data known before the match.

The following inputs exist once for each player.

- Number of Aces
- Number of Service Games
- Number of Service Points
- Number of First Serves In Play
- Number of First Serves Won
- Number of Second Serves Won
- Number of Double Faults
- Number of Break Points Faced
- Number of Break Points Saved

Final Processing and Normalization

Non-numeric data is converted to integers (such as the court surface types). After conversions, addition of the output column, and reversals of player information are complete, dataframes are

converted into numpy arrays. Shuffled arrays are split into training and validation data at a 70/30 ratio.

Normalization first reduces training data values in all inputs by their mean, placing the mean of each input at 0. Each input is divided by their standard deviation, placing all inputs on the same scale. All values are now placed according to their number of standard deviations away from the mean of the input. The same mean and standard deviation are used to normalize the validation data, ensuring all values are placed on the same scale. Inputs with a much larger scale will no longer overshadow other inputs, allowing each similar influence over the weights of the model.

This process is repeated for a second dataset that removes inputs known only after a match ends. While predictions are expected to be weaker, the model could be used before the match takes place and provide an early prediction.

3 Modeling

Binary classification is the best fit for the data, as the output is either a 0 or 1 depending on which of the two players wins a match. Out of accuracy, precision, recall, and the f1-score, accuracy is the best measure.

The three measures other than accuracy each ignore true negatives. In many applications, true negatives drown the useful information. In this case, both true negatives and true positives provide the same answer: the winning player was correctly chosen, whether it was the first or the second in the row. If accuracy were not an option, the f1-score would provide the most valuable information. Similarly to the correctly predicted points, false positives and false negatives provide the same answer: the incorrect player was chosen. The f1-score includes both false positives and false negatives, while precision and recall consider only one of them.

Modeling Results

Figure 1 provides the best learning curves I found during modeling. The curves are smooth and show no signs of overfitting; validation accuracy does not begin to severely decrease as training accuracy increases. The inputs from category three, containing information known only after a match, allow a great deal of accuracy from the predictions. Recall that the outputs are nearly perfectly balanced, and during this modeling process they were split roughly 49% to 51%.

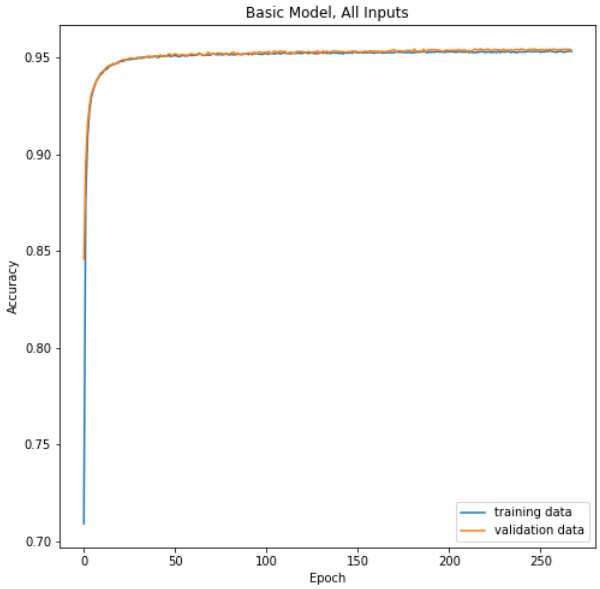

Figure 1: Results from a single layer model built on input from all three categories.

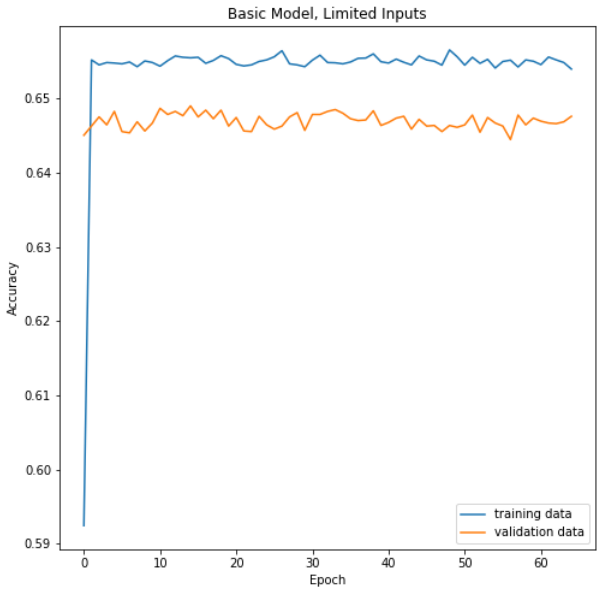

Figure 2: Results from a single layer model built on input from the first two categories.

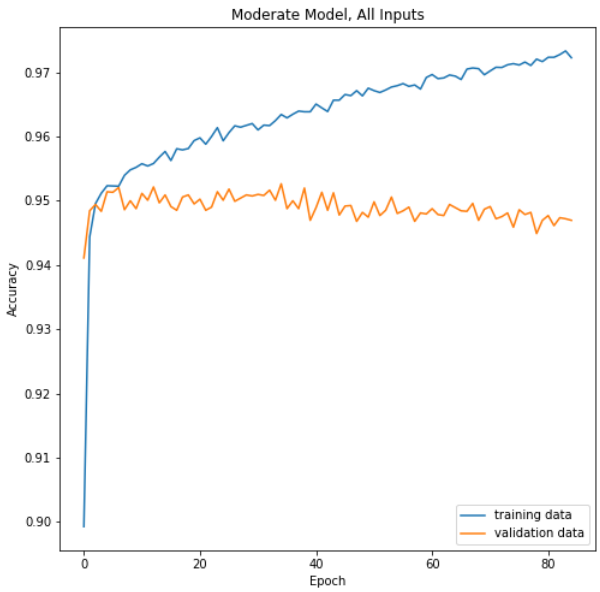

Figure 3: Results from a two layer model built on input from all three categories.

Figure 2 presents the result of removing the third category, which had not been tested during earlier phases of the project. The accuracy drops roughly 30% but remains better than the 51% gained from always predicting one option. There does not appear to be overfitting, as after the first two epochs the two accuracy values do not diverge. Unfortunately learning doesn't really seem to occur. However, this is a single layer model. A more complex model might learn something.

Figure 3 adds a layer with 100 neurons to the first model. Every attempt to add complexity resulted in overfitting, which remains visible in this example. The validation accuracy quickly hits a peak and starts a downward trend with no sign of increasing. The training data continues to climb, a clear case of overfitting.

Figure 4 adds a layer with 200 neurons to the second model. Unlike before, both training and validation accuracy have clear trends despite the violent nature back and forth motion. Unfortunately, it does not appear to succeed in learning anything more about the validation data, and overfits it in the process. As of this report, no better complex model result has been found.

Accuracy	Training	Validation
Figure 1	95.31%	95.47%
Figure 2	65.59%	64.90%
Figure 3	96.67%	95.27%
Figure 4	66.41%	64.98%

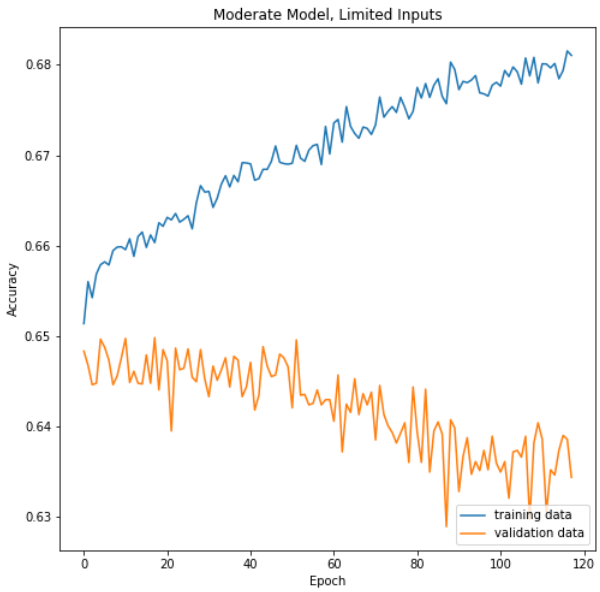

Figure 4: Results from a two layer model built on input from the first two categories.

4 Final Words

Unfortunately, the models with increased complexity have no significant impact on the basic, single layer models. In this case, the validation accuracy of the more complex model on all inputs worsened. No testing thus far has found a complex model significantly better than those presented, with each tending to overfitting.

I hoped that learning would occur using inputs without needing data from after a match. They do remain better than blind predictions, including constantly predicting one winner or the other, so the accuracy still has some merit. Further testing may find a model that can learn based on those inputs.

The model may be usable for predicting the winner of a match before it begins and updating as the match progresses and the new inputs gain value. While I had no training data for matches in progress, I suspect it could start with a less accurate prediction and use the updated data in progress to improve its prediction, providing a moving target. If the model changes its prediction partway through a match, it may even be able to find common turning points given enough data.