Machine Learning for Straight Flat and Jumps Racing Rank Prediction

Adam Gardiner-Hill MSci - Quantum Leap Solutions

May 10, 2022

Abstract

This investigation looked to build upon and extend the work outlined in our previous study, which incorporated machine learning algorithms for the prediction of racehorse rankings. Neural networks similar to those used previously were applied to predict rankings for flat races with no bends and jumps racing. Further data analysis and pre-processing steps are also carried out to improve computational and labour efficiency, and remove some extraneous variables. This included investigating the most appropriate method for populating missing variables associated with first-time runners. For all combined flat racing, the network achieved an exact accuracy of 47.8% and accuracy to within one rank of error of 94.2%. For jumps racing, the network achieved an exact accuracy of 49.4% and an accuracy to within one rank of error of 94.7%.

Investigation

Data and Pre-processing

Data structure was similar to that used in our previous investigation, with some notable differences across race classes. For straight flat races, an identical data structure was used to test the value of the 'Distance to First Bend' variable. After results showed that its omission had little effect on network accuracy, it was removed from all subsequent data sets. The structure of the final data set for all flat races is illustrated in table 1.

For jumps racing, four variables were used; 'Mode of previous ranks', 'Race pace', 'Race distance' and 'Strategy'. There are notably fewer variables for jumps racing as 'Stalls Position' and 'Draw' are not relevant. The structure of the final data set for jumps races can be seen in table 2

MoPR	Draw	Race P.	Stalls Pos.	Race Dis.	Strat.
...	...	...	...	...	...

Table 1: Table illustrating the variables used to train the network for all flat racing after the removal of the 'distance to first bend' variable

MoPR	Race Pace	Race Distance	Strategy
...	...	...	...

Table 2: Table illustrating the considered variables used to train the network for all jumps races.

Neural Networks

For our investigations into the effects of removal of the 'DtFB' data, and the introduction of a distribution to populate unknown variables, a standard sequential neural network identical to that used in our previous investigation was employed. It consisted of an input layer with shape (7,), 4 dense layers with decreasing node numbers (2100 - 210) and a dense output layer with four nodes, one for each classification class. ReLU activation functions were used for the dense layers, and a Softmax activation function was used for the final dense classification layer.

Once we had determined that the 'DtFB' variable could be dropped from the data, the input shape for the network was changed to a (6,) NumPy array. After some brief investigation into the optimisation of network architecture. the decision was made to remove one of the dense layers and decrease the number of sequential nodes. The new network that would be trained for all flat data now had 3 dense layers with nodes (240, 60, 6). This reduced the training time considerably whilst maintaining performance.

The same architecture was then trained on the jumps racing data with the input layer shape altered to (4,) to match the structure of jumps data outlined in table 2.

Platform and Packages

The analysis was completed using open source Python packages and the Spyder IDE. These included NumPy, Pandas, Keras, Sci-Kit Learn, MatPlotLib and Seaborn. A TensorFlow virtual environment was created for this task, and all neural networks were trained on an NVidia GTX 1060 6GB GPU.

'Distance to First Bend' Input Variable

In our first investigation, a 'distance to first bend' variable was included to account for the need of horses in flat races to get onto the best racing line out of the stalls. Collecting the data for this variable is extremely labour intensive and time consuming, as it requires visual mapping of courses over all applicable distances. With the goal of saving time in the future, we investigated the effect of removing this variable from the 'flat races with bends' data and retraining an identical network with the 'DtFB' variable omitted.

Inference of 'Mode of Previous Ranks' for First Time Runners

In our last investigation, the potential effects of assuming the mode of previous ranks to be equal to 3 for first time runners was discussed. If a strong correlation exists between a horse's previous ranks and its future ranks, then this assumption may cause the network to over-predict rank 3s relative to other ranks.

To make the data set more realistic, probability distributions were used to populate these values, derived by examining the true distributions of ranks for first time runners over jumps and the flat. The distribution obtained for first time runners over jumps is shown in figure 1 as an example. The NumPy function, numpy.choices(), was then used with probability weights from the distribution to populate the mode of previous ranks variable for all first time flat and jumps runners.

We then investigated the effects of this sample wide change by training and testing two identical networks on the combined flat racing data - one with a data set where all first time runners were assumed to have a 'MoPR' value equal to 3, and one where these were populated according to the probability distributions. The overall accuracies and granular effects were then compared.

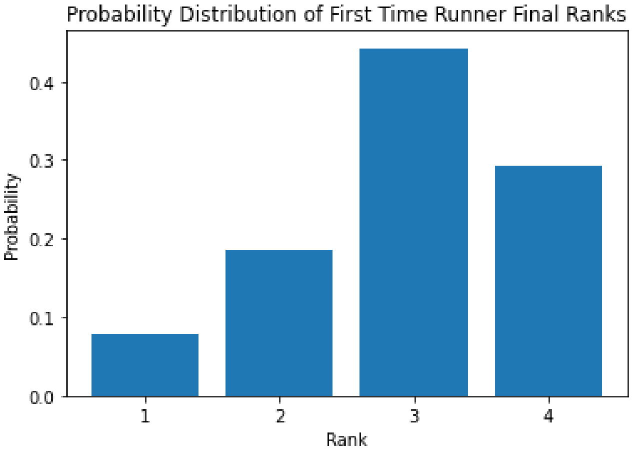

Figure 1: Example probability distribution for true ranks for first time runners over jumps.

Combined Flat Races

Once the impact of the 'DtFB' variable had been established the decision was made to remove it from all future data sets. The distribution of ranks produced by first time runners on the flat was determined and 'MoPR' values for horses with no history populated according to it. The final flat data set consisted of 366217 runner instances, after NaNs and rows with missing data were dropped. This was then split 80/20 into a training and test data set. The network was then trained over 100 epochs and accuracy on the training set recorded. The test data was then used to generate final results.

Jumps Races

The distribution of ranks produced by first time runners over jumps was determined and 'MoPR' values for horses with no history populated according to it. The final jumps data set consisted of 217848 runner instances, after NaNs and rows with missing data were dropped. This was then split 80/20 into a training and test data set. The jumps network was then trained over 100 epochs and accuracy on the training set recorded. The network was then asked to make predictions for the test data to generate final results.

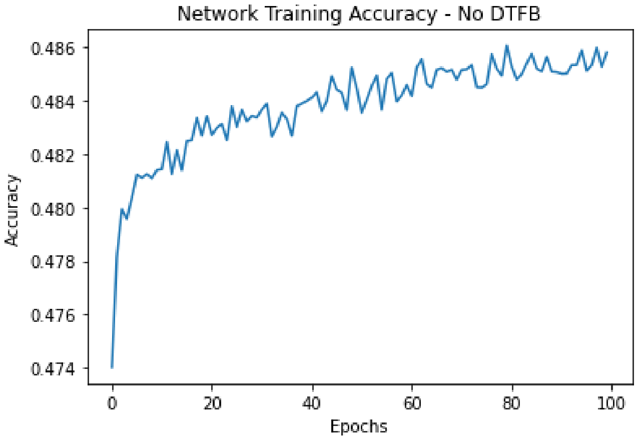

Figure 2: Example plot of training accuracy for the flat network with the 'DtFB' variable removed.

Results

Removing Distance to First Bend for Flat Races

Curiously, the network's overall performance improved after the removal of the 'DtFB' variable. When it was included, the network was able to predict exact rankings with 47.7% accuracy, and to within one rank of error with 91.8% accuracy. With 'DtFB' omitted, the network was able to predict exact rankings with 48.2% accuracy, and to within one rank of error with 94.4% accuracy. The confusion matrices for each data set can be seen in figure 3. Potentially the most notable granular improvement was the increase in correctly identified rank 4s, with 14% being exactly predicted when the data was removed versus just 8.6% with it included. Exact identification of rank 2s also saw a slight improvement to 8.0% from 6.1%.

The improvements in overall accuracy came with some costs in specific areas. Comparing the two confusion matrices, it can be seen that omitting the 'DtFB' data led to a decrease in precise accuracy for rank 3 predictions, and an increase in the proportion of true rank 3s predicted as 2s and 4s. In addition, nearly twice as many true rank 2s were predicted as 4s when the 'DtFB' data was omitted than when it was included.

It was decided that the broad improvements in accuracy, especially the increase in accuracy to within one rank, justified the granular deterioration seen in some cases, and the 'DtFB' data was dropped from all further investigations. The decision was also driven by the desire to save labour time on data entry, as previously mentioned.

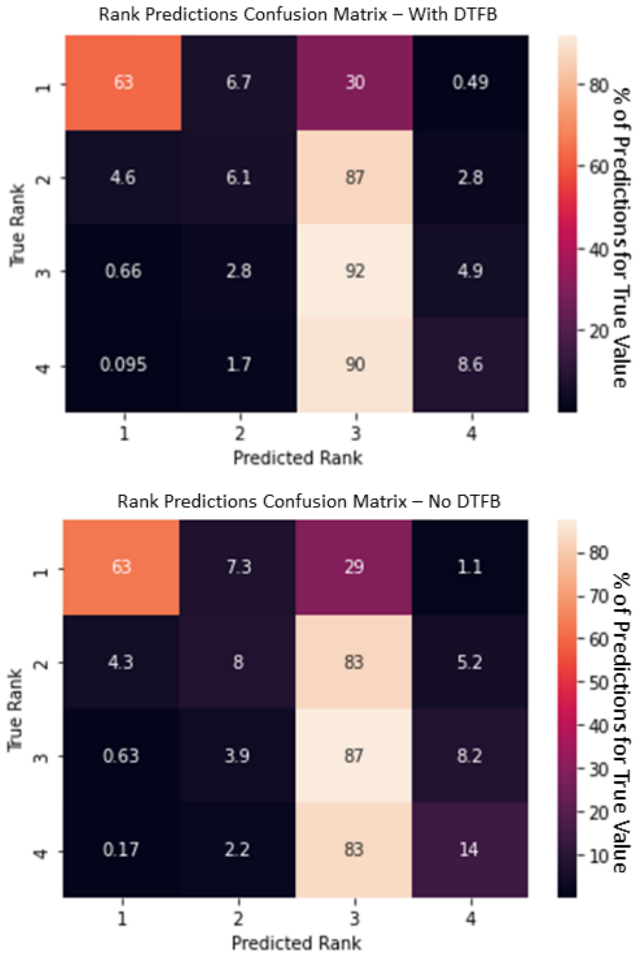

Figure 3: Confusion matrices for networks trained with the distance to first bend variable included and without. The top CM shows the results with the 'DtFB' data included and the bottom CM shows the results produced by an identical network trained with the data omitted.

Using a Distribution for Mode of Previous Ranks

When compared to a scenario where all values were assumed to equal '3', the use of a representative distribution to infer the first time runner 'mode of previous ranks' variable produced some intriguing outcomes. Comparing the results respectively, the networks achieved exact accuracies of 47.5% and 47.8%, and accuracies to within one rank of error of 93.3% and 94.2%. The confusion matrices for these models can be seen in figure 4.

Although the overall accuracies indicate a slight edge in favour of the use of the distribution, the granular changes are more nuanced in nature. Substantially better results were seen for exact predictions of rank 2s and 4s when all values were assumed to equal 3 - 16.0% and 24.0% versus 11.0% and 10.0% with a proportionally representative distribution. This was offset to some degree by a decreased performance for the exact prediction of rank 3s - 76.0% with all values equal to 3 versus 88.0% with a representative distribution. This will be explored further in the discussion section.

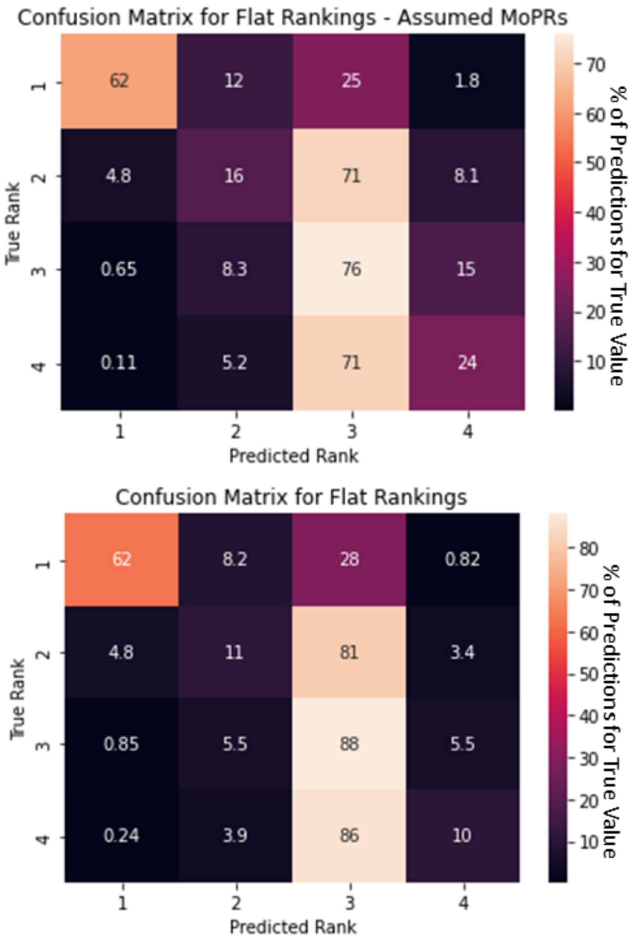

Figure 4: Confusion matrices for networks trained with flat races data. The top CM illustrates a network trained when all first time runners were assumed to have an 'MoPR' variable equal to 3. The bottom CM shows the results after these values were populated using a derived probability distribution.

Given the slight improvements seen in overall accuracy values when a representative distribution was used, all data going forward had its first time runners 'mode of previous ranks' variable populated according to probability distributions.

Combined Flat Races

Since the 'DtFB' variable was found to make little difference to the predictive ability of our network, all flat racing data was combined into one set and used to train a new model. This model achieved an overall exact accuracy of 47.8% and an accuracy to within one rank of error of 94.2%. These values are comparable to the results seen in our previous investigation where a model trained using only races with at least one bend and a 'DtFB' variable produced respective accuracies of 47.7% and 91.8%. The confusion matrix for these results can be seen in the second image of figure 4.

A similar granular pattern to that seen in our previous investigation was found for each ranking. The network performed comparatively well when predicting leaders, but

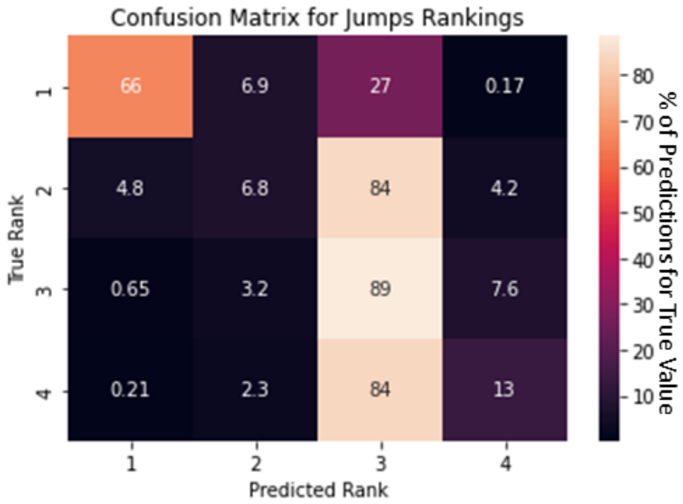

Figure 5: Network training accuracy plot and confusion matrix for jumps racing results.

struggled to differentiate between ranks 2-4, predicting a large majority of values in this range as 3s.

Jumps Races

The model trained using data from jumps races achieved an exact prediction accuracy of 49.4% and and accuracy to within one rank of error of 94.7%. The network training accuracy plot and confusion matrix for these results can be seen in figure 5. The results are comparable to those achieved for flat races with bends and our combined flat races approach. Again, a similar granular pattern emerged. The network was most effective for predicting leaders, successfully identifying 66% of true values for rank 1, but struggled substantially to differentiate with confidence between ranks 2-4.

Discussion of Results

Removal of 'Distance to First Bend' Variable

Given that the only change that was noted in network performance after the removal of the distance to first bend data was a slight improvement in performance, the decision was made to drop that variable moving forward. It may have been the case that the variable had a low correlation to rank outcome and therefore only acted to add noise to the input data. A full correlation analysis to be carried in our future work should help to determine whether this was the case.

In terms of practicality and time saving potential, this is a beneficial result. Mapping the distance to the first bend for all distances over all courses is a time consuming and labour intensive process that no longer needs to be completed. Furthermore, the result showing that a single model for all flat races produces equivalent performance offers many implementation related benefits. The need to have separate models working for straight flat and bending flat races would have incurred financial and labour costs. Two distinct data processing pipelines would have had to have been constructed, and results differentiated from each other before productisation. These can now be combined into a singular, more efficient pipeline and model for all flat race predictions.

Inference of 'Mode of Previous Ranks' for First Time Runners

The results observed when applying a probability distribution to this task were intriguing. In the flat data, there were 18108 instances of first time runners that had their 'MoPR' inferred from the distribution, in the jumps data there were 16393. These, therefore, represent 4.94% and 7.52% of their total data sets respectively, and both are substantial enough to assume that they can influence outcomes.

When closely examining the confusion matrices in figure 4, it appears that the introduction of the distribution primarily acted to make the prediction of a rank 3 more likely. The 'spread' around predictions of a rank 3 became much tighter, with an increase in exact accuracy for true rank 3s from 76% to 88%. The same effect has manifested for true ranks 2 and 4, but around predictions of rank 3. That is to say, the network began predicting more rank 2s and 4s as rank 3s, leading to decreases in exact prediction accuracy for those values.

The reasons for this are complex and a thorough analysis of outcomes and repeat tests would be necessary to determine with any degree of certainty what those causes are. However, speculatively, this 'tightening' may be caused in part by the 'scattergun' approach that results from populating the values according to a probability distribution. While the use of the distribution makes the broad characteristics of the sample more true to life, nothing can be said about the effect it has on an individual horse's profile. It must be considered that the assignment of a 'MoPR' value is random and dictated by the probabilities that we have derived by examining the final distributions. For example, we can determine the probability that a given 'MoPR' correctly reflects the rank that the horse produces in that run, as shown in table 3.

Rank	Prob. MoPR Assigned	Prob. Rank Produced	Prob. Match Correct
1	0.08	0.08	0.0064
2	0.19	0.19	0.0361
3	0.43	0.43	0.1850
4	0.30	0.30	0.0900
Sum	1.00	1.00	0.3175

Table 3: Table illustrating the probabilities associated with the assignment of a 'MoPR' value and the likelihood that a first time runner will produce the matching rank.

As can be seen in the table, a rank chosen to represent the 'MoPR' for a first time runner has a 31.75% chance of matching the rank the horse will actually produce first time out. It should noted, that this means that fewer individual rankings will actually be exactly correct compared to the scenario where all are assumed to be 3s, as 43% of all first time runners produce 3s. This may have made it harder for the network to draw clear distinctions between ranks 2-4, as in the assumed 'MoPR = 3' data, all true rank 3s would have been 'MoPR' rank 3s. In the data populated with the distribution, 68.25% of all final true rank values would not have corresponded to their matched 'MoPR'. This introduces noise on the granular level into the data set, as many true rank 1s may be assigned an 'MoPR' of a 3 or a 4. The way the network manifests this is very complex due to its black box nature, but discrepancies in the trend for 'MoPR' to indicate a particular detailed outcome would make it harder to predict with high precision.

Also interesting are the differences in magnitude between the probabilities of correct matching for each rank. Rank 3 has the highest likelihood of being correctly matched, with an 18.5% chance, whilst rank 1 has only a 0.64% chance of the same. The range seen here may explain to some degree the 'tightening of spreads' effect that was seen when the distribution was introduced to the data. It was previously discussed how the network may find a local minima by over-predicting 3s where it struggles to differentiate between ranks. Given the increased probability (across not just the sample but the whole population) of a true rank 3 being produced, the network may be able to minimise its loss function by predicting a 3 when the task is difficult. With the added noise caused by the randomness of distribution assignment, the network may have reverted to this approach somewhat, minimising its loss function by over predicting rank 3s as they make up 43% of all true ranks.

Combined Flat Races

The results for combined flat races were comparable to those achieved when flat races were split into those with bends and those without. As previously outlined, the primary reason for doing this was because it was suspected that the need for a horse drawn wide to get onto the best racing line early would be important. This required the introduction of the 'DtFB' variable, and made it challenging to model this variable for straight flat races. As it turned out, the variable held little sway on the network's ability to predict rankings, and so was removed from all future data sets.

The option to remove this variable is a fortuitous result and will save us a great deal of time and labour in the future when we expand to incorporate races from outside of Great Britain. Furthermore, all flat races can be combined into one data set and used to train a single model, rather than having one for bends and one for straights.

Jumps Races

The results for jumps races were also comparable to those obtained for data over the flat. The jumps network was approximately 1.6% more accurate with exact predictions and 0.5% more accurate to within one rank of error. This is likely due to more horses being ridden the same way repeatedly over jumps, versus over the flat where strategies can vary more often. This theory is reinforced by the result that the jumps network predicted 4% more leaders correctly than the flat network. Leading with the goal of making all the running is a common strategy for some horses and trainers.

Further Work

There is still great scope for further investigation in this area. One target area for improvement could be the application of the ranking distributions for populating the 'MoPR' data for first time runners. Conditional rules could be set so that maiden races contain rankings in exactly the correct distribution, helping to guarantee the correct distribution of rankings at a more granular level. Other input variables could also be considered given that some trainers set out with the objective of runner first timers with particular strategies. This may help to indicate whether a horse will be held-up, lead or sit prominent - an inference issue that needs to be tackled for 'forward' race prediction before implementation regardless. A combination of these two techniques would hopefully lead to a synergistic interaction, where the high probability that one horse may run at the front due to its strategy would eliminate the possibility of other horses in the race being predicted a rank 1, further improving overall prediction accuracy.

We mentioned in our previous study that approaching this problem using ordinal regression methods could help to improve network accuracy. Now that all major race types have been accounted for and we have an implementable product, testing ordinal techniques to try and optimise predictive performance would be a logical next step.

Beyond this, a full correlational analysis between a multiplicity of variables and a horse's final rank should be carried out. This will provide insight regarding which inputs may be beneficial to add to data and which may be extraneous.

Conclusion

This investigation expanded considerably on the findings from our last paper on the application of machine learning algorithms to race horse ranking prediction. We began by investigating the effects of removing the labour intensive variable, 'distance to first bend' to save time, and introducing more representative distributions to populate the 'mode of previous ranks' variable for first time runners. It was found that the 'DtFB' could be removed without degrading network performance, so it was dropped from all subsequent data sets. The introduction of a true to life probability distribution for first time runner ranks to infer the 'MoPR' variable yielded some interesting results. A slight improvement in general network accuracy was seen, with some degradation at the granular level for specific rankings. Due to the slightly improved performance, this methodology was then deployed for all subsequent data sets.

For all combined flat races, the trained network achieved an exact accuracy of 47.8% and an accuracy to within one rank of error of 94.2%. The model trained on jumps data achieved an exact accuracy of 49.4% and an accuracy to within one rank of 94.7%. Overall, these results were comparable to those obtained in the previous investigation, and confirm that two separate models to cover all race types will be sufficient upon implementation.

Further work should focus on the introduction of ordinal regression techniques to investigate whether they improve network accuracy, and a full correlational analysis of input variables and final ranks should be completed.

Notes

The aforementioned previous paper can be found at: https://www.quantumleapsolutions.co.uk/wp-content/uploads/2022/04/Quantum-Leap-Solutions-Machine-Learning-Rankings-Analysis.pdf