# BEATING BETFAIR
# WITH
# MACHINE LEARNING

**No Programming or Machine Learning Knowledge Required**

**Mark Littlewood**
**MySportsAI@SmarterSig**



# Contents

1 Getting Started with Machine Learning

2. MySportsAI First Contact

3 My First Model

4 Saving a Model and Making Daily Predictions

5 Hyper Parameter Tuning

6 More Train/Test Features

7 Outliers

8 Feature Scaling Binning and One Hot Encoding

9 What Should we Predict

10 How Many Input Features

11 Price Trading

12 Is Machine Learning Betting Profitable

13 Modelling Other Sports

14 The Future

15 Appendix A Input Features

16 Appendix B Algorithm Abbreviations

# Preface

When The Minstrel crossed the line in the Dewhurst Stakes in 1976 beating Saros I had no idea that this beautiful sport would shape my life for the next 50 or so years. The Minstrel was the first horse I really developed affection for, an emotion that does not go hand in hand with making money from betting. Back then the sources of useful information on how to bet intelligently were limited. Today there are a plethora of avenues one can venture down for decent schooling on what makes a good bettor.
My awakening came much later than 1976. Sure I did OK up to that point betting mainly in areas that interested me, group and listed races. The penny only started to drop however after I read Alan Potts book Against The Crowd along with joining a little known email group called SmartSig.
My first piece of education within SmartSig was when I posted an opinion on the up and coming Champion Hurdle and found that no one was interested in the thread. It took me a while to realise that members were interested in making money and doing so did not have to mean focusing on the top races. They pay the same for a 3/1 winner at Wolves as they do at Cheltenham on Champion Hurdle day.
I started to take betting seriously around the year 2000. Betting tax had melted away and access to large volumes of past data was becoming available. It was time to get the betting boots on. Back in those days it took a lot longer for a firm to ban you. I recall Stan James going a full 18 months before they caved in and admitted they could not beat me.
Flipping online accounts kept me going until around 2007 after which I switched to betting shops, naively thinking that they offered greater anonymity. I quickly found that shops are equally alert to you winning money and I found myself banned from all the city centre betting shops. The final rounds of my battle with bookmakers came in the form of me employing students as runners. I would set up base on a laptop in Costa Coffee in the city centre and dispatch runners to place bets and retrieve any winnings from the previous day. This led to Costa Coffee banning me from their coffee shops due to the fact that young men leaving and returning with wads of cash looked suspiciously like drug running. None of this mattered in the long run as the students were quickly banned.
I realised that a fresh approach to betting was needed. I had been having around 3,500 bets per year and making between 10% and 15% ROI on turnover per year. This was no longer sustainable with no available accounts. While I figured out how to continue betting I decided to run a tipping line. I was effectively saying to other people you figure out how to get bets on, here are the bets. After 4 years I had figured out what was needed. A realisation that a much lower ROI on Betfair was the goal but off a much higher number of bets. The automation of bet production and bet placement needed to be coded, not a problem for someone from my background. My early efforts were based

around ratings produced from my own AE values. You can google these if you are not sure what they are.

That wasn't the end of the journey however. I was increasingly aware that we were entering a new phase in betting. Artificial Intelligence was making huge strides generally and betting was an obvious area of utilisation.  I had been using Machine Learning (ML) in my betting but became aware of the learning curve required for non Computer Scientists or even non Computer Programmers. I decided to implement a GUI based approach to ML, a piece of software that would allow non coders to simply select, click and produce an ML model that could be used for personal betting. This software is called MySportsAI and can be used for betting on other sports other than horse racing. Check my blog below on examples using Soccer and Tennis by simply inputting tennis or soccer in the search box.

https://markatsmartersig.wordpress.com/

# Introduction

A subset of AI is Machine Learning and this book will introduce you to the concepts and methods behind Machine Learning as applied to horse betting. You will not need to be a coder, no knowledge of computer programming is required. I will be holding your hand through the production of Machine Learning models to attack the betting market via software that I have developed myself.

MySportsAI requires no coding, just the ability to click buttons and select things from drop down menus. Something we have all had plenty of practice with on web sites. MySportsAI will enable you to enter this wonderful new world of Machine Learning and betting and make no mistake this technology is revolutionising the betting world. Machine Learning algorithms have conquered Chess and the more complicated game of Go, defeating world champions in both games. Sports betting will not be immune.

The intention of the book is to produce a ready made winning model for you so you can retire from the grind of the day job. This book will introduce you to the methods behind MySportsAI and ML in general. It is not a full depth dive into Machine Learning concepts, rather an overview, a flavour of what Machine Learning is with sports betting as the underlying subject matter. When you have finished you will be confident enough to feel that you can create your own Machine Learning models and there is an almost infinite number of them.

The first section will show you how to install the software. You can read through the book without using the software, simply reading the description of how to use it will introduce you to the world of Machine Learning but to get the most out of the book I advise you install and use the software.

The next section covers how to create your first model. The purists will tell us that data exploration should come first and they are correct but we are going to sample the fun bit first and come back to data.

After creating a model we look at how we can save that model so we can bring it back each day and use it to make predictions on the data for a day's racing.

Any model has under the hood parameters that can influence how well the model will work. We next look at how we can fine tune these so-called hyper parameters.

We then move on to take a look at some of the reporting features in MySportsAI, for example the different ways we can view the results produced by a model and also how we can get the software to perhaps suggest the best input data to feed into a model.

Getting back to data we then check out some of the data engineering features of the software. How can we change data so it's more applicable to a Machine Learning algorithm ?

Outliers within data can be a problem for some algorithms, we look at how to detect and deal with them.

What should we predict, win, place, distance beaten ? and how many input features should we use to make predictions. There are no absolute correct answers, just options to be explored.

Can Machine Learning be used to trade prices and what about sports other than Horse Racing ? I will demonstrate how it can be used for sports like Tennis and Soccer.

Let's get started.

# Getting Started with Machine Learning

We often stick with ideas and procedures that we adopted many years ago. Take the keyboard sat in front of you. You are probably aware that it's called a QWERTY keyboard because those are the first set of characters on the top row. You may not be aware that the layout was originally designed like this in order to slow typing down. The problem designers originally encountered was that the mechanics behind the keys would stick if hit too quickly in succession. Laying the keyboard out like this slowed typing down and hence resulted in fewer incidents of sticky keys. This problem no longer exists but we have stuck with the non sticky layout despite the fact that other layouts are more efficient.

Now ask yourself how has your race prediction methods changed over the years? Have you made any radical changes in direction or are you still treading the same paths in the hope that the blueprint you are using will somehow start to produce real long term profits ? Warren Buffet said the five most dangerous words in business are "Everyone else is doing it", the same applies to betting. It is time for a change.

**What is Machine Learning ?**

Let's kick of with the dictionary definition

"The use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyse and draw inferences from patterns in data."

Not very helpful so let's try and simplify a little further. You are probably familiar with what a computer program is. They are simply lines of programming code that bring about a desired final outcome. A computer program may well have calculated your monthly salary or your gas bill. These kinds of programs are written so they tell the computer what to do. Typically they pick up pieces of data eg your hours worked and then step by step calculate your pay for the month.

Now let's apply that idea to a racing system. We could write a program to pick up the data for tomorrow's runners, sift through them and pick bets based on certain set criteria. What we are doing here is telling the computer what to do in order to come up with the selections. We are giving it the recipe, for example select male horses that have run less than 30 days ago and are not wearing headgear for the first time.

Machine Learning approaches the problem differently. With ML you give the ML algorithm the historical data and ask it to figure out the chance of each horse winning the race it is competing in. You are not prescribing how to produce these results, you are asking the ML algorithm to do it for you. Notice also that the ML approach will produce probabilities or ratings if you like with the top rated horse in a race having the highest probability.

So far I have referred to the 'ML algorithm' when in fact I should say an ML algorithm as there are more than one of these mystical creatures. One you may have heard of is a Neural Network or perhaps you have heard of Linear Regression ?

Most Machine Learning text books define the process of creating a model as a series of steps.

1. Get hold of or create the data. No need MySportsAI does this for you in terms of historical data and today's runners
2. Visualize the data to gain insights. No need to write code for this MySportsAI enables you to look at the data in terms of visual plots
3. Prepare or clean the data. No need to do this MySportsAI has the data not only clean but in a format suitable for Machine Learning
4. Select and train a model. Again no need for code MySportsAI will do this at the click of a few buttons
5. Fine tune the model. MySportsAI takes care of this, fine tuning those bells and whistles under the hood of the model
6. Run the model and make predictions. Yes it does this too allowing you to get predictions for todays runners from your model

**Setting things up**

Enough of the introductions, things will become clearer as we dip our toe into the ML waters. You could read through his book without trying out MySportsAI but you will get more out of the process if you actually use the software. To do this we need to download and install MySportsAI onto your laptop. To purchase MySportsAI you will need to make a payment of £90 via the link below. This gives you lifetime access to the program and 3 months access to the daily data downloads that enable you to predict the days racing.

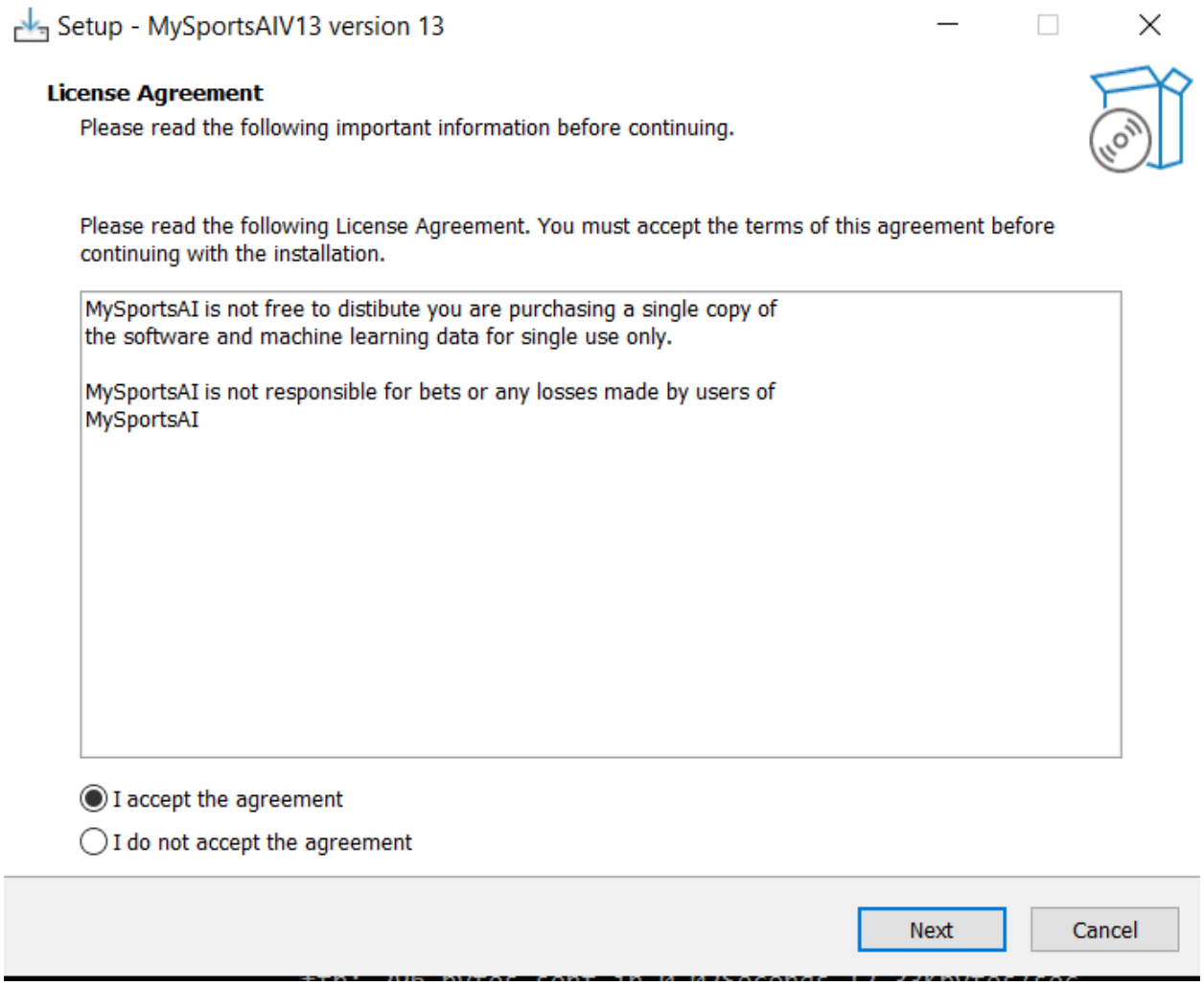
http://www.smartersig.com/membership.php

So you have made your subscription payment and you are now ready to install the software. Log in to the web site and go to the utilities section of the web site and click on the latest version of MySportsAI in the listing. As I write the latest version is MySportsAIV13.




| Title | Author | Description |
|---|---|---|
| MySportsAIV13 | MySportsAI | Upgrade which incorporates the facility to create an oddsline that incorporates BFSP Bill Bentner style |
| MySportsAIV12 setup | MySportsAI | Installer and data for MySportsAIV12. If it downloads as a crdownload file just rename it as mysportsaiv12setup.exe and execute |
| Todays AM Odds | MySportsAI | Todays 9am and 9:15am bookmaker odds |
| OddsLineSheet | MySportsAI | Excel file that allows you to calculate an oddsline |
| MySportsAIV11 | MySportsAI | Latest update no data files included so just copy across your data files from V10 if you are a new user email me |
| dataflat23am.csv | MySportsAI | A selection of races from 2023 which include best AM price with big bookmakers, bookmaker SP and BFSP. They are headed BFSP,SP and BFSP2 respectively. Change these as needed |
| MLdarts.csv | MySportsAI | A MySportsAI ready darts data set results up to October 2023 |
| createmlfootie | MySportsAI | A program to create a combined football results and expected goals file for use in model building within MySportsAI. Read https://markatsmartersig.wordpress.com/2022/12/06/simple-xgoals-model-3/ |
| histmatchweights | MySportsAI | One of two files needed by createmlfootie |
| repteamnames | MySportsAI | One of two files needed by createmlfootie |
| myDailyRun | MySportsAI | Creates your models daily ratings without having to enter MySportsAI. Make sure you have copies of your models in the MyDailyRun folder along with a file userpass.txt containing two lines, your SmarterSig userid and password and a file |

You will be prompted for your user id and password which will have been supplied to you when you subscribed. The installation software will take a few minutes to download onto your computer. Once the download has finished we need to install the software. The installation software should be in the downloads folder of your PC under the name of mysportsaiv13setup.exe . Simply double click the file and you will be presented with the following screen.




Click I accept the agreement followed by Next

Click Next on the following screen

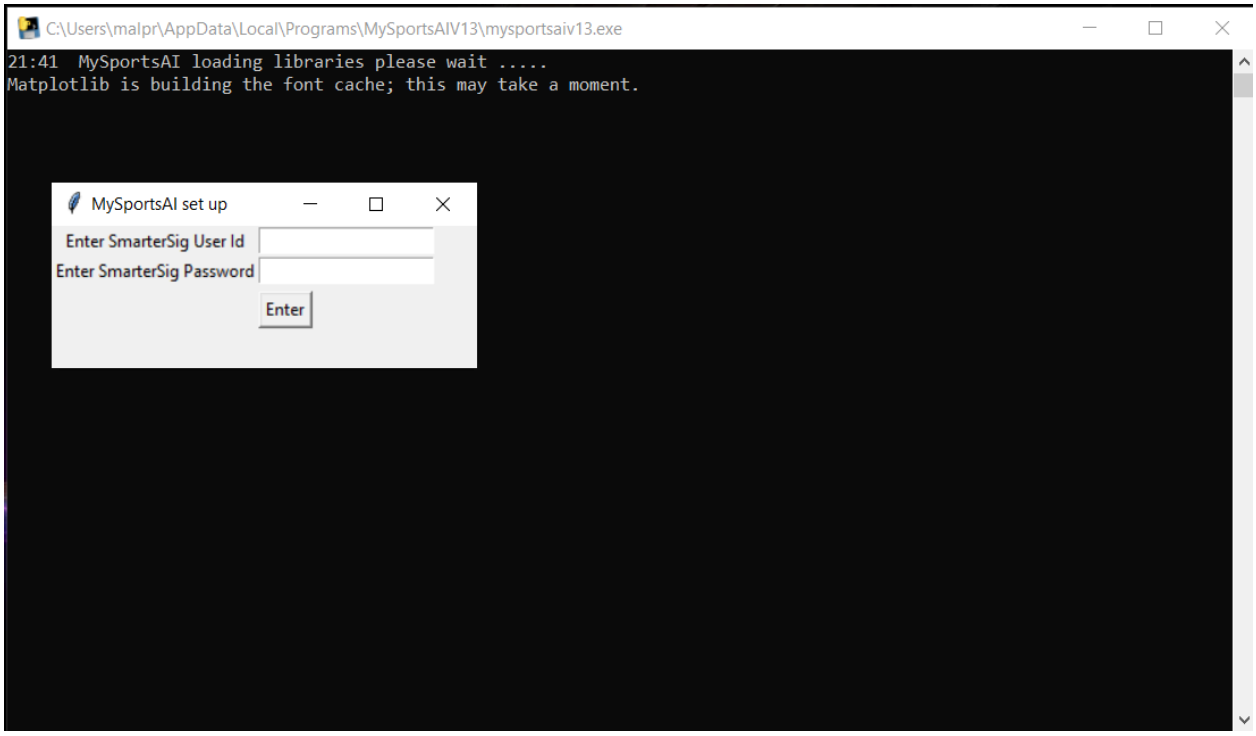
Now click the checkbox ‘Create a desktop shortcut’ followed by Next

On the final screen click install to set the install procedure in motion. It will take a few minutes to install the software. Once installed if you right click on the MySportsAI icon now on your desktop and then select properties it will show you exactly where the software has been installed.

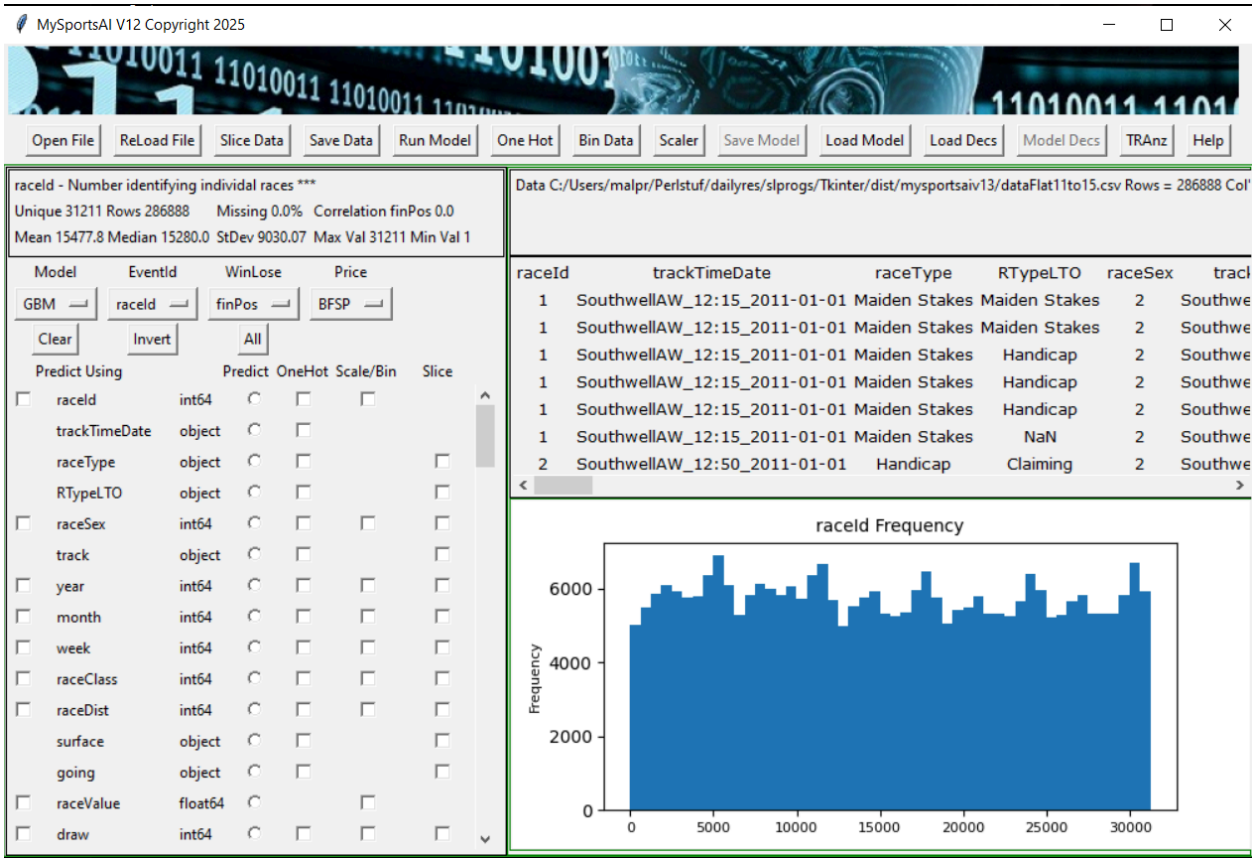
We are now ready to begin our journey into Machine Learning and horse race betting.

# MySportsAI First Contact

You have installed the software and you are now ready to test drive it. Launch the software by double clicking on the MySportsAI desktop icon (note the software may self start after the installation if you chose that option). After a minute or so the first request from the software will be for you to enter your user id and password, see below



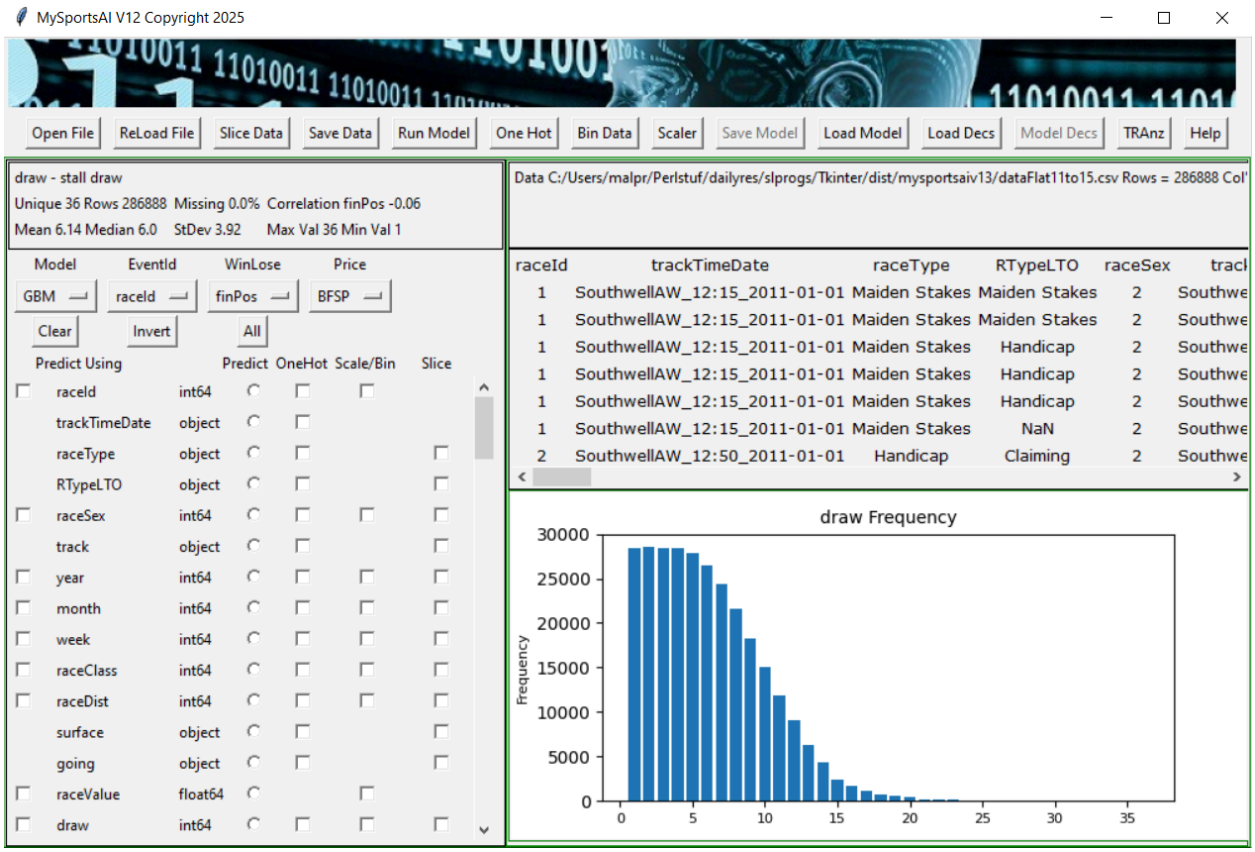

Once you have entered your id and password the following main screen will be launched. If the screen below appears but with no data showing, don't panic simply click on file and select dataflat11to15.csv to load up that file.



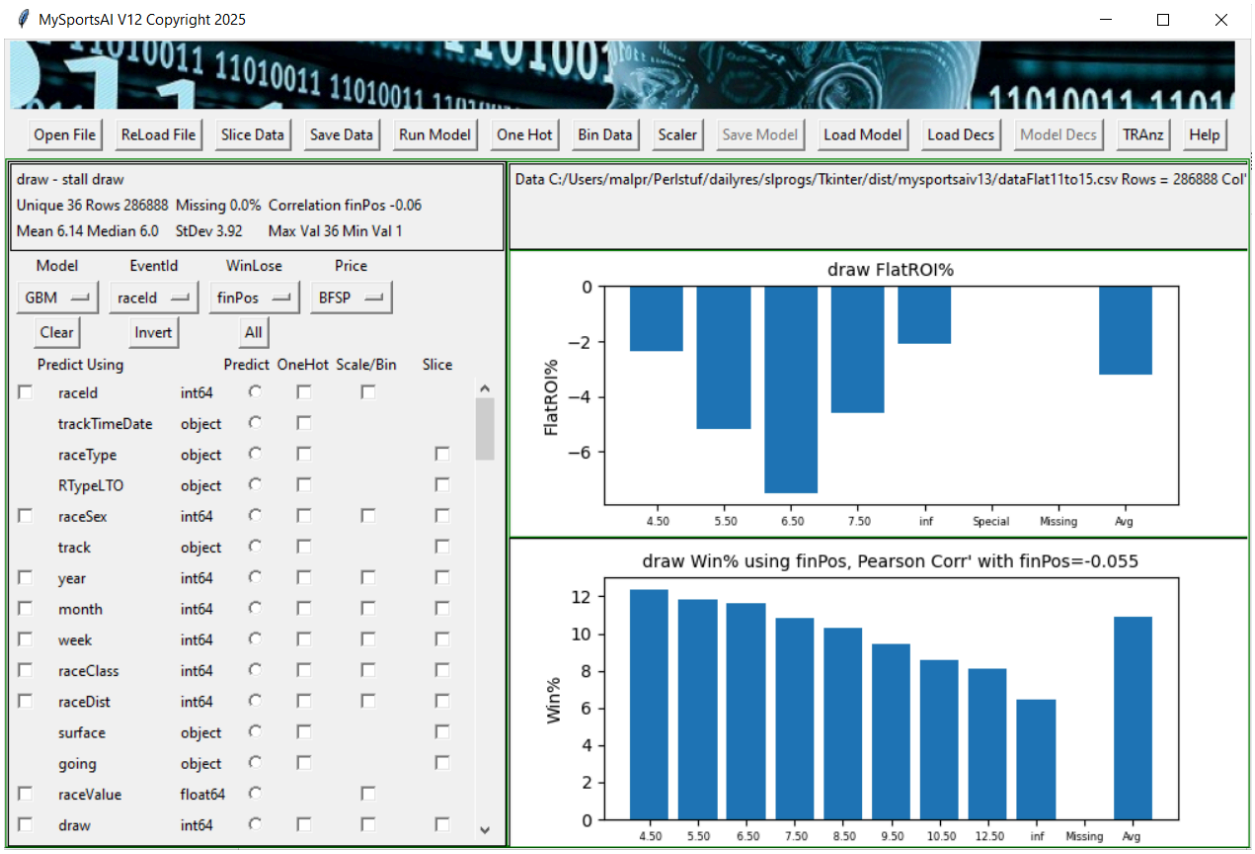

Let us get a feel for what this screen is telling us. We will start in the top right corner section. This area is telling us that we have loaded up flat data from 2011 to 2015 and it consists of 286,888 rows of data.

Looking at the section below we can see the first 7 rows of data and if we scroll across we can inspect all the columns. Each row contains the details of a horse that ran in a particular race. You can see that the first 6 rows are a maiden race, the 7th row is the start of a handicap race. It is worth looking at the 4th column namely RTypeLTO. This refers to the race type the horse ran in last time out. Notice horse 6 has NaN entered. This is Machine Learning language for essentially a blank entry. Given it is a maiden race the horse clearly is making its debut.

We will now take a look at the bottom right of the screen but to get a more meaningful graph displayed I have placed the cursor on the draw feature at the bottom left of the screen. This will make the graph change to details about the draw.




This shows the frequency of the various values for draw in the data. Not surprisingly draw 1 has the most occurrences and there are steadily fewer draw values as the draw increases. If we click on the int64 value next to draw we get perhaps a more revealing pair of graphs.




This shows us the win percentage of each draw, albeit across all courses and the return on investment betting £1 on each draw. The top graph may look a bit odd with the first draw value being 4.5. This is because the graph is constructed by placing the values into buckets. The first bucket is all draws up to 4.5 which is clearly draws 1 to 4. Draw 1 to 4 made the least loss blindly along with draws 8 and above.

Moving onto the main left hand section. This contains all the input features we can feed into a model. Each feature also has a type descriptor. We can see that draw is an int64 meaning it's an integer or whole number. Looking above we can see that raceValue is a float64 which means it is a floating point number. Features described as object means they are not numerical values. The columns to the right of the type descriptors will be covered when we start to build a model.

Finally the top left section contains details of the feature you have just placed the cursor over. You can see from the above that draw has 36 unique values (maximum field size found in the data). Missing values stand at 0.0% so no rows have NaN in the draw column. The correlation with finPos is -0.06. A negative correlation means that as the values get bigger the less likely a horse is to win. A positive correlation would mean as the draw gets bigger a horse is more likely to win but clearly this is not the case here.

The value i.e. 0.06 is the strength of the correlation. Larger values whether they are positive or negative means a stronger connection. The final line of information is this section is fairly self explanatory.

OK enough of the guided tour let us get our feet wet and build a model.

# My First Model

In Zen Buddhism there is a principle known as Shoshin or beginner's mind. The Zen teacher Shunryu Suzuki wrote "In the beginner's mind there are many possibilities, in the expert's mind there are few". A beginner's mind can bring fresh thoughts and ideas. When J K Rowling submitted her first draft of Harry Potter to publishers she received rejection after rejection. It was not until it landed on the desk of the chairman of Bloomsbury Publishing that the potential was noticed but not by the chairman. He gave a sample to his eight year old daughter who lapped up the contents and begged for more. It cost only £2,500 for Bloomsbury to get the rights to the book, probably one of the best publishing investments of all time. MySportsAI does the magic, you bring the imagination and the Shoshin.

MySportsAI comes with data for flat horse racing 2011 to 2015, 2016 to 2017, 2018 to 2019 and 2020 to 2024. The reason for these split into different files will become clear as we start to build a sample model. The purpose of this book is however not to show you how to build the world's greatest predictive horse betting model, after all if it did that everyone would have access to it and hence overbet the model into lower or non profit status. The purpose of the book is to show you how to build a model and introduce you to key Machine Learning concepts. In the same way a woodwork class does not show you how to solely build a wardrobe but rather teaches you key concepts and techniques that allow you to build your wardrobe of your design.

Before we dive into model building let us first get a feel for what we mean by a model. Take a look at the following numbers and see if you can spot the pattern.

| x1 | x2 | y |
|---|---|---|
| 4 | 2 | 10 |
| 1 | 4 | 13 |
| 0 | 3 | 9 |
| 2 | 1 | 5 |

Now what is the value of y for the following numbers

| | | |
|---|---|---|
| 3 | 1 | ? |

Hopefully you detected that x1 + 3 X x2 = y

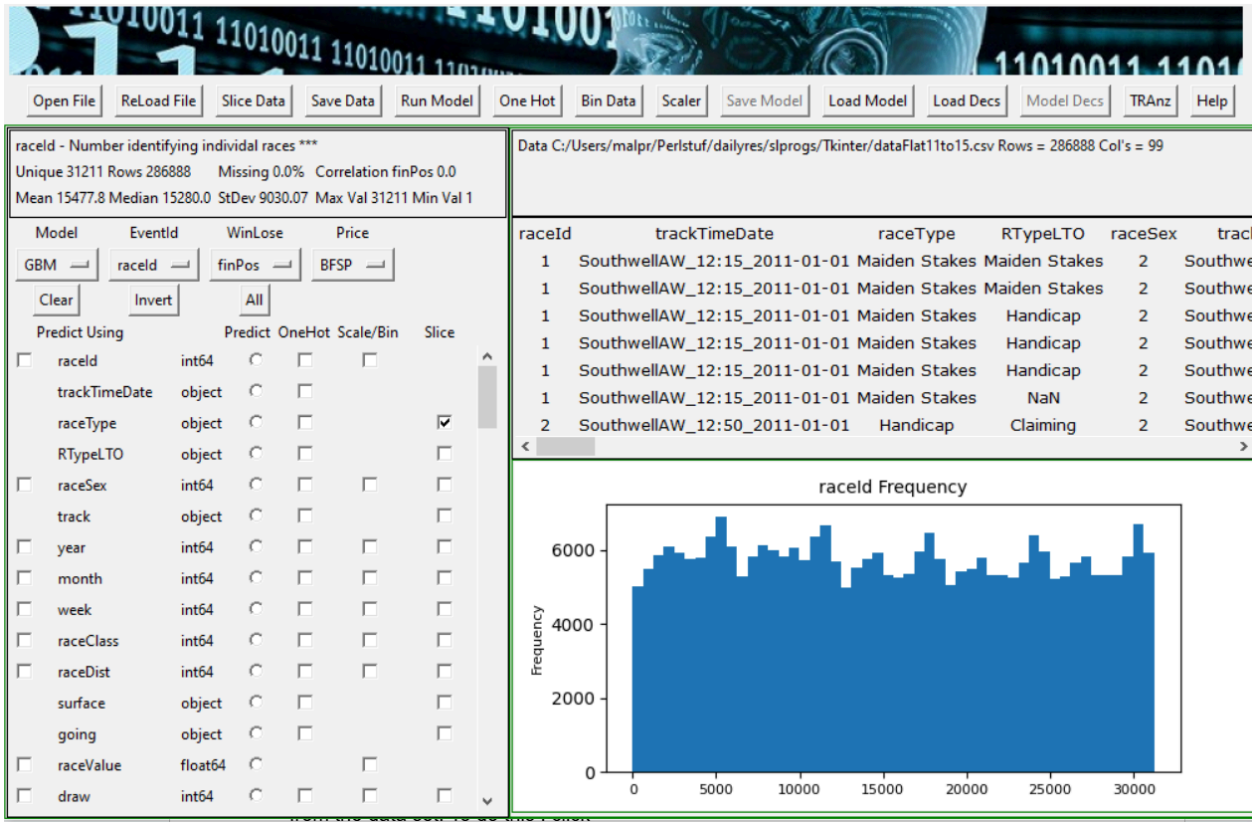
We can write this more succinctly as y = x1 + 3x2

Well done you have found one relationship between x1, x2 and y. Furthermore this relationship you have found is called a linear relationship. We can see that before adding things together we need to multiply x2 by 3 and x1 by 1. We call these values 3 and 1 coefficients or weights of the linear relationship between x1, x2 and y. We can now use this relationship to discover the ? on the last line, in other words make a prediction of 6.

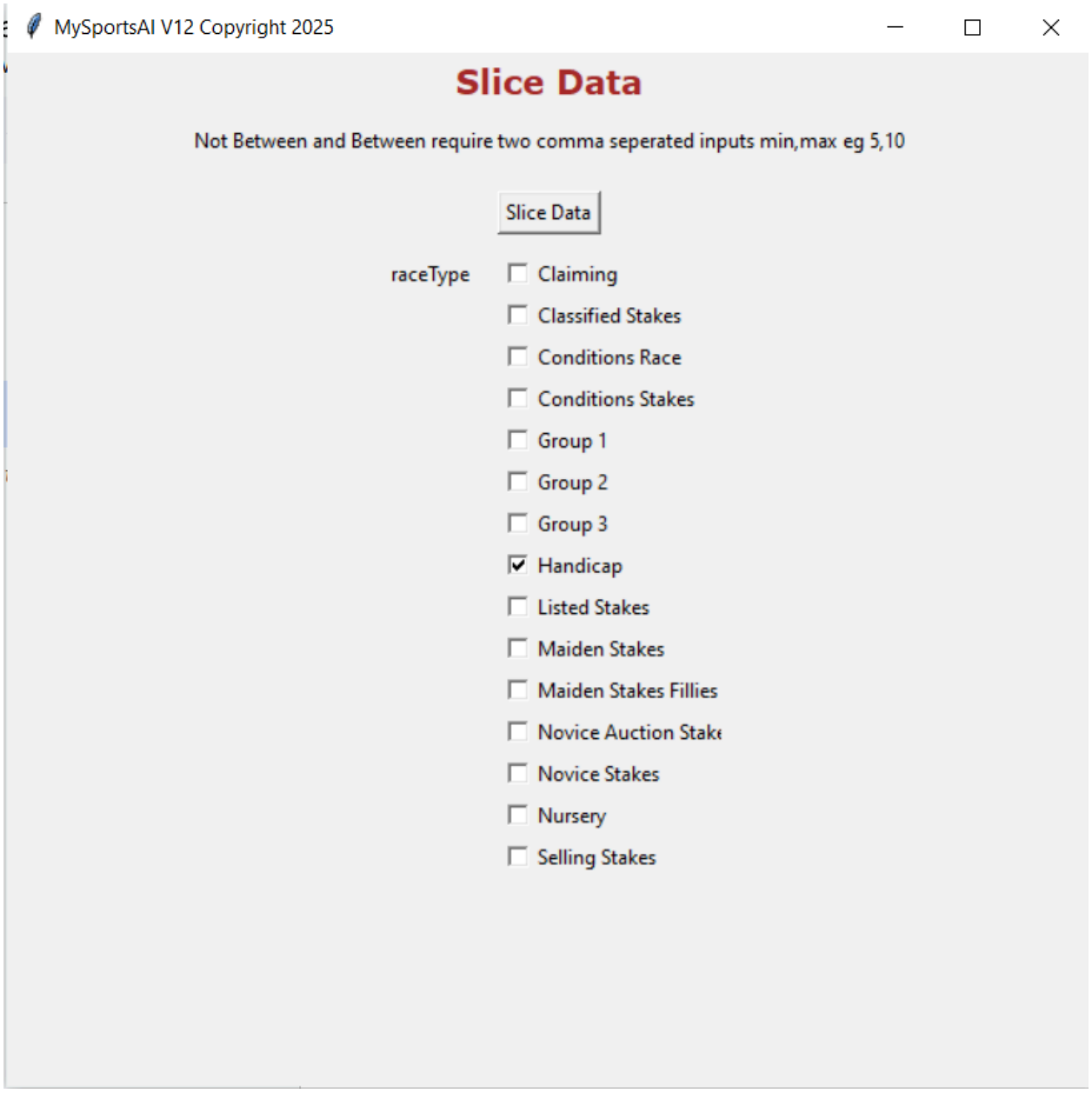
So far everything seems a bit abstract but imagine x1 was the number of rooms in a house and x2 was the floor space of a house. Maybe with these two variables and enough examples of data we could come up with a meaningful formula to predict the sale price of the house, y in other words.

Of course we are interested in sports predictions not house sales. Our variables may well be days since a horse last ran for horse racing or average expected goals from the last three matches for soccer. The y value could be did the horse win or lose or perhaps the distance the horse was beaten by. For soccer predictions it could be win or lose or goal superiority. Eyeballing data when only two variables are involved is something humans are capable of or at least we can get our heads around such data but when data expands to more than 3 variables we tend to have trouble even imagining the hyperplanes that the relationships exist on. This is why we need Machine Learning algorithms to search for the patterns within the data and produce a meaningful relationship or model.

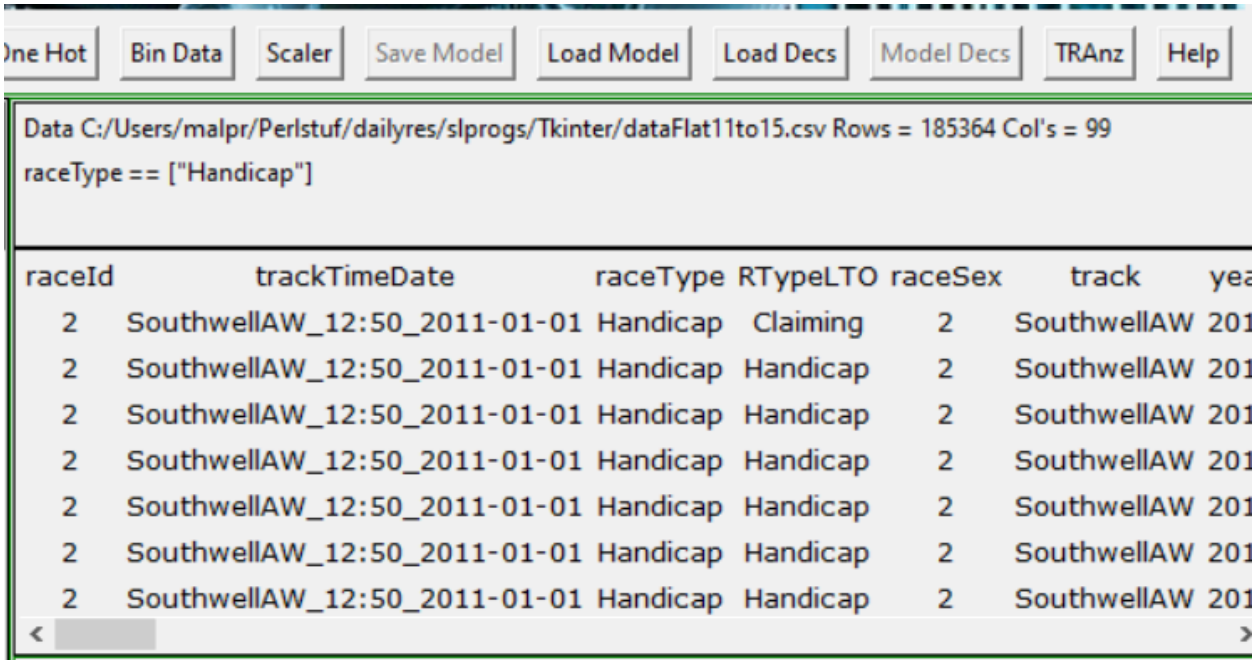
Our first step for our first models is we need to select some input features for our model. Feature selection is a subject in itself with techniques developed to improve this process but we are simply going to use our knowledge of racing and plump for a few input features to demonstrate model building. Before I select my features I first need to decide whether it would be a good idea to build a model on all types of races or a model for a specific type. The effects that input features have may vary for different race types. For example if I select the trainer strike rate (wins / runs) this will be different for a trainer who specialises in handicaps rather than early 2yo winners. For this example I am going to select handicap races only from the data set. To do this I click the checkbox in the slice column next to the feature name raceType



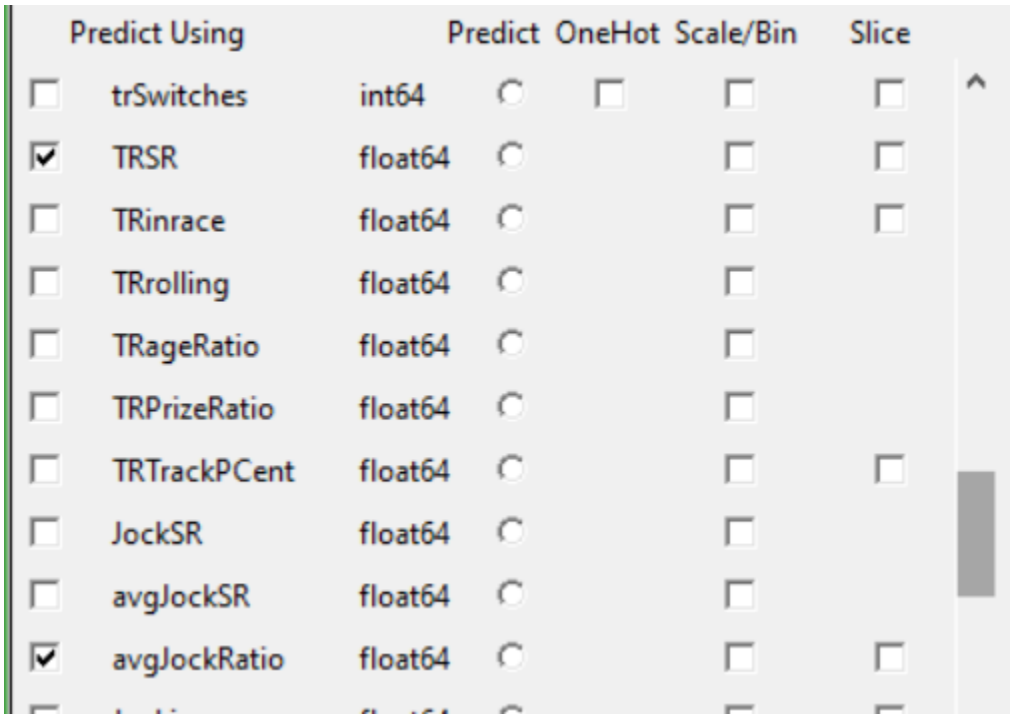

I then click the slice data button along the top button bar which presents the slice data menu. Here I select handicap and click the slice data button




You can now see that in the main screen the top right section shows that I have sliced the data and it now only contains handicap races.



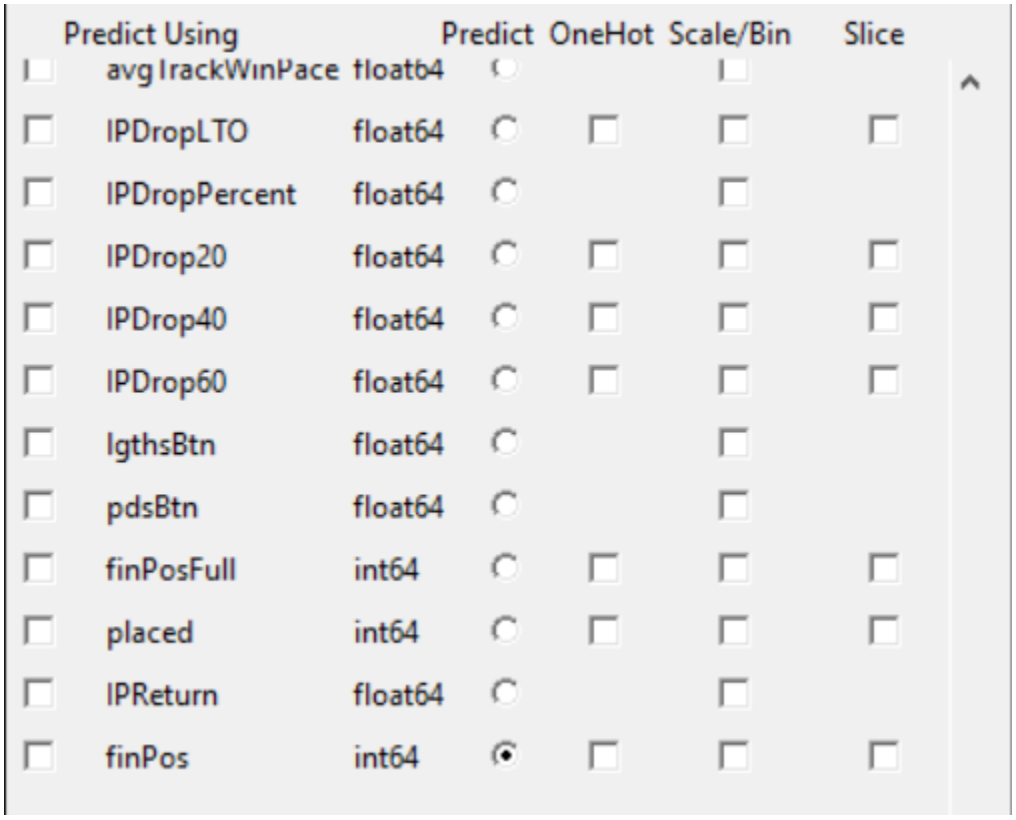

I am now going to select just a couple of input features by scrolling down the input feature section and selecting TRSR and avgJockRatio in the Predict Using column.




These selections are for demonstration purposes, there is no magic correct number of features to choose, sometimes small is better than big. TRSR means trainer strike rate

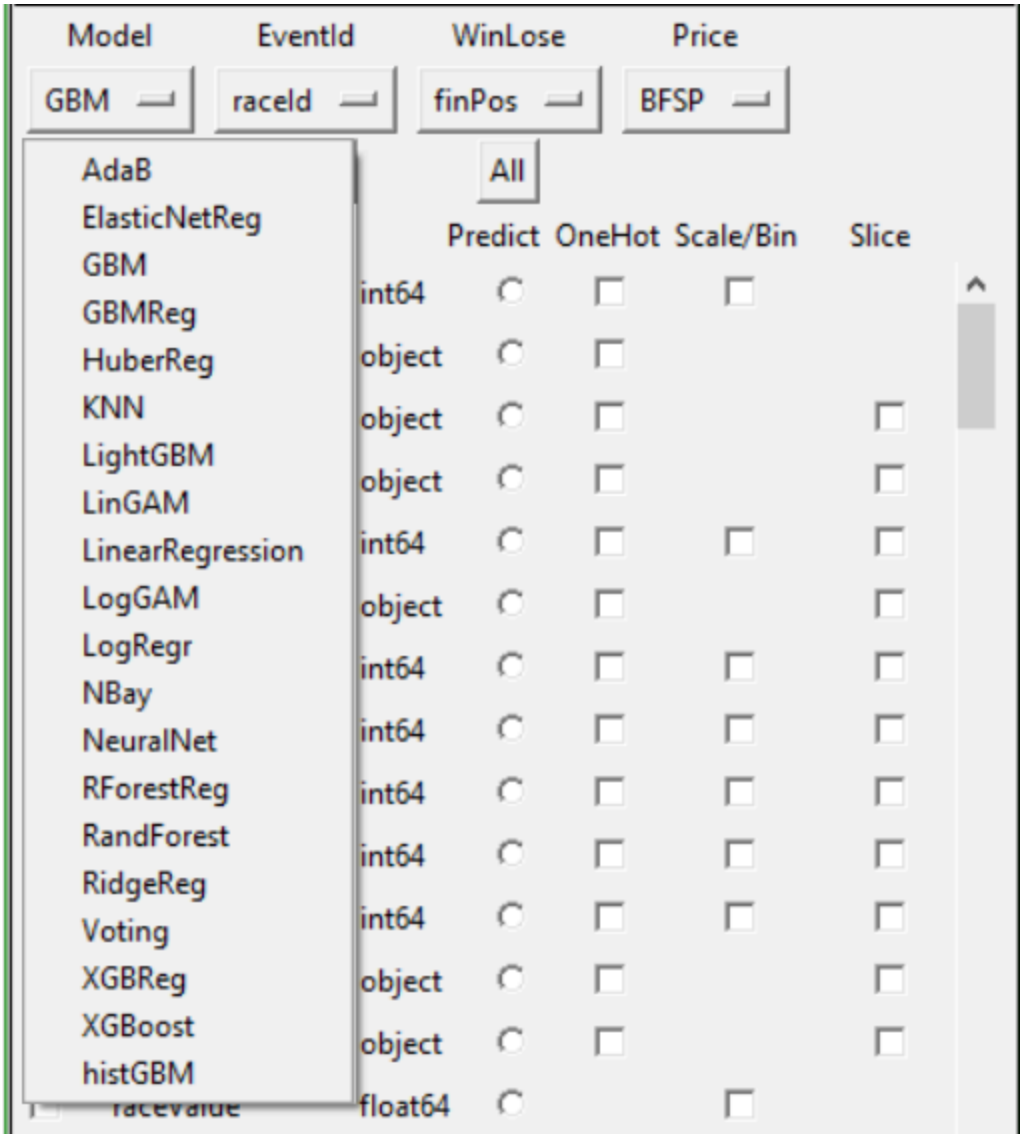
and avgJockRatio is the ratio of the jockeys strike rate to the average strike rate of all the horses previous jockeys. Both of these are accurate on each line of data at the moment of that particular race. A description of each feature is displayed in the top left when you place the cursor over the feature and in the help files shown when you click the help button.

Next I need to decide what I am going to predict. Scrolling to the bottom of the feature section we can see that finPos is already selected for us.




The predict column is where we select what we want to predict. finPos contains either a 1 if the horse won or 0 if it lost. Predicting something that contains either a 1 or a 0 is called a Binary Classification model. We could have selected to predict lengths beaten (lgthsBtn) this would have made our model a regression model as we are trying to predict a continuous variable. Distinguishing whether you are dealing with a classification or regression problem is important when you come to select an algorithm to use. Some are designed for one or the other but not both.

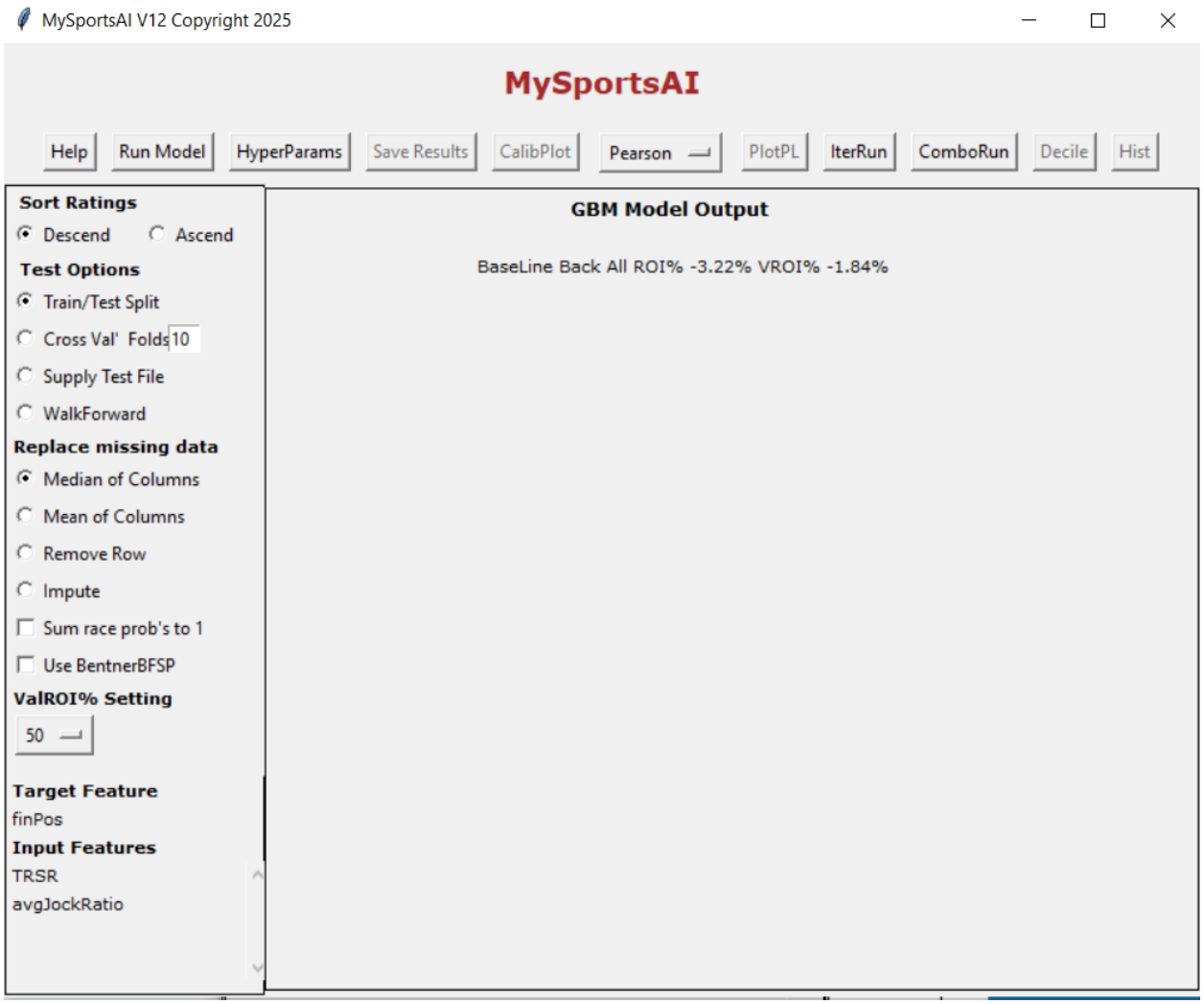
That brings us on to algorithm selection. If you click on the model button towards the top of this section you will be offered a choice of a number of ML algorithms.




We are not going to do a deep dive at the moment into any one of these. We can see however that some of these are clearly for regression problems and others for classification. For now let us simply select GBM which stands for Gradient Boosting Machine, it is a good choice for this type of data but more on this topic later.

Finally we need to make sure that all the other drop down menus next to the Model drop down we just set are all OK. The Evenetid refers to which input feature will allow the software to differentiate one event from another. In this case raceId clearly does this. The WinLose drop down allows us to choose which feature allows the software to decide whether a horse has won or lost, obviously finPos does this.Finally the price drop down allows us to set which feature contains the price of a horse, in this case it is BFSP (Betfair SP). These all seem fairly obvious but there are scenarios where you

may want to choose other features. We are now ready to click the Run Model button which will take us to the model training screen.



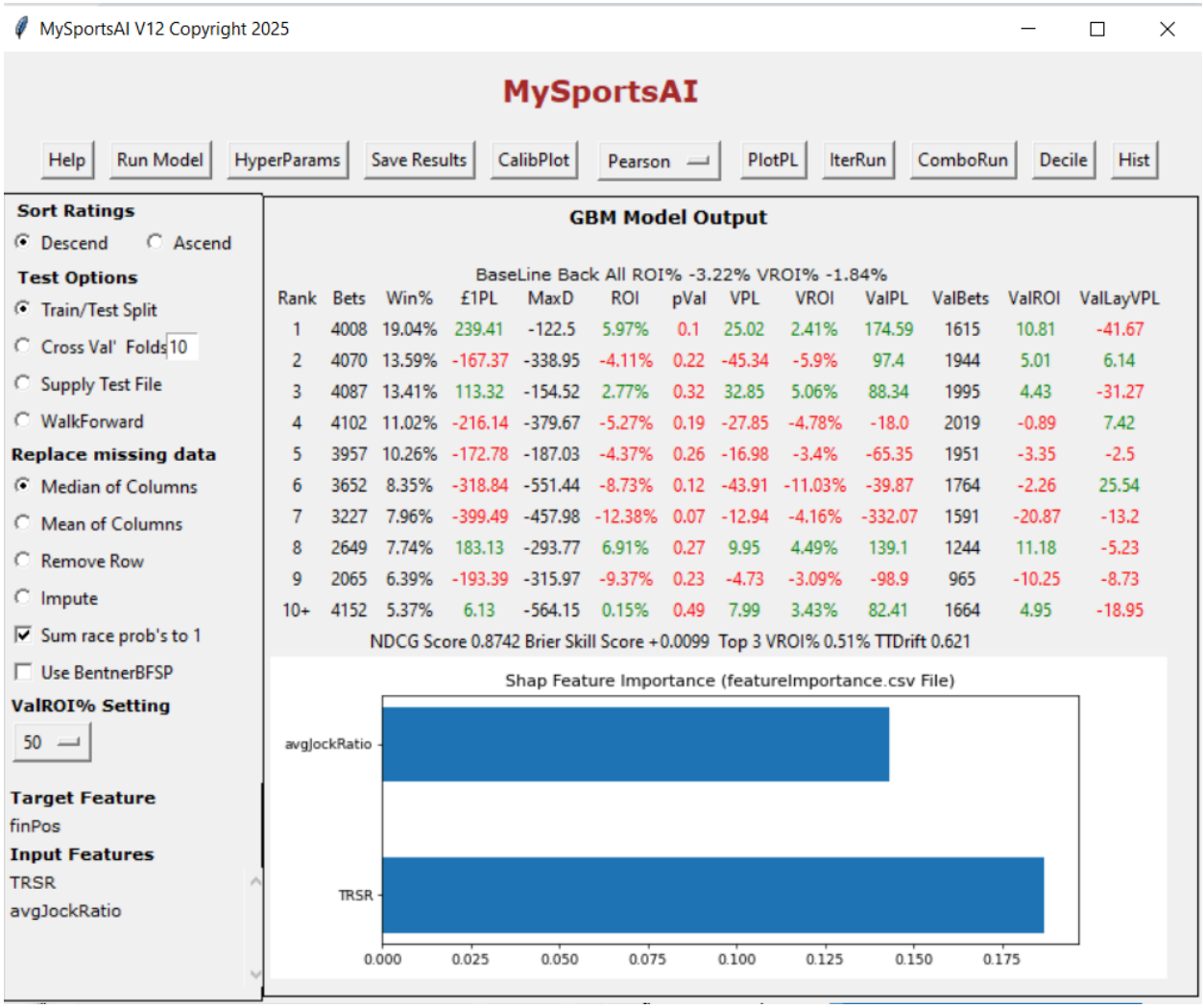

The left hand side of this screen needs a little explaining. GBM will produce probabilities for each horse it makes a prediction for. We therefore want to sort ratings into descending order as a horse with a probability of 0.2 clearly has a better chance than one in the same race with 0.15. Imagine however if we had chosen earlier to predict lengths beaten using a regression model. In this case we would opt for ascending order as a prediction of a horse to be beaten 1 length is better than a horse predicted to be beaten 3 lengths.
In test options we will leave the default train/test split checked. This is a very important aspect of ML. We train the data on one set of data and then test its predictive worthiness on a fresh set of data. Train/test split will split the 2011 to 2015 data into 80% for training the model and 20% for testing it. Importantly it will select the first 80% for training as with race data the ordering of data is clearly important. Think of a new

trainer's strike rate evolving over this period. Training the model on the last 80% and then testing it on the first 20% would give false results in that the model has learnt from future data. Train test splits are a little bit like those past paper exam questions you used to do at school. You trained yourself on past exam questions and then sat the exam. Of course if the exam consisted of the same past exam questions you had trained on, your results would be outstanding and you would be off to Oxford or Cambridge. On arrival however you would fall flat on your face. The same applies to train and testing model's, train on one set of data and test on another.

We will cover the other 3 train options in a moment but first the missing data section. It is quite possible that any feature we choose could have missing data in a row. For example trainer strike is not formed unless the trainer has had at least 50 runners. Where a trainer has not yet reached that number one of those NaN values is entered. We need to choose how we want the ML algorithm to deal with those rows. There are essentially 4 options, replace with the median value of the column (That is the median column of all other trainers), the mean value, remove the row or impute the value. The first 3 are fairly self explanatory. Impute means the algorithm will try and predict the missing value by using the other predictors or predictor in this case as we only have 2 features. Which one is best will depend on the nature of your data. For now we will stick with the default but do not be afraid to experiment. One thing I have learnt is that a great deal of ML science is in fact experimentation. Do not think there is one correct answer or method.

The next checkbox reads 'Sum race probs to 1'. Remember that the GBM algorithm produces probabilities for its predictions for each horse. Checking this box will ensure that after it has produced the probabilities it then adjusts them to make sure that they all within each race add up to 1. We will check this box before training the model.

The use of BentnerBFSP and ValROI% entries I will skip for now and come back to when we have some results up on the screen. The final sections simply lists the input features and the target feature as a reminder. OK let us train the model by clicking the Run Model button.




Great, we have some results, let's run through this report. First the baseline numbers, backing all horses would have resulted in a loss of 3.22% and to variable stakes a loss of 1.84%. All results are to BFSP after 2% commission is paid on winners. The top rated horse according to the trained model however produced 4008 bets with a win percentage of 19.04% and a profit of £239.41 for £1 staked. The max drawdown was -£122.5 and the ROI was 5.97%. Backing to win £1 with variable stakes produced a profit of £25.02 with a variable ROI of 2.41%.

There were 1615 value top rated bets producing an ROI% of 10.81% and if we had laid horses that were poor value we would have lost £41.67. Each row after that shows the performance of 2nd rated, 3rd rated etc.

Below this we have some performance metrics. In the case of NDCG score and Brier Skill Score, bigger is better. For a full explanation of these two check out my blog at

https://markatsmartersig.wordpress.com/

The Top3 VROI% shows the variable stake return backing all the top 3 rated horses. The final metric is the TTDrift. This is a measure of how much the test data resembles the training data. Ideally we would like to see this number at 0.5, this is rarely the case but the bigger it gets the more cautious we should be as this means the distribution of the data has drifted from the training phase to the test phase.

What does that mean in plain English ? Allow me to give a concrete example. One input feature is days since the horse last ran. We are probably all aware that in 2020 racing was closed down for 3 months or so due to Covid. That meant that every horse that reappeared was at the very least showing a day's since the last run of 93 days and yet your model, if it used this feature, would have been trained on data that had the usual mixture of days since last run. Clearly the data has drifted and results are going to be unpredictable. This is why I warn users to be wary of 2020 data.

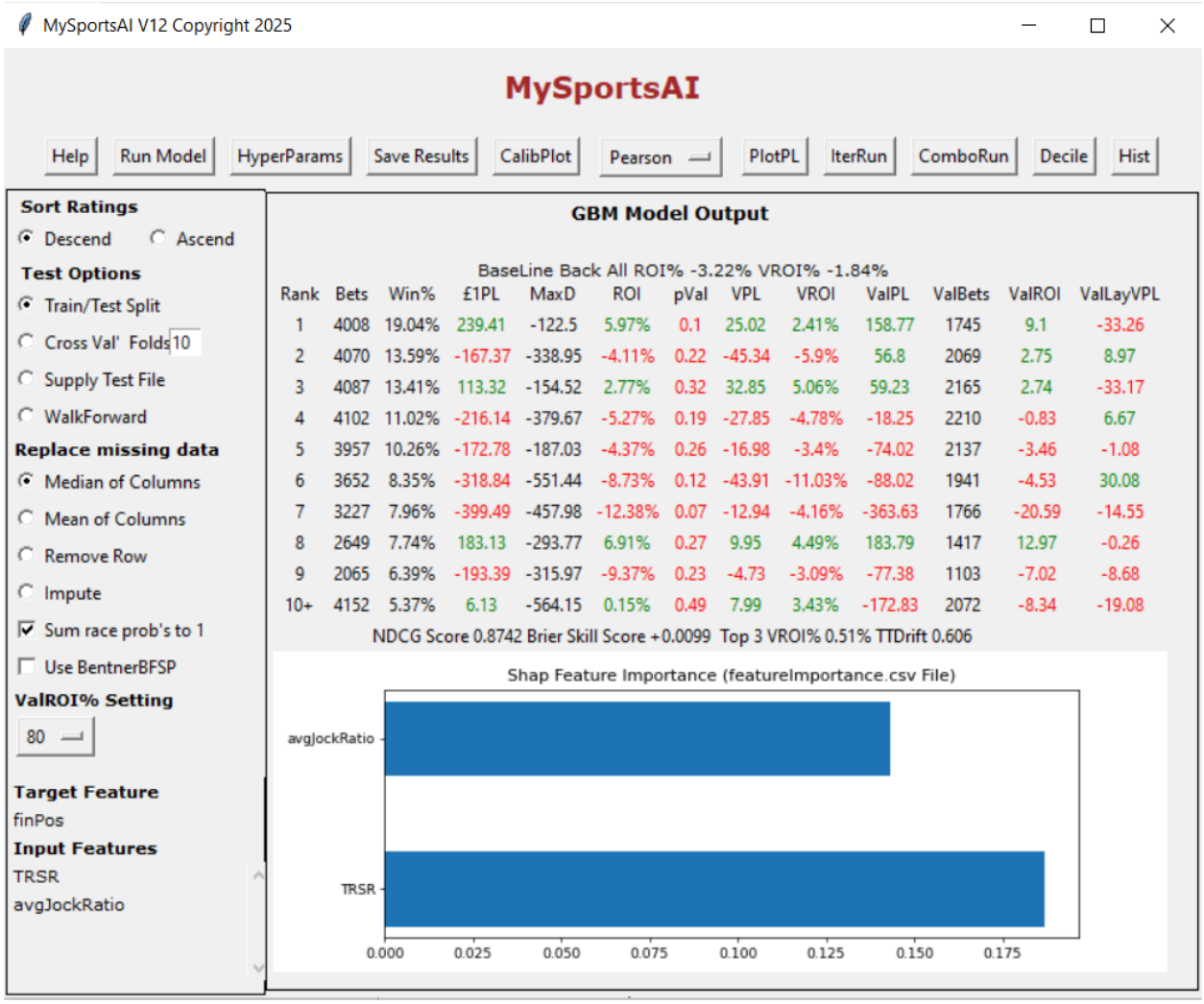
The final section shows a bar chart of which features were found to be the most important when it constructed the model. We can see that TRSR was the dominant feature.

Now let us cover the ValROI% setting. We could deem a value bet as simply when the model's probability for a horse is greater than the BFSP implied probability. For example if our model deems a horse to have a 0.34 chance and the BFSP is 3.0 which equates to a probability of 1 divided by 3 or 0.33 then we could say it's a bet. If the model came up with 0.32 then it is a no bet. The problem with this is we probably want a bigger edge than this and at larger BFSP prices we would want a bigger edge than with smaller BFSP prices. We would not want to back a 25.0 BFSP horse just because our model is saying it's price is 24.5. The software compromises by allowing you to state a ValROI% value (50 shown at the moment) which is then applied to create a threshold via the following equation.

Threshold = 1 + BFSP/ ValROI%

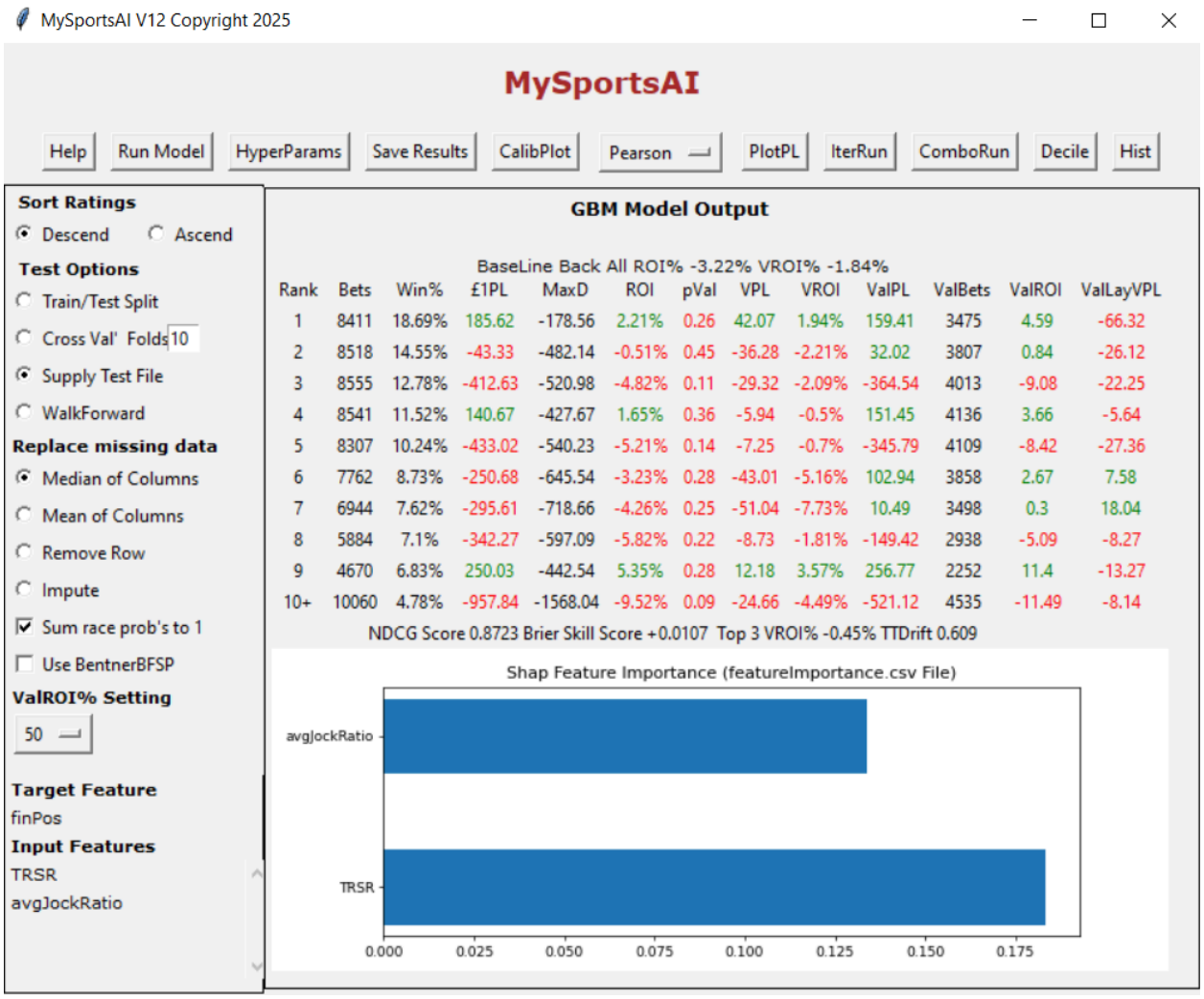
A value bet is now logged if the model probability is greater than (BFSP probability X Threshold)

Do not worry too much about these details for now, suffice to say that adjusting the ValROI% upwards increases the number of value bets as we can see below.




We have not covered all the test options yet. Cross Val' Folds is an option I do not use much. It involves creating a train test split based on 90% and 10% of the data if we leave it set to 10 as above. It will then run this in the usual way but instead of stopping there it will shuffle the 90% selection forward 10% and the remaining 10% will be the test portion. It repeats this until all 10% portions have been used as the test data whilst accumulating the results as it progressed. As you can probably see this involves training on future data and testing on past data and for this reason I tend not to use it.

Supply test file allows us to train on all the data from 2011 to 2015 and then test it on another data set, for example 2016 to 2017. Let's try that and see how things stand up. After being prompted to choose a file from my MySportsAI folder and choosing dataflat16to17.csv the following results were produced.

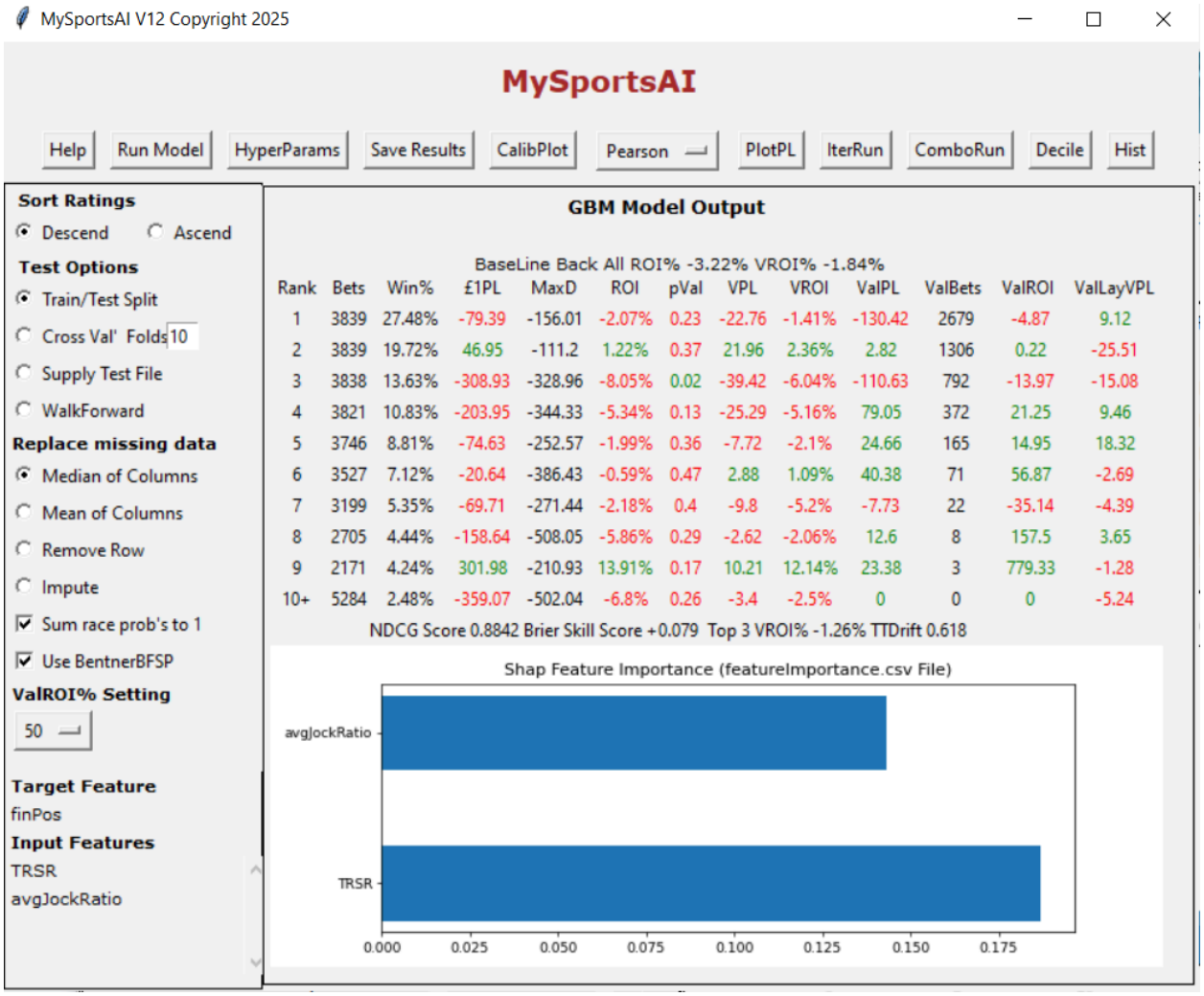



The results have stood up reasonably well in 2016/17 with the top rated still making a profit, the drift at 0.609 is however a little concerning.

The final test option is walkforward. Walkforward will train on 2011 and test on 2012, train on 2011 and 2012 test on 2013 and so on, the combined results will be displayed.

Finally we have the BentnerBFSP option. First of all apologies to BIll for spelling his name wrong, I must correct this, note to self. Benter was a highly successful pro gambler who operated in Hong Kong, his models were highly successful which is impressive given that the PM take out in Hong Kong is around 17% compared to just over 2% on Betfair. Bill however found that he was unable to crack the problem until he admitted to himself that the market carried information that no amount of modelling could account for. It was not until he incorporated the market price into his model that he then started to produce profits. Clicking this option in MySportsAI will incorporate

Benter's formula into the probability formation and hence influence the value bets. You can still slide the ValROI% number to increase or decrease the number of value bets.

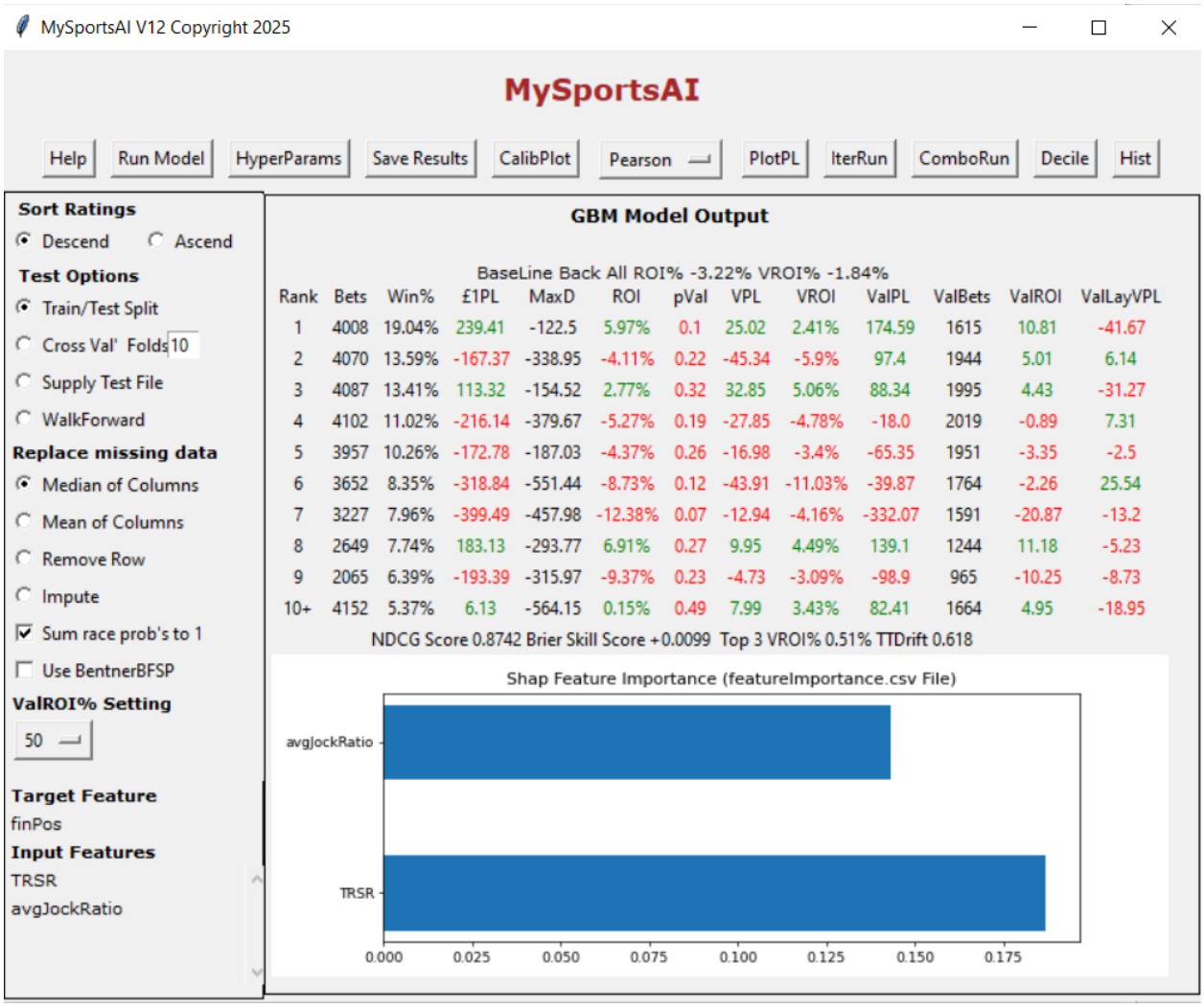



We can see that incorporating the market on this model has not really improved performance. The win percentage of top rated horses has gone up as the ratings are pulled closer to the market but profit has not generally improved. Model building is often about this relationship between accuracy and profit. If we are betting to an oddsline we want accuracy. It may be only marginally profitable to BFSP but we have lots of opportunities with prices fluctuating before BFSP emerges. On the other hand if we intend to simply bet top rated horses then consistent profit across all test sets is something we are probably focusing on.

# Saving a Model and Making Daily Predictions

We are now going to take a look at how to save a model for future use along with how to create daily predictions.

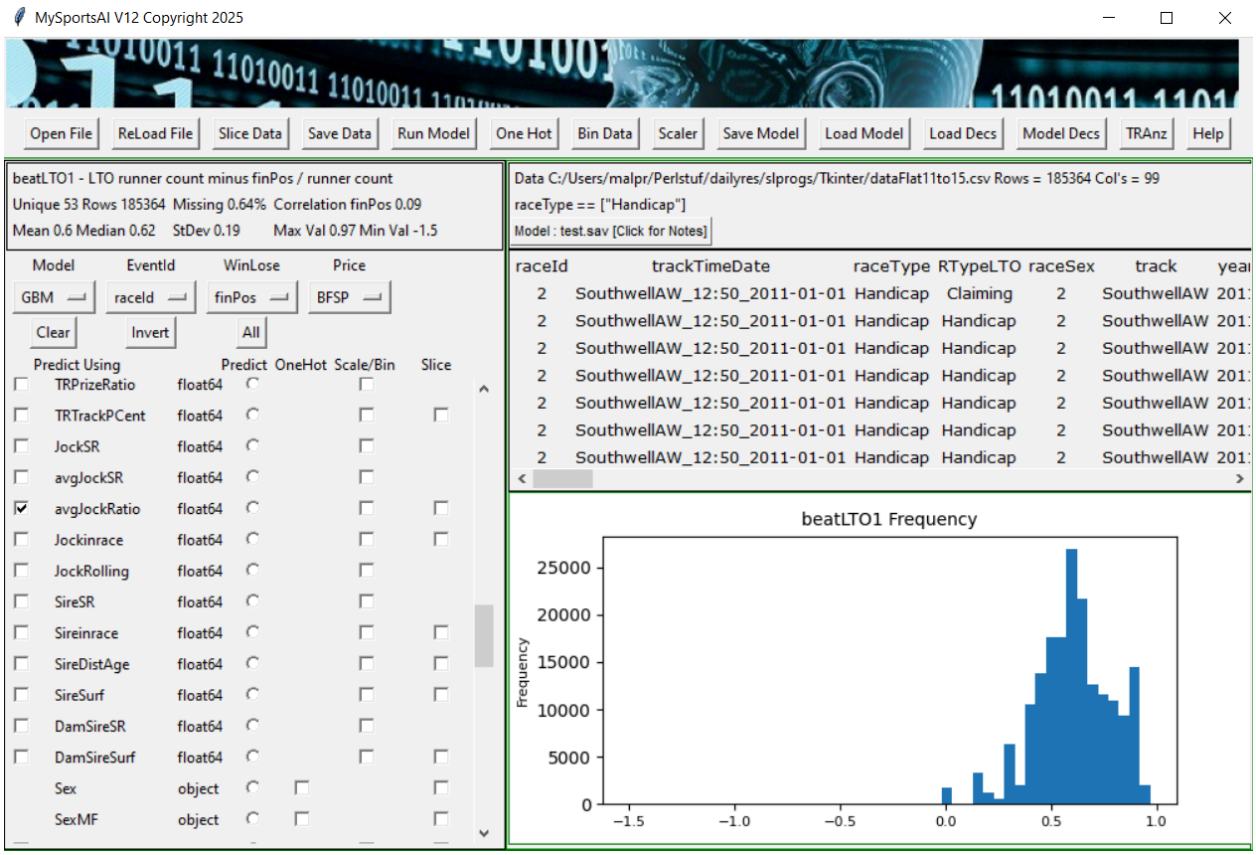
I have retrained the model using GBM thus flushing out the KNN model so to speak.



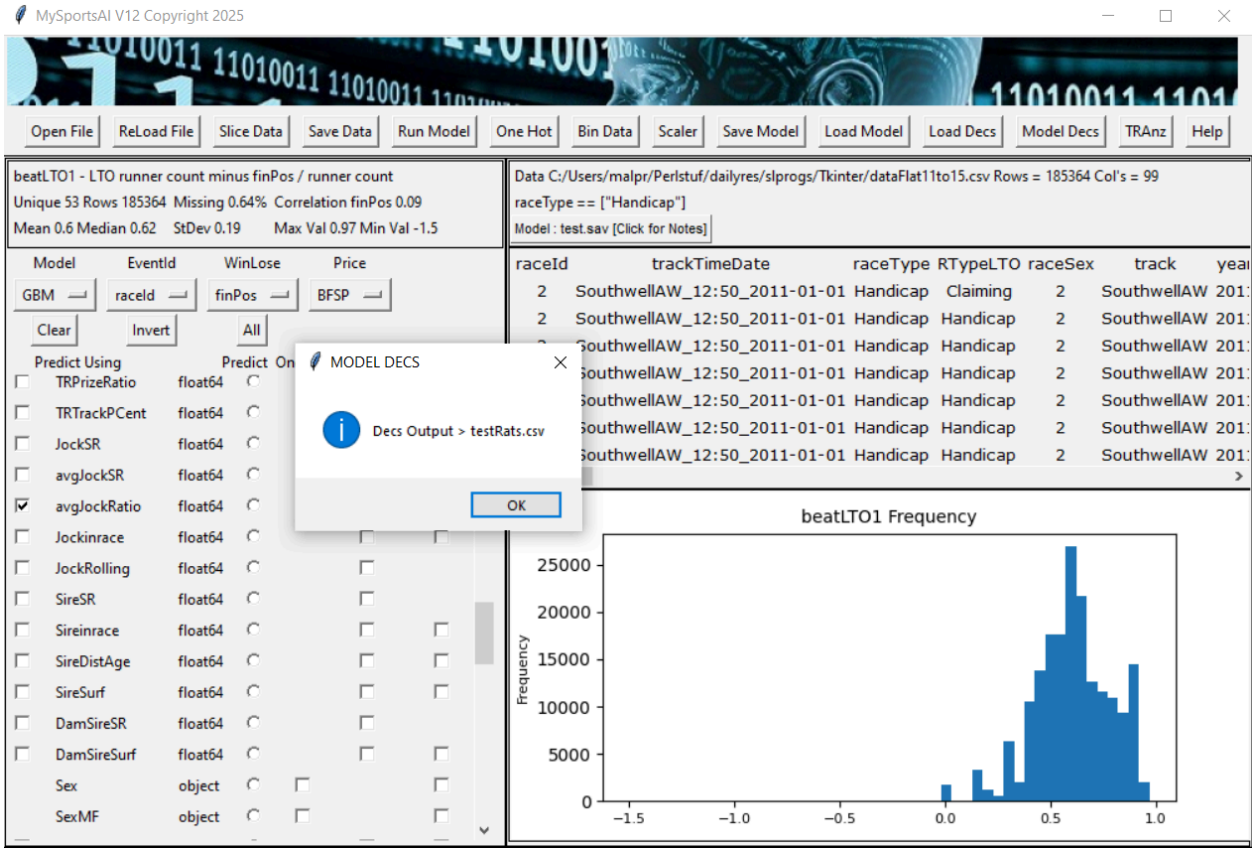

I would probably want to test this model on more future data files eg dataflat18to19.csv and beyond. For this it would be best if I used a file containing data from 2011 to 2017 to do the training. At some point if we are happy with the model we would want to save it so let's look at how to do that.

It is simply a case of going to the main window and clicking the Save Model button. You are then prompted for a file name and location. Once saved the file name will appear in

the main top left window section along with a link to allow you to enter notes into a file that will be associated with the model. This allows you to come back later and recall what stage you were at with its development, concerns you may have and further testing required.




Let us now look at how we can make predictions on today's runners. At the moment we have the model loaded but if we were coming into MySportsAI after closing down we would simply load the model up by clicking Load Model on the main window. The next thing we need to do is load up the data for today's runners by clicking Load Decs. Finally we now click Model Decs and we get the following response.




The predictions can now be found in the file testRats.csv in the same folder as our MySportsAI installation. Here is a snapshot of the first few lines of the output.

| TrackTime | Horse | Rating | BFPrice |
|---|---|---|---|
| | | | |
| Kempton_20:30_2025-03-05 | Oslo | 0.215082 | 4.65 |
| Kempton_20:30_2025-03-05 | Almosh'her | 0.16226 | 6.16 |
| Kempton_20:30_2025-03-05 | Arqoob (IRE) | 0.15021 | 6.66 |
| Kempton_20:30_2025-03-05 | Victory Shout (FR) | 0.148914 | 6.72 |
| Kempton_20:30_2025-03-05 | Buxted Too | 0.123469 | 8.1 |
| Kempton_20:30_2025-03-05 | Gaassee (IRE) | 0.112007 | 8.93 |
| Kempton_20:30_2025-03-05 | Haku (IRE) | 0.088058 | 11.36 |
| | | | |
| Kempton_20:00_2025-03-05 | Rebel Empire (FR) | 0.111331 | 8.98 |
| Kempton_20:00_2025-03-05 | Grenham Bay (IRE) | 0.10828 | 9.24 |
| Kempton_20:00_2025-03-05 | Nogo's Dream | 0.103655 | 9.65 |
| Kempton_20:00_2025-03-05 | Perennial | 0.096115 | 10.4 |
| Kempton_20:00_2025-03-05 | Supreme King (IRE) | 0.094469 | 10.59 |
| Kempton_20:00_2025-03-05 | Dashing Harry | 0.093312 | 10.72 |
| Kempton_20:00_2025-03-05 | Shallow | 0.09205 | 10.86 |
| Kempton_20:00_2025-03-05 | Abbey Heights | 0.089357 | 11.19 |
| Kempton_20:00_2025-03-05 | Four Adaay | 0.079376 | 12.6 |
| Kempton_20:00_2025-03-05 | Dark Dreamer | 0.072994 | 13.7 |
| Kempton_20:00_2025-03-05 | Ultramarine (IRE) | 0.05906 | 16.93 |
| | | | |

Oslo is top rated in the last race at Kempton.

# Hyper Parameter Tuning

The input features we select for our model will strongly determine its accuracy and profitability. There is however another set of features which can influence output. All Machine Learning algorithms have some under the hood parameters that dictate how the algorithm will operate. Finding the optimum setting for these parameters can improve model performance. Let us take an example using one of the simpler algorithms to understand, the K nearest neighbor algorithm.

Imagine we are creating a model based on trainer strike rate and sire strike rate. There is a good reason why putting these two together may not be a good choice but more of that later. Now suppose our first row in our test data consists of

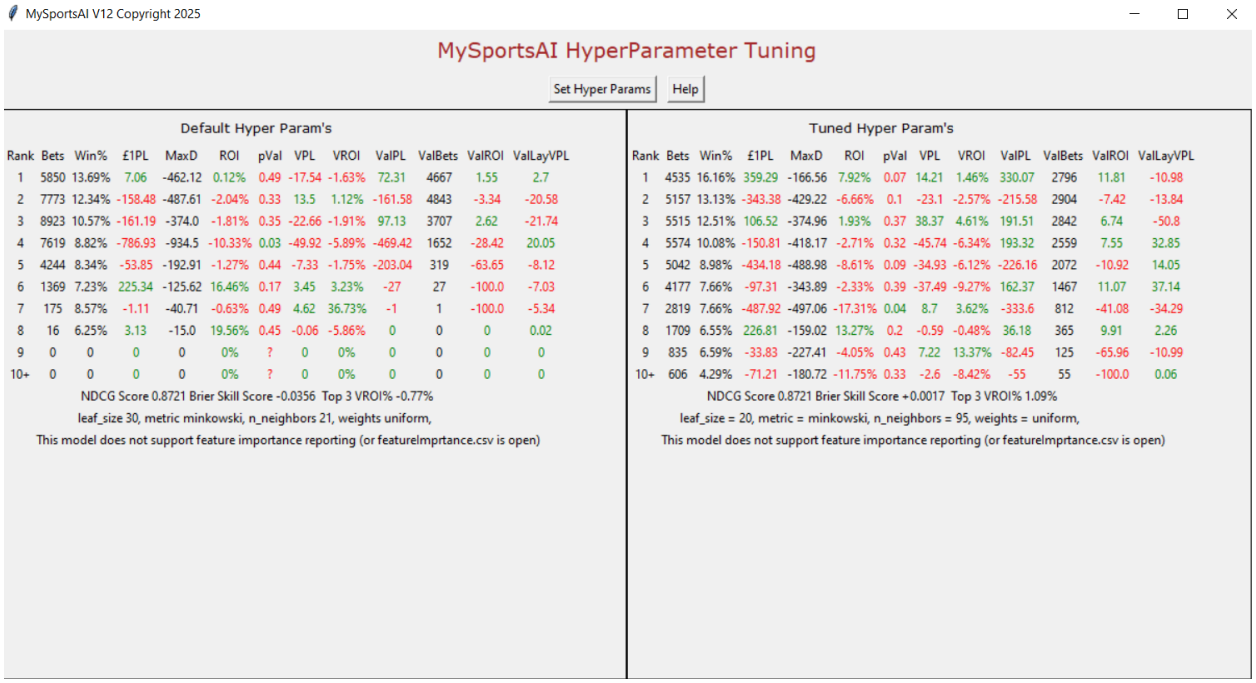
| TRSR | SireSR | finPos |
|---|---|---|
| 12.5 | 13.5 | 0 |

The algorithm will go looking for the best or should I say closest fit to those two numbers in the training data and it will look for the k instances or rows in the training data that best fit. If k=101 it will look for the closest 101 rows that mirror 12.5 and 13.5 and then it will create a percentage chance of winning based on those 101 rows of data from the training data. If there are say 10 other occurrences of 12.5 and 13.5 then clearly they will go into the 101 sample. But it will also look for next closest fits to complete the 101 sample. Now we can see the first obvious hyper parameter, what is the best choice for k. Too small and you do not get a representative number to base a percentage on. Too big and you may be drifting too far away from the 12.5 and 13.5. Tuning a KNN algorithm involves finding the best fit for k. Another hyper parameter is the method used to calculate how far away each row in the training data is from 12.5 and 13.5. There is more than one way to do this. The four main ones are Manhattan, Minkowski, Euclidean and Cosine similarity.

In MySportsAI you can fine tune these hyper parameters by clicking on the HyperParams button at the top of the train/test screen.



Tuning can take a while depending on the size of the file. Using a KNN algorithm we see some improvement in performance with the tuned parameters compared to the defaults.




Top rated horses have improved and the increase in neighbors considered from 21 to 95 has resulted in a greater spread of ratings. We could now click Set Hyper Params if we want to go ahead and use those before perhaps saving the model.

# More Train/Test Features

Let us now  briefly cover some of the other command features in the train/test window.

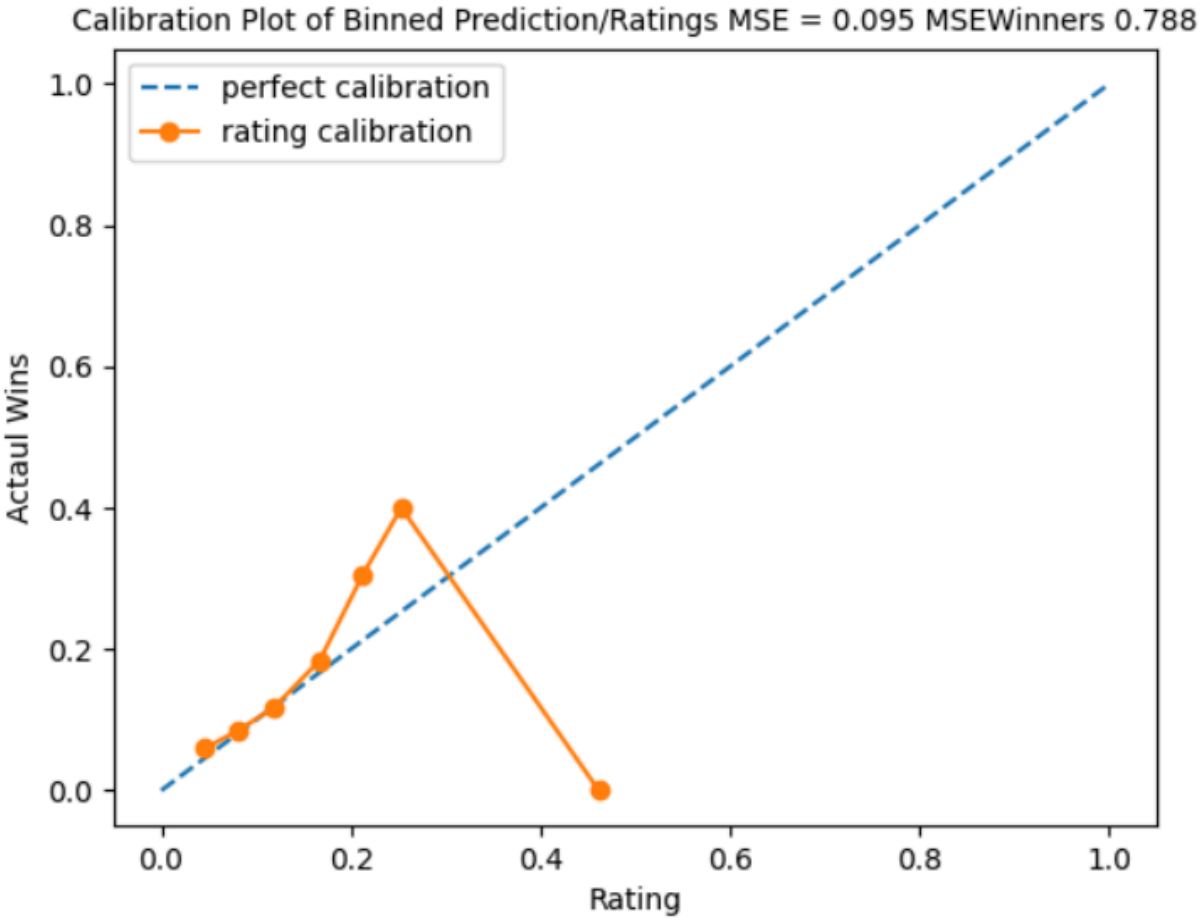



Save Results does what it says, it will create a csv file of all the rows in the test file along with the predictions for each row and the results. Here is a snapshot of the first few rows of the results file.

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Model test.sav GBM | Slice raceType == ["Handicap"] | | | | | | | | | | | |
| 2 | Input Features > | TRSR | avgJockRatio | | | | | | | | | | |
| 3 | HyperParams | ccp_alpha | criterion fr | init None | learning_r | loss deviar | max_deptl | max_featu | max_leaf_ | min_impur | min_samp | min_samp | min_weigh |
| 4 | | | | | | | | | | | | | |
| 5 | raceId | horse | finPos | BFSP | rating | finPosFull | Normalise | HHindex | BFSPStDev | ratStDev | PL | BFSPRank | ratRank |
| 6 | 25119 | Urban Cas | 1 | 3.8 | 0.172002 | 1 | 0.172002 | 813.5017 | 326.4868 | 0.031547 | 2.744 | 2 | 1 |
| 7 | 25119 | Bathwick S | 0 | 9.4 | 0.129428 | 3 | 0.129428 | 813.5017 | 326.4868 | 0.031547 | -1 | 3 | 2 |
| 8 | 25119 | Nolecce | 0 | 37.22 | 0.121073 | 7 | 0.121073 | 813.5017 | 326.4868 | 0.031547 | -1 | 7 | 3 |
| 9 | 25119 | Dancing Pr | 0 | 180 | 0.120663 | 6 | 0.120663 | 813.5017 | 326.4868 | 0.031547 | -1 | 8 | 4 |
| 10 | 25119 | Archipeligo | 0 | 2.51 | 0.117914 | 5 | 0.117914 | 813.5017 | 326.4868 | 0.031547 | -1 | 1 | 5 |
| 11 | 25119 | Lineman | 0 | 19.5 | 0.100889 | 2 | 0.100889 | 813.5017 | 326.4868 | 0.031547 | -1 | 6 | 6 |
| 12 | 25119 | Bushel (US | 0 | 10.55 | 0.097193 | 8 | 0.097193 | 813.5017 | 326.4868 | 0.031547 | -1 | 4 | 7 |
| 13 | 25119 | Celestial B | 0 | 16.55 | 0.076403 | 4 | 0.076403 | 813.5017 | 326.4868 | 0.031547 | -1 | 5 | 8 |
| 14 | 25119 | Bogey Hol | 0 | 1000 | 0.064435 | 9 | 0.064435 | 813.5017 | 326.4868 | 0.031547 | -1 | 9 | 9 |
| 15 | 25120 | Bangers (IF | 0 | 8.67 | 0.136971 | 9 | 0.136971 | 599.411 | 28.02869 | 0.028617 | -1 | 4 | 1 |
| 16 | 25120 | Happy Pur | 0 | 5.65 | 0.126408 | 5 | 0.126408 | 599.411 | 28.02869 | 0.028617 | -1 | 3 | 2 |

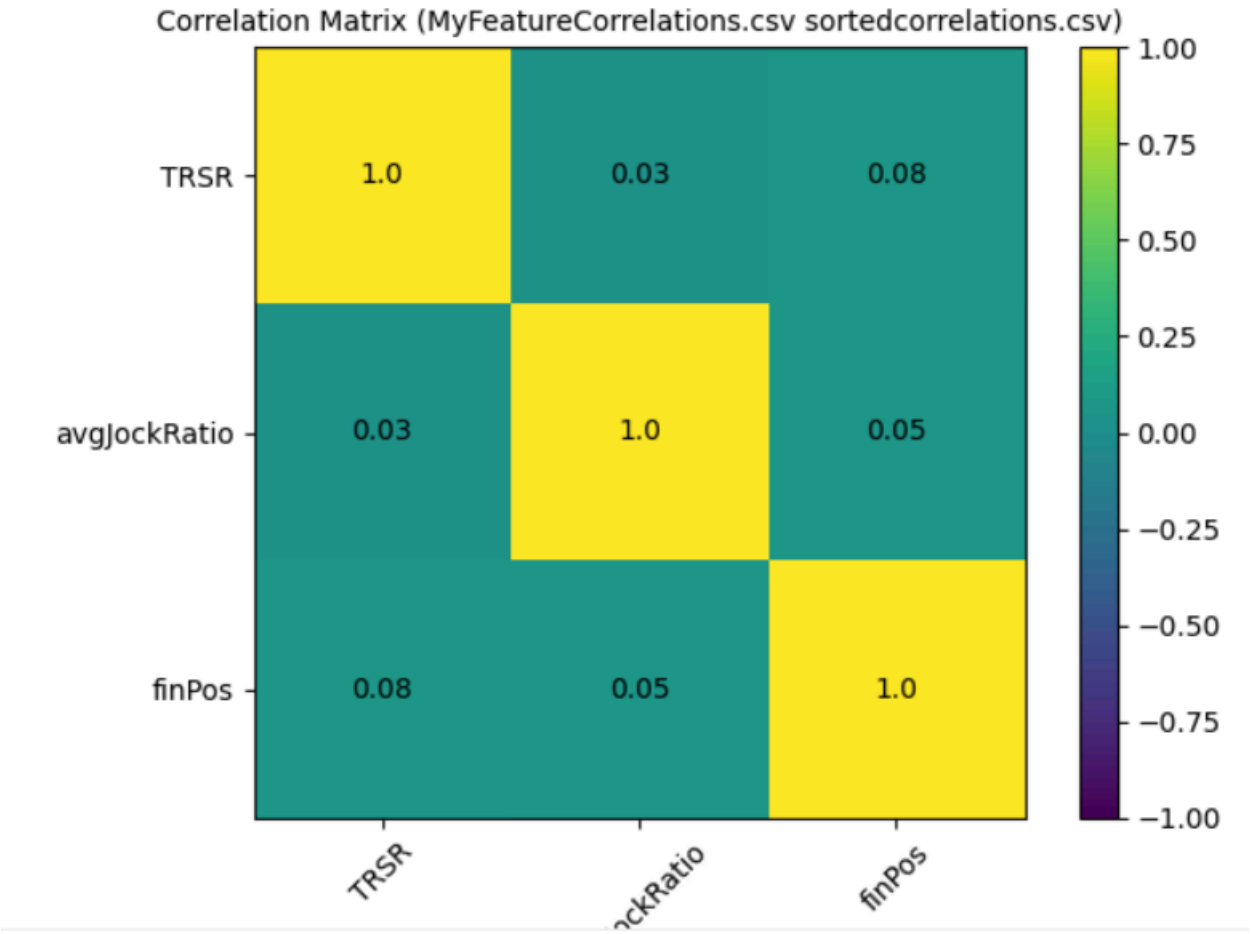
The first 3 rows contain information about the model, all the following rows show the ratings and the results of each horse in each race. We can see that the top ranked horse Urban Cas' won the first race.

Clicking CalibPlot will give you a calibration plot for the probabilities (ratings) produced for the test file.



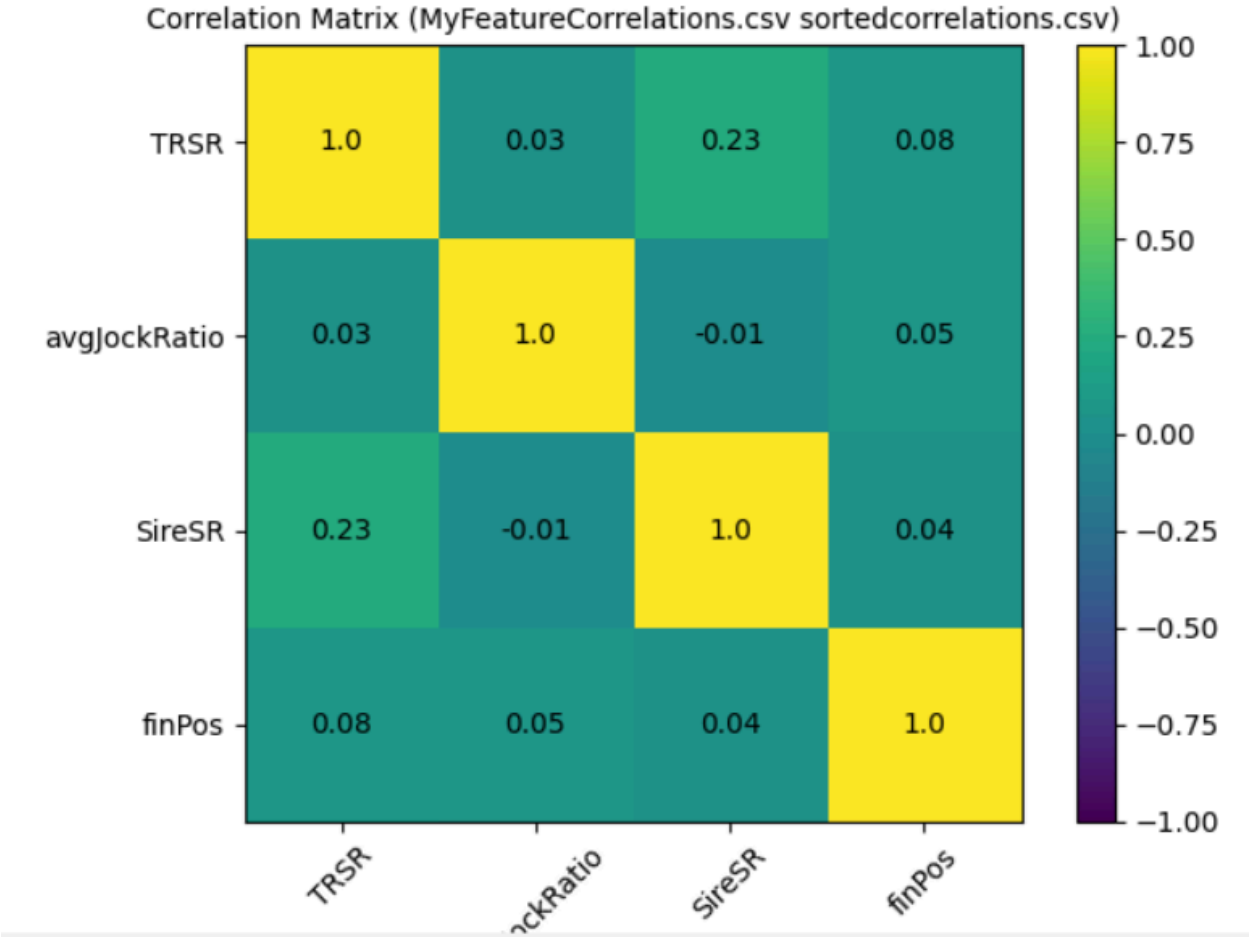

The closer the orange line is to the blue dotted line the more accurate the probabilities were to the actual winning chance. We can see that they did quite well up about 0.18

Clicking the Pearson button will show us the correlation between the features. We have mentioned this earlier in the book. This diagram however will show us the correlation between input features along with input features and the target feature finPos.



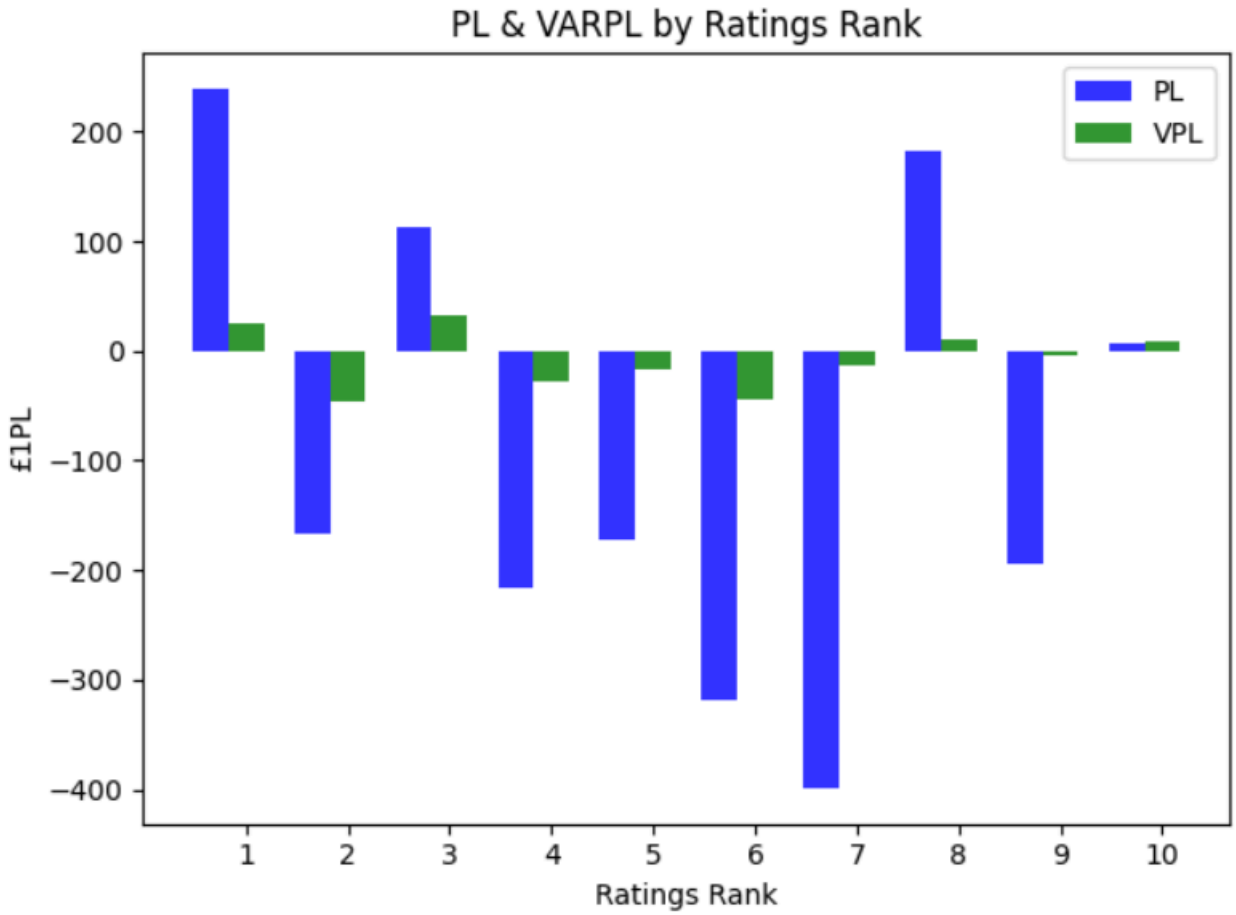

Reading across and then down we can see that TRSR has a correlation with finPos of 0.08, in other words as TRSR go's up so does the chance of winning a race. The avgJockRatio also has a positive correlation but not as strong at 0.05. We can also see that the two input features have a positive correlation with each other at 0.03.

We should be mindful of features that have a very strong correlation with each other. Take for example the following.




I have added SireSR (sire strike rate), to illustrate that better trainers have better bred animals in their stables and this seems to be the case with a much larger correlation of 0.23 between TRSR and SireSR. Including both of these in a model may amount to double counting or over fitting on a particular feature.

The PlotPL button will give you a visual snapshot of the profit and loss by rating ranking.




We can see that the PL generally deteriorates with an increase in rating ranking.

IterRun if clicked will train/test a model using all the input features you have selected, it will then remove the least predictive feature and re train/test and so on. On finishing it will report back to you which was the best model. ComboRun will train/test all combinations of your input features, so for 3 input features it would train/test all 3 and then the 3 combinations of 2 input features selected from the 3. You can throw as many input features as you like at ComboRun but obviously the time taken to complete gets higher as you increase the number of features. Once again it will report back on the best combination.

The Decile button will show you the performance of the ratings on the test file by grouping the rating values into deciles, here is a sample for our saved model.

**Results By Rating Deciles**

| Decile | Bets | PL | ROI% | VPL | ValPL | WIN% | Max & Min Rating |
|---|---|---|---|---|---|---|---|
| 1 | 3597.0 | 141.3 | 3.93 | 31.8 | 161.17 | 22.38 | [0.555 0.1653] |
| 2 | 3596.0 | 7.93 | 0.22 | -22.31 | 65.3 | 14.91 | [0.1653 0.1381] |
| 3 | 3598.0 | -75.92 | -2.11 | -22.97 | 22.52 | 12.7 | [0.1381 0.1211] |
| 4 | 3597.0 | 18.79 | 0.52 | -10.83 | 81.14 | 11.31 | [0.1211 0.1089] |
| 5 | 3596.0 | -287.09 | -7.98 | -10.94 | -219.26 | 10.26 | [0.1089 0.0985] |
| 6 | 3597.0 | -170.81 | -4.75 | -32.96 | 176.53 | 8.56 | [0.0985 0.0888] |
| 7 | 3597.0 | 73.16 | 2.03 | 12.62 | 81.31 | 8.73 | [0.0888 0.0797] |
| 8 | 3597.0 | -262.83 | -7.31 | -8.07 | -216.76 | 7.31 | [0.0797 0.0701] |
| 9 | 3597.0 | -461.94 | -12.84 | -26.63 | -130.18 | 5.62 | [0.0701 0.0583] |
| 10 | 3597.0 | 91.38 | 2.54 | 14.35 | 5.89 | 5.09 | [0.0583 0.017 ] |

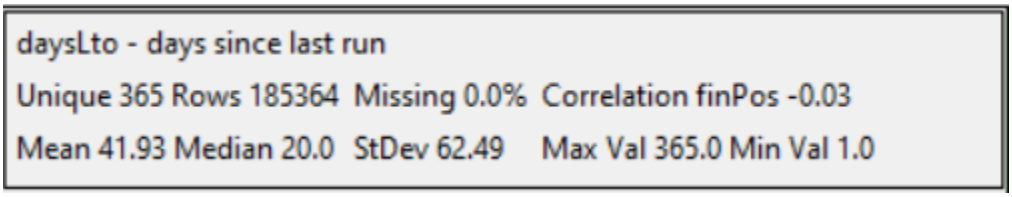
We can see that ratings between 0.55 and 0.1653 did very well. This information can help to highlight where a model is working well at a more granular level than simply looking at top ranked, second top etc.

Finally the Hist button allows you to look at the results summary of each model you have train/tested. This is useful when you have trained a model and then changed it and retrained it. Rather than try and remember how the first model performed you can go to Hist and look back at the previous outputs of the model.

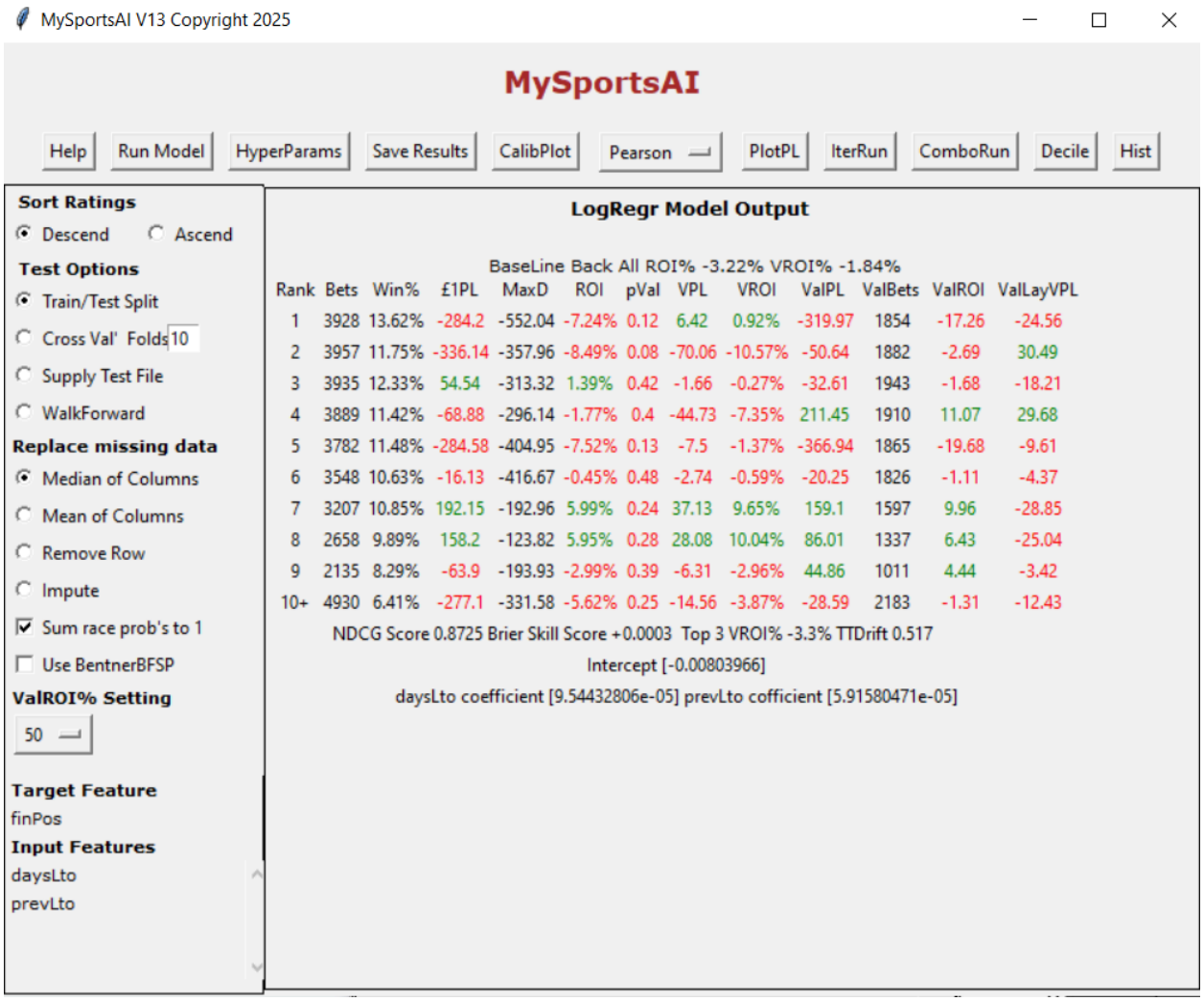
# Outliers and EDA

In Malcolm Galdwell's best selling book Outliers his examples of out of the norm people or behaviour are connected with success. The Beatles trip to Hamburg as young inexperienced musicians resulted in a period when they performed twice a day for long hours for many months. The result was a fine honing of unrealised talent that UK bands were not experiencing. They came back as far better musicians, certainly far better than they would have been if they had spent the same time in the UK playing two or three gigs a week. Outliers in Machine Learning however are often something to be more cautious of depending on the algorithm you choose to use to create your model.

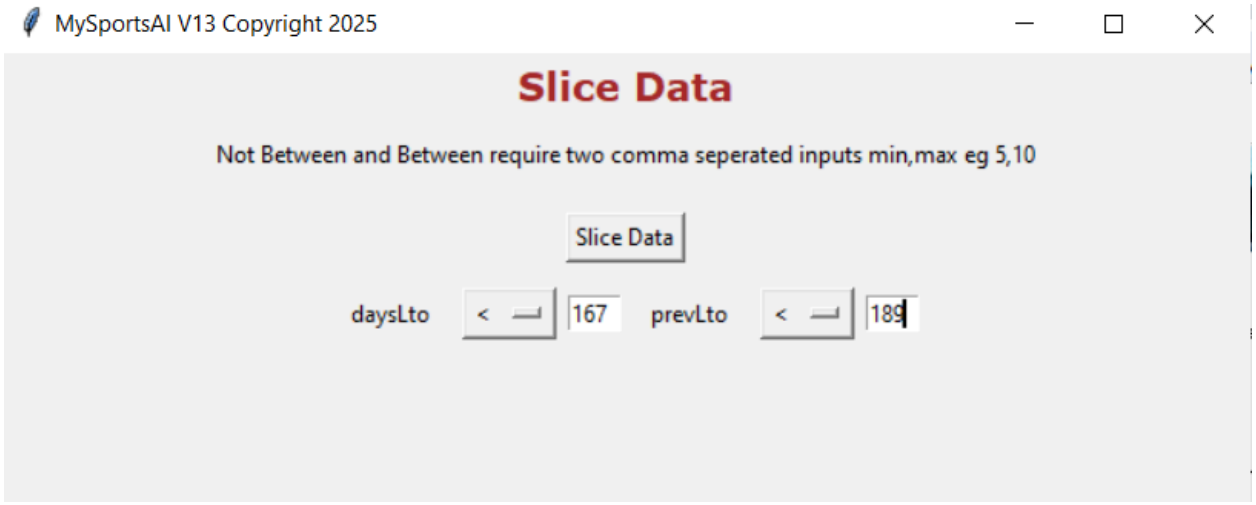
Outliers are values in our data that lie outside the normal range of values. They are of concern to us because algorithms like KNN and Logistic Regression will not handle them as well as other algorithms such as Tree based algorithms GBM and XGBoost.. To illustrate this we will take a look at Logistic Regression and two input features daysLto and prevLto. Here is the data information for Handicaps 2011 to 2015.



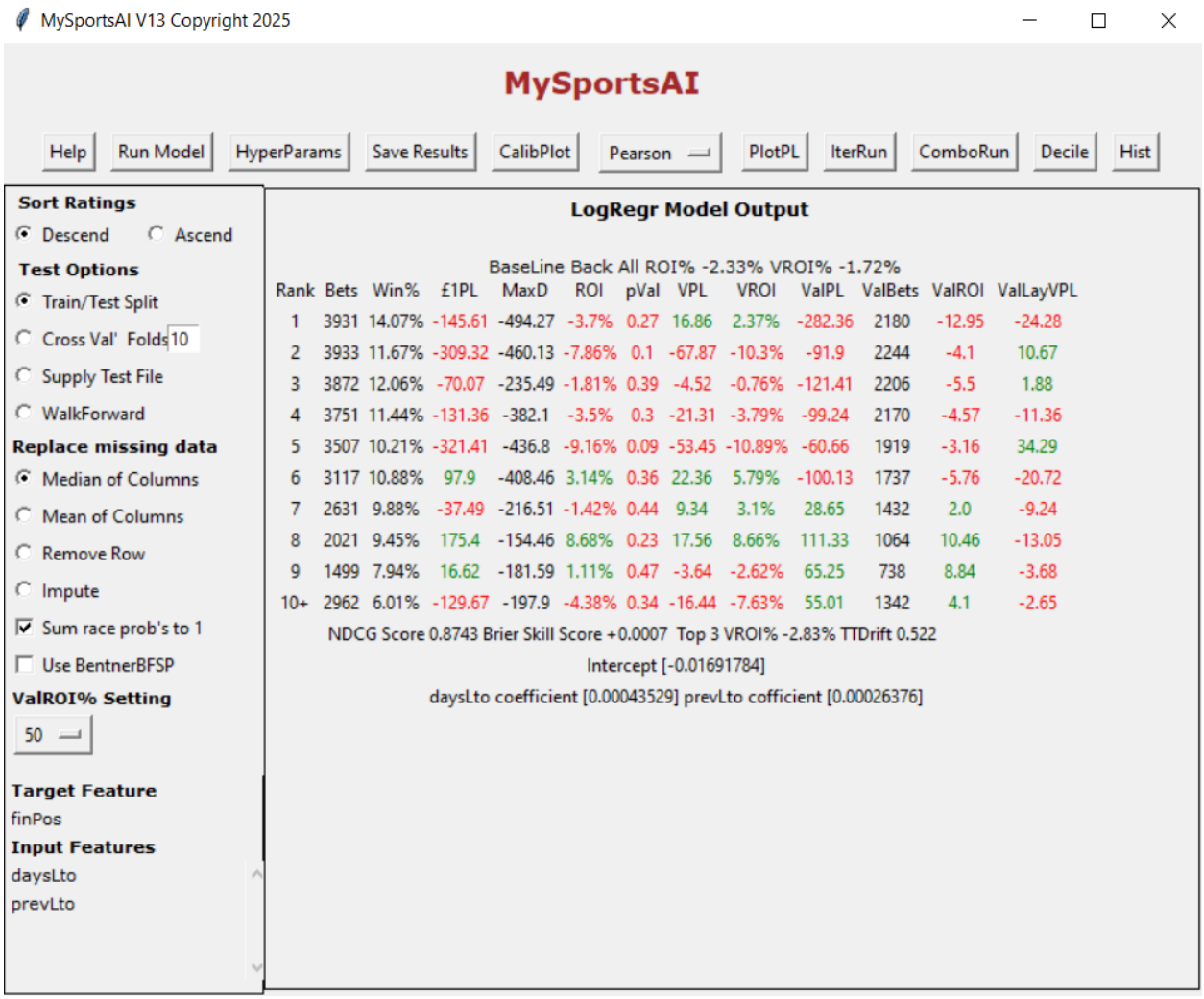

The question is which of these values could be considered as outliers. Those above 250, above 200 ? A rule of thumb is that anything more than 2 x the standard deviations from the mean is an outlier. In that case anything above 166.91 would be an outlier for daysLto. Before we act on this let us first create a Logistic Regression model on the data as it is using daysLto and prevLto, here are the results.




Not particularly encouraging, the top rated made a small variable profit but a big level stake loss. Now let us remove the outliers for both daysLto and prevLto by slicing the data.




Now if we train/test the model again on the sliced data we get more encouraging results from the Logistic Regression algorithm.




The performance of the top rated has improved as has the Brier Skill Score.

Now contrast that with the GBM algorithm for the data without outliers removed and then with.

**GBM Model**

raceType == ["Handicap"]

| Rank | Bets | Win% | £1PL | MaxD | ROI | pVal | VPL | VROI | ValPL | ValBets | ValROI | ValLayVPL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4245 | 14.98% | 282.73 | -267.89 | 6.66 | 0.14 | 45.16 | 5.58 | 81.72 | 2155 | 3.79 | -58.72 |
| 2 | 4411 | 11.74% | -550.03 | -707.27 | -12.47 | 0.02 | -66.43 | -9.19 | -430.1 | 2186 | -19.68 | 45.43 |
| 3 | 4427 | 11.43% | -323.57 | -428.42 | -7.31 | 0.11 | -30.55 | -4.58 | -194.09 | 2173 | -8.93 | 13.23 |
| 4 | 4276 | 11.32% | -152.55 | -303.43 | -3.57 | 0.29 | -28.02 | -4.37 | -16.13 | 2025 | -0.8 | -5.85 |
| 5 | 4084 | 11.85% | 184.69 | -274.51 | 4.52 | 0.26 | 24.87 | 4.37 | 55.76 | 1915 | 2.91 | -12.78 |
| 6 | 3726 | 10.14% | 102.65 | -363.91 | 2.75 | 0.36 | -5.26 | -1.15 | 105.68 | 1824 | 5.79 | 5.4 |
| 7 | 3125 | 9.5% | -111.78 | -293.38 | -3.58 | 0.34 | 10.65 | 3.13 | -47.99 | 1495 | -3.21 | -29.46 |
| 8 | 2504 | 8.51% | 136.45 | -192.16 | 5.45 | 0.31 | -0.18 | -0.08 | 116.28 | 1181 | 9.85 | -8.67 |
| 9 | 1890 | 6.88% | -186.58 | -367.04 | -9.87 | 0.2 | -17.18 | -10.26 | -14.75 | 913 | -1.62 | -0.11 |
| 10 | 3281 | 6.03% | -308.02 | -430.13 | -9.39 | 0.18 | -9.01 | -3.85 | -179.78 | 1454 | -12.36 | -6.49 |

NDCG Score 0.874 Brier Skill Score +0.0023 Top 3 VROI% -2.36% TTDrift 0.53

**GBM Model**

raceType == ["Handicap"] and daysLto<167 & prevLto<189

| Rank | Bets | Win% | £1PL | MaxD | ROI | pVal | VPL | VROI | ValPL | ValBets | ValROI | ValLayVPL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4145 | 15.01% | 259.24 | -213.59 | 6.25 | 0.16 | 44.51 | 5.64 | 110.74 | 2450 | 4.52 | -60.72 |
| 2 | 4289 | 11.75% | -242.49 | -484.34 | -5.65 | 0.18 | -60.82 | -8.75 | -133.54 | 2489 | -5.37 | 51.74 |
| 3 | 4236 | 10.91% | -587.25 | -689.05 | -13.86 | 0.01 | -51.29 | -8.13 | -457.1 | 2398 | -19.06 | 2.29 |
| 4 | 3990 | 10.93% | -241.02 | -376.45 | -6.04 | 0.18 | -31.47 | -5.48 | -92.61 | 2162 | -4.28 | 8.48 |
| 5 | 3642 | 11.29% | 94.02 | -422.77 | 2.58 | 0.36 | 14.59 | 2.96 | 69.9 | 1951 | 3.58 | -24.99 |
| 6 | 3142 | 10.22% | 72.06 | -130.16 | 2.29 | 0.39 | 0.66 | 0.17 | 125.54 | 1670 | 7.52 | -0.2 |
| 7 | 2525 | 9.43% | 74.94 | -350.62 | 2.97 | 0.38 | 11.33 | 4.26 | 94.65 | 1330 | 7.12 | -13.62 |
| 8 | 1850 | 8.54% | 72.65 | -214.44 | 3.93 | 0.38 | 6.13 | 3.53 | 95.1 | 945 | 10.06 | -12.81 |
| 9 | 1285 | 6.54% | -257.54 | -349.15 | -20.04 | 0.06 | -19.24 | -16.79 | -133.45 | 615 | -21.7 | 10.09 |
| 10 | 2120 | 5.52% | -99.6 | -309.19 | -4.7 | 0.36 | -15.52 | -10.57 | 92.29 | 966 | 9.55 | 2.37 |

NDCG Score 0.8769 Brier Skill Score +0.0023 Top 3 VROI% -3.2% TTDrift 0.532

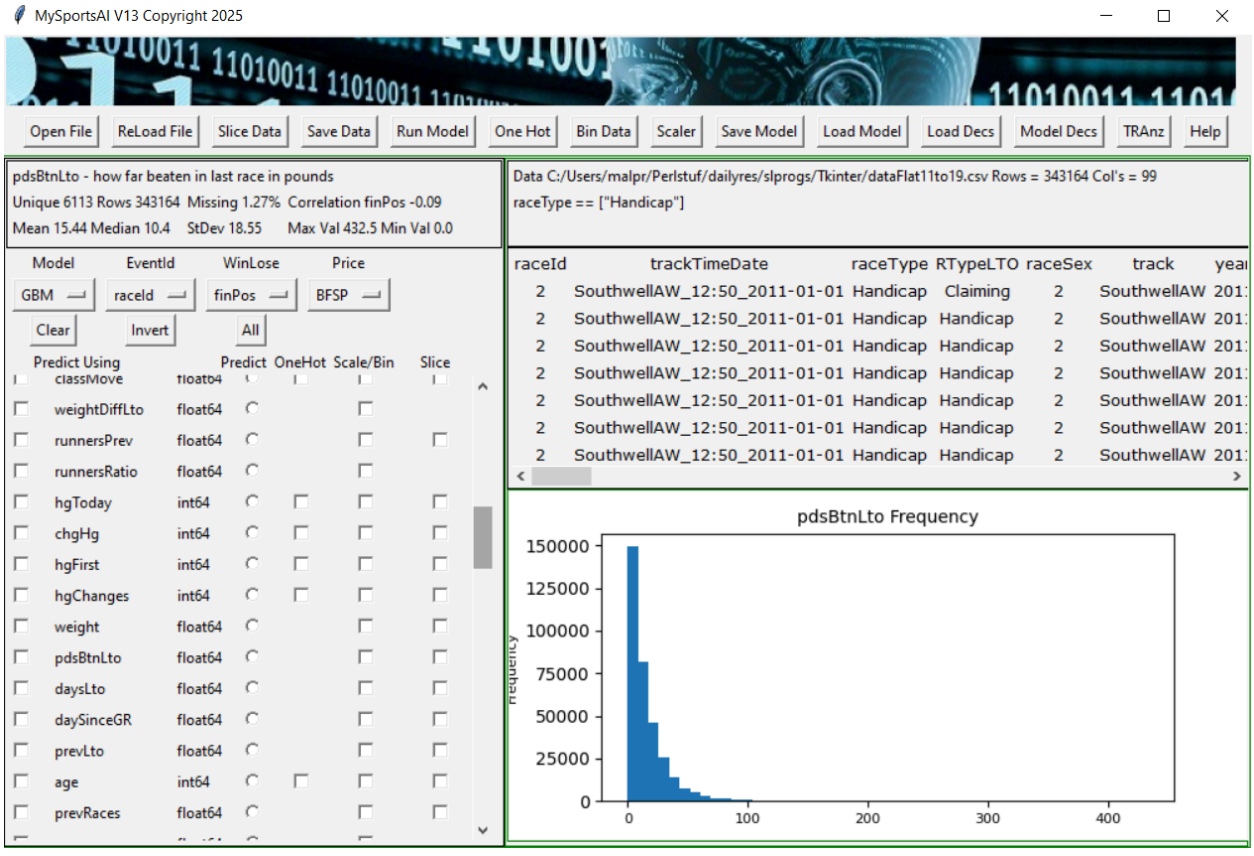
We do not see the same kind of improvement for treating outliers as we did when using Logistic Regression.

The process of identifying outliers forms part of a stage in model development known as Exploratory Data Analysis (EDA). Purists may well be upset that I have not touched on

this subject first as it certainly should be the first port of call when developing a model. You need to get to know your data in order to get the best out of any possible model. One advantage we have here is that you are almost certainly knowledgeable in your sport of choice. Here is an example I blogged about which qualifies as EDA but was born out of my understanding of horse racing.

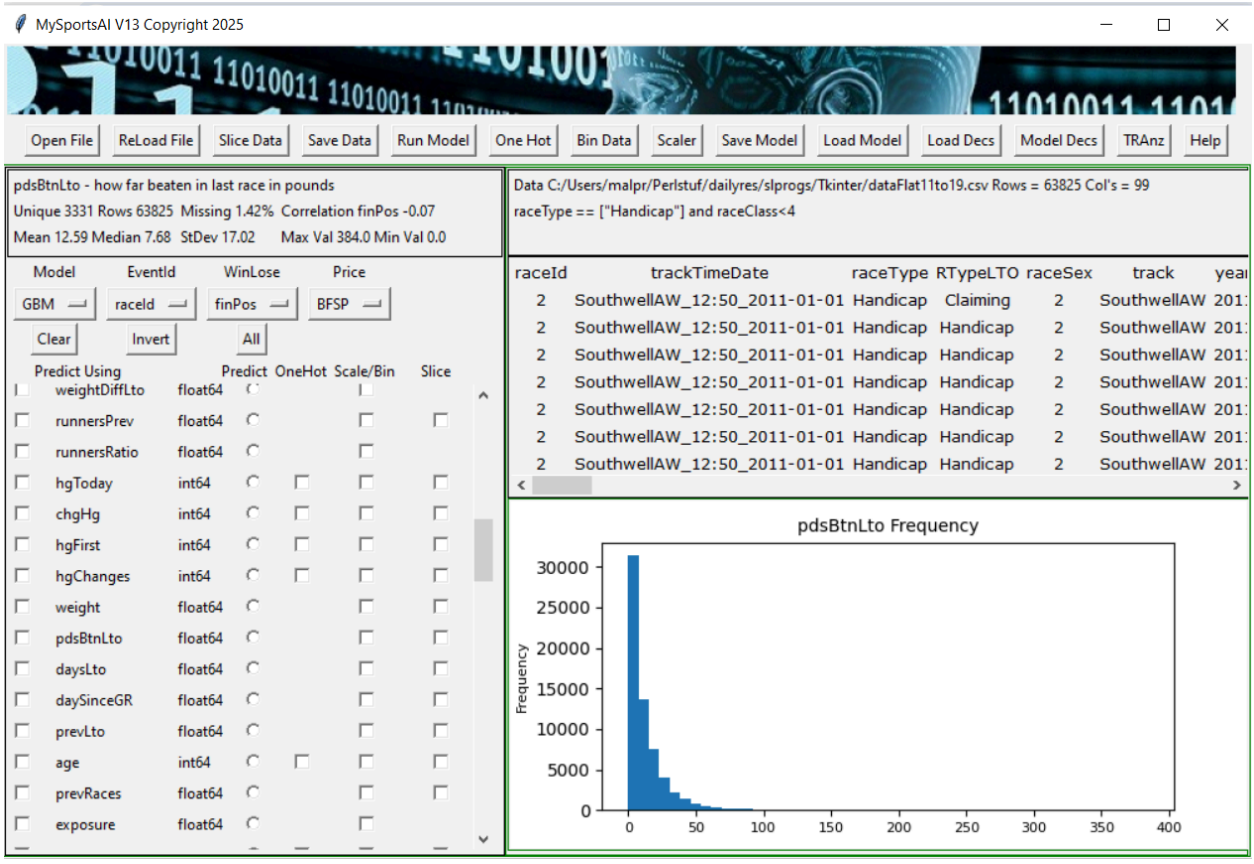
Yesterday saw the William Haggas trained Godwinson race to victory in the Lincoln, first big handicap of the season. It was tipped up by the Wisdom of Models model. I want to draw your attention to recent form and Handicap class. Clearly Haggas mapped out this race for Godwinson as on his previous run he finished half way down the field beaten 12 lengths. In fact his penultimate run was a similar effort. Coupled with the fact that he had not run for 161 days, which I admit is not unusual for a Lincoln runner, one could not say he was in form. Contrast that with the favourite who on his previous run had come 2nd beaten a length.

The question is does being in form matter more or less in higher grade handicap races ? We can answer this using MySportsAI. A look at all handicaps shows the following.

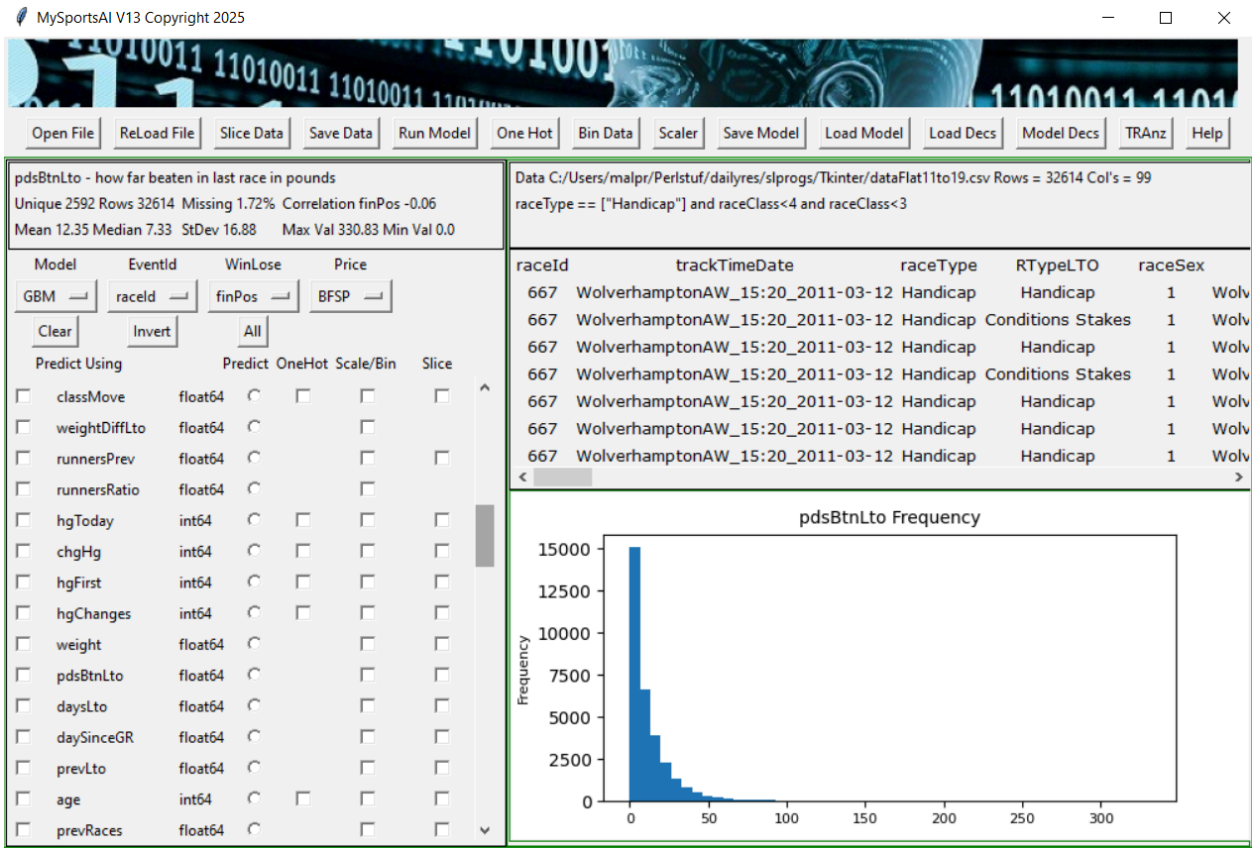


By placing the cursor over pdsBtnLto (pounds beaten last time out) we can see in the top left corner that the correlation with finPos (winning) is -0.09

This means that as pdsBtnLto increases the likelihood of winning decreases.
How does this compare with higher class handicaps ? Let us take a look at class 3 and below. To do this I simply slice the data using the slice data button to reduce the data to only races below class 4.

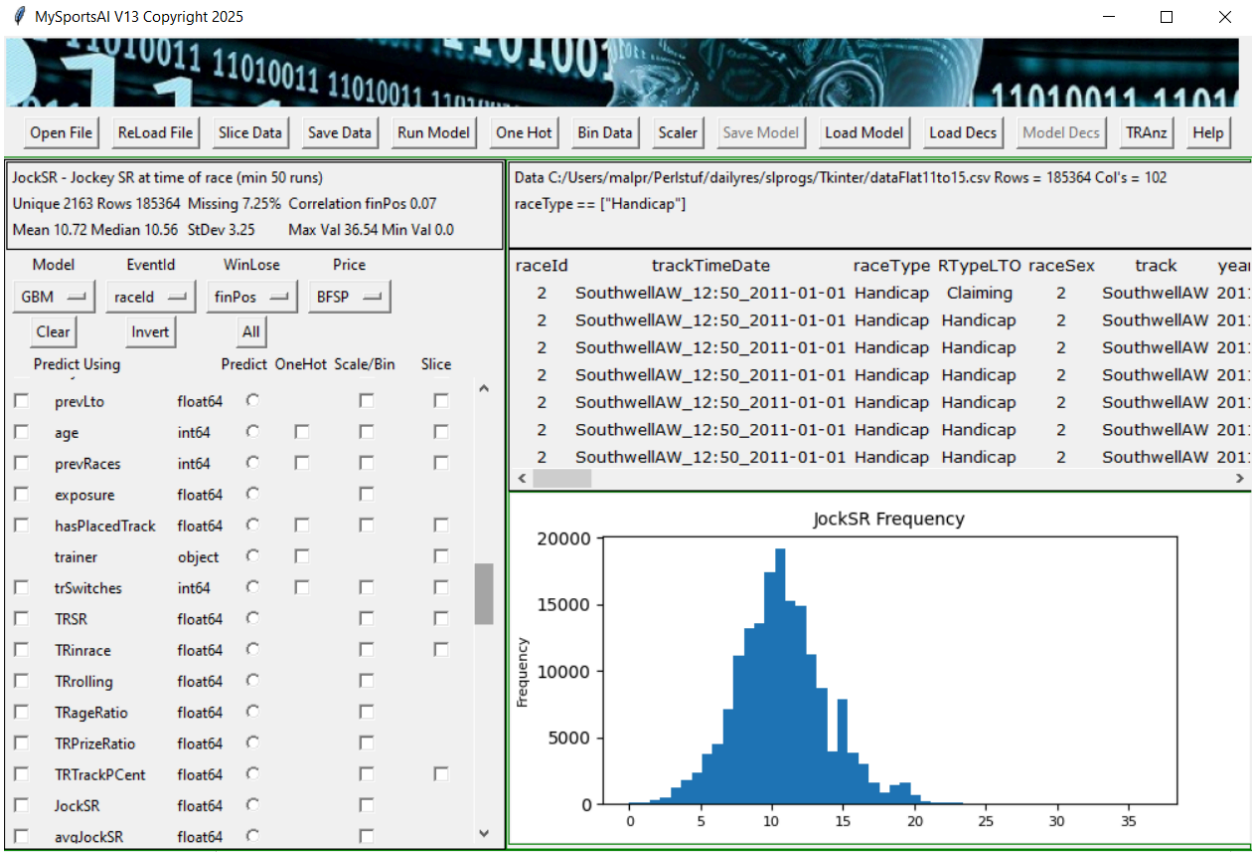



We can see that the correlation has dropped to -0.07, which of course means in races of class 4 and above the correlation has strengthened. This confirms that pdsBtnLto is less relevant in class 3 or less handicaps. In fact if we slice the data to class 2 and below the correlation reduces further to -0.06

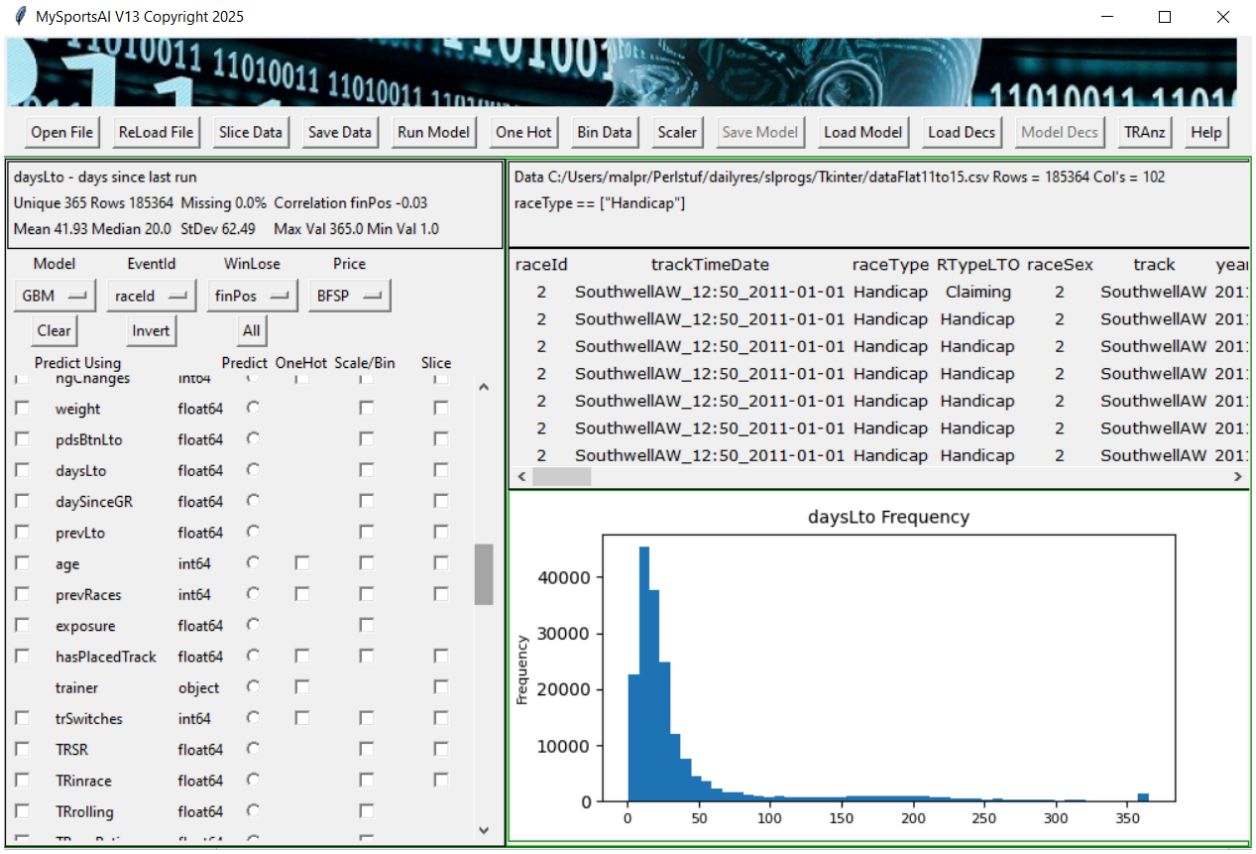



The conclusion seems to be that in higher grade races do not be afraid to get creative.The winner is less likely to jump out at you from recent form. It will come as no surprise therefore if I admit that pdsBtnLto is not a feature in my heritage handicap model.

Another important part of EDA is getting to know the distribution of the data you intend to use as input features. Looking at the screenshot below we can see that JockSR (Jockey strike rate) is evenly distributed about the mean. We call this a normal distribution.

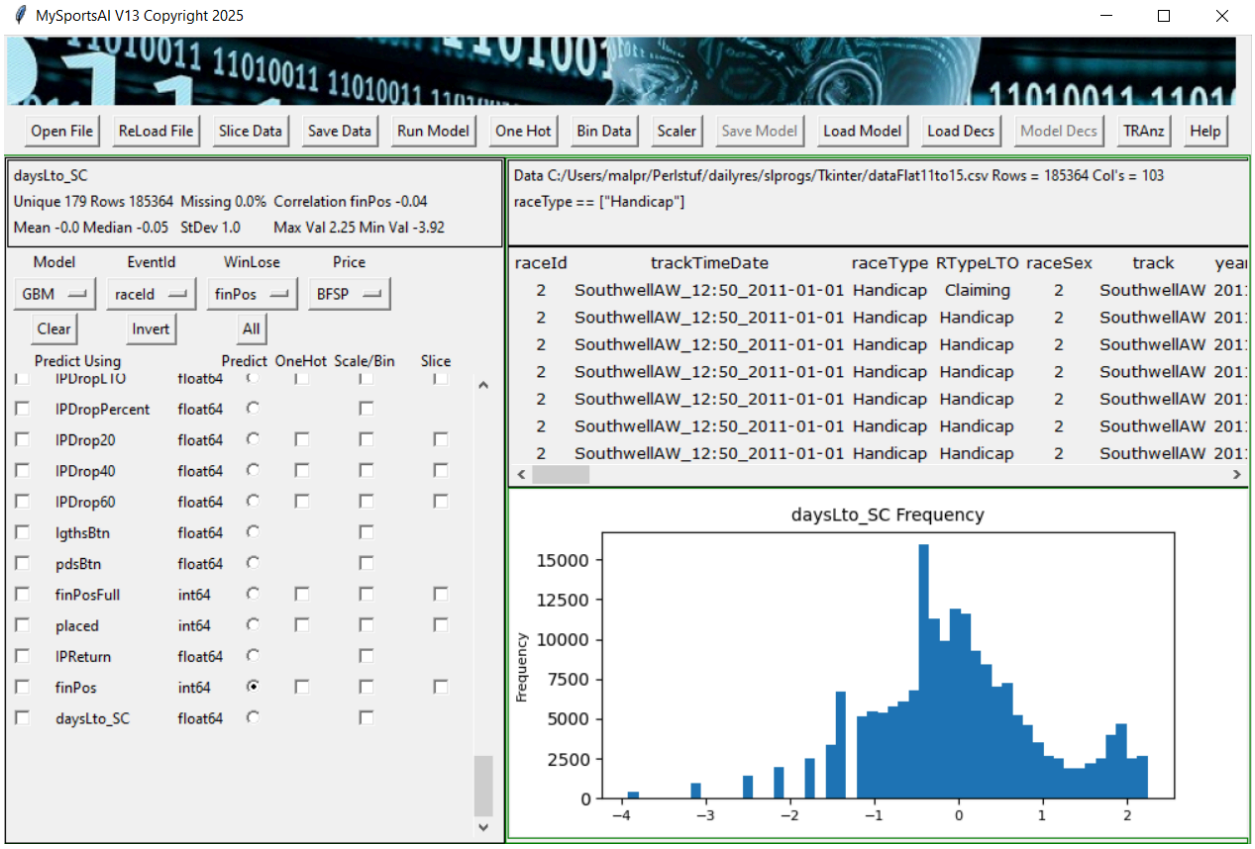



By contrast take a look at daysLto (days since last run)



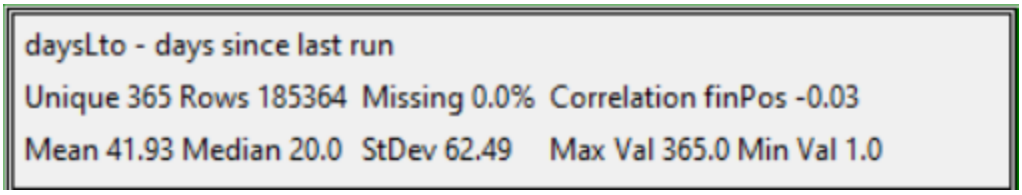

This has a right hand skewed tail. The reason for examining these distributions is that some algorithms handle skewed data better than others whilst algorithms like Linear Regression, Logistic Regression and Naive Bayes work better if the data is normally distributed. We will take a look at what we can do about this in the chapter on scaling.

Using the Scaler button we can transform the a data item such as daysLto so that the transformed data has more normal distribution


MySportsAI V13 Copyright 2025
Open File
ReLoad File
Slice Data
Save Data
Run Model
One Hot
Bin Data
Scaler
Save Model
Load Model
Load Decs
Model Decs
TRAnz
Help
daysLto_SC
Unique 179 Rows 185364 Missing 0.0% Correlation finPos -0.04
Mean -0.0 Median -0.05 StDev 1.0 Max Val 2.25 Min Val -3.92
Data C:/Users/malpr/Perlstuf/dailyres/slprogs/Tkinter/dataFlat11to15.csv Rows = 185364 Col's = 103
raceType == ["Handicap"]
Model
EventId
WinLose
Price
GBM
raceId
finPos
BFSP
Clear
Invert
All
Predict Using
Predict
OneHot
Scale/Bin
Slice
IPDropPercent float64
IPDrop20 float64
IPDrop40 float64
IPDrop60 float64
IgthsBtn float64
pdsBtn float64
finPosFull int64
placed int64
IPReturn float64
finPos int64
daysLto_SC float64
raceId trackTimeDate raceType RTypeLTO raceSex track
2 SouthwellAW_12:50_2011-01-01 Handicap Claiming 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
daysLto_SC Frequency
Frequency
15000
12500
10000
7500
5000
2500
0
−4
−3
−2
−1
0
1
2

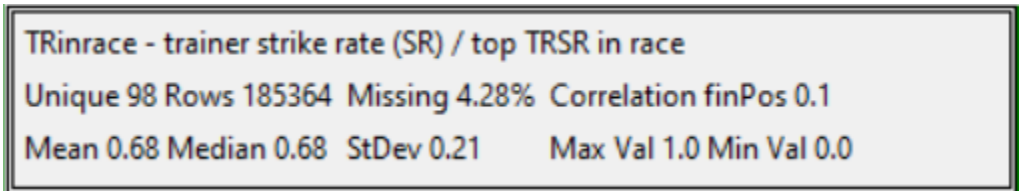
# Feature Scaling, Binning and One Hot Encoding

Looking back at the main window we can see earlier that placing the cursor over a feature name will give us a breakdown in the top left corner of the window and a graph of the distribution of the feature. Take daysLto or days since last run.




We can see that this value can be anything from 1 to 365 (above 365 is capped at 365).

Contrast that with TRinrace or Trainer Strike Rate within each race

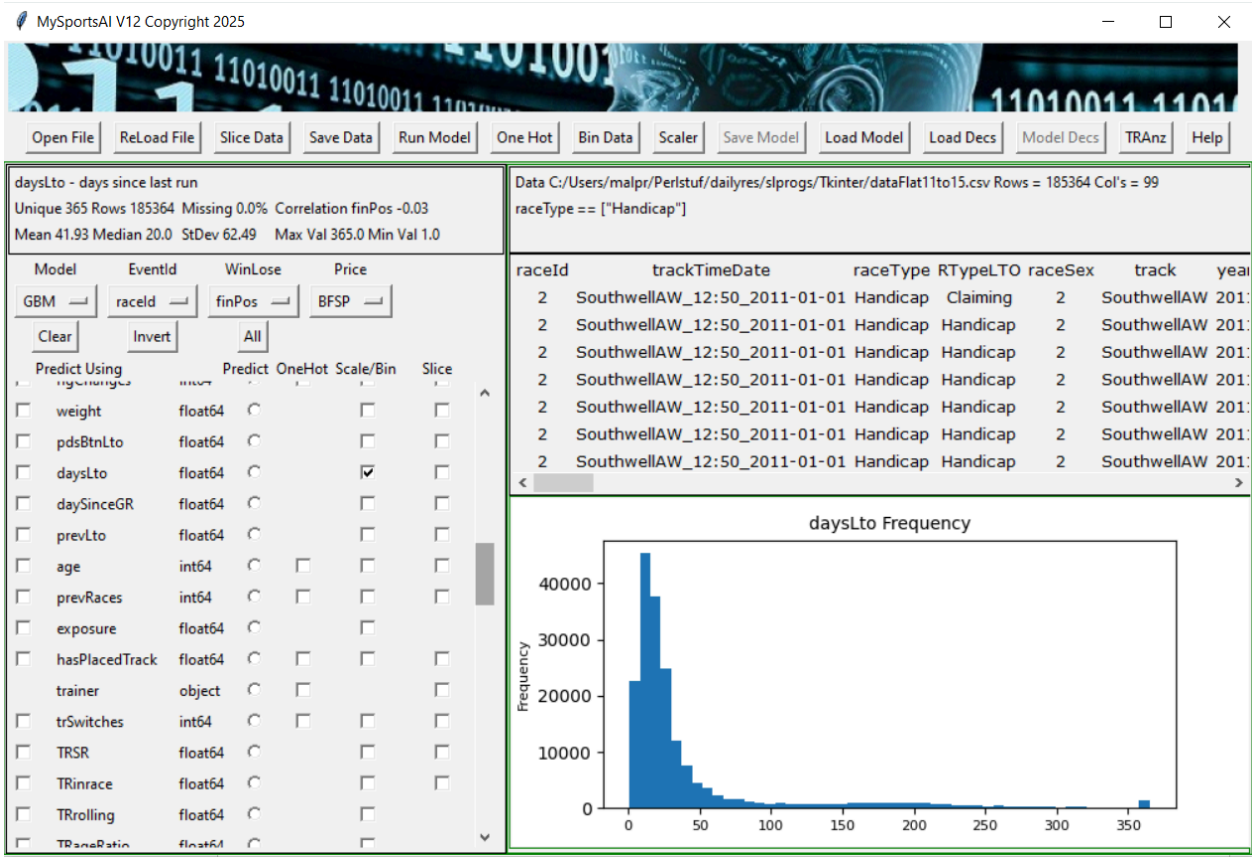



This value has a very different range, a minimum of 0 and a maximum of 1 as you would expect if you divide a trainer's strike rate by the top trainer strike rate in the race.

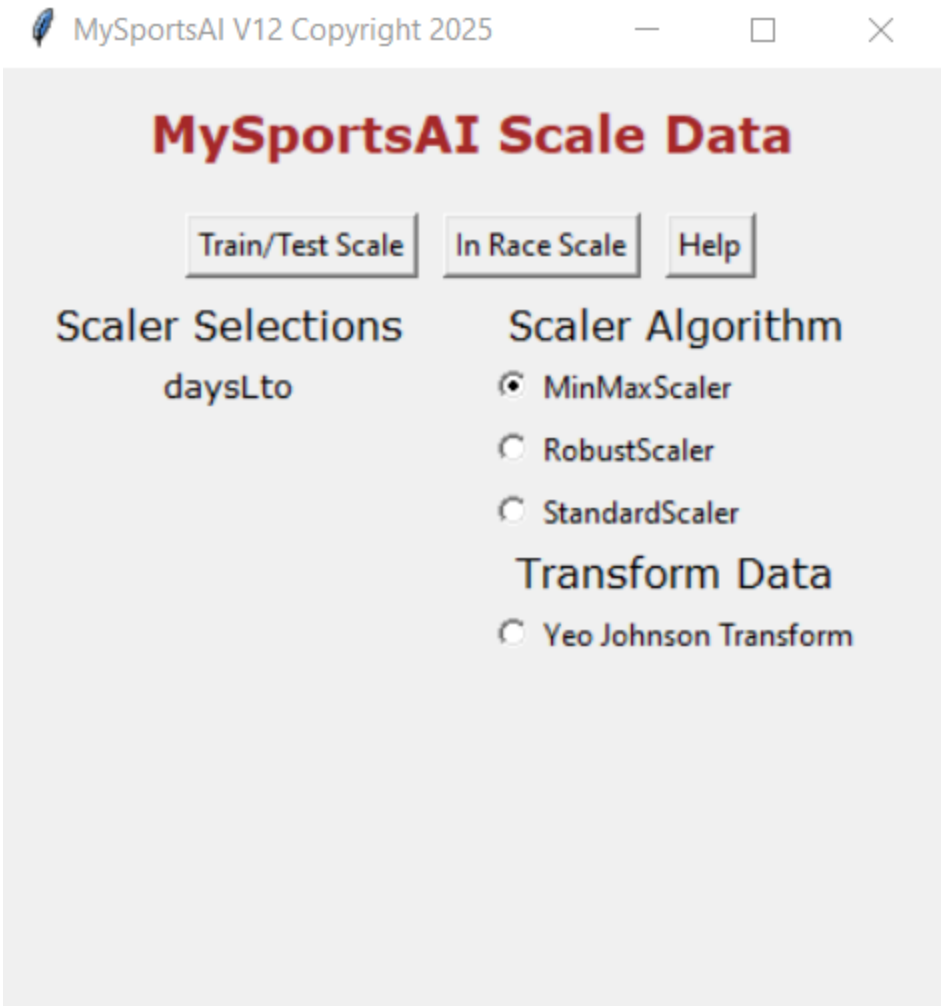
These very two different distributions can cause problems for some algorithms and less so for others. Tree based algorithms like GBM and XGBoost cope pretty well with varying data distributions. Others like KNN and Logistic Regression need a little help.

We can see why this happens by considering KNN. Let us imagine we are asking KNN to predict a horse with daysLto of 156 and TRinrace of 0.75. To do this it will go looking for k (you can specify k) occurrences that most closely match 156 and 0.75. There is more than one algorithm KNN can be asked to use to find the nearest k neighbors but they all suffer from the same problem. How to balance the fact that say 140 is far bigger in difference to 156 than 0.7 is to 0.75. It is in danger of saying nothing + or - 1 either side of 156 is a worthy match. If it cannot find k it may push it to + or - 2. But we can see that this heavier weighting towards daysLto makes no sense. To help KNN we need to place both daysLto and TRinrace on the same scale ie 0 to 1.

We can do this by clicking the checkbox in the scale/bin column next to daysLto and then click the Scaler button at the top of the main screen.



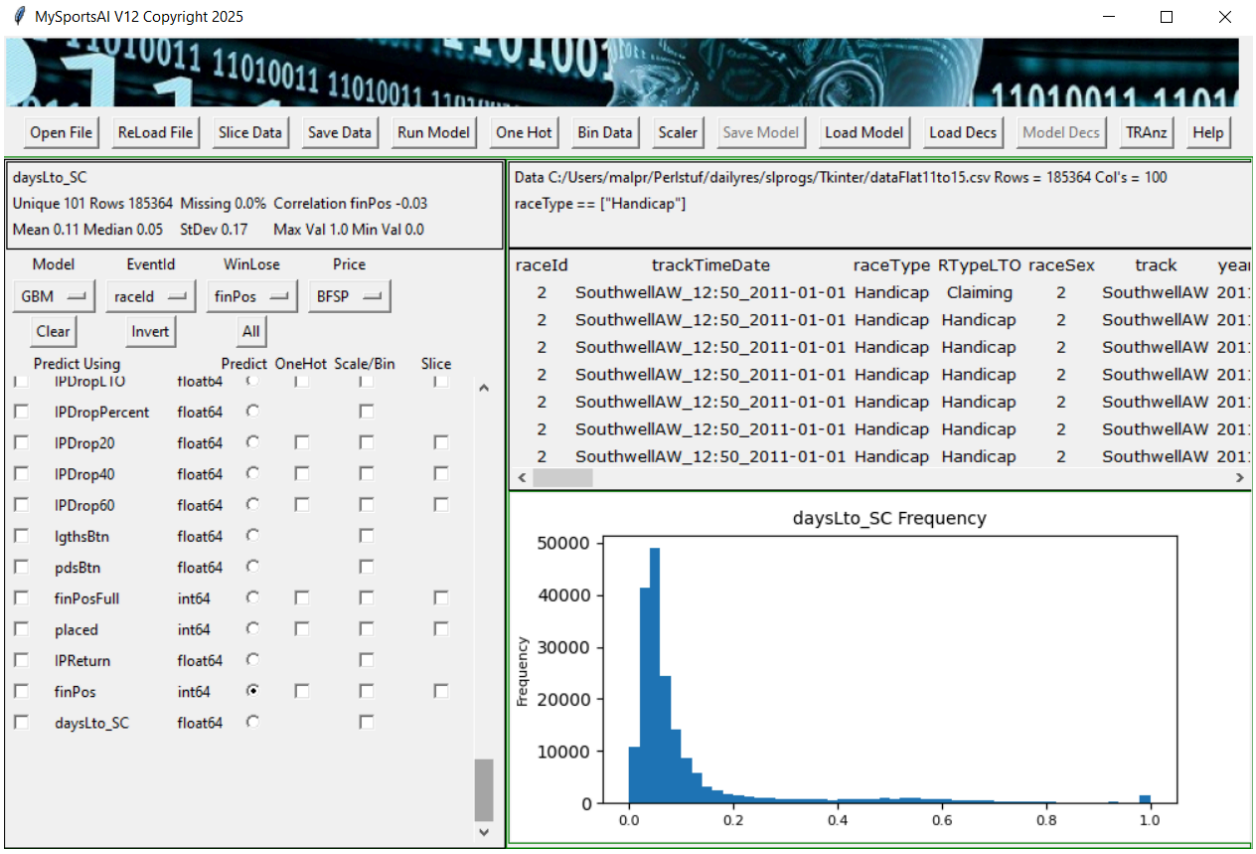

We are then presented with the scaler window.




There is more than one type of scaling available and you can check exactly what they do and how they differ via the help pages. For MinMaxScaler you are offered the choice of in race scale or train/test scale. The former will give the same treatment to daysLto as TRinrace received whereby the horse in each race with the longest layoff had a value of 1 and the others progressively less than 1. Train/test scale uses values across the whole of the train data which means any horse with a layoff of 365 will have a value of 1 and others progressively less than 1.

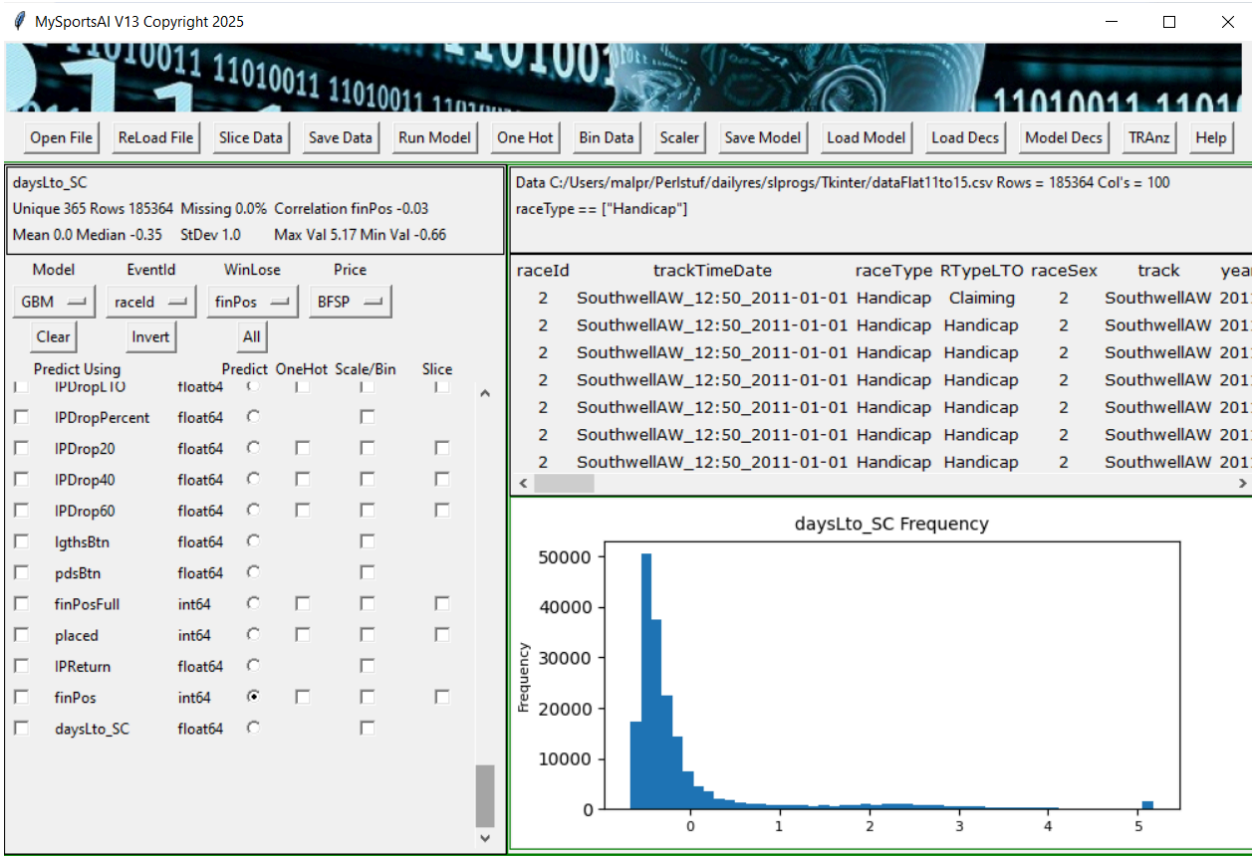
Here is where I re emphasise my earlier message, there is no right or wrong option. Many aspects of model building or try it and see. Having said that, MinMaxScaler is considered not the best choice for scaling when a data item contains outliers (see previous section on outliers). The fact that the max is an outlier distorts the scale. A better choice for daysLto may well be the Standard Scaler

Once we have scaled daysLto a new feature appears at the bottom of the feature list.




Notice the new scaled feature now has values between 0 and 1, a more suitable scale for using alongside 0 to 1 features and an algorithm like KNN.

Contrast that with daysLto scaled using Standardization.

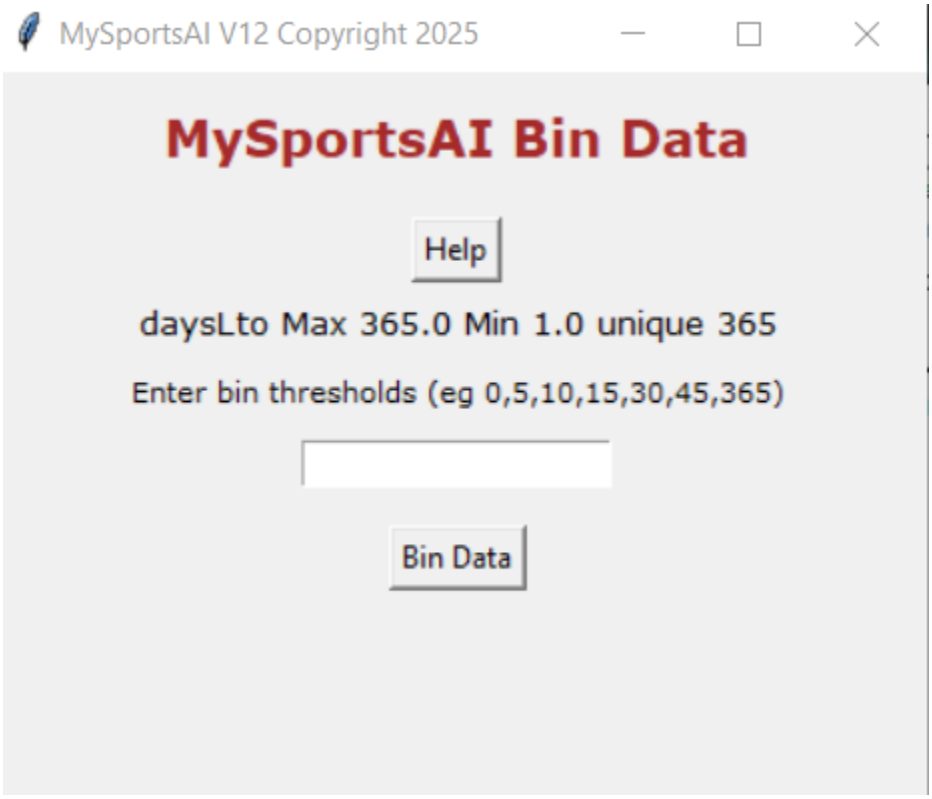



We now have values below zero this is because Standardization creates a range with a mean of zero and a standard deviation of 1

Let us go back to the unscaled version of daysLto because we are going to use an algorithm like GBM that handles data on different scales more readily. Nevertheless we may feel that a data item has some predictive value to it in different value segments. For example I may want to try creating individual features for buckets or bins of daysLto values. Let us say I want a feature for daysLto less than 6 which will contain a 1 if the horse ran less than 6 days ago and 0 otherwise. I also then need a feature bin 6 to let us say 10. Maybe the complete list of bins looks like this

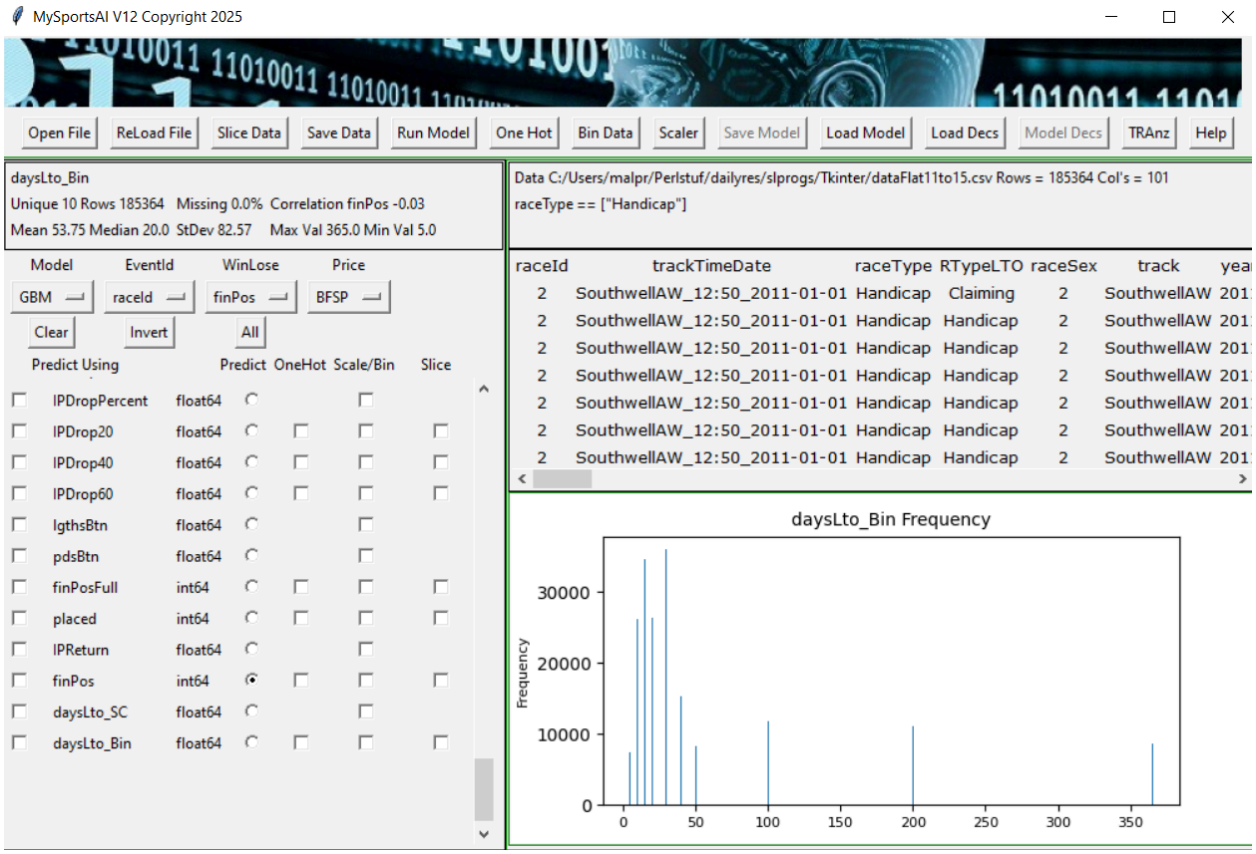
1 to 5
6 to 10
11 to 15
16 to 20
21 to 30
Etc etc

As before we check the checkbox in the slice/bin column next to daysLto and then click the Bin Data button where we are presented with



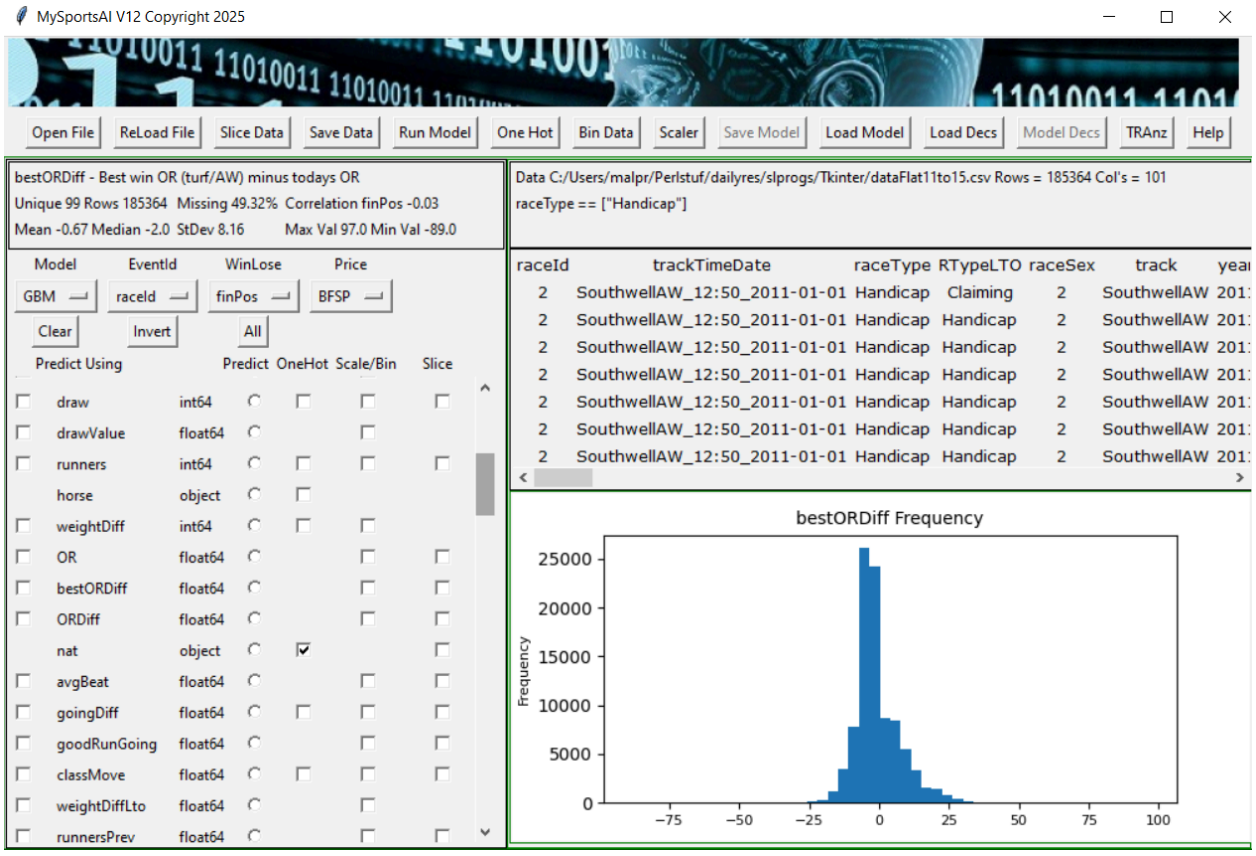

Now simply type in the boundaries of each bin eg- 0,5,10,15,20,30,40,50,100,200,365 and then hit Bin Data.

We now have a single feature that contains one of the bin limits we have set depending on the original daysLto value.

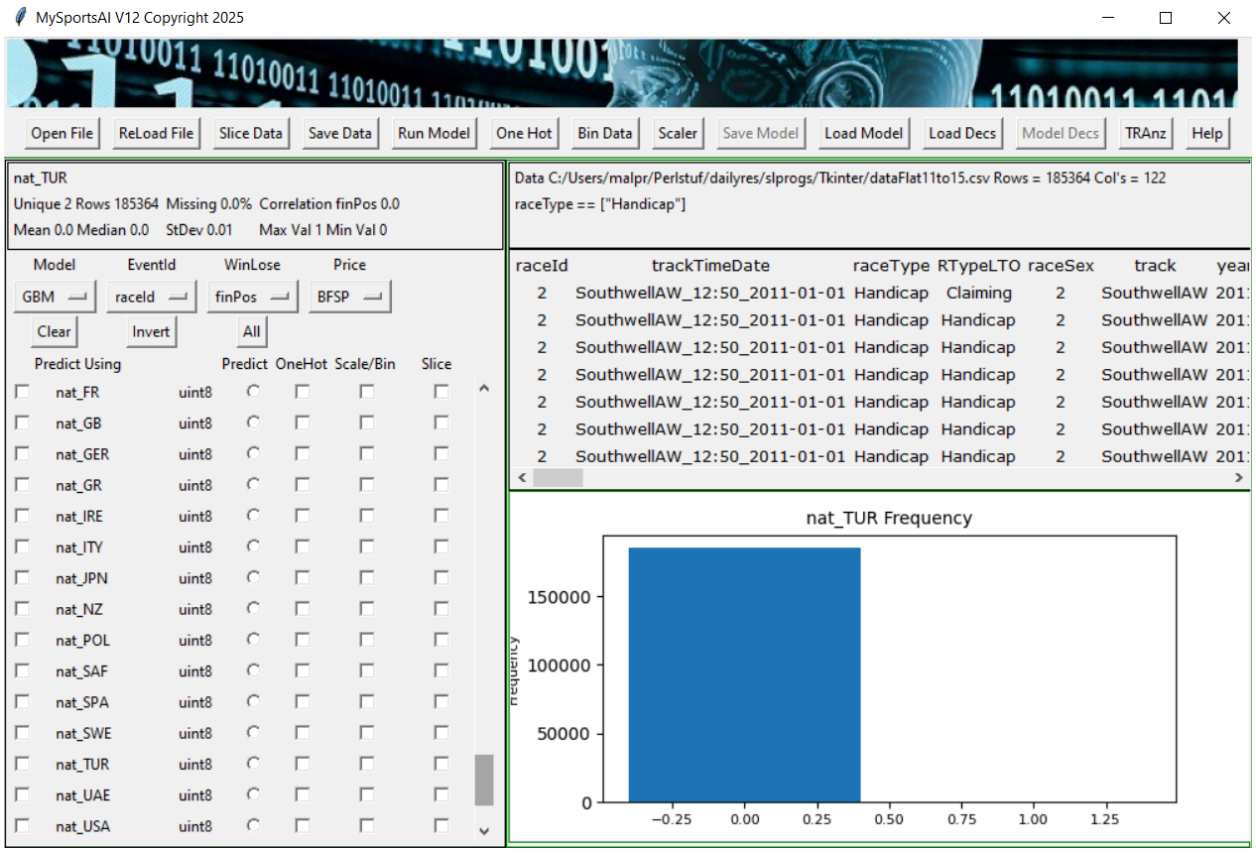



The graph on the right shows the frequency of each bin in the new feature.

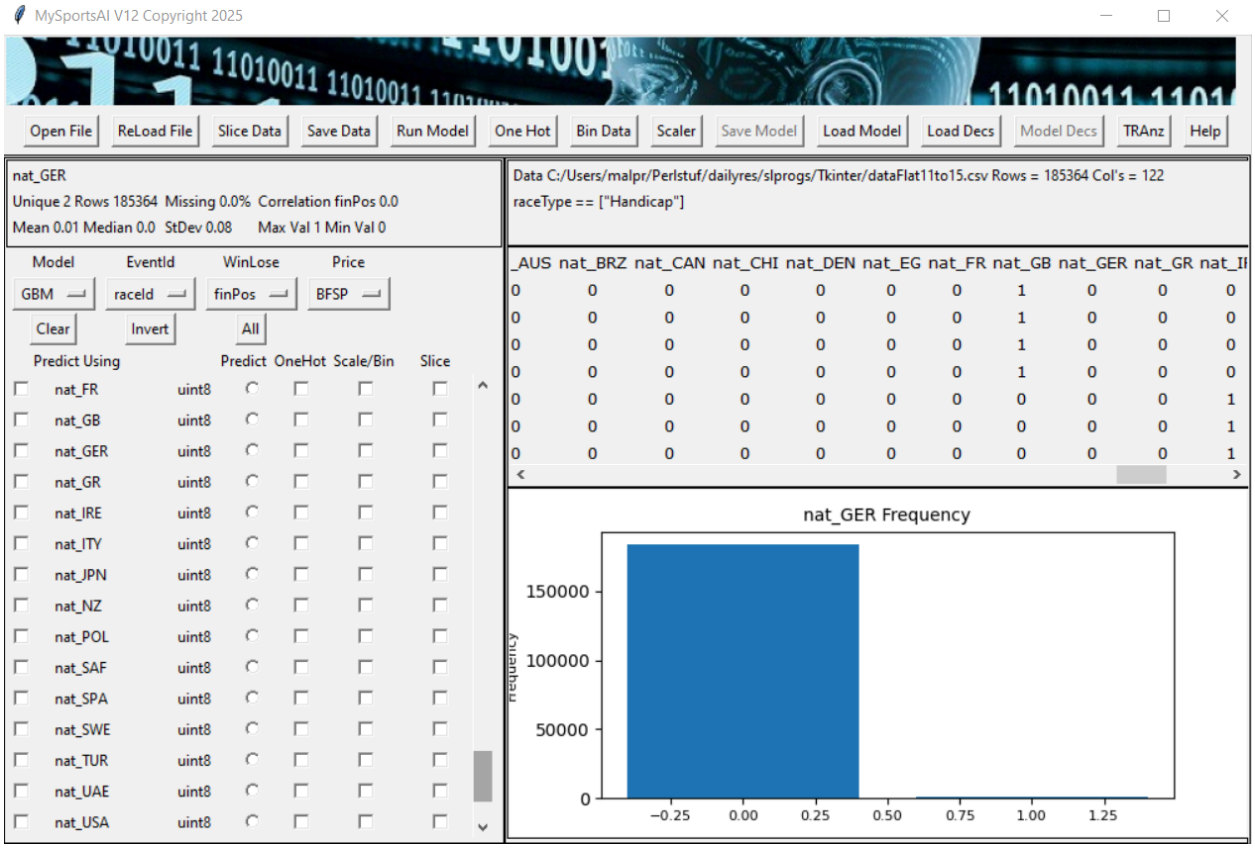
All input features to a Machine Learning algorithm have to be numeric values. So what do we do if we want to include horse nationality into the model. We can see below that nat is an object which means it is not a numeric value. We need to somehow convert the various values that nat takes into numeric values. To do this we check the checkbox in the one hot column next to nat and then click the One Hot button.



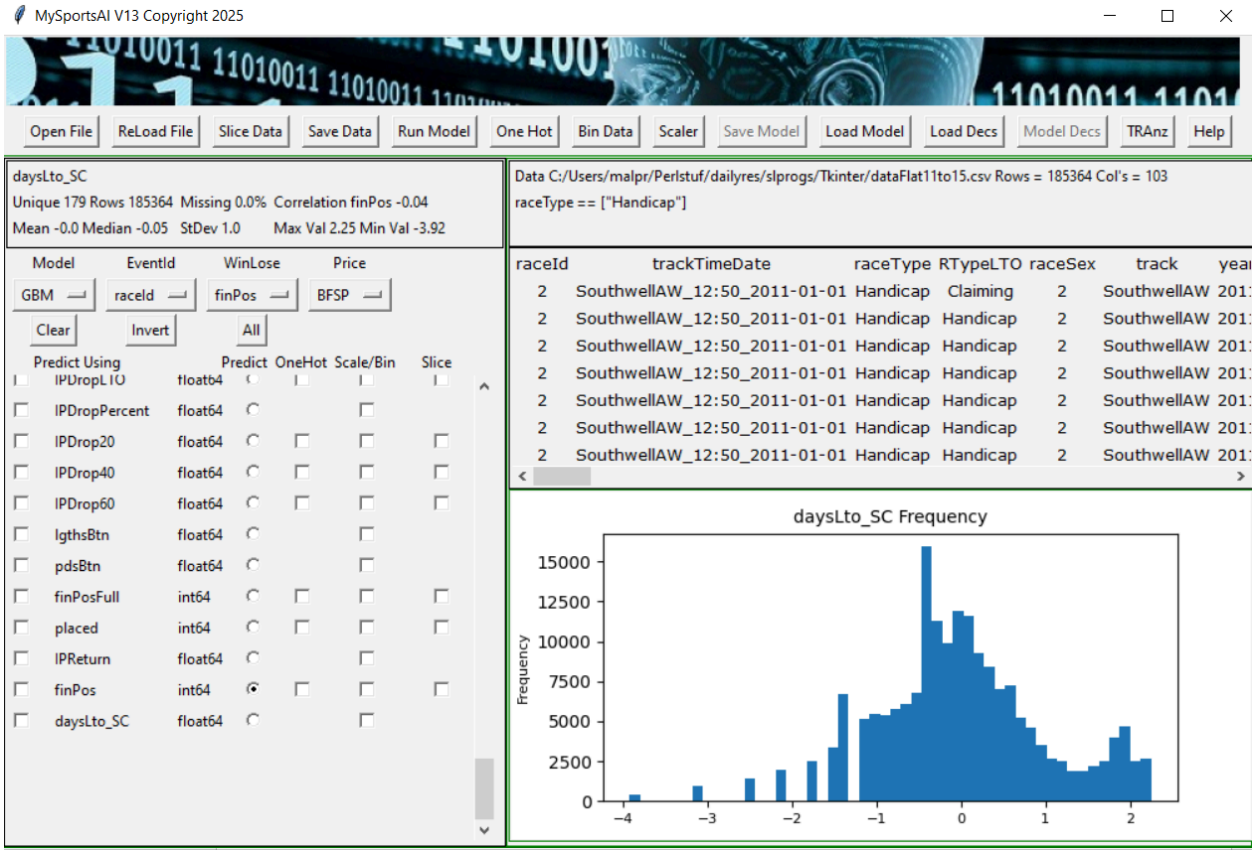

The one hot encoder has now created a whole raft of features, one for each nationality that exists in the data.




All these new features are actually added to the data, if I scroll across the window displaying the first few rows of the file we can see the extra fields and that the first four horses were all bred in GB.




In the previous chapter on outliers I mentioned that some algorithms like Logistic Regression work best with data that has a normal distribution ie a bell shaped curve. What can we do about daysLto if it clearly does not have this bell shaped curve. If we scale the data item using Yeo Johnson scaler we get the following new distribution.

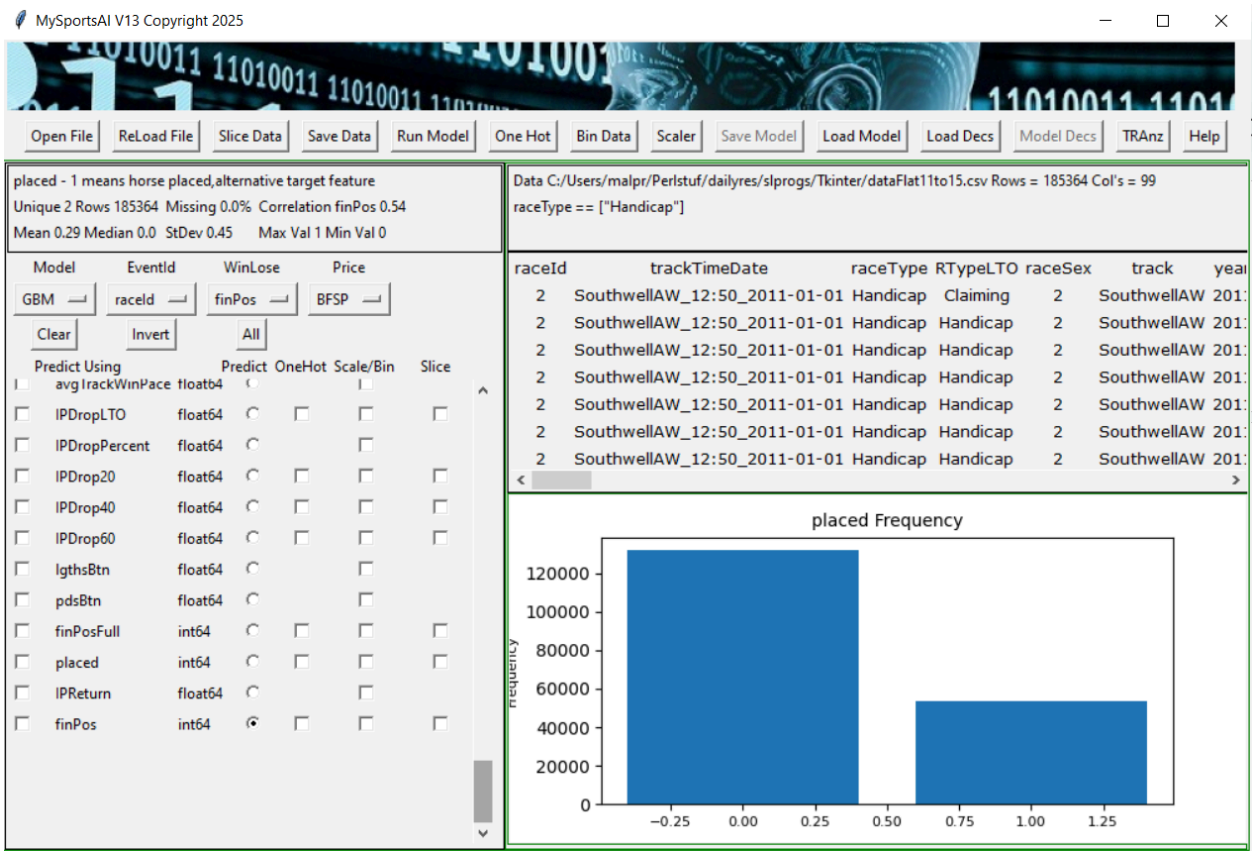



We can see that the scaled version is more normally distributed between -4 and 2. Each real occurrence of daysLto will have been mapped onto that scale and provided the daysLto attached to today's runners is also mapped then the algorithm will make the right predictions.

# What Should We Predict

So far I have introduced a single target feature finPos. The model we looked at was trying to predict finPos which contains a 1 if a horse won within the training data and a 0 if it lost. Any algorithm is trying to find meaningful patterns within the training data that bare a relationship between the input features you have chosen and the target feature finPos in this case. Predicting a 1 or 0 or in our case creating a probability of 1 or 0 is called a classification problem. We will now look at another avenue we can explore, regression

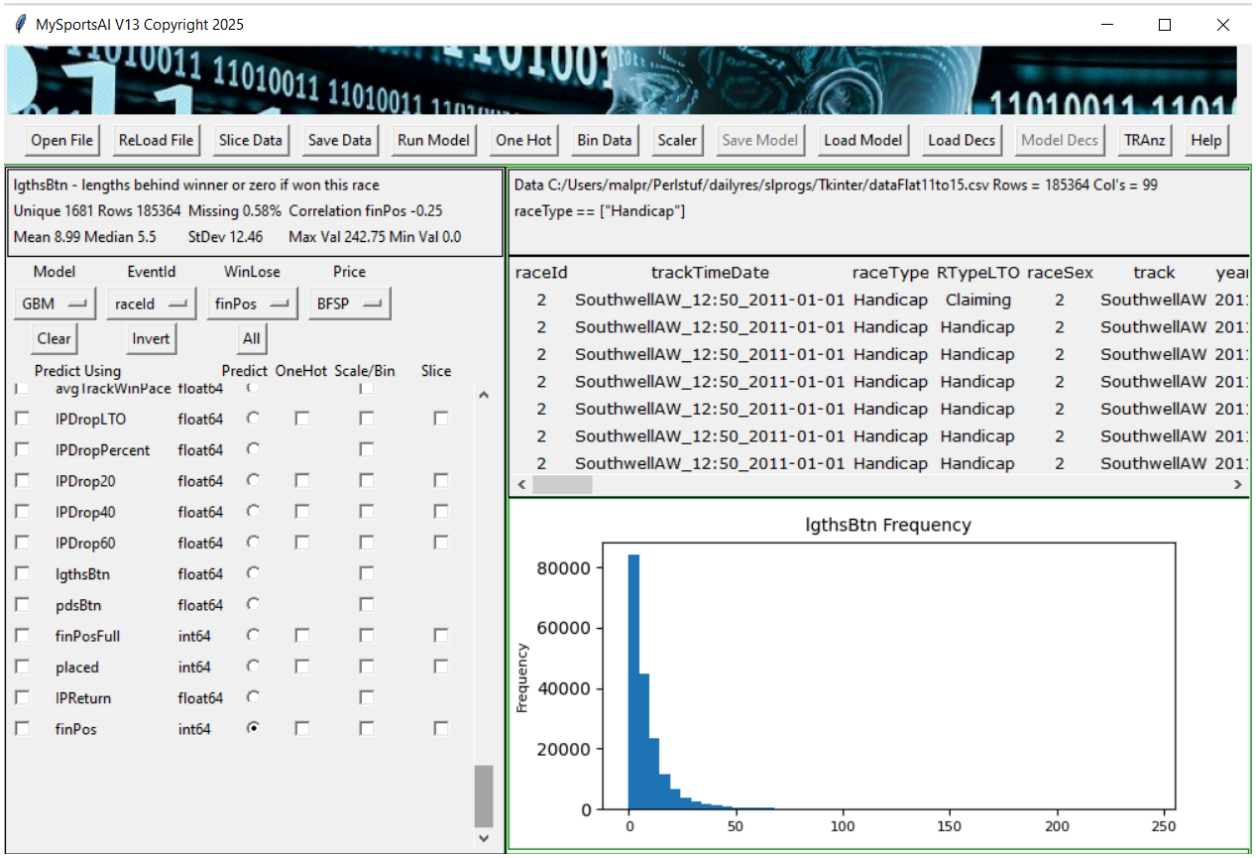
There are other target features we could choose. Let us take a look at the bottom of the feature list on the main window.



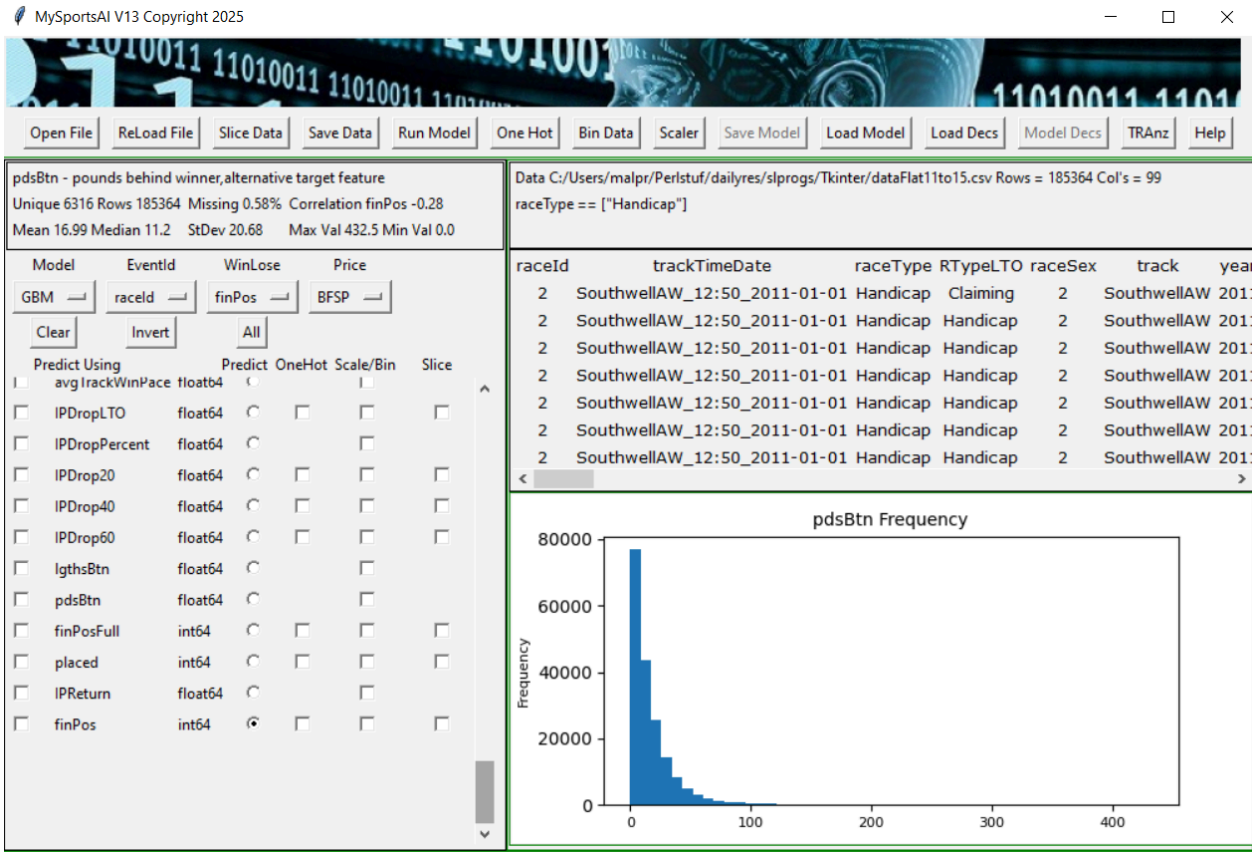

The bar chart on the right and the description at the top left show we are looking at the placed feature which contains a 1 if the horse placed 0 otherwise. The placed feature will obviously have more one's in the data set than finPos does which leads us on nicely to the concept of imbalanced data. A balanced data set would be one in which there are an equal number of one's and zero's. For some algorithms a balanced data set is

preferable but tree based algorithms eg GBM,XGBoost, ADABoost all work well with imbalanced data. There are tricks and methods for balancing data should it be needed but they are beyond the scope of this book. We can see that placed is more balanced than finPos but it is still a classification target as we are still predicting a one or a zero.

Lengths behind winner lgthsBtn would be an example of a regression target feature. It contains a zero if the horse won but a real value if beaten. For example 3.5 would mean the horse was beaten 3.5 lengths. The problem with lengths is that being beaten a length over two miles is not the same as being beaten a length over five furlongs.

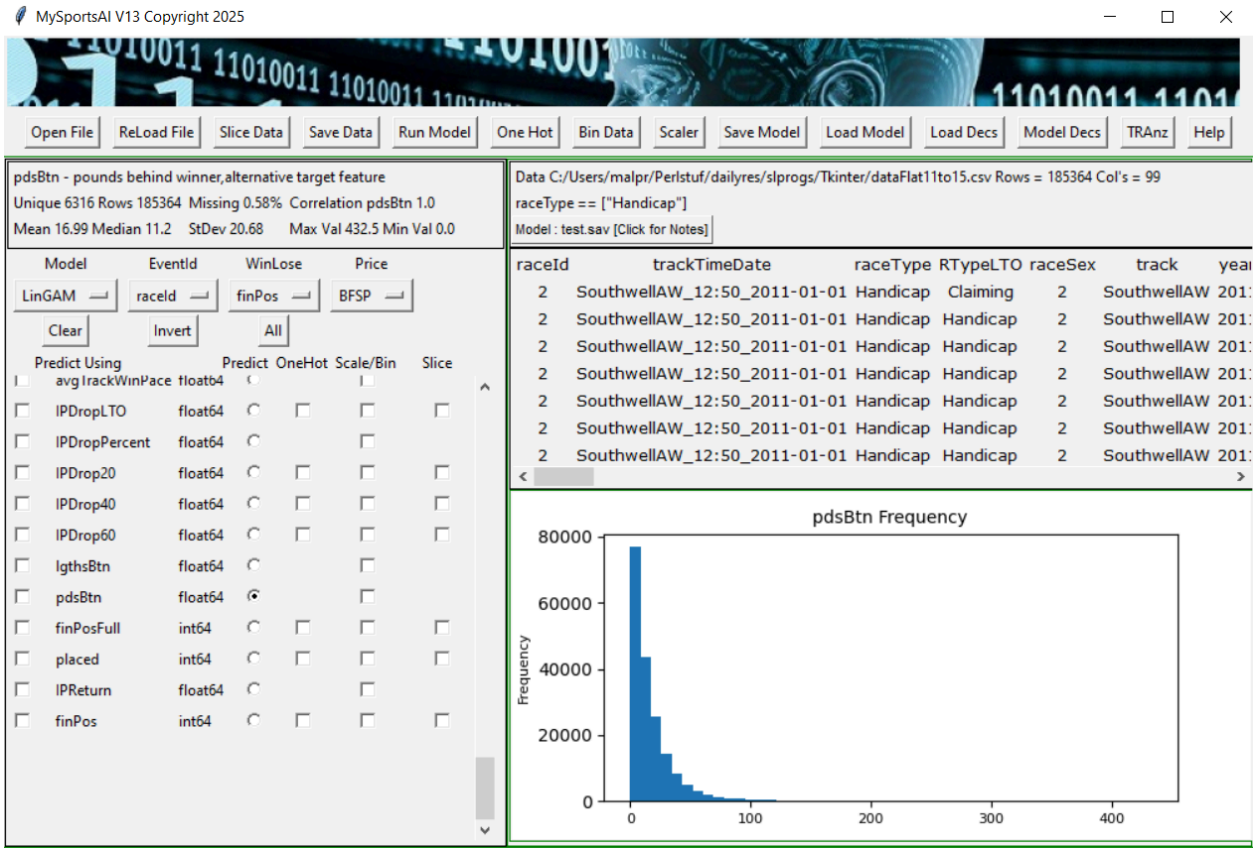



Perhaps a better alternative is to convert those lengths beaten into pounds beaten which will allow for the fact that 1 length over five furlongs converts to a different pound's beaten value than over a two mile trip.

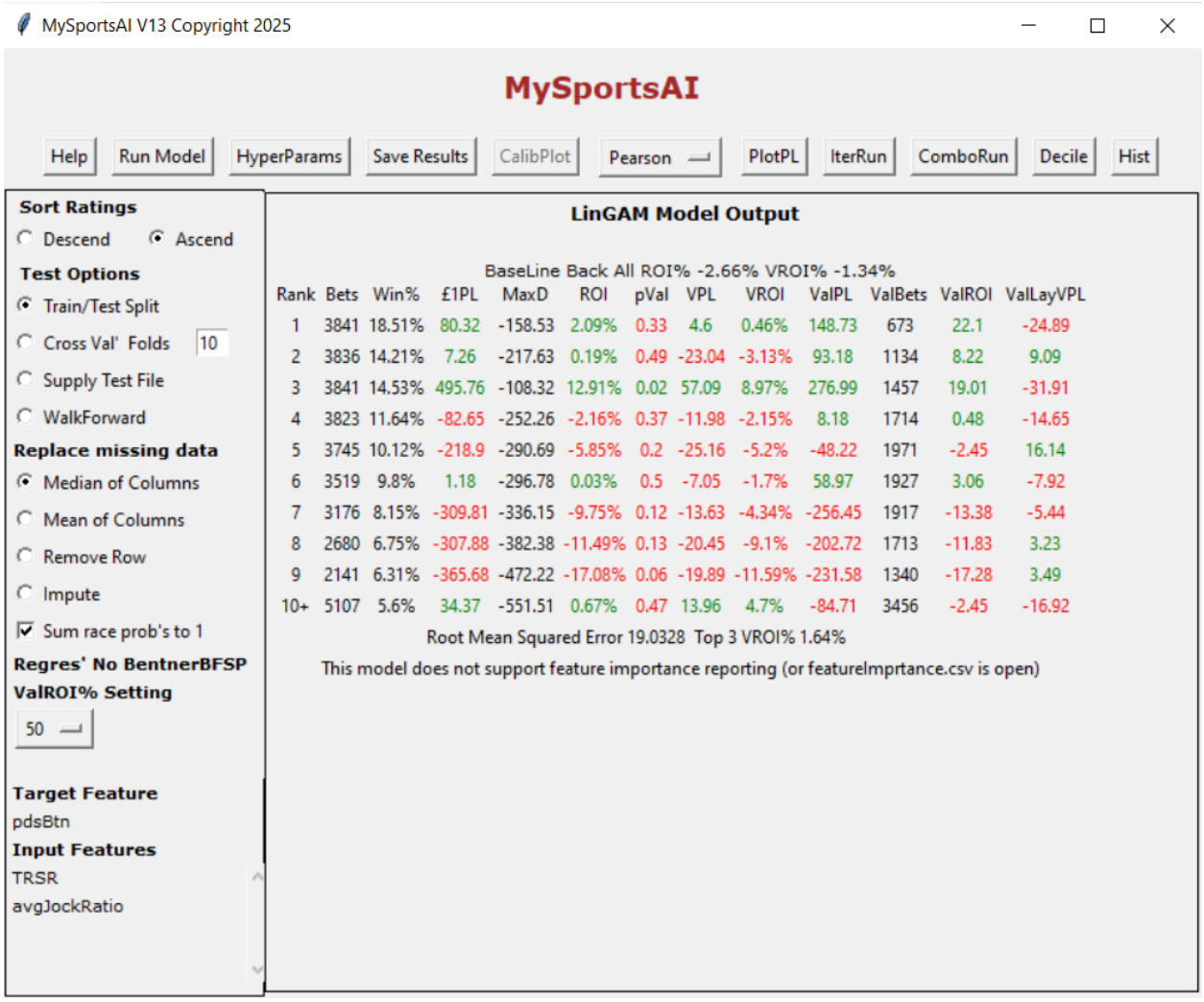



If we are going to use pdsBtn as our target feature we will need to use a regression algorithm from the choice of algorithms. Here I have selected LinGam as my regression algorithm and psdBtn as the target feature. So a quick recap here, classification algorithms predict discrete values like finPos 1 or finPos 0 (won or lost) while regression algorithms predict continuous values like lengths beaten or pounds beaten. MySportsAI will check your target feature and correct you if you choose the wrong algorithm. For example if you try to predict pdsBtn with GBM it will inform you that it is not the right algorithm.

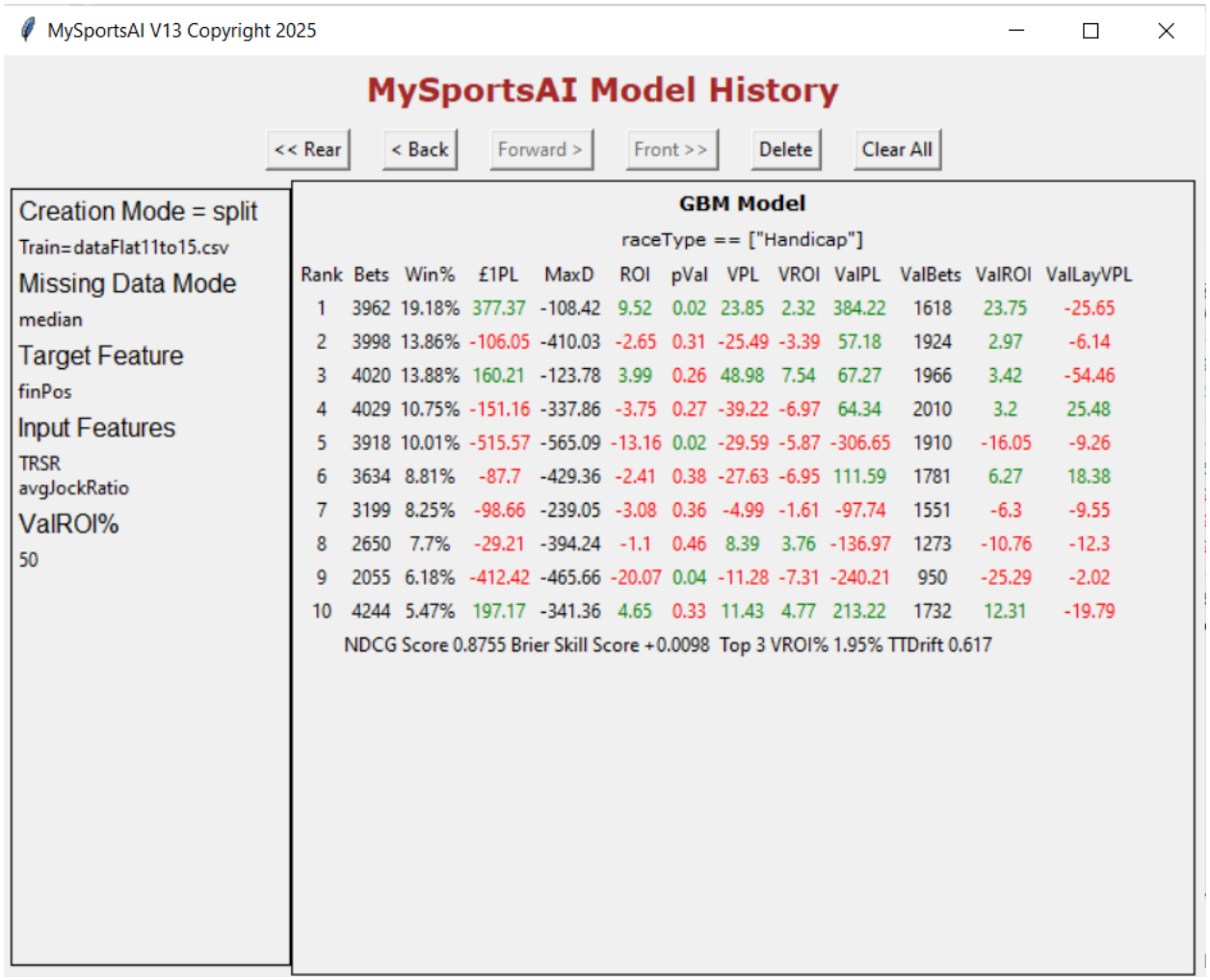
There is another good reason why using pdsBtn as the target feature may be preferable to simple old 1 for win 0 for lose. With the latter we are throwing away a great deal of information with the losers. Consider today's (as I write this) Lincoln Handicap. It is quite possible that some horse will come second from 22 runners and yet get beaten a short head. This will be logged as a zero, a loser as far as finPos is concerned and yet the horse is as close to winning as you can get. If we could consistently predict plus or minus short head runs we would be googling beach front properties.

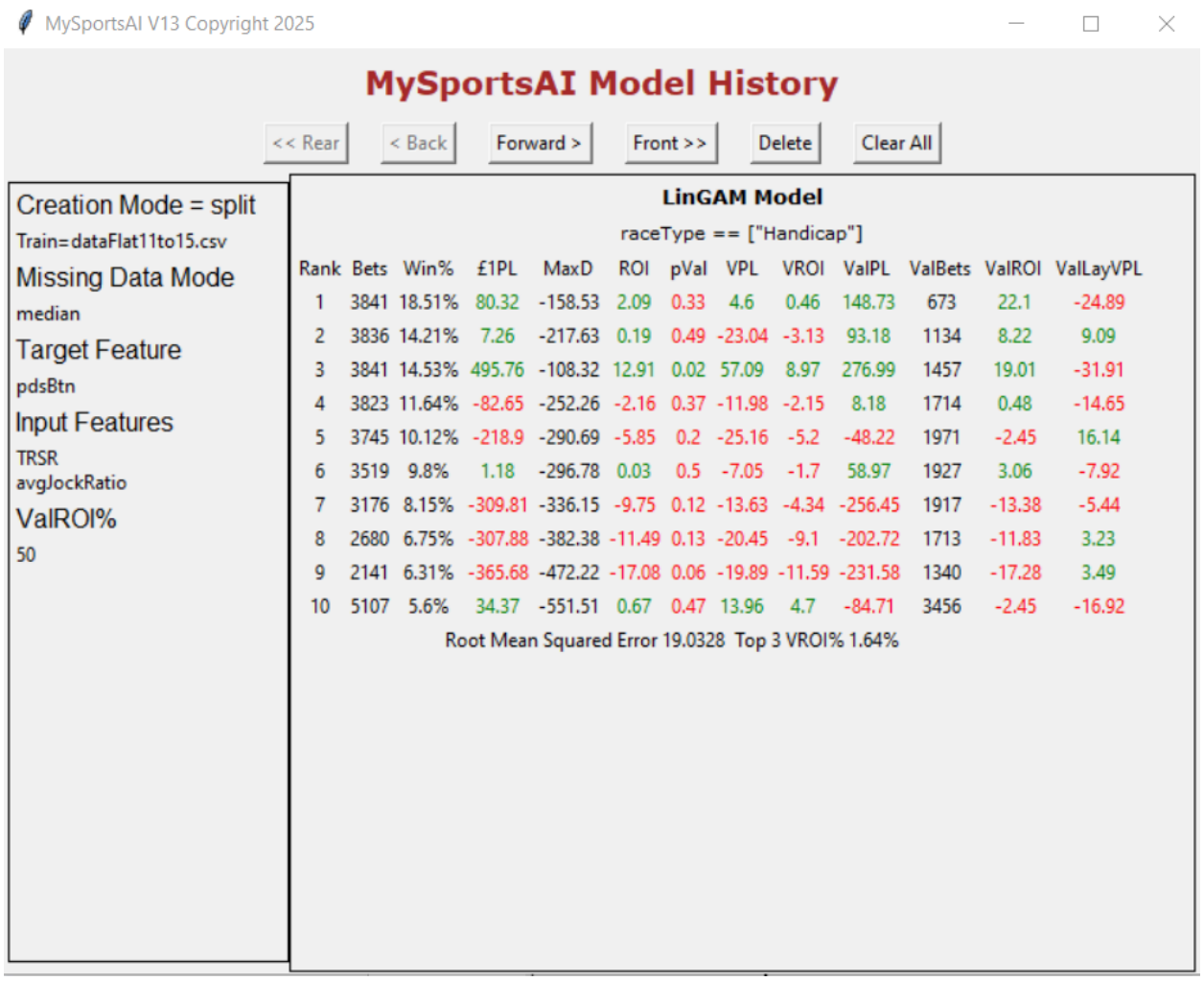



Clicking run model and then going through the same clicks on the train/test window as before we can see the results. Note that I had to set sort ratings to ascending. This is because we are predicting pdsBtn and of course a lower predicted pdsBtn is better than a higher so our top rated in a race should have the lowest predicted pdsBtn.




I reran the GBM model and then clicked on the Hist button in the top right. This displays the results of your previously run models during a session. It is a useful way to check back and compare. Here are the GBM results and the LinGam results.


MySportsAI V13 Copyright 2025
MySportsAI Model History
<< Rear
< Back
Forward >
Front >>
Delete
Clear All
Creation Mode = split
Train=dataFlat11to15.csv
Missing Data Mode
median
Target Feature
finPos
Input Features
TRSR
avgJockRatio
ValROI%
50
GBM Model
raceType == ["Handicap"]
Rank Bets Win% £1PL MaxD ROI pVal VPL VROI ValPL ValBets ValROI ValLayVPL
1 3962 19.18% 377.37 -108.42 9.52 0.02 23.85 2.32 384.22 1618 23.75 -25.65
2 3998 13.86% -106.05 -410.03 -2.65 0.31 -25.49 -3.39 57.18 1924 2.97 -6.14
3 4020 13.88% 160.21 -123.78 3.99 0.26 48.98 7.54 67.27 1966 3.42 -54.46
4 4029 10.75% -151.16 -337.86 -3.75 0.27 -39.22 -6.97 64.34 2010 3.2 25.48
5 3918 10.01% -515.57 -565.09 -13.16 0.02 -29.59 -5.87 -306.65 1910 -16.05 -9.26
6 3634 8.81% -87.7 -429.36 -2.41 0.38 -27.63 -6.95 111.59 1781 6.27 18.38
7 3199 8.25% -98.66 -239.05 -3.08 0.36 -4.99 -1.61 -97.74 1551 -6.3 -9.55
8 2650 7.7% -29.21 -394.24 -1.1 0.46 8.39 3.76 -136.97 1273 -10.76 -12.3
9 2055 6.18% -412.42 -465.66 -20.07 0.04 -11.28 -7.31 -240.21 950 -25.29 -2.02
10 4244 5.47% 197.17 -341.36 4.65 0.33 11.43 4.77 213.22 1732 12.31 -19.79
NDCG Score 0.8755 Brier Skill Score +0.0098 Top 3 VROI% 1.95% TTDrift 0.617

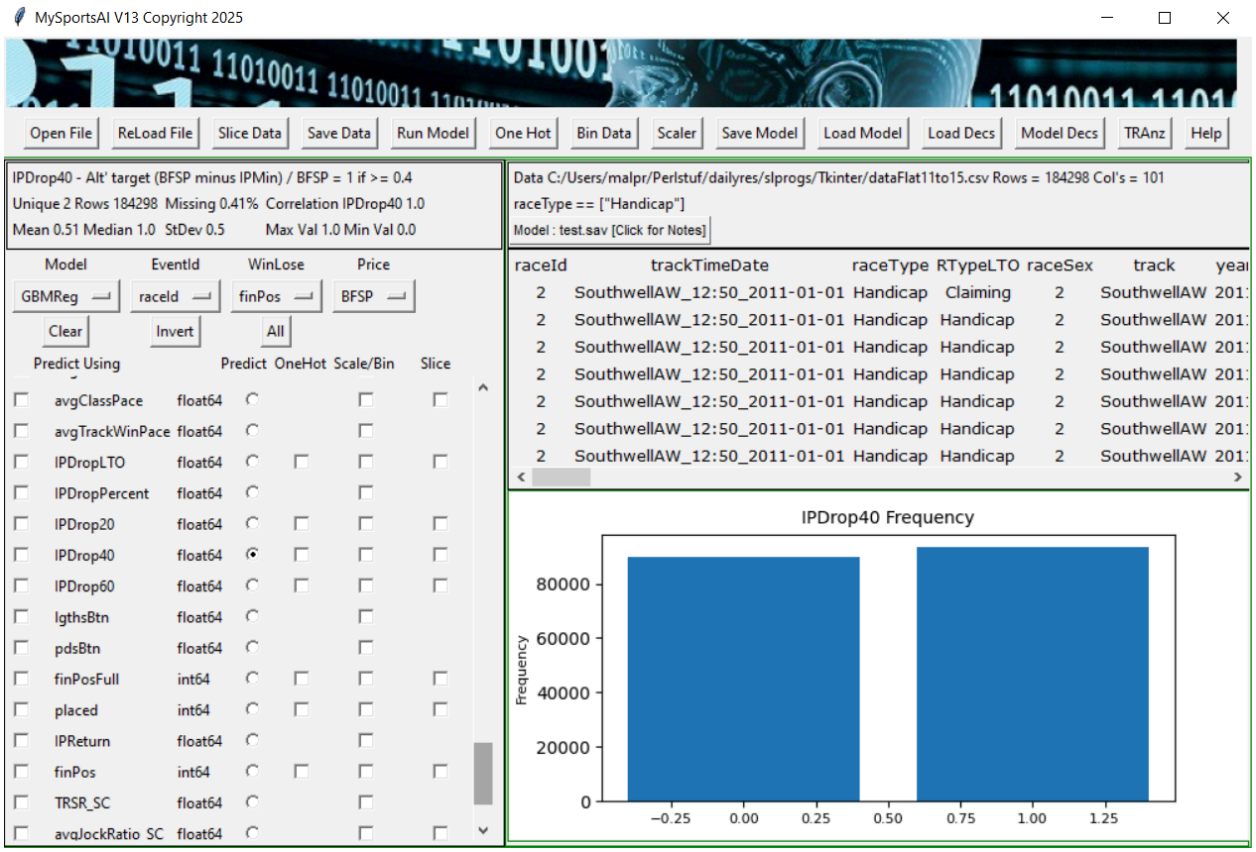



The GBM model appears more profitable than the LinGAM model

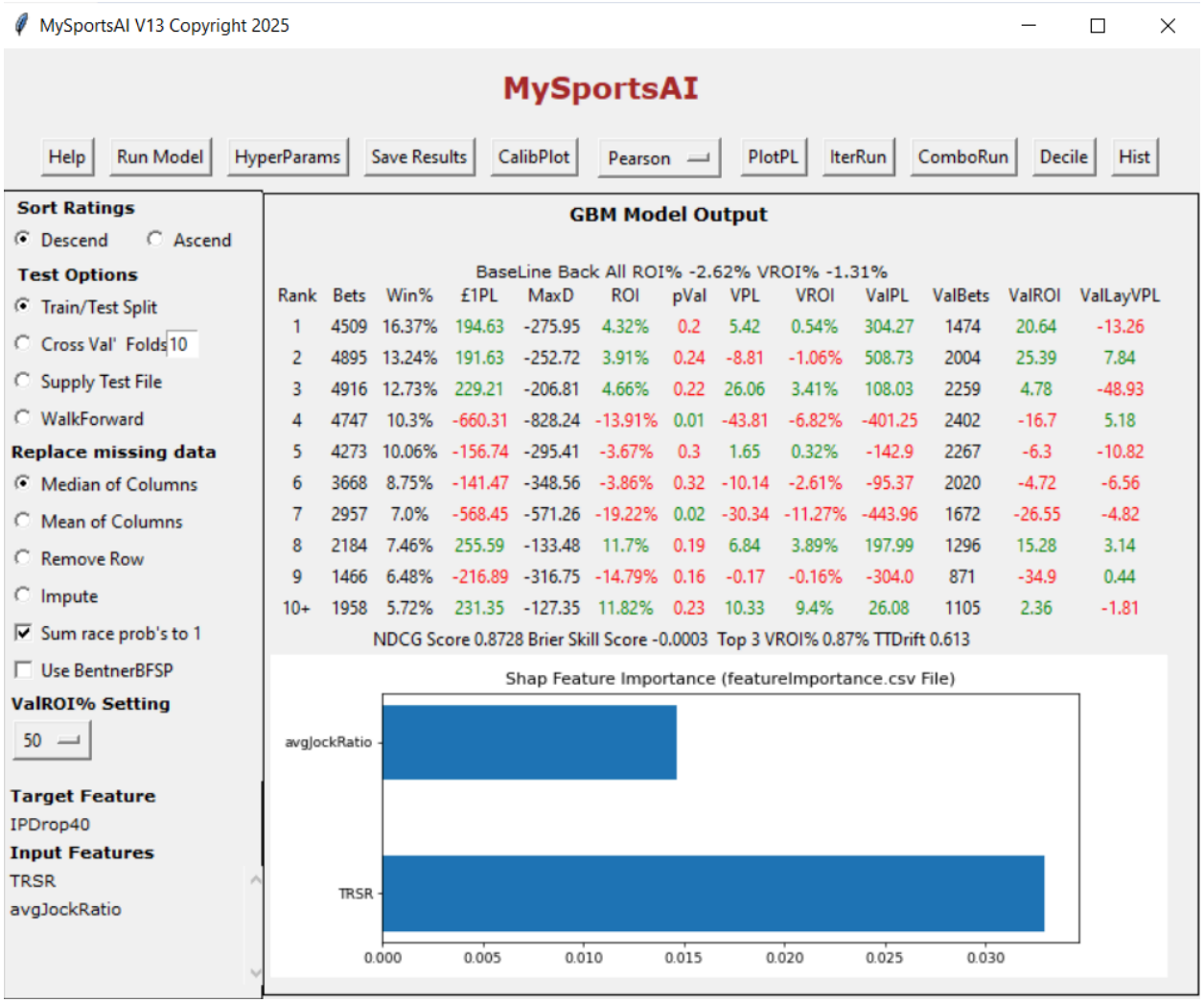
Is there anything we can predict that is more balanced than predicting placed horse's. We could use IPDrop40. This contains a one or a zero depending on how far the horse dropped in price in running according to the following formula.

(BFSP minus IPMin) / BFSP = 1 if >= 0.4

IPMin refers to the inplay minimum price. This can be a proxy for how well a horse ran as well as being a useful target feature for trading strategies.




We can see from the bar chart that IPDrop40 is pretty well balanced. Let us see how it performs as a target feature. Remember we are trying to predict IPDrop40 but we still have WinLose towards the top left of the screen set to finPos so our results are still based on whether a horse wins or loses.


MySportsAI V13 Copyright 2025
MySportsAI
Help
Run Model
HyperParams
Save Results
CalibPlot
Pearson
PlotPL
IterRun
ComboRun
Decile
Hist
Sort Ratings
Descend
Ascend
Test Options
Train/Test Split
Cross Val' Folds 10
Supply Test File
WalkForward
Replace missing data
Median of Columns
Mean of Columns
Remove Row
Impute
Sum race prob's to 1
Use BentnerBFSP
ValROI% Setting
50
Target Feature
IPDrop40
Input Features
TRSR
avgJockRatio
GBM Model Output
BaseLine Back All ROI% -2.62% VROI% -1.31%
Rank Bets Win% £1PL MaxD ROI pVal VPL VROI ValPL ValBets ValROI ValLayVPL
1 4509 16.37% 194.63 -275.95 4.32% 0.2 5.42 0.54% 304.27 1474 20.64 -13.26
2 4895 13.24% 191.63 -252.72 3.91% 0.24 -8.81 -1.06% 508.73 2004 25.39 7.84
3 4916 12.73% 229.21 -206.81 4.66% 0.22 26.06 3.41% 108.03 2259 4.78 -48.93
4 4747 10.3% -660.31 -828.24 -13.91% 0.01 -43.81 -6.82% -401.25 2402 -16.7 5.18
5 4273 10.06% -156.74 -295.41 -3.67% 0.3 1.65 0.32% -142.9 2267 -6.3 -10.82
6 3668 8.75% -141.47 -348.56 -3.86% 0.32 -10.14 -2.61% -95.37 2020 -4.72 -6.56
7 2957 7.0% -568.45 -571.26 -19.22% 0.02 -30.34 -11.27% -443.96 1672 -26.55 -4.82
8 2184 7.46% 255.59 -133.48 11.7% 0.19 6.84 3.89% 197.99 1296 15.28 3.14
9 1466 6.48% -216.89 -316.75 -14.79% 0.16 -0.17 -0.16% -304.0 871 -34.9 0.44
10+ 1958 5.72% 231.35 -127.35 11.82% 0.23 10.33 9.4% 26.08 1105 2.36 -1.81
NDCG Score 0.8728 Brier Skill Score -0.0003 Top 3 VROI% 0.87% TTDrift 0.613
Shap Feature Importance (featureImportance.csv File)
avgJockRatio
TRSR
0.000
0.005
0.010
0.015
0.020
0.025
0.030

# How Many Input Features

So far I have demonstrated MySportsAI with just two input features. The reason for this is that all the models so far are just aimed at demonstrating the software. You are probably wondering however, how many input features should we use as well as which ones are best. Like a lot of things in ML there is no absolute correct answer to the first question. Increasing the number of input features does not guarantee a better model and in fact can lead to a case of overfitting. This is an ML term used to describe a situation where the algorithm is behaving as though it is memorizing the training data. Think of it in terms of a student going through past Maths exam questions but memorizing the answers to these past questions without understanding the method behind solving them. When the student faces fresh questions that do not look exactly like the past questions he fails to solve them.

Throwing more and more input features at a model can lead to overfitting whilst too few features can lead to underfitting where no underlying patterns are learnt by the algorithm. A model needs to generalize to fresh data whilst acknowledging that there will always be some grey area that it fails to capture and understand. What I would be sure of is that faced with two models of similar performance where one has many features and the other only has perhaps two or three I would choose the latter. Fewer features means fewer points at which the model can break. Occam's Razor sums this up "When confronted with 2 or more competing explanations, Occam's Razor suggests that the explanation that makes fewer assumptions or has fewer entities is the more likely to be true".

I can tell you that there is one feature in MySportsAI which when in race scaled and used alone produces a top rated profit in every test set that I tested. It would not be for the faint hearted but has produced profit in 2016/17, 2018/19 and 2021/24. Not bad for a single feature model.

One possible alternative approach is to split input features into logical groups and then create individual models for each and then use the probabilities of each model as inputs to a final ensemble model.A concrete example will make this clearer.

Perhaps we first create a model to estimate a horse's general fitness by incorporating just two input features, daysLto and prevLto. We build a classification model using the GBM algorithm training the model on 2011 to 2015 data and then predicting results onto the 2016 to 2017 data. We save the results for the test data into a file and then load these up into Excel. Here are the first few lines.

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model GB | Slice raceType == ["Handicap"] | | | | | | | | | | | |
| Input Feat | daysLto | prevLto | | | | | | | | | | |
| HyperPara | ccp_alpha | criterion fr | init None | learning_ra | loss deviar | max_deptl | max_featu | max_leaf_ | min_impur | min_samp | min_samp | min_weigh | n_ |
| | | | | | | | | | | | | |
| raceId | finPosFull | raceDates | horse | finPos | BFSP | rating | HHindex | BFSPStDev | ratStDev | PL | BFSPRank | ratRank | |
| 1 | 5 | ######## | What Coul | 0 | 5.95 | 0.108604 | 783.4556 | 9.935722 | 0.00772 | -1 | 2 | 1 | |
| 1 | 3 | ######## | Ythan Wat | 0 | 7.7 | 0.10648 | 783.4556 | 9.935722 | 0.00772 | -1 | 4 | 2 | |
| 1 | 1 | ######## | Moon Rive | 1 | 9.87 | 0.105726 | 783.4556 | 9.935722 | 0.00772 | 8.6926 | 7 | 3 | |
| 1 | 2 | ######## | Clockmake | 0 | 6.6 | 0.105153 | 783.4556 | 9.935722 | 0.00772 | -1 | 3 | 4 | |
| 1 | 6 | ######## | Tiger's Hor | 0 | 5.28 | 0.10107 | 783.4556 | 9.935722 | 0.00772 | -1 | 1 | 5 | |
| 1 | 4 | ######## | Best Tama | 0 | 8.26 | 0.099684 | 783.4556 | 9.935722 | 0.00772 | -1 | 5 | 6 | |
| 1 | 7 | ######## | Myboydan | 0 | 8.71 | 0.089426 | 783.4556 | 9.935722 | 0.00772 | -1 | 6 | 7 | |
| 1 | 8 | ######## | Kodiac Lac | 0 | 35.26 | 0.088532 | 783.4556 | 9.935722 | 0.00772 | -1 | 8 | 8 | |
| 2 | 8 | ######## | Frantical | 0 | 11.42 | 0.146183 | 302.6561 | 263.5517 | 0.01418 | -1 | 6 | 1 | |
| 2 | 3 | ######## | Roger Tho | 0 | 14.63 | 0.108604 | 302.6561 | 263.5517 | 0.01418 | -1 | 8 | 2.5 | |
| 2 | 9 | ######## | Indomitab | 0 | 9.32 | 0.108604 | 302.6561 | 263.5517 | 0.01418 | -1 | 4 | 2.5 | |

If we remove the first four model description rows and then sort the data into horse alphabetical order within each race (you will see why in a second) we can then copy and paste into a new sheet the following columns, raceId,horse,rating,finPos and BFSP.

| A | B | C | D | E |
|---|---|---|---|---|
| raceId | horse | finPos | BFSP | rating |
| 1 | Best Tamayuz | 0 | 8.26 | 0.099684 |
| 1 | Clockmaker (IRE) | 0 | 6.6 | 0.105153 |
| 1 | Kodiac Lady (IRE) | 0 | 35.26 | 0.088532 |
| 1 | Moon River (IRE) | 1 | 9.87 | 0.105726 |
| 1 | Myboydaniel | 0 | 8.71 | 0.089426 |
| 1 | Tiger's Home | 0 | 5.28 | 0.10107 |
| 1 | What Could She Be | 0 | 5.95 | 0.108604 |
| 1 | Ythan Waters | 0 | 7.7 | 0.10648 |
| 2 | Basingstoke (IRE) | 0 | 6.27 | 0.100735 |
| 2 | Eeny Mac (IRE) | 0 | 140 | 0.089426 |
| 2 | Excelling Oscar (IRE | 0 | 5.02 | 0.098098 |
| 2 | Frantical | 0 | 11.42 | 0.146183 |

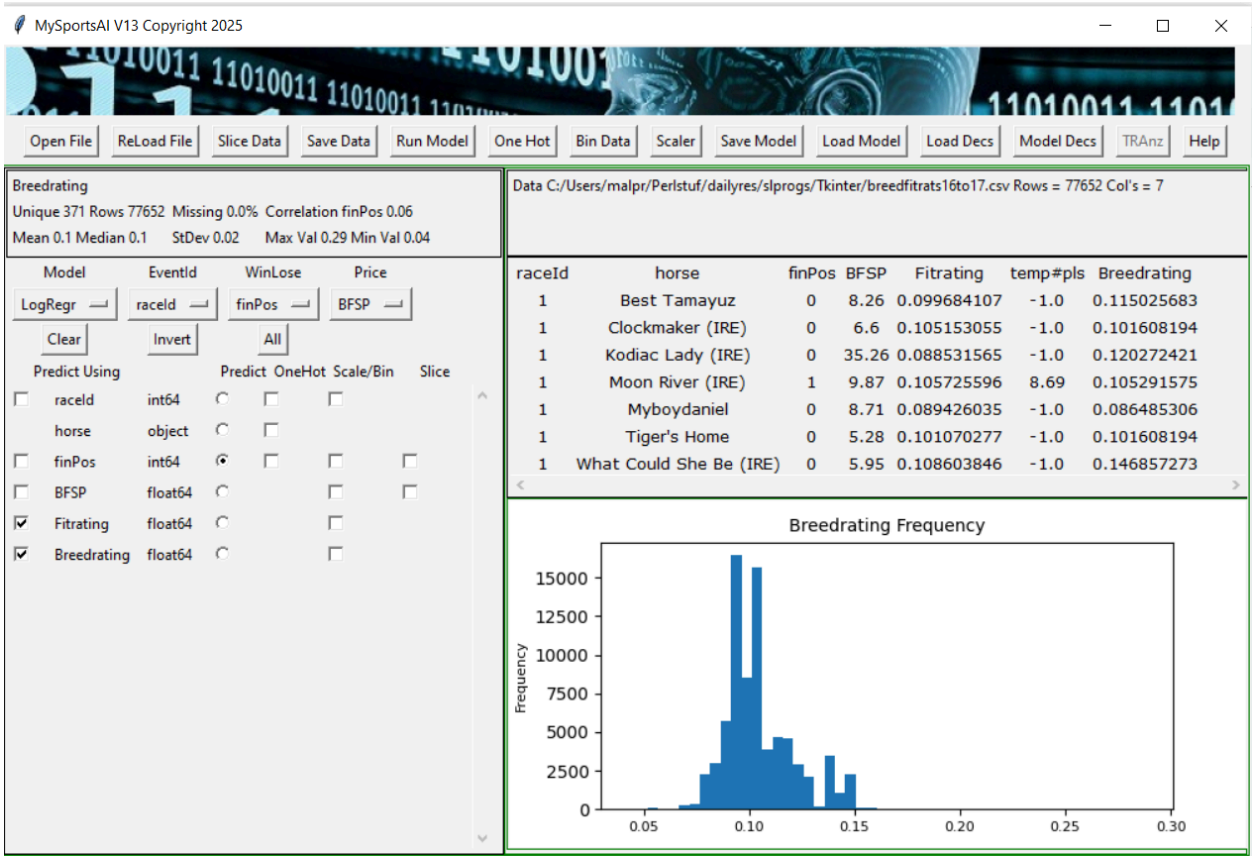
Now let us create a model using breeding. Our input features will be Sireinrace and DamSireSurf. As before we train 2011 to 2015 and then test on 2016 to 2017. Save the results and sort within Excel on horse name and then copy across the ratings into the new sheet we just created.

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| raceId | horse | finPos | BFSP | rating | rating |
| 1 | Best Tamayuz | 0 | 8.26 | 0.099684 | 0.115026 |
| 1 | Clockmaker (IRE) | 0 | 6.6 | 0.105153 | 0.101608 |
| 1 | Kodiac Lady (IRE) | 0 | 35.26 | 0.088532 | 0.120272 |
| 1 | Moon River (IRE) | 1 | 9.87 | 0.105726 | 0.105292 |
| 1 | Myboydaniel | 0 | 8.71 | 0.089426 | 0.086485 |
| 1 | Tiger's Home | 0 | 5.28 | 0.10107 | 0.101608 |
| 1 | What Could She Be | 0 | 5.95 | 0.108604 | 0.146857 |
| 1 | Ythan Waters | 0 | 7.7 | 0.10648 | 0.099695 |
| 2 | Basingstoke (IRE) | 0 | 6.27 | 0.100735 | 0.125692 |
| 2 | Eeny Mac (IRE) | 0 | 140 | 0.089426 | 0.101238 |
| 2 | Excelling Oscar (IRE | 0 | 5.02 | 0.098098 | 0.130593 |

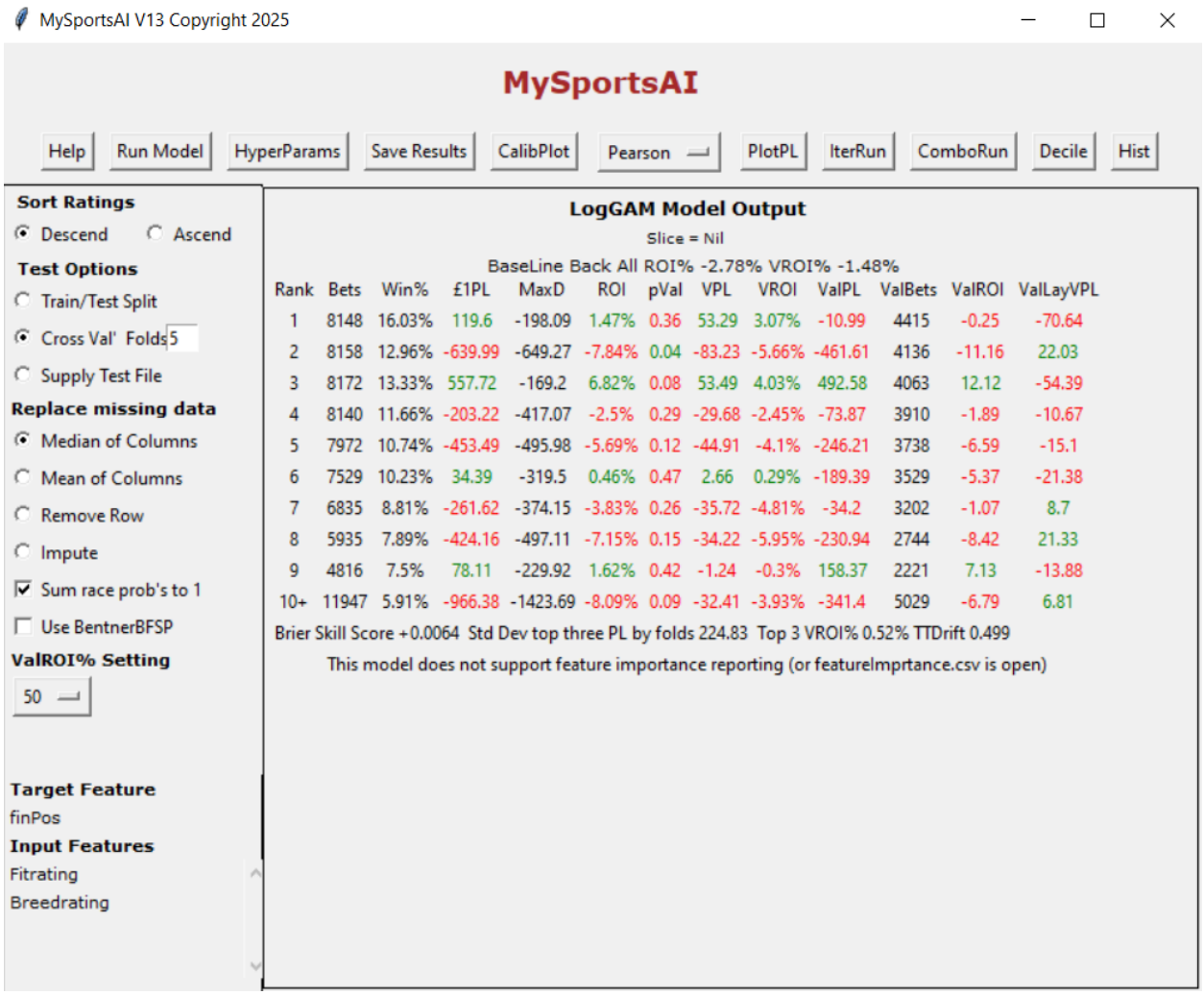
You can probably see why I sorted on horse name within each race now, to ensure that the ratings in the two columns align with the same horse in each race. Now I need to give the ratings different header names.

| raceId | horse | finPos | BFSP | Fitrating | Breedrating |
|---|---|---|---|---|---|
| 1 | Best Tamayuz | 0 | 8.26 | 0.099684 | 0.115026 |
| 1 | Clockmaker (IRE) | 0 | 6.6 | 0.105153 | 0.101608 |
| 1 | Kodiac Lady (IRE) | 0 | 35.26 | 0.088532 | 0.120272 |
| 1 | Moon River (IRE) | 1 | 9.87 | 0.105726 | 0.105292 |
| 1 | Myboydaniel | 0 | 8.71 | 0.089426 | 0.086485 |
| 1 | Tiger's Home | 0 | 5.28 | 0.10107 | 0.101608 |
| 1 | What Could She Be | 0 | 5.95 | 0.108604 | 0.146857 |
| 1 | Ythan Waters | 0 | 7.7 | 0.10648 | 0.099695 |
| 2 | Basingstoke (IRE) | 0 | 6.27 | 0.100735 | 0.125692 |
| 2 | Eeny Mac (IRE) | 0 | 140 | 0.089426 | 0.101238 |
| 2 | Excelling Oscar (IRE | 0 | 5.02 | 0.098098 | 0.130593 |

Saving this file and then loading it into MySportsAI I can now train/test a model based on the rating inputs from the two models.Here is the file loaded up.

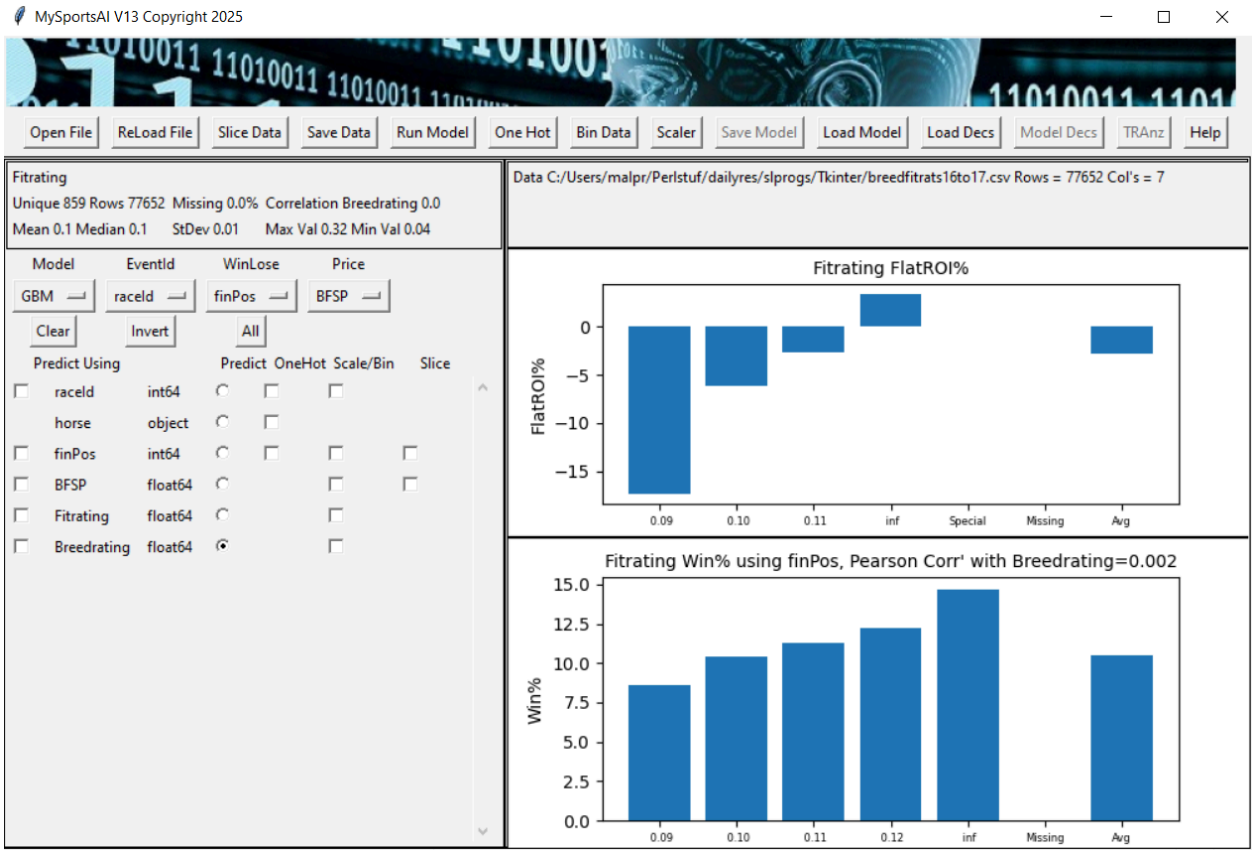


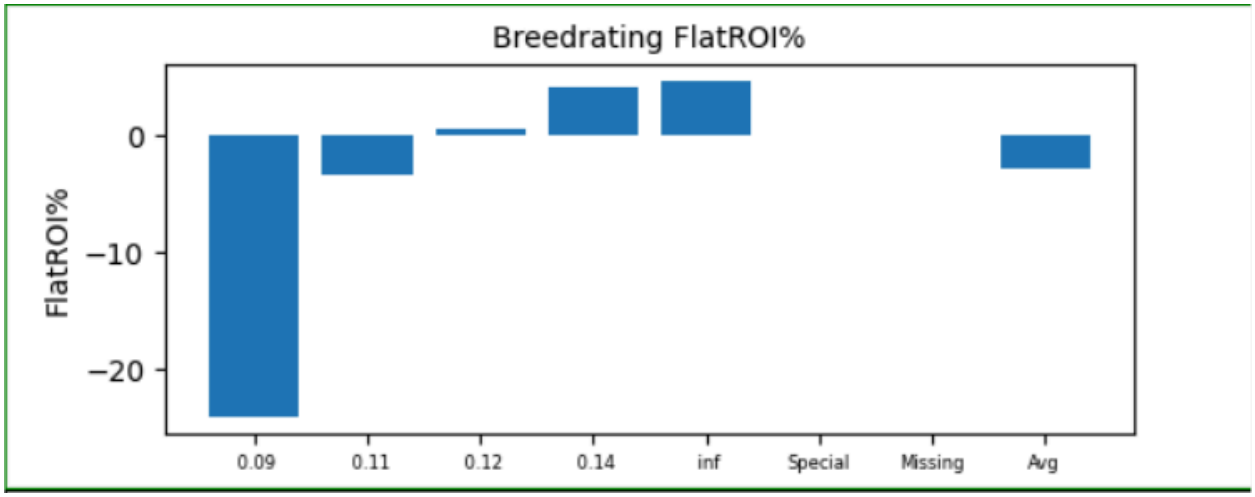

Now we can train/test this model using K fold cross validation to get a decent sample size as the ordering is not important.




Using a LogGAM algorithm the combined ratings as inputs have managed to create a profitable top ranked horse set. This ensemble of ratings is the basis of an experiment called Wisdom of Models which I will expand on in the ‘Is ML Betting Profitable’ section of the book. The above example is just two model inputs which I chose randomly. The idea can obviously be expanded to more than two.

This approach may also offer a hybrid of system betting and modelling when it comes to bet selection. We can see from the two charts below that both input ratings are profitable above 0.11

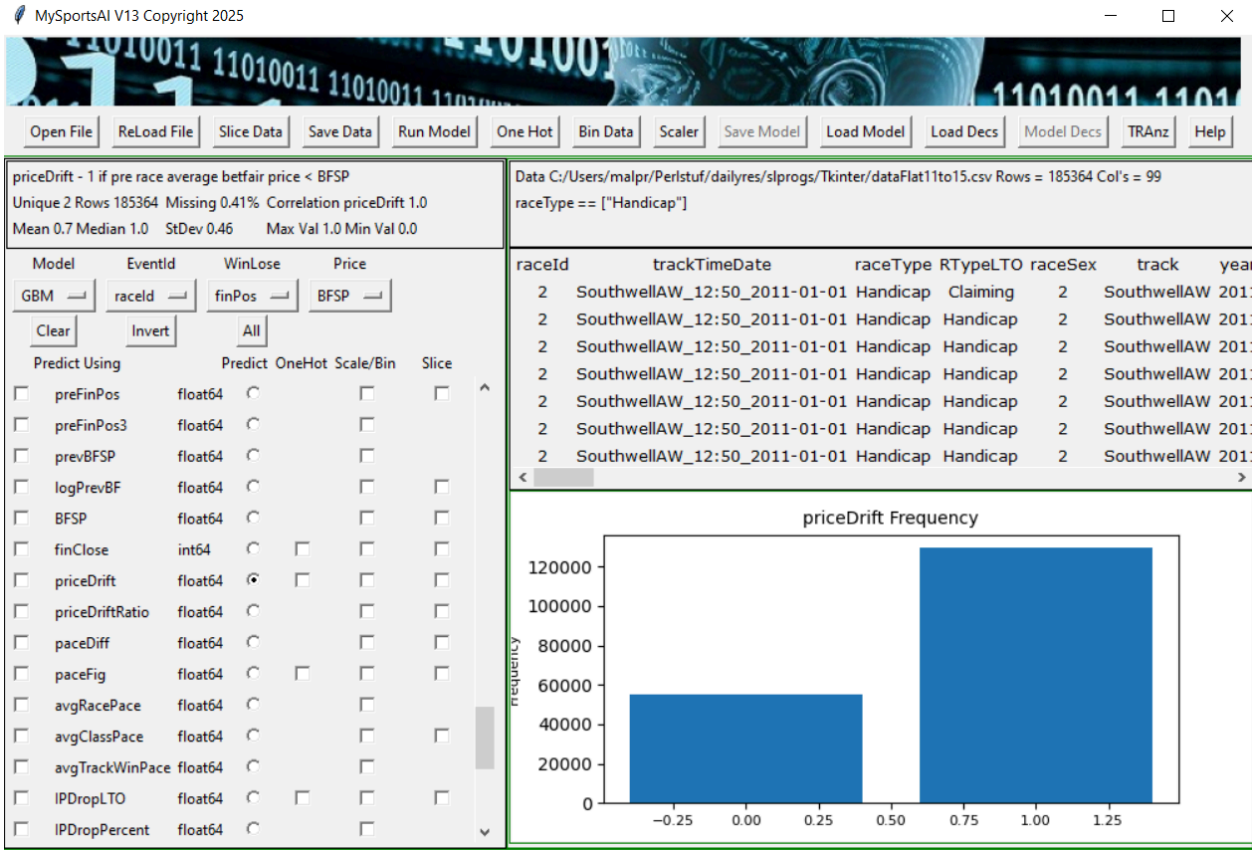


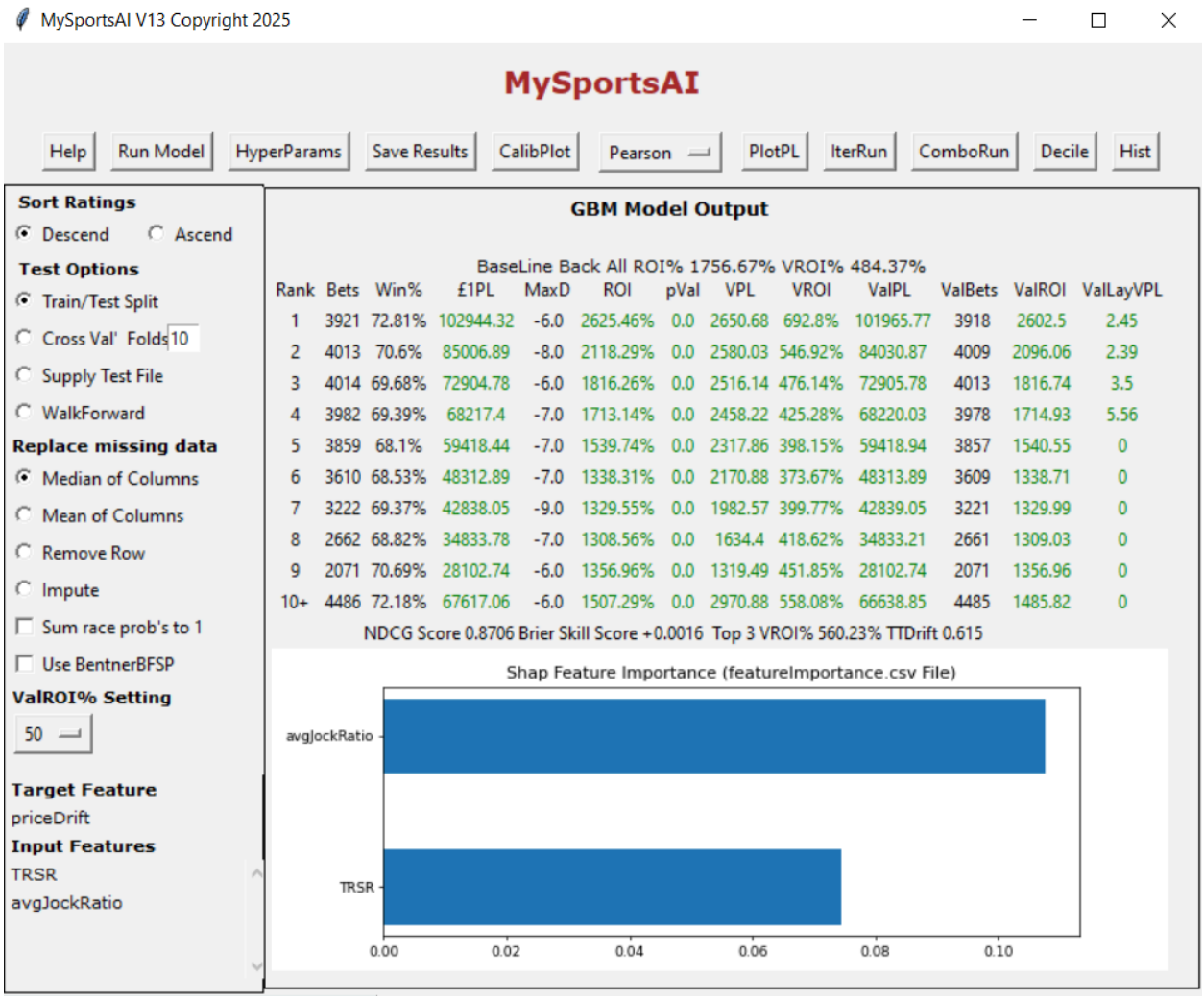

Backing horses that showed both ratings above 0.11 made over a 3% return after commission. Obviously you would want to look at future years to see if this holds up.

# Price Trading

Traders may well ask how can I use MySportsAI to trade prices. We need to employ an ML algorithm to predict price movements whether that be drifters or droppers. We have a feature called priceDrift which contains 1 if the pre-race average price is less than the BFSP. We can use this as the target to predict drifters.

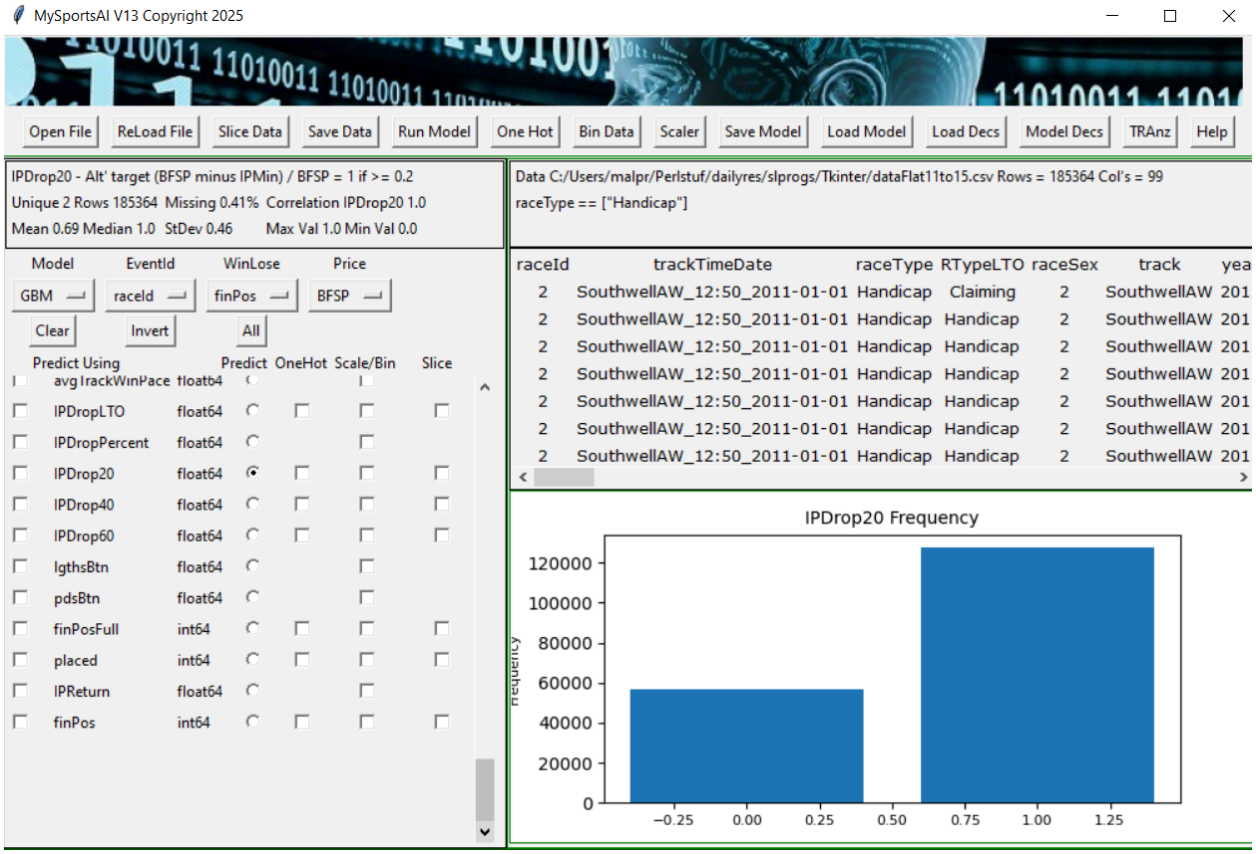



As you can see from the bar chart there are more drifters than droppers which is not too surprising given that the overround tends to come down to just over 100 before the off. Notice also that I have winLose set to priceDrift because I am not interested here in gauging how often horses win or indeed the profit or loss. Instead I am interested in how good it is at predicting drift percentage wise. Notice also that the mean for priceDrift is 0.7 or 70%. Let us see how our two input features perform when asked to predict priceDrift.



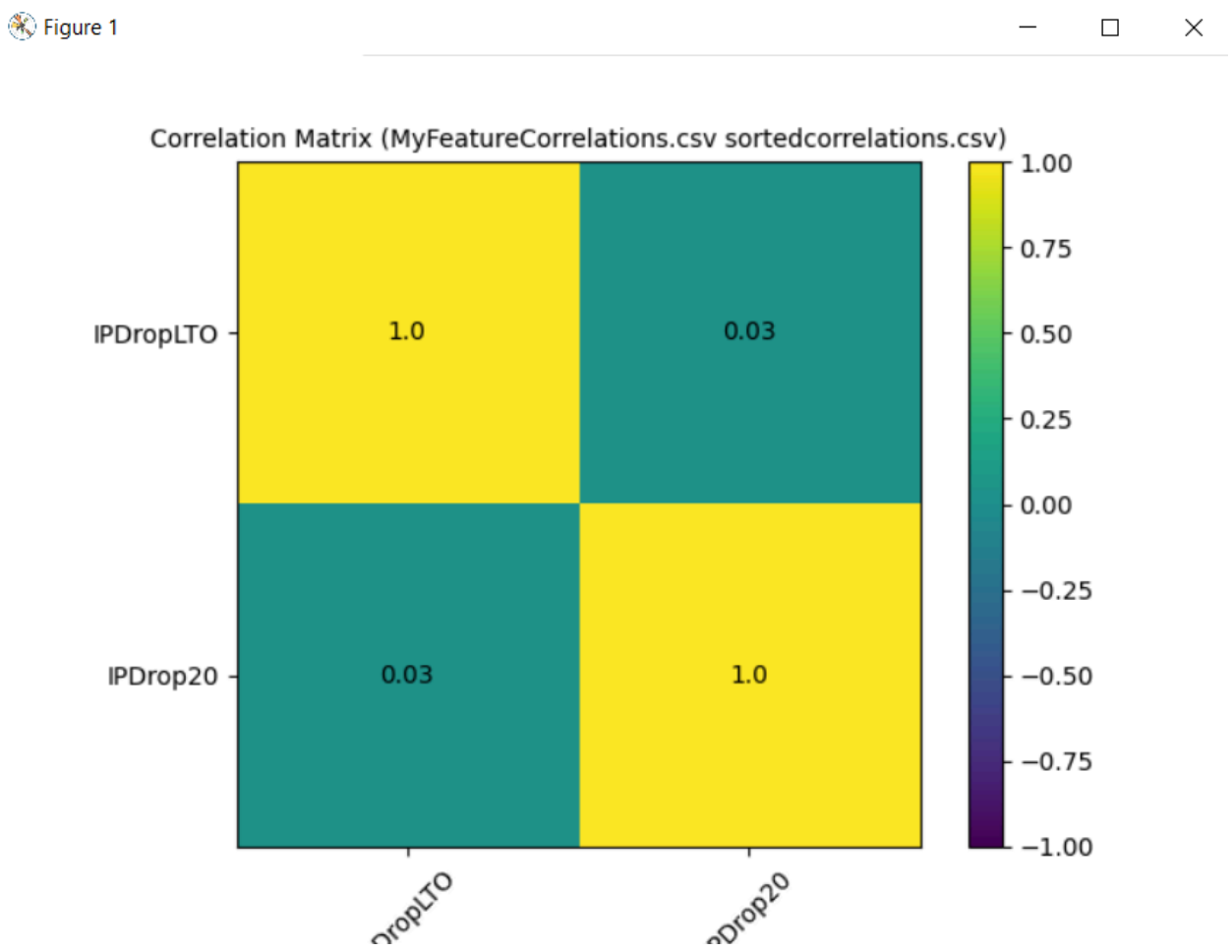

First of all ignore all the green here, this is just a by-product of setting WinLose to priceDrift. What we are really interested in is the fact that the top ranked drifted to the tune of 72.81% compared to an overall drift of 70% for all horses. I am certain this can be improved upon by selecting features that are more suitable to predicting this outcome.

What about in play trading ? We have three features we can work with here, IPDrop20, IPDrop40 and IPDrop60.

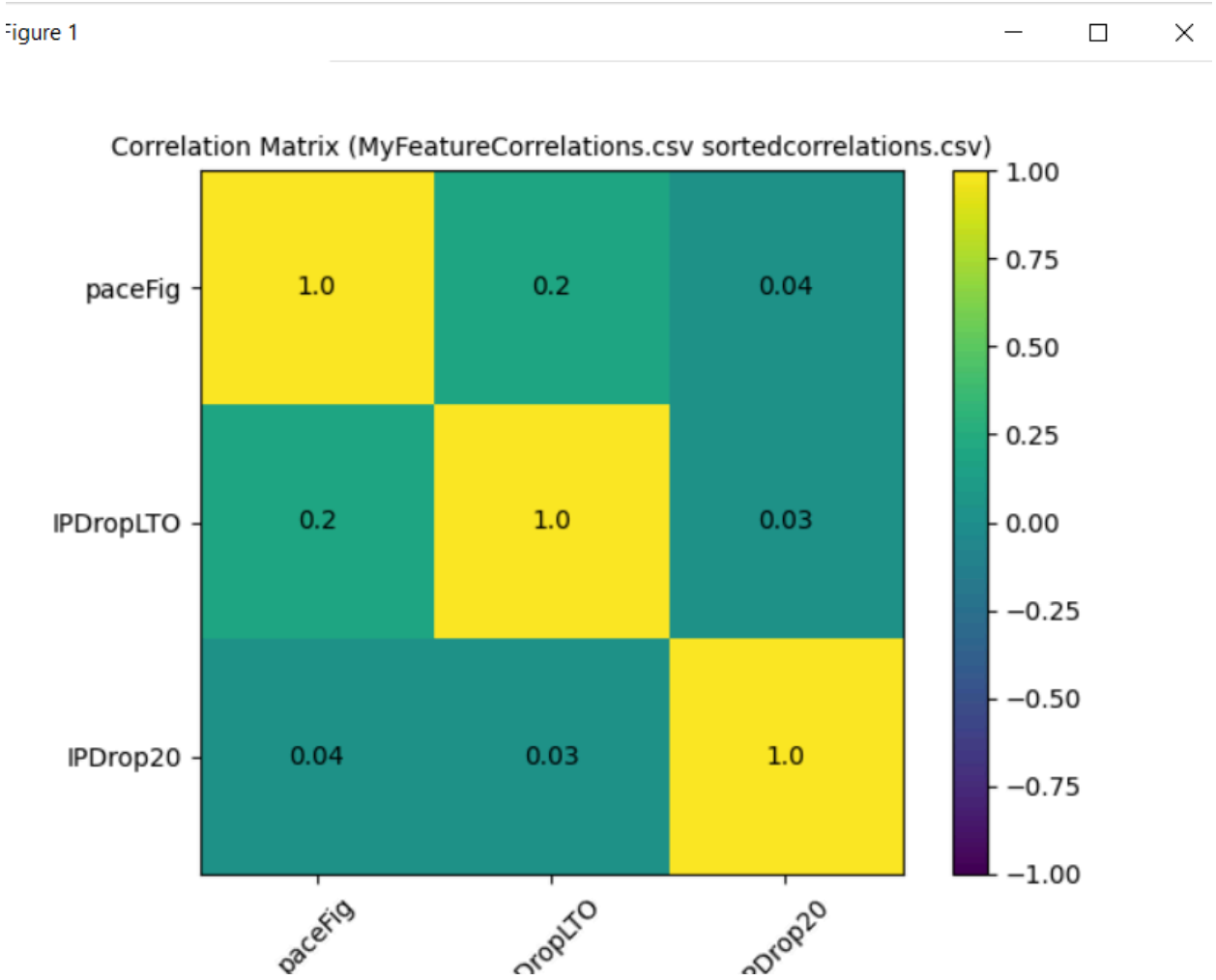



You can see from the top left of the screen that each of these three features depicts not only whether a horse went below its BFSP but by how much. If we set up IPDrop20 as a predict feature we can see which other features are good at predicting whether a horse drops in price in running. Our starting point is that on average 70% of horses have an IPDrop of 0.2 or more as shown by the mean value. Let us take a look at a few possible candidates for predicting IPDrop20.

Selecting IPDropLTO (did the horse go below its BFSP last time it ran) we can click on the Pearson Correlation option in the train/test window and see that it has a Pearson Correlation of 0.03 with IPDrop20.



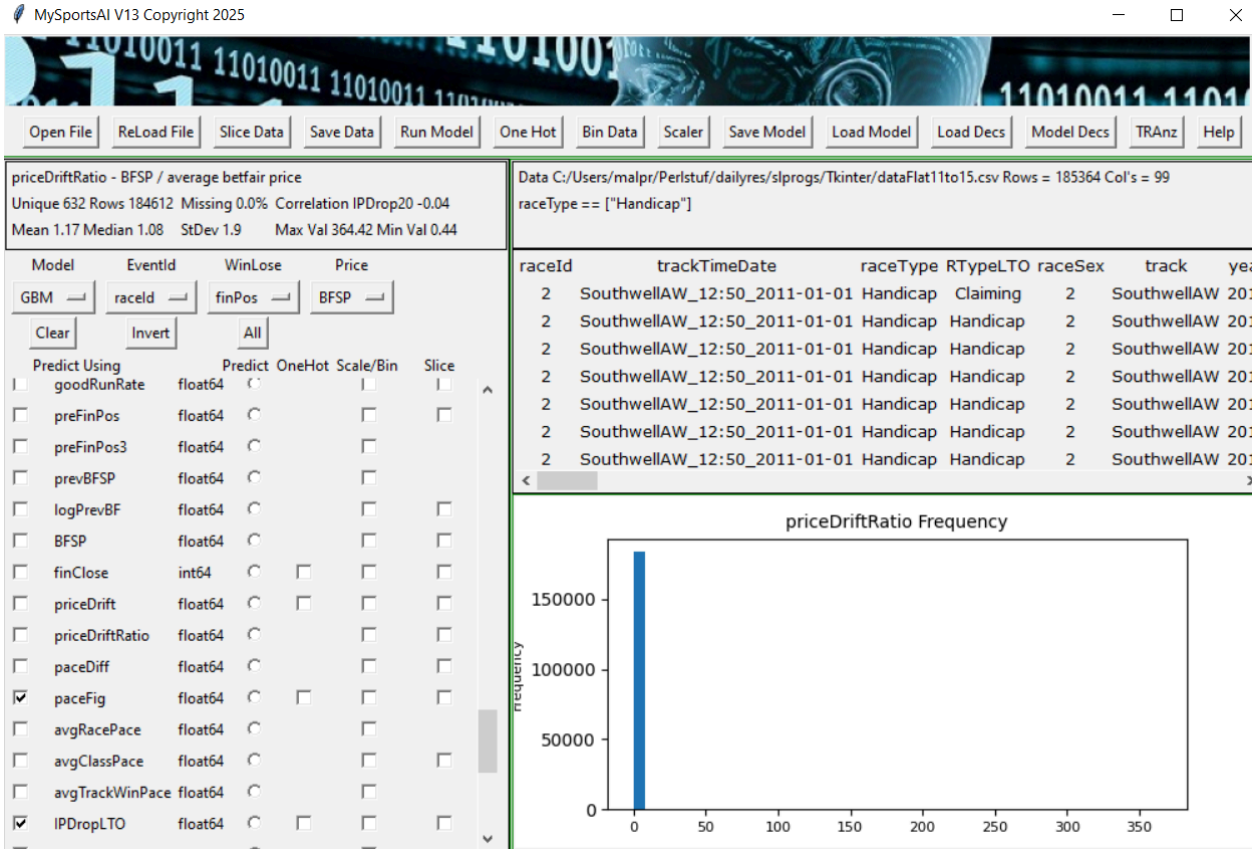

But are there better predictors of dropping in play? What part does the pace play in getting horses to drop in play after all one would think prominent runners might be better candidates for dropping in price in running.




Pace figures from the last run do seem to have a stronger correlation at 0.04. It is a positive correlation so this means the higher the pace figure the more likely a horse is to drop in running and a higher pace figure also means a prominent running style last time out.

Of course we do not have to invoke the above diagram to check out the correlation. If we simply place the cursor over any input feature we can see in the top left the correlation with the target feature, in this case IPDrop20 (screenshot below). Below we can see that priceDriftRatio has a -0.04 correlation with IPDrop20. In other words horses that drift have a negative correlation with dropping in running so we would be best advised to consider price droppers as potential in running droppers.

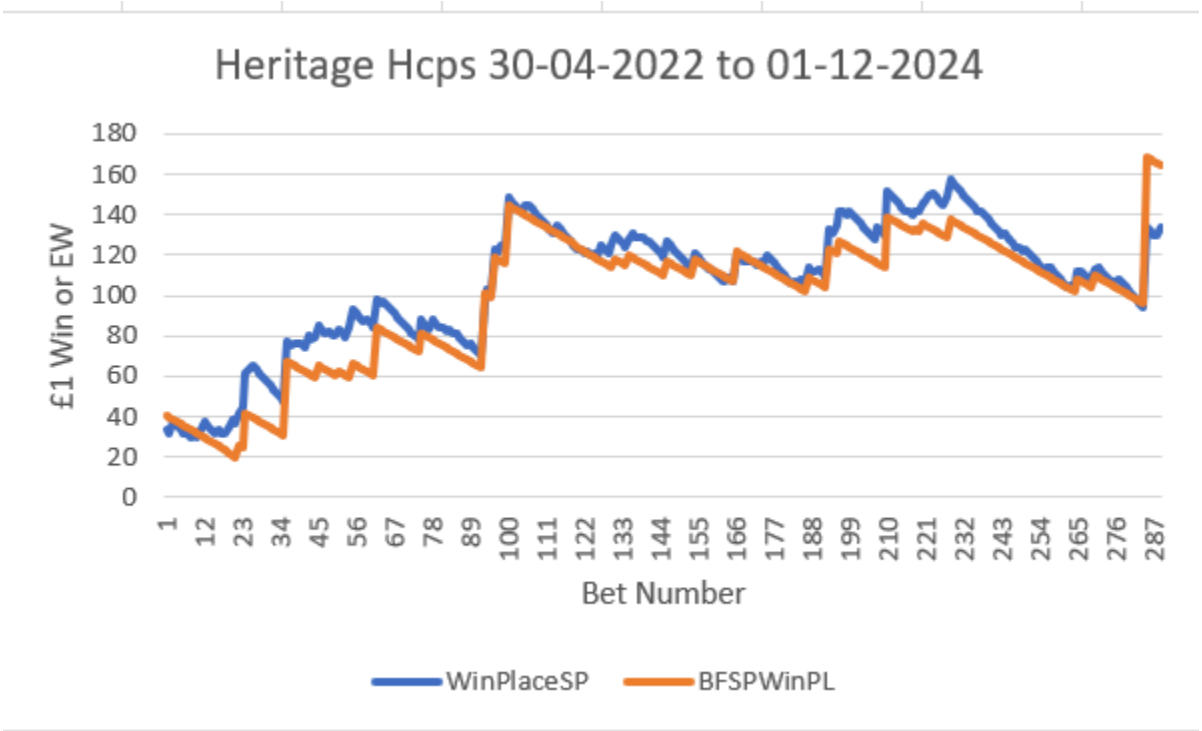
We have touched on just a few pointers here that can guide us in developing in running prediction models for traders who choose to trade in running. Any model devised for this purpose can be used to predict daily runners most likely to be trading candidates.


MySportsAI V13 Copyright 2025
Open File
ReLoad File
Slice Data
Save Data
Run Model
One Hot
Bin Data
Scaler
Save Model
Load Model
Load Decs
Model Decs
TRAnz
Help
priceDriftRatio - BFSP / average betfair price
Unique 632 Rows 184612 Missing 0.0% Correlation IPDrop20 -0.04
Mean 1.17 Median 1.08 StDev 1.9 Max Val 364.42 Min Val 0.44
Data C:/Users/malpr/Perlstuf/dailyres/slprogs/Tkinter/dataFlat11to15.csv Rows = 185364 Col's = 99
raceType == ["Handicap"]
Model
EventId
WinLose
Price
GBM
raceId
finPos
BFSP
Clear
Invert
All
Predict Using
Predict
OneHot
Scale/Bin
Slice
goodRunRate float64
preFinPos float64
preFinPos3 float64
prevBFSP float64
logPrevBF float64
BFSP float64
finClose int64
priceDrift float64
priceDriftRatio float64
paceDiff float64
paceFig float64
avgRacePace float64
avgClassPace float64
avgTrackWinPace float64
IPDropLTO float64
raceId trackTimeDate raceType RTypeLTO raceSex track
2 SouthwellAW_12:50_2011-01-01 Handicap Claiming 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
2 SouthwellAW_12:50_2011-01-01 Handicap Handicap 2 SouthwellAW
priceDriftRatio Frequency
Frequency
150000
100000
50000
0
0
50
100
150
200
250
300
350

# Is Machine Learning Betting Profitable

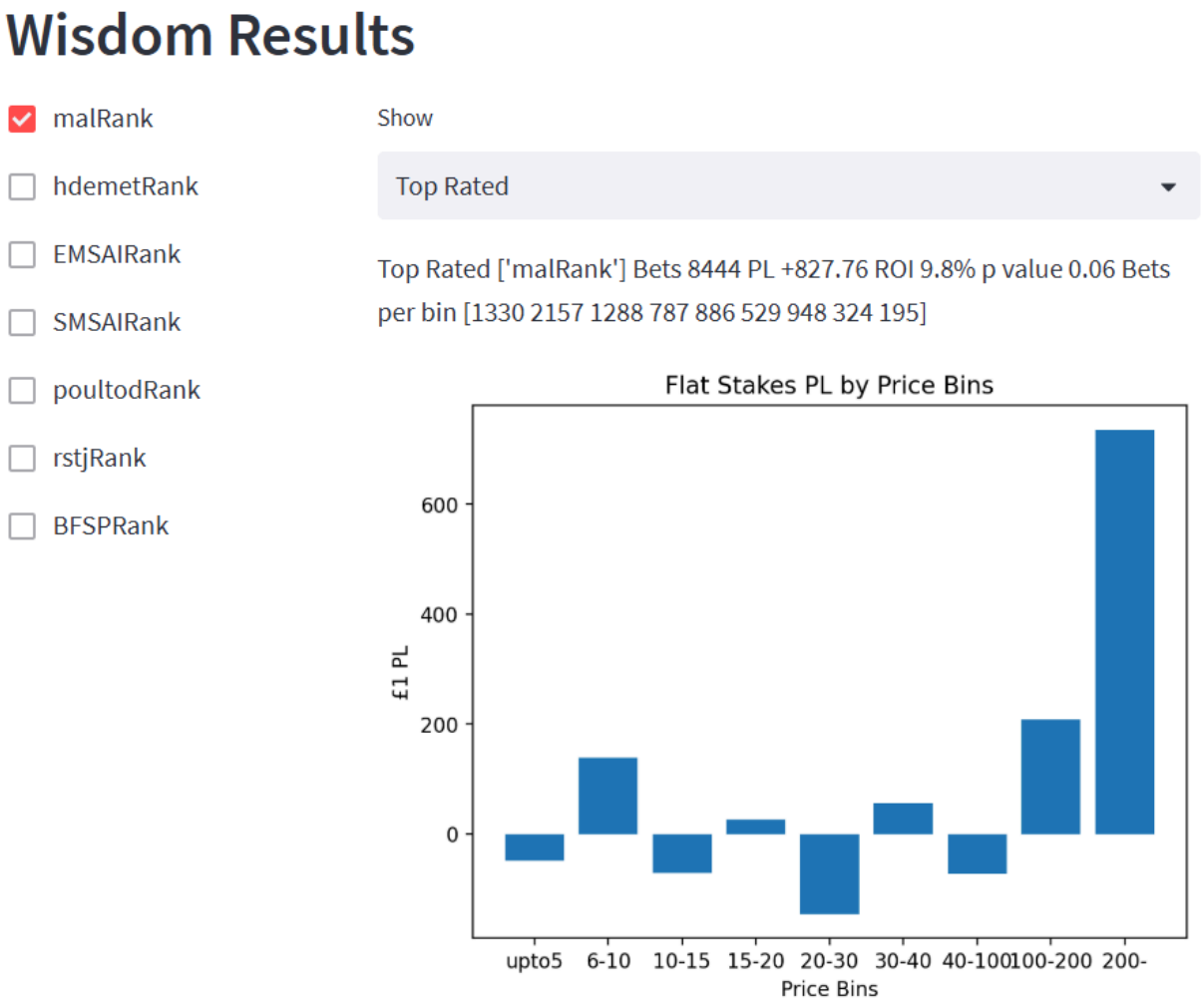
Can Machine Learning models beat the Betfair Market or indeed the bookmakers. I can cite some real world examples where MySportsAI models are indeed achieving profits. The first one is a model designed to predict Heritage Handicaps on the flat. This was in response to a Twitter based competition set up in 2022. The basic idea was that entrants had to make two predictions in all the flat heritage handicaps that year. You could specify each way or win. My model was created using MySportsAI and I tracked all its selections both to Betfair SP and to EW betting at bookmaker place offers but not counting SkyBet who always offer stand out terms beyond anyone else.

After the first year my model was laying in 7th place out of approximately 500 entrants. At the end of 2023 it had moved up to 5th place out of approximately 370 entrants that had tipped in both years. The competition folded before 2023 got under way but I continued to track it where it had another profitable year, one that would have certainly moved it up into a higher position, possibly even first place. Here is the model's performance.




There is probably an element of over performance here but the results are still suggestive of long term profit.

The next example is a collaborative project set up in 2022. Five members of MySportsAI pooled their model's ratings by downloading them to SmarterSig. One member submits two models so we have 6 sets of ratings for the day's flat racing handicaps. Each day I run some code to pick up the various sets of ratings in order to compile a master rating for each horse in each race. A kind of voting system if you will, which went under the name of The Wisdom of Models. Here are results so far for each model's top rated horse to Betfair SP after 2% commission paid as at 1/3/2025



Top Rated

Top Rated ['hdemetRank'] Bets 8128 PL +0.25 ROI 0.0% p value 0.5 Bets per bin [4249 2917 611 200 99 29 22 1 0]

Top Rated

Top Rated ['EMSAIRank'] Bets 7293 PL +368.11 ROI 5.05% p value 0.05 Bets per bin [3262 2792 699 271 167 51 43 8 0]

Top Rated

Top Rated ['SMSAIRank'] Bets 7283 PL +437.28 ROI 6.0% p value 0.04 Bets per bin [3207 2643 715 297 188 73 112 34 14]

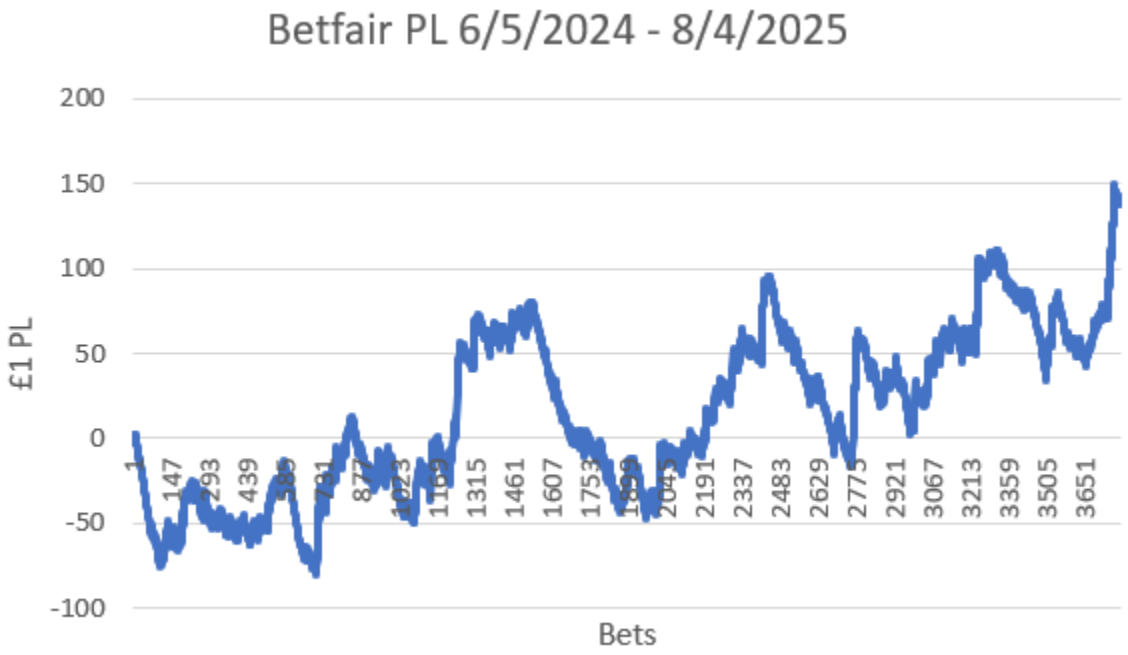
Top Rated

Top Rated ['poultodRank'] Bets 7410 PL -184.32 ROI -2.49% p value 0.24 Bets per bin [2836 2608 880 402 347 138 172 19 8]

Top Rated

Top Rated ['rstjRank'] Bets 7944 PL +342.67 ROI 4.31% p value 0.17 Bets per bin [2136 2615 1178 585 562 292 420 108 48]

Here are the results of a model I have recently been working with. It runs on all Flat Handicaps and is novel in that it is not designed to predict winners but rather to predict price droppers. The profit is to actual bets logged 10 minutes before the off time minus commission at 2% on Betfair.




For those people who prefer their own judgement on all things betting, Machine Learning can still play a part. For example consider an ML model that creates a short list of contenders in a race and from past analysis of the model you feel confident that backing them all blindly breaks even to Betfair SP. This would be a wonderful position to then use your own skill and judgement to analyse those contenders and decide if there is a bet amongst them. If you cannot convert that break even scenario into profit then you are likely not bringing any skill to the table.

Since early 2025 I have made available a free app called MySportsAILite. The app allows you to choose any combination of inputs from a choice of 6 and then create a model based on these for a single race today (the highest value handicap of the day). The screenshot below gives you an idea



# MySportsAILite

## Choose Inputs

- [x] Class Move
- [x] Days Since Last Run
- [x] Trainer Strike Rate
- [ ] Jockey Strike Rate
- [ ] Sire Strike Rate
- [x] Pace Figure

Predict

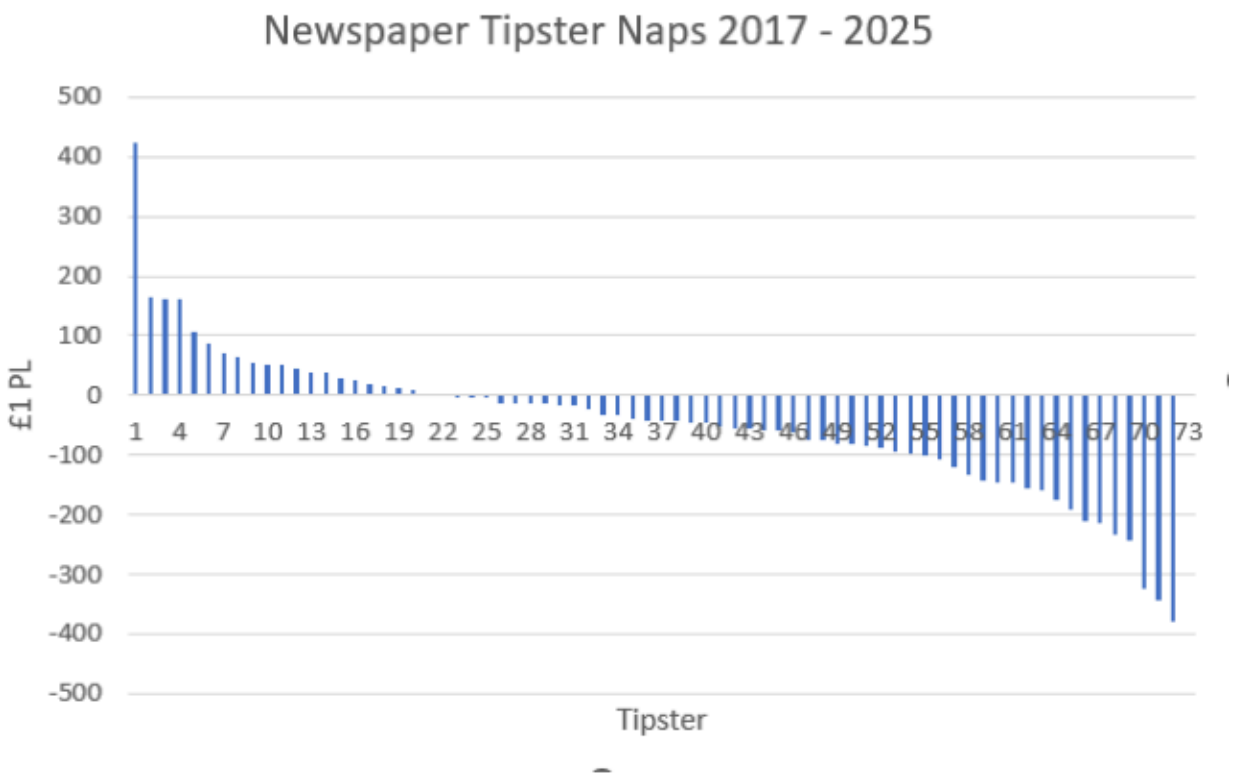
MySportsAI has over 90 predictors to choose from
www.smartersig.com/mysportsai.php

## Ratings Chep 15:20

| Horse | Rating |
|---|---|
| Next Trick (IRE) | 19.2673 |
| Outrace (IRE) | 11.6859 |
| Happy Chandler | 10.4687 |
| Dappled Light | 8.9544 |
| Royal Justice (IRE) | 8.8038 |
| Letter Of The Law (IRE) | 8.8038 |
| Glencalvie (IRE) | 8.8038 |
| Port Noir | 8.5346 |
| Johnjay | 8.358 |

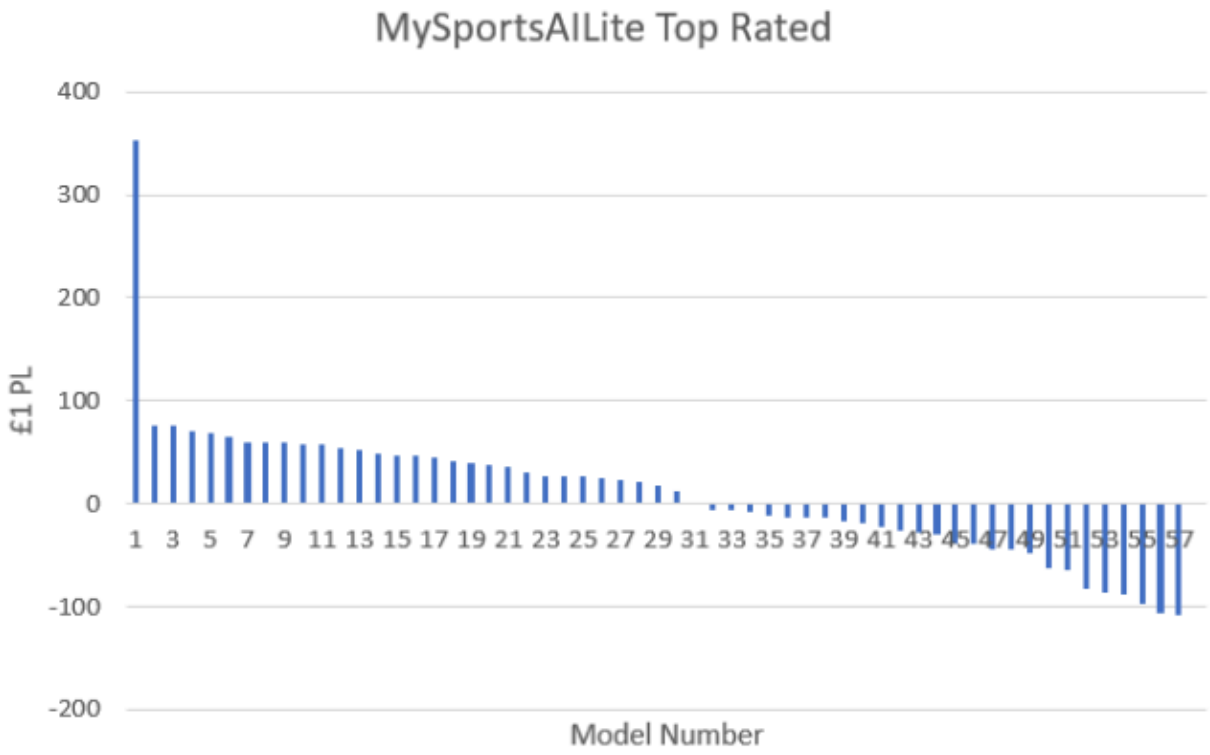
We can see from the above that choosing class move, days since last run, trainer strike rate and pace figure created a set of ratings with Next Trick top rated.

One Twitter user asked me at the beginning of May how the ratings were performing. Well let us take a look at that and compare their performance with how Newspaper tipsters perform with their naps. All profit and loss numbers are to Betfair SP.

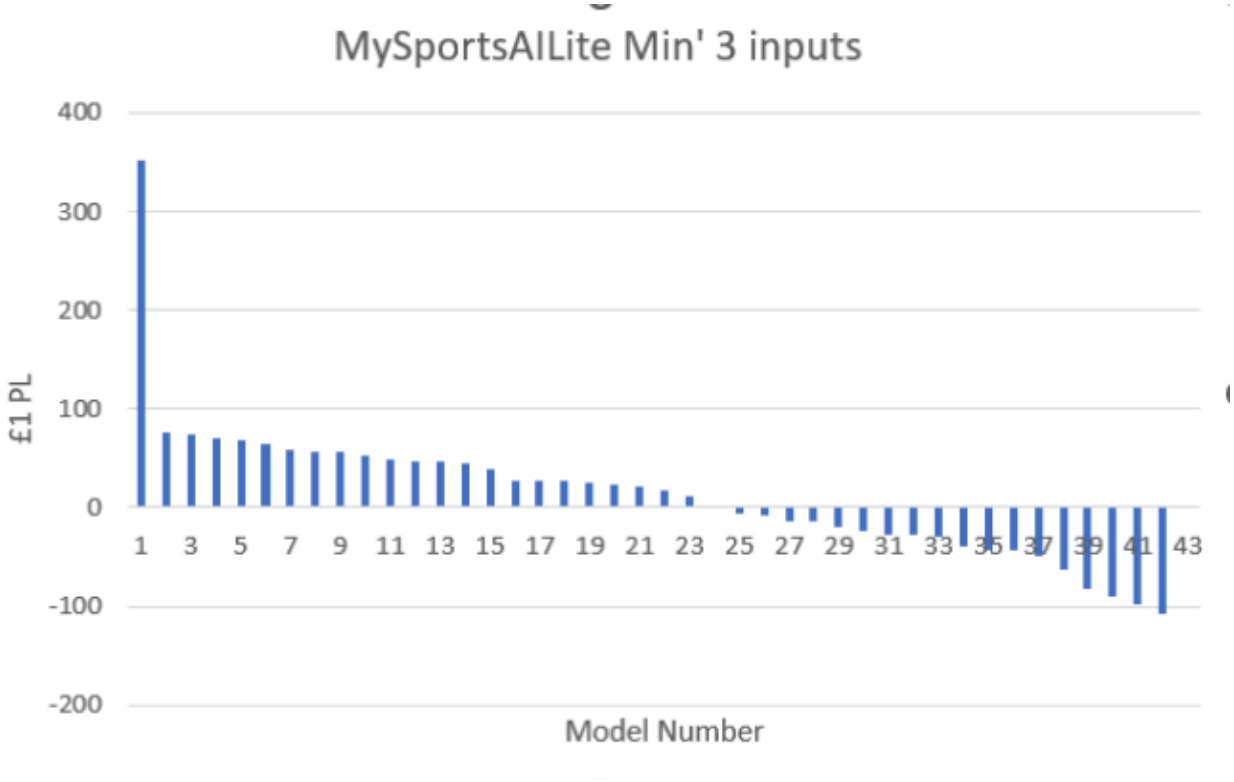



We can see that there is a skew towards a negative return. They are sorted into PL order and tipster 1 has made over £400 to a £1 stake but most of them are making a loss. In fact the average loss is -£46.75.

Now let us take a look at how the MySportsAILite top rated horse's have been doing so far.




We can see that the skew is in the favourable direction, in fact the average PL amongst all the models is +£9.25. The above shows the results for all combinations bar single inputs eg a model based solely on pace figure. If we increase the minimum input




number to 3 inputs we get further improvement.

The percentage of profitable models has gone from 52.6% to 53.4% and the average profit has risen to +£13.27.

You may now well ask which are the best combinations of inputs, well that is for you to find out by tracking or better still subscribe to the full package at http://www.smartersig.com/mysportsai.php where there are over 90 inputs to choose from instead of just 6 and you can back test the performance of your models. Another alternative is to subscribe to the MySportsAILitePay app which has the same 6 inputs but creates ratings for all the days class 4 and below handicaps.

MySportsAILite – https://mysportsaifree-4bwgag7mvr5t7utfqeiqqp.streamlit.app/

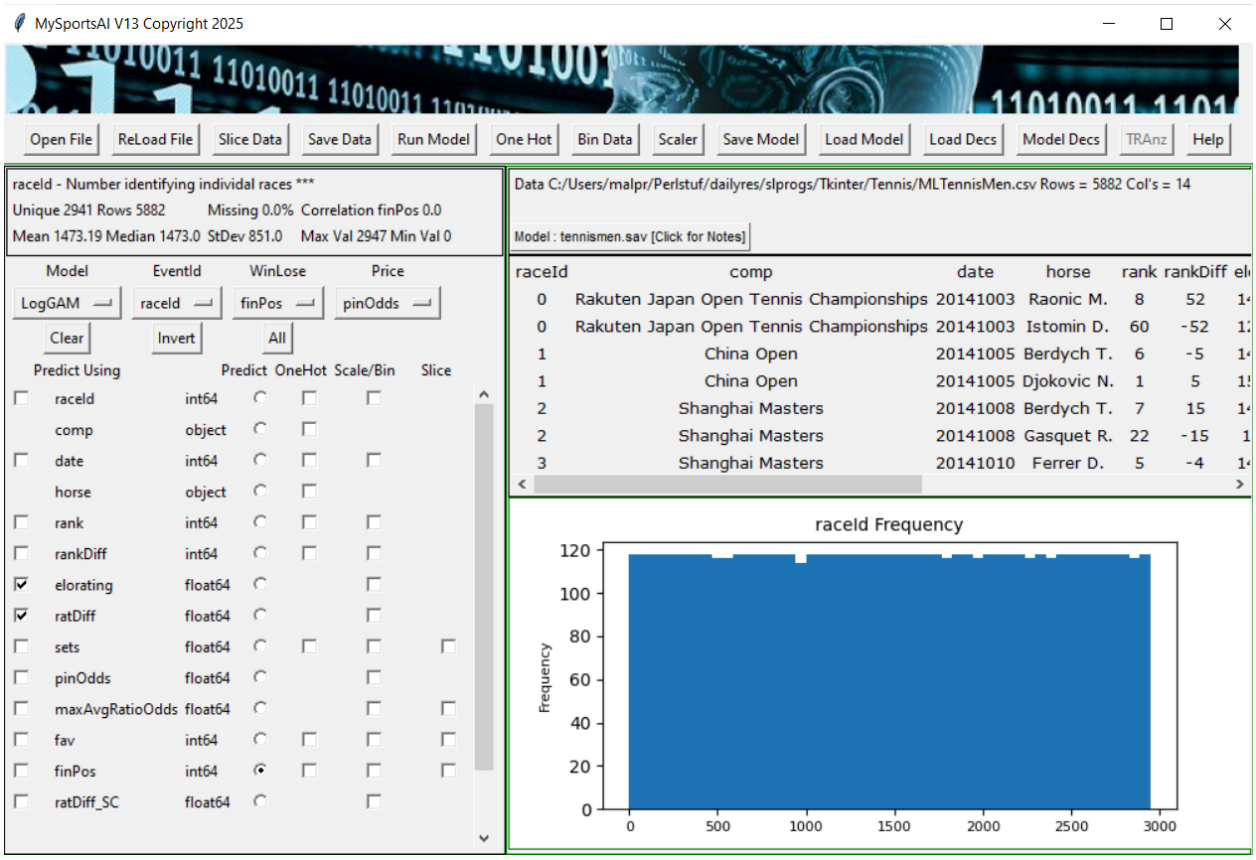
MySportsAILitePay – https://mysportsailitepay.streamlit.app/

If you still need convincing then maybe you are unaware of some of the leading betting syndicates who are employing ML methods, check out https://starlizard.com/

# Modelling Other Sports

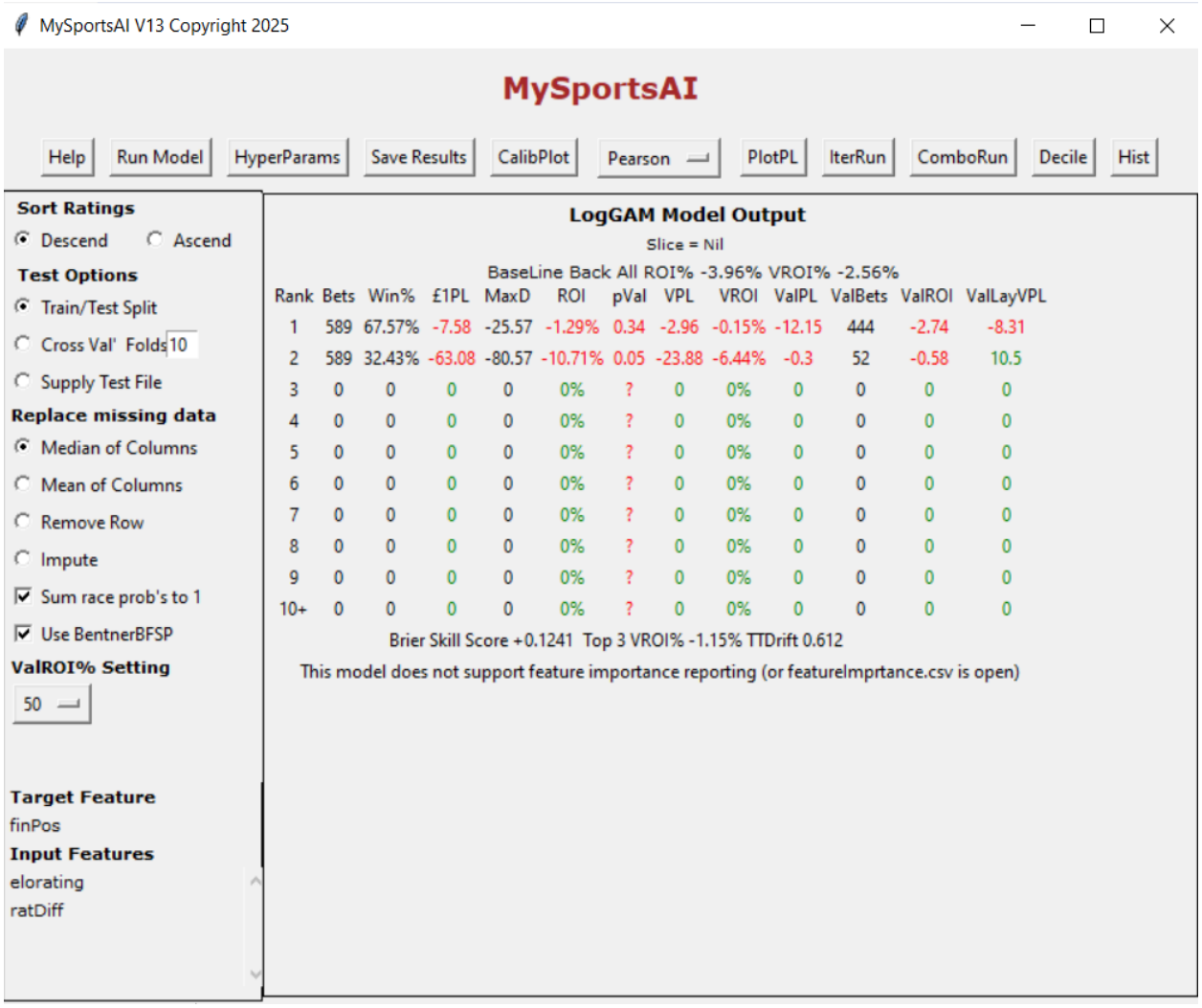
So far I have only discussed horse racing and you may well have betting interests in other sports. Can Machine Learning and MySportsAI in particular be applied to other sports such as soccer or Tennis. The answer is yes and here I will give you a flavour of how you can tackle other sports. Let us begin with Tennis.

Compared to Horse Racing tennis is a simpler problem, by that I mean there are only two competitors and only one of two outcomes. Player A wins or player B wins there is no draw to consider. There are some good data sources out on the web notably [www.tennis-data.co.uk](http://www.tennis-data.co.uk). Here is an example of some tennis data I curated from this source which I then enhanced with an ELO rating for each player.




MySportsAI is designed primarily for horse racing but as long as you adopt a few MySportsAI naming conventions you can model other sports. You may notice that I have called what is in effect the match ID raceId and each player is in a column called horse. I have also set the price to pinOdds as this is the name of the odds column which contains Pinnacle final prices. Anyone will tell you that Pinnacle is an extremely efficient

bookmaker and beating their final odds is not easy, but even this very simple model continuing input features elorating and RatDiff (difference between the two players ELO ratings) does a reasonable job of sifting out the bets.

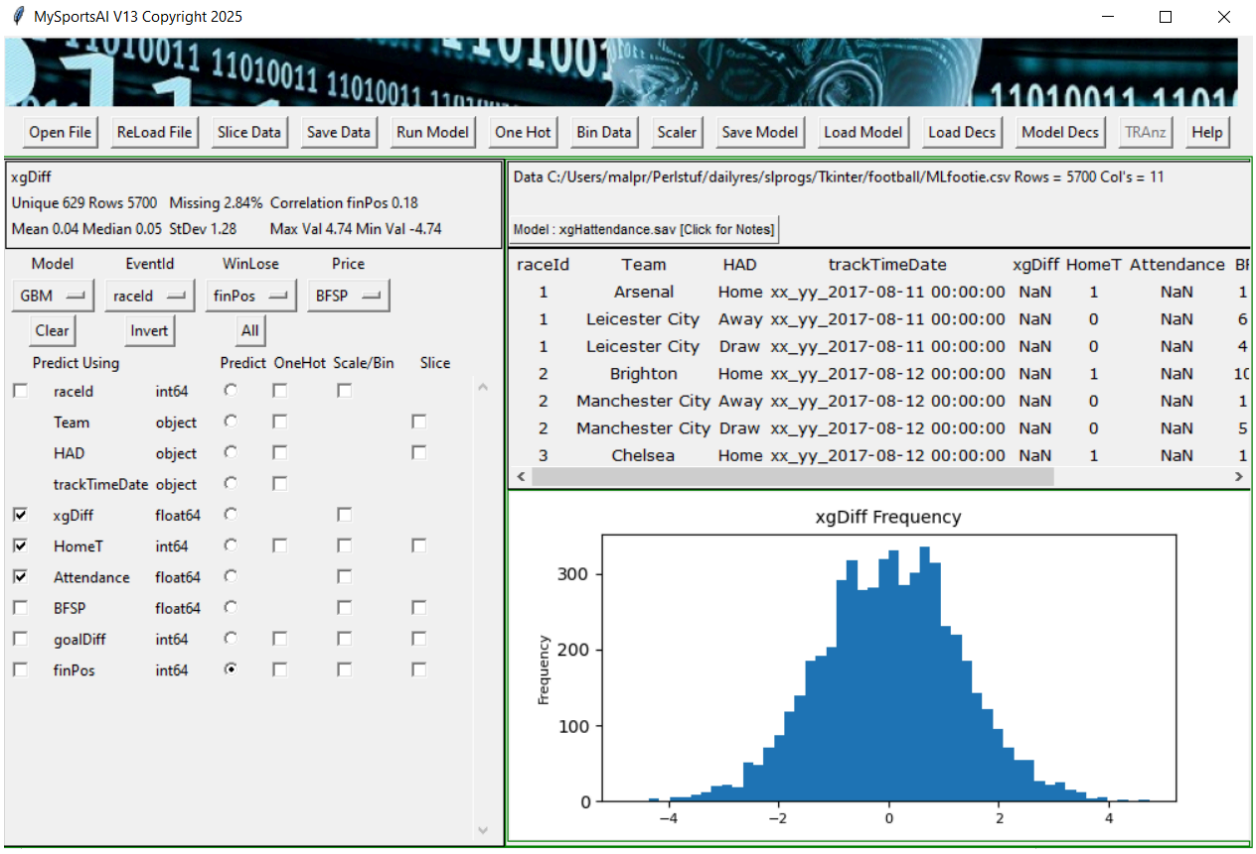



The model managed to reduce a loss of 3.96% if you just backed everything to a loss of 1.29%. No cigar but the purpose of this example was to show how we can structure data as input to a model.

Let us now take a look at Football or Soccer as the world knows it. There has been a lot of work done on modelling Soccer and one feature that has risen in prominence over the last few years is xG or expected goals to give it its full name. xG is a calculation in a game already played of how many goals a team could have expected to score. When a player attempts to score the past history of all attempts is considered in order to give a probability of scoring from that distance and that angle. So a shot from 2 meters out will

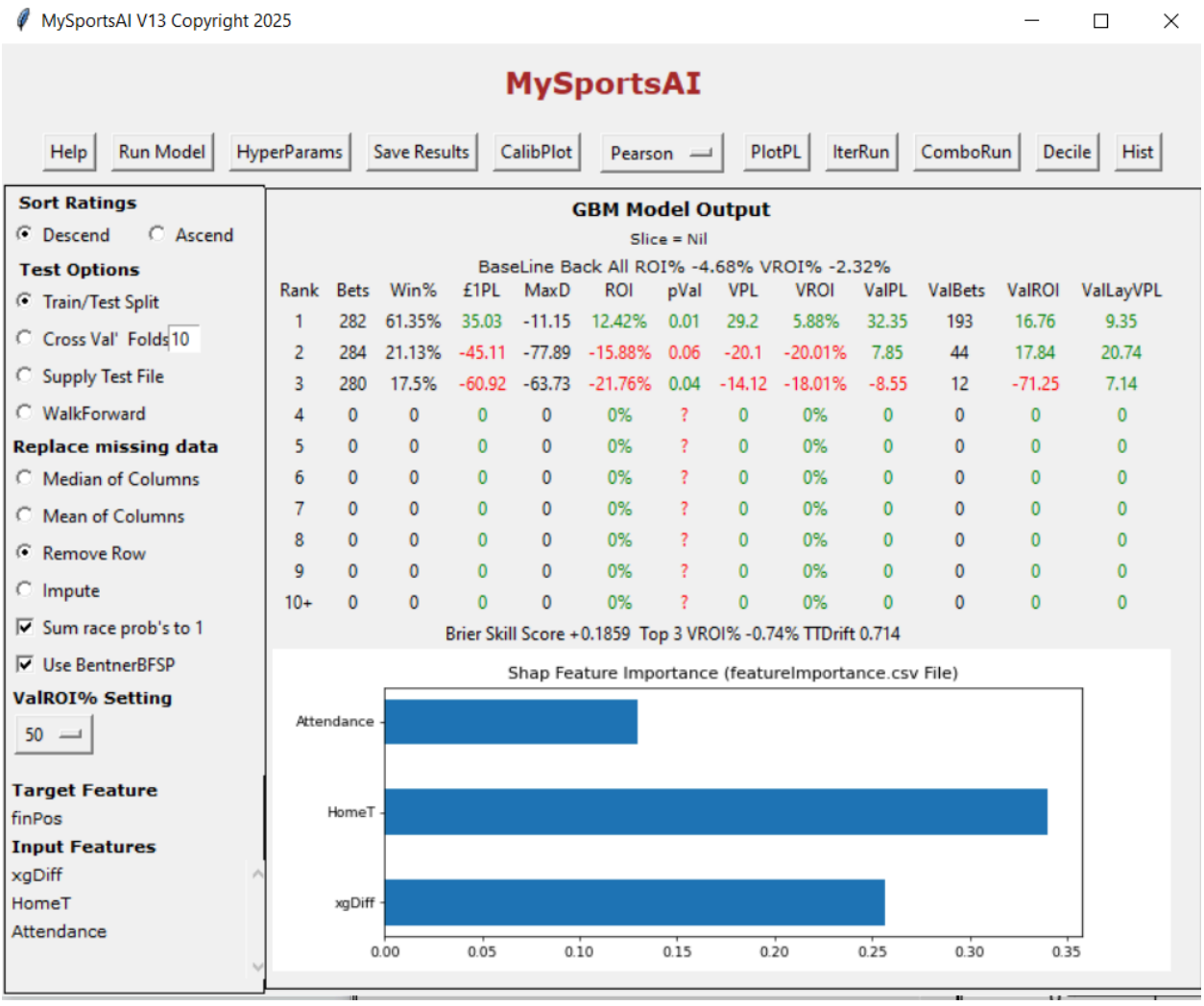
have a very high xG whilst an attempt from outside the box and out wide will have a lower xG. Using this you can calculate a teams average xG over the last 3, 5 or however many matches you wish. This can give you a variable with which to measure a team's ability, especially if taking the strength of opposition into consideration.

There are web sites that offer data on matches and include xG. I put together a data set to test the predictive power of xG and loaded this into MySportsAI.




I have set up 3 rows per game, the home teams data, the away teams data and the draw. The xgDiff feature is the difference between the home teams xG and the away teams. The first few rows of the data show NaN in the sgDiff column because I used the first 3 games of the season of each team as a bleed in (I am using the average of the last 3 games for a teams xG value).

Now using xgDiff, whether a row is a home team or not and match attendance as input features I trained and tested the model.




The model performed well on this data with the top ranked selection producing a profit 12.42% from 282 bets. This is almost certainly over performing from this sample which incidentally was Premier League games from 2017 to the end of 2022 season.

A good friend of mine Dr Alun Owen gained his PhD in soccer data modelling and he had previously mentioned that the Premier League appeared to be the best league for this area of modelling. He also mentioned that the Bundesliga was the next best and I also found this to be true.

I have only scraped the surface here on Soccer and Tennis modelling but the intention was to show that you can feed sports data into MySportsAI and create models. I have also carried out the same exploration and model building on darts and with the vast array of prop markets eg over and under 2.5 goals the potential seems endless.

# The Future

I could go on about how AI is making inroads into so many human endeavours. Some of these claims are exaggerated, others or not. Betting on sports is just a game in some ways similar to many other games. It has rules, it has variables and it involves probabilities. Machine Learning usually in the form of Deep Learning Neural Networks have conquered many complex games such as Chess and Go. Sports betting will not be immune to its influence.

There is another reason however why you should be exploring this avenue of bet creation. The bookmakers have long since put a stop to winning punters. This has been to the detriment of Racing as well as the punter. If there are no winning punters a large number of the next generation of punters will simply look somewhere else outside of horse race betting. For us it means we are forced to bet solely on the exchanges where winning punters are not banned. We can no longer shop around for a price, we have to be on the exchanges and this usually means during the ten minutes before the off time of a race. In order to generate meaningful profits we must go down the path of increased turnover and lower ROI% returns. This is not as bad as it sounds after all a 3% return on 10,000 bets equates to a profit of 300 points whilst a ROI% of 25% on 300 bets a year gives you 75 points profit. I know which I would rather have.

This kind of turnover requires automation. A pro' punter like Alan Potts can happily create 300 bets a year and make 25% to bookmaker prices, but if I asked him to create 10,000 bets his head would explode. It also requires the automation of bet placing, something that is beyond the scope of this book.

The other advantage to bet automation is the built in discipline that it brings to your betting. It is terribly difficult to bet with personal interpretation, form reading if you will, and remain objective. Your view of today's racing may be different simply because you slept badly last night, let alone because you are on the mother of all losing runs. Machine Learning does not suffer from those influences. The bets it produces today will be the same regardless of whether the Mother in Law is coming round for tea or not.

Hopefully this book has given you an insight into some of the principles of Machine Learning and the terminology surrounding it. You hopefully have a clear idea what Machine Learning is and how it can be used for betting without having got bogged down in the task of learning how to code a Machine Learning model in Python or R.

If you want to learn more become a MySportsAI member and join the friendly email forum, there is even a 'selections' sub forum where members post up selections they

have produced and backed so that others can avail themselves of any remaining juice in a bet.

Membership can be attained via http://www.smartersig.com/membership.php

# Appendix A

INPUT FEATURES

*** required with same name or other name if user supplying own data

raceId - Number identifying individual races ***
trackTimeDate - Useful for those that want to merge in own data
raceType - type of race eg Handicap
RTypeLTO - race type last time out
raceSex - Male only = 1 mixed = 2 female only = 3
track - track
year - year of race
month - month of year
week - week of year
raceClass - race class
raceDist - race distance
surface - Race surface eg turf
going - race going
raceValue - race win purse / 1000
draw - stall draw
drawValue - Value of draw at track and dist
runners - Number of runners in race
horse - horse name
weightDiff - top weight minus horse weight
OR - Official Rating
bestORDiff - Best win OR (turf/AW) minus todays OR
ORDiff - Difference between top OR in race
nat - horse nationality
avgBeat - Lifetime horses in front / horses behind
goingDiff - 1 means yes 0 means no from last race
goodRunGoing - On todays going previous good runs / runs
classMove - race class minus last race class
weightDiffLto - todays weight minus weight LTO
runnersPrev - number of runners in previous race
runnersRatio - runners in previous race / runners today
hgToday - wearing headgear in current race 1 = yes 0 = no
chgHg - change of headgear from last race 1 = yes 0 = no
hgFirst - First time headgear 1 = yes 0 = no
hgChanges - Count of headgear changes from all prev' runs
weight - horse weight / topweight

pdsBtnLto - how far beaten in last race in pounds
daysLto - days since last run
daySinceGR - days since last good run
prevLto - days between last run and previous run
age - horse age
prevRaces - Number of previous races for code ie Flat
exposure - Number of previous races / race average
hasPlacedTrack - has placed form at the track
trainer - trainer name
trSwitches - horse lifetime trainer switches
TRSR - Trainer strike rate at time of race (min 50 runs)
TRinrace - trainer strike rate (SR) / top TRSR in race
TRTrackPCent - trainer track wins / trainer track runs
TRrolling - rolling win & place SR, sample based on trainer runs
TRageRatio - SR% with this horse age / TR overall SR%
TRPrizeRatio - Trainer average win prize money / race prize
JockSR - Jockey SR at time of race (min 50 runs)
avgJockSR - Average Jock SR derived from all previous runs
avgJockRatio - ratio of this jock SR to all previous average SRs
jockRatio - this jock strike rate / previous jock strike rate
Jockinrace - Jock strike rate / top Jocksr in race
JockRolling - rolling win and place SR, roll based on jockey runs
SireSR - Sire strike rate at time of race
Sireinrace - sireSR / top sireSR in race
SireDistAge - % of placed runs at distance by sire for 2yo, 3yo+
SireSurf - placed runs on surface turf fibre poly / surf runs
DamSireSR - Dam sire strike rate at time of race
DamSireSurf - placed runs on surface turf fibre poly / surf runs
Sex - sex of horse eg g c f m
SexMF - sex M male F female
beatLTO1 - LTO runner count minus finPos / runner count
beatLTO2 - as above for second last run
beatLTO3 - as above for third last run
entries - number of 5 days entries for horse
jockSame - 1 if same jock to previous run 0 if not
jockMinus - 1 if jock SR% is less than last run jockey
distUp - 1 if race dist greater than last race dist 0 if not
distSame - 1 if same race dist to previous run 0 if not
goodRunRate - ratio of good runs to runs
preFinPos - All previous fin positions / Previous opponents

preFinPos3 - Last 3 fin positions / last 3 oppponents
prevBFSP - BFSP in last race
logPrevBF - natural log of chance according to last BFSP
BFSP - Betfair SP in this race ***
finClose - SeeHelp 1 if horse won or < rDist/10 lgths of winner
  can be used as alternative to finPos as target to give more balanced data
  do not use as an input feature
priceDrift - 1 if pre race average betfair price < BFSP
priceDriftRatio - BFSP / average betfair price
paceDiff - pace figure from last race minus pace from pelutimate
paceFig - RunStyle from last flat race 0 heldUp to 5.6 madeAll
avgRacePace - average race pace figure for this race
avgClassPace - average winning pace fig for handicap class
avgTrackWinPace - average track winn pace fig for handicaps
IPDropLTO - 1 if horse dropped below its BFSP in running LTO
IPDropPercent - % of horse IP Min < BFSP all prev races
dropLast3Avg - average of last 3 runs in running drop %
IPDrop20 - Alt' target (BFSP minus IPMin) / BFSP = 1 if >= 0.2
IPDrop40 - Alt' target (BFSP minus IPMin) / BFSP = 1 if >= 0.4
IPDrop60 - Alt' target (BFSP minus IPMin) / BFSP = 1 if >= 0.6
lgthsBtn - lengths behind winner or zero if won this race
pdsBtn - pounds behind winner,alternative target feature
finPosFull - full numeric finishing position
placed - 1 means horse placed,alternative target feature
IPReturn - finPos minus (1/BFSP) alternative target feat
finPos - 1 for winner 0 for loser ***

# Appendix B

ALGORITHM ABBREVIATIONS

GBM - Gradient Boosting Machine (classification)
LightGBM - LightGBM (classification)
NeuralNet - Neural Network (classification)
RandForest - Random Forest (classification)
RForestReg - Random Forest (regression)
HuberReg - Huber Regression (regression)
KNN - K Nearest Neighbor (classification)
AdaB - AdaBoost (classification)
LogRegr - Logistic Regression
NBay - Naive Bayes (classification)
GBMreg - Gradient Boosting Regression
LogGAM - Logistic Generative Model (classification)
LinGAM - Linear Generative Model (Regression)
XGBoost - Extreme Gradient Boosting (classification)
XGBReg - Extreme Gradient Boosting Regression
Voting - Combined ensemble voting predictions from your choice of models
histGBM - GBM classifier that bins feature values, faster than GBM
RidgeReg - Ridge Regression (regression)