Factors in Tennis

John Soderstrom

March 15, 2020

1 About the Project

The link for the notebook is found here.

The dataset is sourced from Kaggle. It contains statistics for tournaments from the Association of Tennis, Professionals Matches, from 2000 to 2017. Years 2016 and 2017 are missing more data than previous years and were not parsed correctly by Pandas, so they are not included. However, this still leaves 16 years of tournament data to work with.

Part of the dataset focuses on the matches themselves, including the date, number of matches, and number of sets to win the match. The greater part focuses on the players in each match, separating data by the winner and loser.

The data separated by individual players may be further split into data known before the match starts and data known as the match completes. Age, handedness (left or right), and height are examples known before the match starts. The number of aces, double faults, and number of break points are examples of data not fully known until the match ends. However, a trend may be visible partway through the match.

After removing rows with missing values, there are 40413 rows in the dataset. After removing columns with almost no present data and splitting the date, the dataset cuts to 46 columns. This is further reduced to 37 columns, including the output. The dataset originally placed the winner before the loser in every row, but half of them have been flipped with an extra column describing which player won.

This dataset was one of two I originally considered, and I shifted to this one due to lack of interest and small number of columns in the first. Even though I can't really play tennis these days, it is one of the few sports I continue to have an interest in.

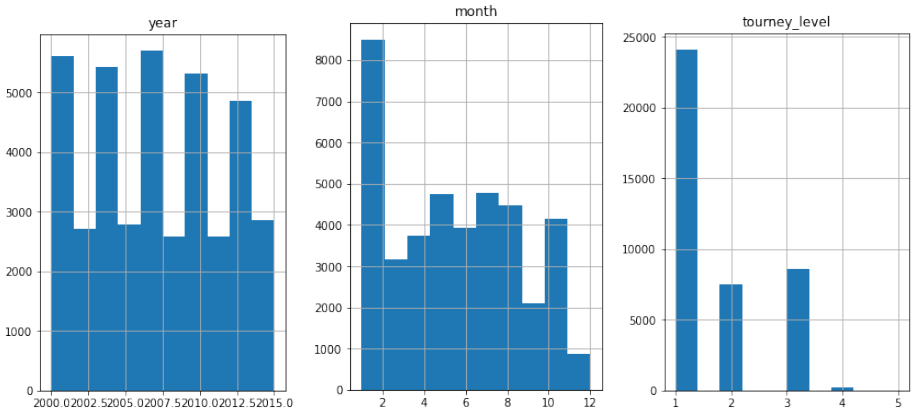
2 The Dataset

2.1 Description of Data

The dataset has 40413 rows and 37 columns after processing from an original 46 columns. Some inputs were altered to use purely numeric values in place of categorical values. The output was not originally part of the dataset; I added it as a column with binary classification on the winner. If the first player in the row is the victor, it outputs a 1. Then I reversed half of the inputs and their output so the same label does not appear for each row.

- Year: the year the tournament takes place
- Month: the month the tournament takes place
- Draw Size: the number of players in the tournament
- Minutes: length of time of the match
- Hand: relationship of dominant hands of players (LR, LL/RR, RL)
- Surface: type of surface (Hard, Clay, Grass, Carpet)
- Tournament Level
- Round: point in tournament (Finals, Semifinals, and earlier matches)
- Number of Sets
- Heights by Player
- Ages by Player
- Ranks by Player
- Rank Points by Player
- Service Aces by Player
- Double Faults by Player
- Service Games by Player
- Service Points Scored by Player
- First Serve Success by Player
- First Serve Won by Player
- Second Serve Won by Player
- Break Points Saved Against by Player
- Break Points Faced by Player
- Output: Which Player Won

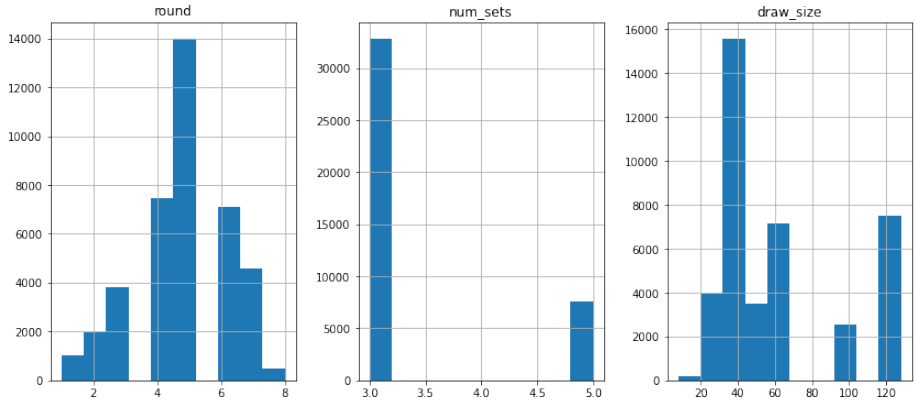
2.2 Input Data Histograms



The histogram for the year each tournament is held is slightly misleading. The maximum number of tournaments held in a single year is 2850, and most are between 2000 and 3000. The number of tournaments falls in the last five years, down to a minimum of 926 in 2015.

Similarly, the greatest number of tournaments in a single month is 4765, in July. The fewest are held in November (650) and December (230), which fits the colder months in the holiday season. They were more focused in the middle of the year, but well spread outside of those two months.

The greatest number of tournaments were type A, which I believe are the Tour Series. I think those would be the most common tournaments while the others are more focused on players of higher skill breaking their way into the tournament.

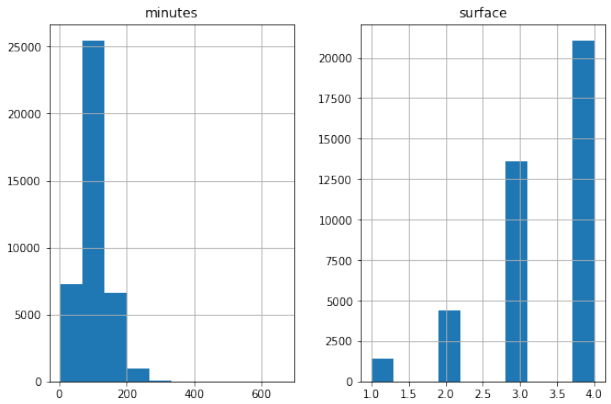


The round marks which round of the tournament a match took place in. 32 player tournaments were the most common, and in general the number of matches each round is expected to double each step down the tournament lad-

der. Larger tournaments are less common, so matches during those rounds are less common; round 5 (the 32 player stage) is thus the largest category. Round-robin tournaments (category 8) are the least common and have few matches in comparison.

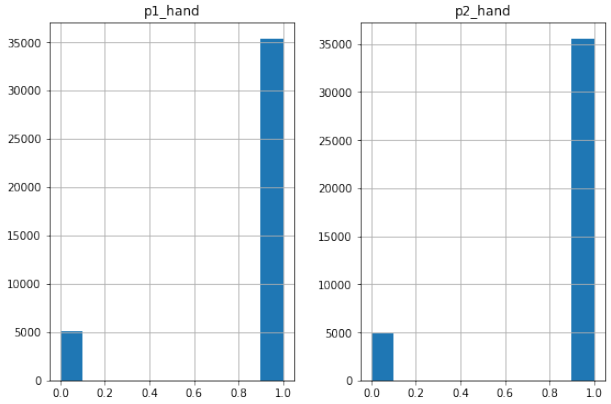
3 set matches contain the overwhelming majority of matches. They are shorter which allows the players to exert more energy in a single match, while 5 set matches are more likely to happen in later rounds of a tournament.

The draw size represents the number of players in the tournament. As expected from the round histogram, the largest category is 32 players (15564 instances). The rarest is 16 (15 instances).

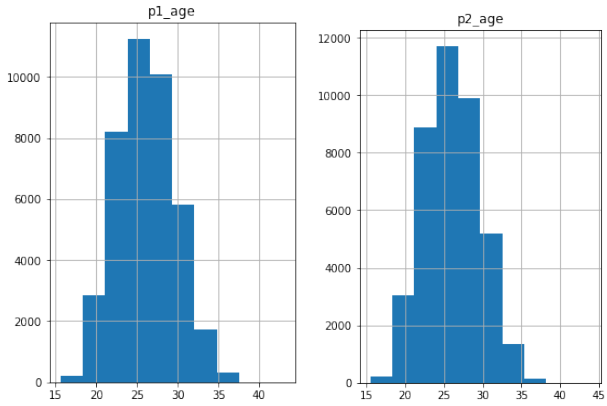


The minutes represent the time spent in each match. It is concentrated around the 70 to 80 minute mark, so most matches end well before 2 hours have taken place. These are likely connected most to 3 set matches and matches in earlier rounds, before weaker players are removed from the tournament.

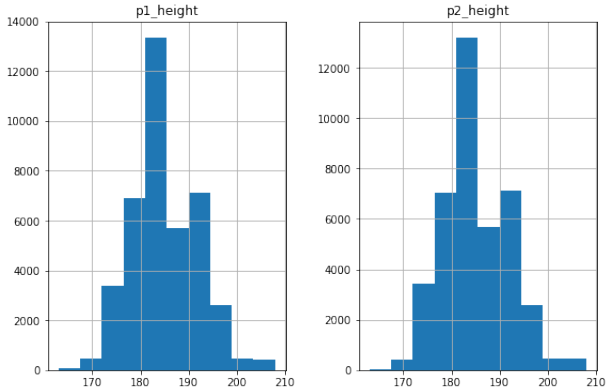
Hard surfaces are the most common at 21042 instances, followed by clay surfaces at 13582 instances. The harder surfaces allow the ball to bounce more, affecting each player's play. Carpet surfaces are the rarest at 1413 instances.



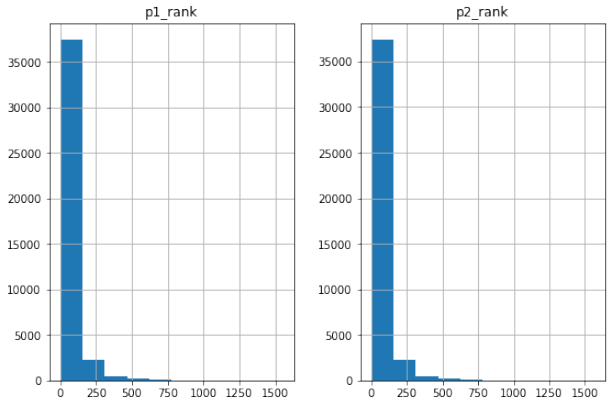
Players have been swapped such that 50% of the time, player 1 wins. Right handed players are far more common. Left handed players (often called Southpaws) may have a reversed stance that can throw off opponents.



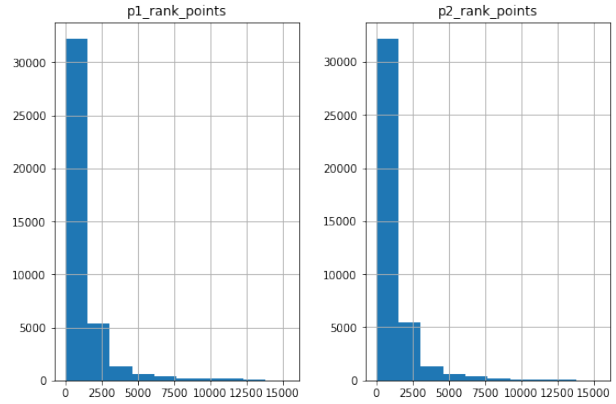
Most tennis players are in their mid to late 20's. The mix of experience and physical condition at those ages tend to result in a more powerful player who is more likely to take part in more matches, especially in the same tournament. Players weaker in either category may get into a tournament but not last as long, giving more weight to the previous group. Regardless of the reason, the ages form something similar to a bell curve, with fewer players showing up less often as the age deviates further from that center.



Heights are measured in centimeters, and appear to concentrate mostly around 185 cm. While more height can help a player reach a ball more often and add more power to their strikes, a lesser height may be more common. They appear to create something like a bell curve with a bit more weight to the greater heights.

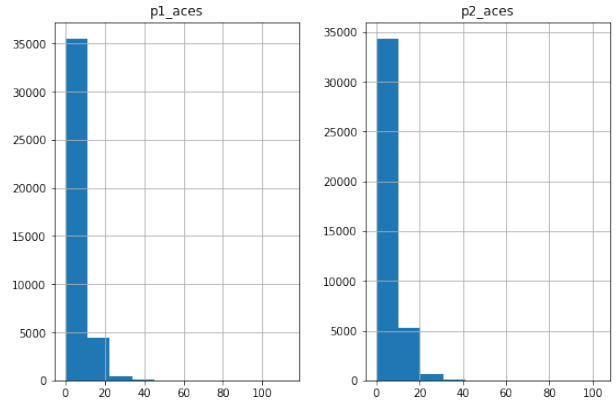


The higher ranks are most present, skewing the distribution towards them. Higher ranked players are more likely to win more matches, considering how they achieved their rank, allowing them to show up in more matches. Further, tournaments that are more selective will pull players from the higher ranks, providing them even more opportunities to play matches, skewing even further toward them.

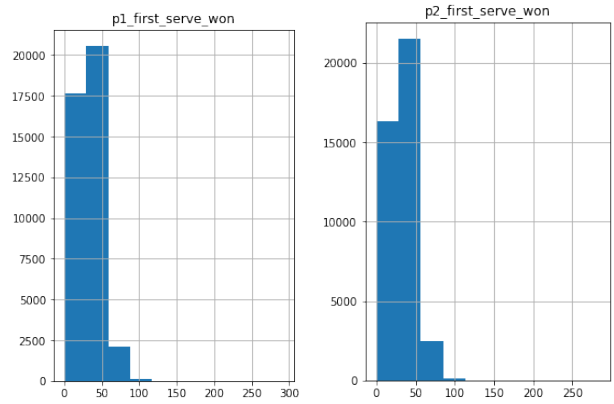


Including rank points may be redundant with the inclusion of rank (skewing toward the better ranked players in the same way), but it provides another type of measure. Rank provides a discrete measure where the gaps are always the same. Rank points add a continuous measure that may take into account a wider gap between two ranks than another two, weighing toward the larger gap.

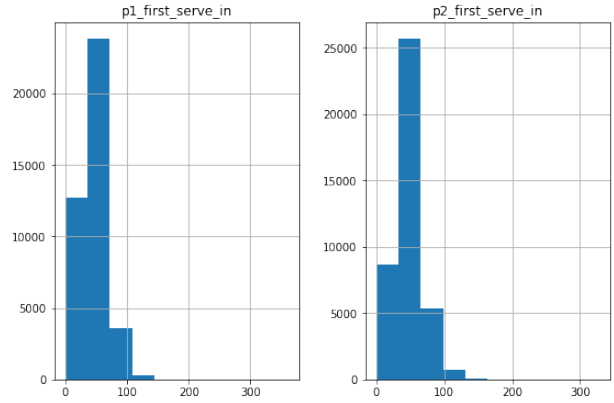
The next inputs may only be fully included after a match is complete, and the winner decided.



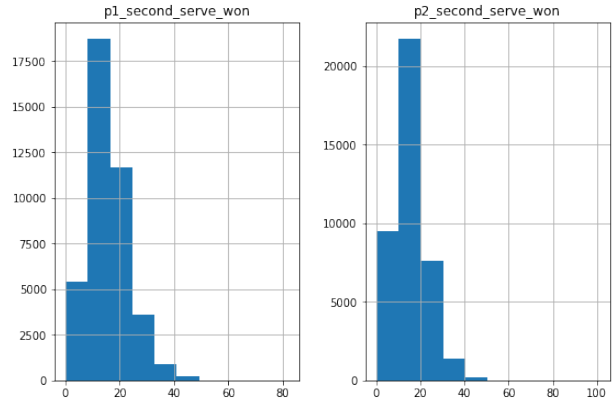
Aces are points won on the first serve alone, where the opponent does not manage to return the ball at all. Aces are difficult without a large gap in skill and skew to a low number, though if a player is able to get 1 ace they can probably get a few more. In both cases, 2 aces are the most common, and they drop after 5.



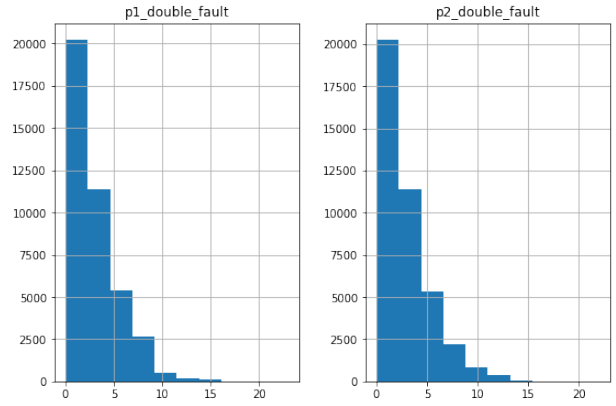
This category includes some of the aces, but also any point won off of a serve even if the opponent returned it. It skews toward the mid to high 20's, which makes sense as the minimum number of games in a 3 set match (the more common type) is 12. In the best case, a player will need to win 6 of those games (winning the returning games against the opponent or server), or 24 points won. The number will increase when considering second serves won, but they are less common. Earlier histograms showed the imbalance in 3 and 5 set games towards the former, and it was less common for matches to extend past 80 minutes. A shorter match is more likely to mean fewer points were needed to win.



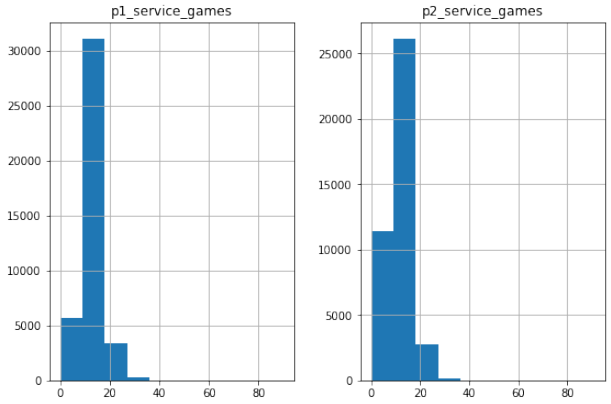
This category includes all of the previous, as to win on the first serve it must go in. In both cases, the number of first serves that make it in weigh towards the mid to high 30's. For every 4 first serves that makes it in, roughly 2 or 3 points are won (no major statistics run, just looking where each category weighs). That makes sense because the game favors the server. The server decides where to begin their attack, and the receiver has to try to seize the advantage.



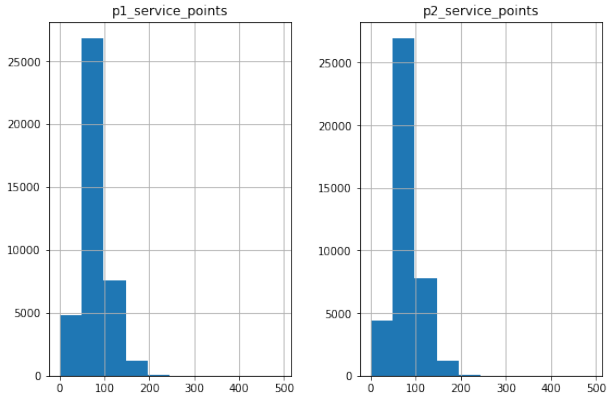
If the server fails to hit their first serve into play, they get one more chance before losing the point. Second serves are less common, and the server may play more cautiously if they need to use one, which gives the receiver a better chance to respond. Both sets weigh toward the low 10's. When added to the first serves won, we reach a rough estimate of 40 points won per match. Considering that not all points will lead to a won game, the number seems more appropriate for a full match.



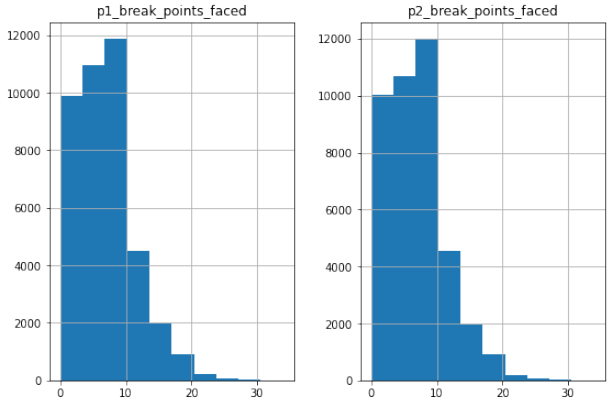
A double fault means that both the first serve and second serve failed to go into play. Since players tend to make their first serve in and take more caution on the second serve, double faults tend to be rare. They skew to the left, 2 double faults the most common at almost 8000 instances in both cases.



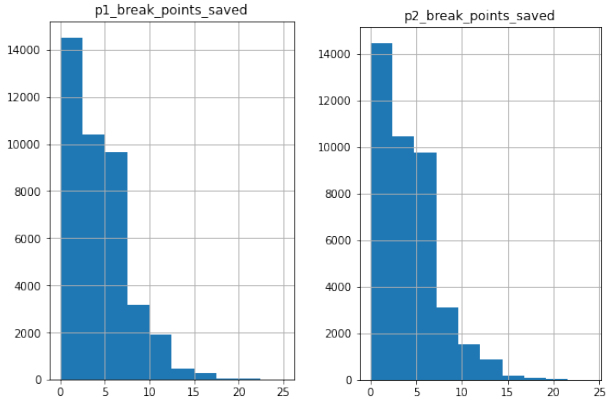
While less concentrated in one small group of numbers, both tend toward 10 to 17 service games. As stated previously, during a 3 set match, in the best case a player will win 6 games they serve in (service games). The number of service games point to more common difficulty in winning those games, and increase the number of points expected to win a match. Despite the greater spread across the main group of numbers of service games, there is a steeper drop on either side.



The number of service points (points the server wins during their service game) concentrates in the 50s and 60s. At minimum, 4 service points are needed to win a service game. Deuces (a tie that needs 2 points won to win the game) can increase that. A server may also win service points in service games they do not win.

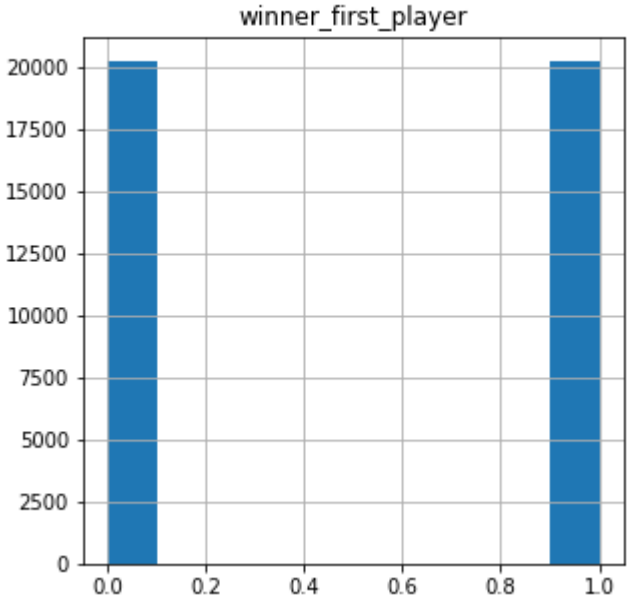


Break points occur when the receiver is one point away from winning a game. We know already that service games favor the server, since they get to make the initial attack. Yet break points are needed to win a set and a match, since the winner must be ahead by at least 2 of the points or games. Break points are nearly inevitable, but they still skew to the left. They concentrate in high values under 10 in both cases, and begin sharply dropping as the number increases.



Break points must be faced in order to be saved, and some break points must be won in order to finish the game. Our total number here will be less than the number of break points faced, and it skews even further to the left than the previous category. 3 saves are the most common in both cases, and they begin sharply dropping after 5 saves.

2.3 Output Data Histogram



The output values had to be added and positions of players swapped to ensure different values existed. They were split in half, so this result is not surprising. They are perfectly balanced as a result, which helps ensure training and validation data will have a spread of each value.

2.4 Data Normalization

I chose Z-score normalization after converting all input columns to numeric values if they were not already. The learning process can go faster with each point closer to their axis, and outliers remain obvious for human observation.

3 Modeling

3.1 Data Splitting

After randomly shuffling the data, it was split into training and validation sets using 70% and 30% respectively. The training data consisted of 28290 rows and the validation data 12123 rows.

3.2 Basic Results of Models

Layers	One		Two		Three	
Data	Training	Validation	Training	Validation	Training	Validation
Accuracy	.9530	.9506	.9684	.9468	.9982	.9348
MSE	.4423	.4434	.4689	.4688	.4469	.4436

More detailed results for binary models in section 3.5.

The single layer neural network appears to be the most useful, as the other two categories have a greater disparity between the training and validation errors. In the next section, the difference between the models becomes more apparent.

The output data fits a classification model better, with only two possible results for each input row. The linear model is not as appropriate, but the results show a similar story. Surprisingly, despite the difference in loss between training and validation data, the validation error gets smaller in comparison to the training error as complexity increases.

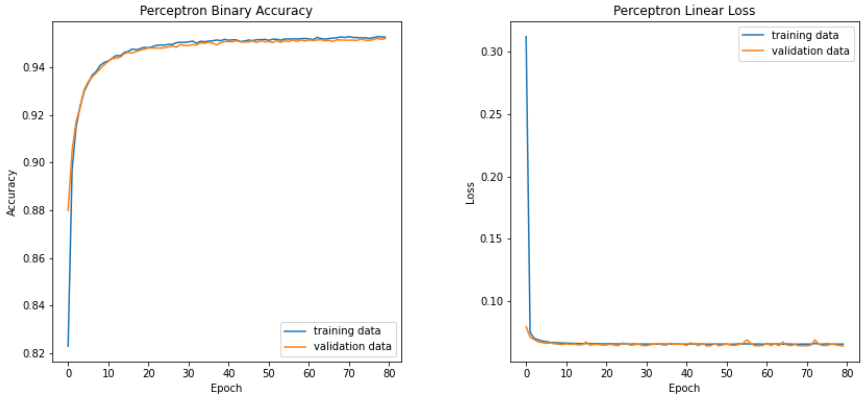
For this dataset, the outputs are balanced, and a true positive has the same importance as a false negative. After all, whichever player wins the match, it's still just one of the two players winning. Recall and precision are then less important values to choose a model. Either accuracy or the f1 score may be used, and those values turned out to be close.

Layers	Two		Three	
Data	Training	Validation	Training	Validation
Accuracy	.9651	.9461	.9981	.9279
MSE	.4472	.4447	.4213	.4189

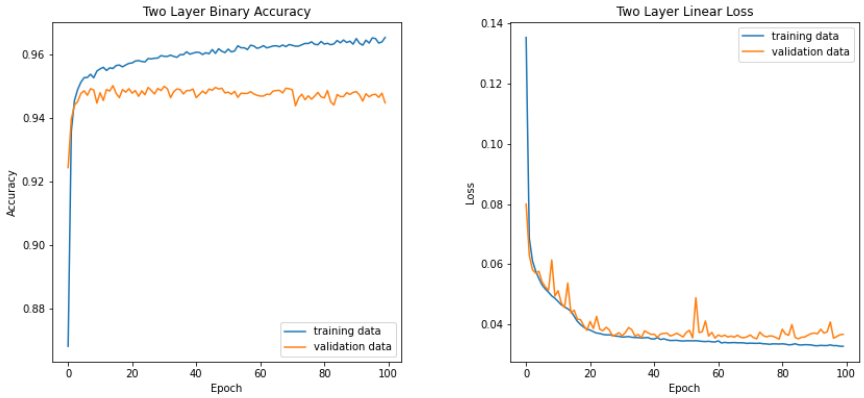
Modifying the neural networks such that all neurons use a sigmoid activation results in minor loss in accuracy on the training data and a slightly more significant loss in accuracy on the validation data. More strikingly, the learning curve does not change at quickly during the early stages of learning.

Strangely, changing the activation of all neurons to linear results in slightly less error. The loss in the learning curves is closer, though the validation loss is slightly more volatile.

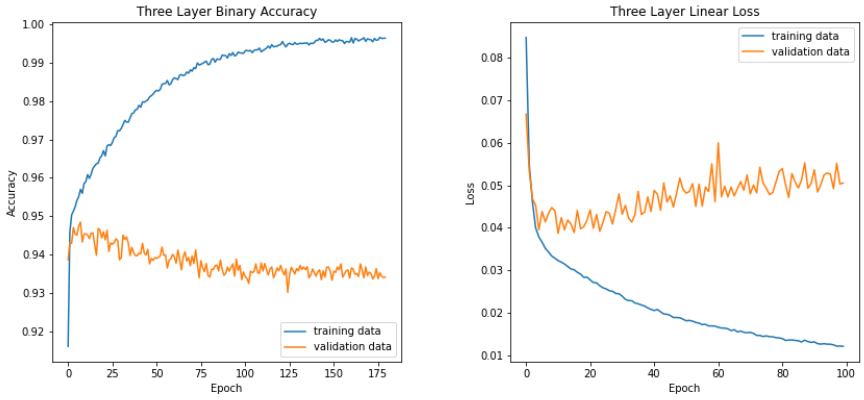
3.3 Learning Curves



The perceptron, or single layer model, achieves similar accuracy and loss values across both training and validation data. It still seems to improve with more epochs, though the gain is becoming insignificant.



The two layer neural network appears to overfit from an early stage, especially in the binary model. Success on the validation data gains little after the first few epochs before stagnation.



The three layer neural network overfits with greater severity than the two layer version. Success on the training data becomes near perfect. Meanwhile, success on the validation data doesn't just stagnate but decreases.

3.4 Overfitting

The two and three layer neural networks already show overfitting. In the three layer network, the first layer contains nodes equal to four times the number of columns (148 nodes). The second layer, and first of the two layer network, contains nodes equal to the number of columns (37 nodes). Clearly the threshold for overfitting lies somewhere between a single layer and two layers with 37 initial nodes.

When the output is added to the input data for learning, the accuracy for both training and validation data jumps to 100% even with a single layer. Adding additional layers and nodes appears to make this happen sooner.

3.5 Binary Model Predictions

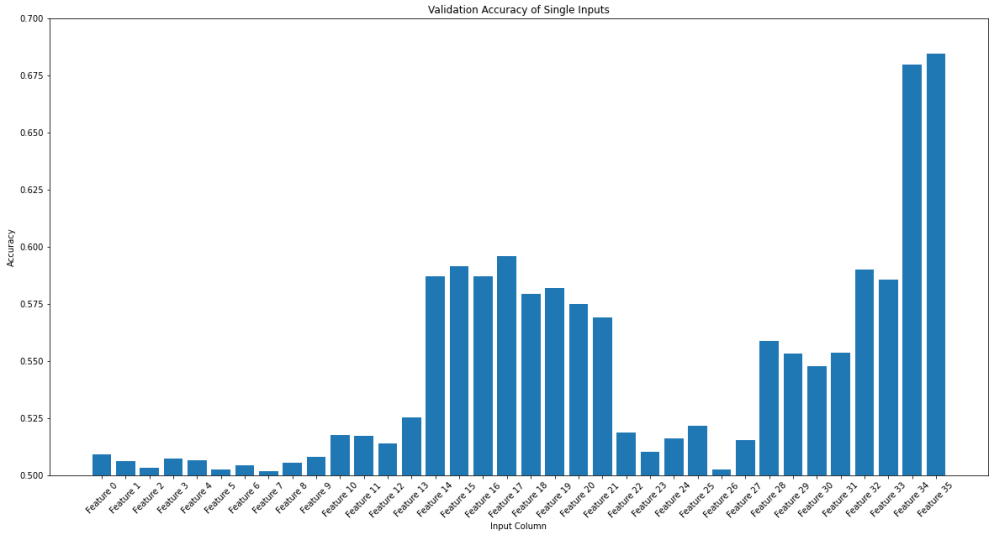
Layers	One		Two		Three	
Data	Training	Validation	Training	Validation	Training	Validation
Accuracy	.9526	.9519	.9663	.9473	.9963	.9350
Recall	.9516	.9547	.9669	.9475	.9955	.9293
Precision	.9533	.9499	.9655	.9483	.9972	.9432
F1 Score	.9524	.9523	.9662	.9479	.9963	.9362

Due to the nature of false positives and false negatives, the f1 score is a more appropriate measure than either. False positives and false negatives both represent an incorrect choice of the winner, with equal importance.

4 Feature Reduction

4.1 Feature Importance

All 36 inputs were run through a simple model individually and ranked against each other. Differences are emphasized, as the range is less than .2 (accuracy ratings may range from 0 to 1). Testing pairs for player specific features may be more useful.



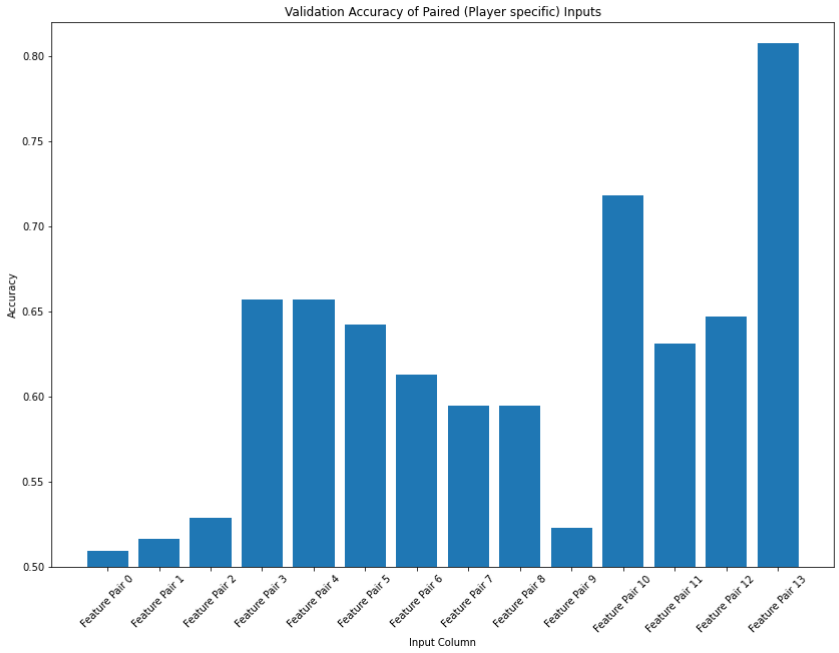
The weakest features include all inputs common to both players, such as the time of the match and surface of the court. It is not surprising that such features would be weaker in predicting the winner. They were originally left in to see if, combined with other features specific to one player, they could improve the prediction.

They also include inputs specific to each player: dominant hand, age, height, number of service games, number of service points, and the number of first serves in. The first three compose most of the information available before a match starts, leaving only information related to their ranks. I expected age might have a greater impact, despite the experience gained from those who are older. The latter three are not completely surprising. The number of first serves in is less likely to represent better players as they may attempt tougher serves as necessary. The number of service points is likely varied regardless of who wins.

The next group of features contain most of the player specific inputs: rank and rank points, number of aces, number of double faults, number of first and second serves won, and number of break points saved against. The number of first and second serves won are the weakest members of this group, but restricting them to one player leaves a weaker connection where the lower it is, the

more likely the player lost. The other features all have a stronger connection where better players are more likely to have better numbers, and they may be less affected by the length of the match.

The final group contains only the break points faced by each player. The threshold for losing players is much firmer, as the losing player will face at least two break points in a three set match and most likely faces more. The winning player could face as few as zero, and is far more likely to face few break points than the loser. While not impossible for the values to be reversed, it is unlikely.



Pairing the player specific inputs allows the important differences to impact the accuracy, and grant a stronger idea of their importance. Pairs are sorted in the following manner:

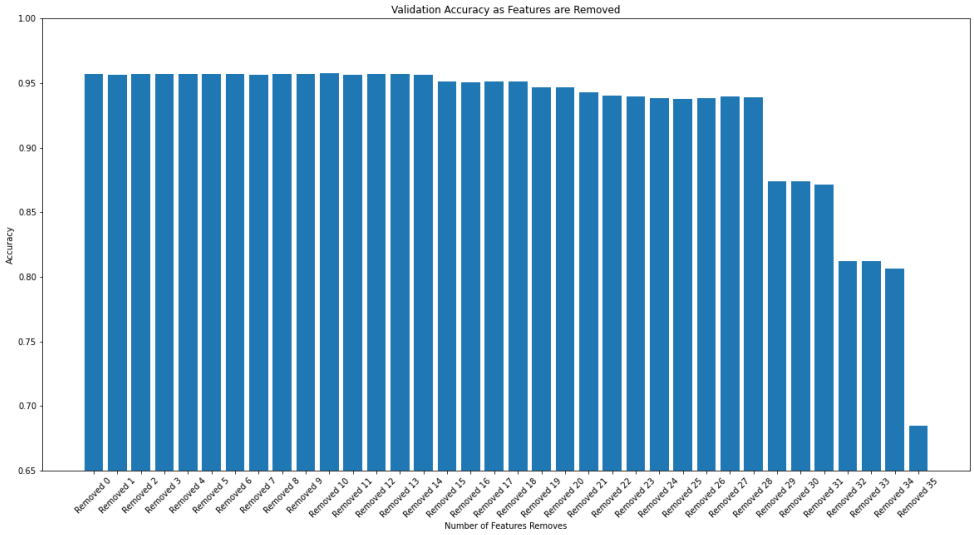
0. Dominant hand
1. Height
2. Age
3. Rank
4. Rank points
5. Number of aces
6. Number of double faults
7. Number of service games
8. Number of service points
9. Number of first serves in
10. Number of first serves won
11. Number of second serves won
12. Number of break points saved
13. Number of break points faced

Dominant hand, height, age, and number of first serves in are the clear weak points. Break points faced and number of first serves in are clear winners, follow by rank and rank points. Rank and rank points are likely redundant, though rank points allow larger gaps between players than merely the rank. They are also the only high scoring inputs that are known before a game.

4.2 Performance After Removal

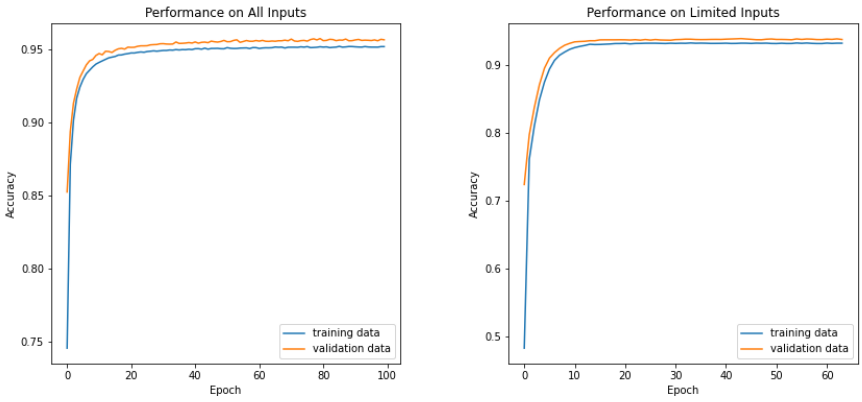
The first 15 features removed have no significant impact on performance, and the next 14 have minimal impact; this group of 14 includes information known about specific players before the match starts, apart from their rank and rank points. The last feature removed before a major drop is the number of aces from Player 2. The first feature removed to cause a major drop is the number of break points saved by Player 2.

I did not expect that the last 8 features causing the drops in accuracy were already the top 4 pairs. As such, the graphs in 4.3 already take into account the pairs I would have reinserted for the limited inputs.



The next major drop occurs when the number of break points saved by Player 1 is dropped. The final occurs when the only remaining feature is the number of break points faced by Player 1. The drops appear occur mostly when features related to break points are removed.

4.3 Performance Comparisons



The learning curves of both sets of input are strikingly similar, though this was expected with the minimal drop until the last 8 inputs. The learning on limited inputs slows down a little early into the process, but continues steadily and stops with a slightly lower accuracy than the full set of inputs.

In both cases, early stopping with a patience value of 20 still led to an early cutoff, and the limited inputs stopped in just over half the time. The model had far less to process, and hit a peak much earlier that it never overtook.