R is For Racing

Colin Magee

January 2019


3

Overview

- Introduction, context and history
- Data for programmatic horseracing analysis
- Exploring horseracing data
- Building models with historic horseracing data
- Using daily data to find profitable predictions
- Using abettor to place bets into Betfair
- What next? Avoiding gambler's ruin!

About

- Horseracing enthusiast, punter, sometime owner
  - Currently member of Horseracing Bettors' Forum http://ukhbf.org/about/ supported by the BHA
- Careerwise, exec in software and data businesses
  - Currently on board at Doorda www.doorda.com
- Set up www.betwise.co.uk in 2007
  - Initially to support book Automatic Exchange Betting – first book on automating betting process via Betfair API
  - Created Smartform for horseracing / data enthusiasts
  - Currently working on *R is for Racing* with Jay Emerson http://www.stat.yale.edu/~jay/

Context

- Horseracing is in many ways the archetypal predictive analytics problem
  - Constant series of experiments (races)
  - Bookmakers and punters have always used information to assign probabilities of success to each contender
- No surprise either that the history of betting on horseracing is closely intertwined with the roots of computer science, programming and statistics…

Babbage – designed first computer?

Lovelace – first programmer?

But did you know….?

From the 1840s, Ada turned her prodigious talents toward gambling and programming the outcomes of horse races. A mysterious “book” was passed between Lovelace and Babbage once a week that probably contained a program designed to predict horse-race results.

On May 17, 1850, Ada wrote to her mother:

“I am afraid you will take no interest in what interests me much just now,—viz: the winner of the Derby.”

Markus, Julia. Lady Byron and Her Daughters

Apparently, Ada lost **£3,200*** betting on the Derby.

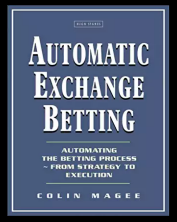
c. **£425,000 in 2019*

I have no idea what they were up to

- There was very little data as we understand it – possibly they were trying to calculate the effects of weight in slowing down horses of different ages over different distances, but who knows?
- What is true is that in this century one might *assume* we are in a much better position to master this domain by applying a classic data science approach – with modern computing tools, domain knowledge, useful data and modeling expertise

…Modern Opportunities and Challenges

- Betting exchanges – given API access - indeed provide opportunity to be quantitative in betting analysis and execution
  - Wrote Automatic Exchange Betting in 2007 about this



- More challenging was the 'fundamental data' situation to understand the domain, create models and understand what to bet on
  - Licensing situation is expensive (largely B2B) and restrictive
  - No programmable data source – lots of websites or GUI driven databases relied on webscraping or manual processes for daily data extraction

- So … for convenience we created Smartform, an end user licensed, programmable database for horseracing for enthusiasts and data scientists alike

Smartform

- https://www.betwise.co.uk/smartform
  - MySQL database
  - 15 years of horseracing history in UK and Ireland, updated daily with results and upcoming races
  - Easy to automate, import and connect from most any language
- R is a natural choice for analysis environment
  - Minimize interation with MySQL and get straight to data exploration, analysis, modeling etc
  - And packages available for Betfair API etc.

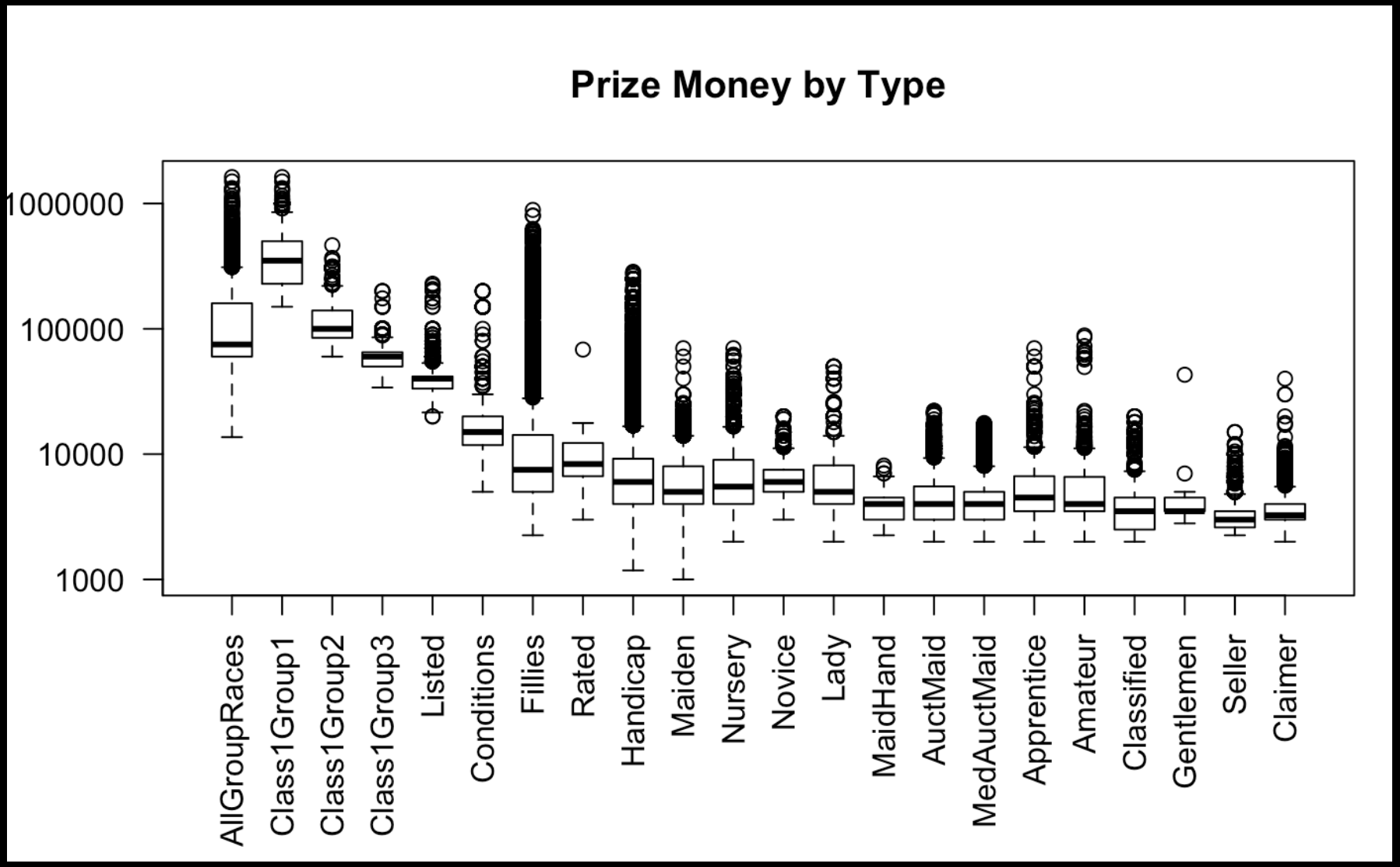
```
# Connect to database, easy with RMySQL
con  <-  dbConnect(MySQL(),
             host='127.0.0.1',
             user='name',
             dbname='smartform')

# Database is simple, flat structure – non-normalised – making it easy to slurp into R
> dbListTables(con)
[1] "daily_races"                    "daily_runners"
[3] "historic_races"                 "historic_runners"
#….Plus some extra tables for betfair historic prices and easy mapping of entities (more later) ….

# For data analysis and modeling we only really care about historic tables – race_id is common key
query <- paste("select * from historic_races;", sep="");   # All races
x  <-  as.data.frame(dbGetQuery(con, query), stringsAsFactors = FALSE)
dim(x)
191867  40
query2 <- paste("select * from historic_runners;", sep="")  # All runners in those races
y  <-  as.data.frame(dbGetQuery(con, query2), stringsAsFactors = FALSE)
dim(y)
2140306  66
```

Exploring horseracing data

- So, we have about 2 million horses (c. 162k are unique) spread amongst 200,000 races, with results for each race
- Despite look of data being “flat”, the time / history element of racing and interdependent entities (eg. races, horses, sires, dams, trainers, jockeys, owners, courses etc) enables us to create thousands of derived dimensions for deeper modelling
- But data can still be messy: Split by jurisdiction, split within jurisdiction, from different sources: Vigilance is required!
- Let’s look at some preprocessing / exploration etc. (Demo)



Feature engineering overview

- Basic clean up tasks – removing non-runners etc.
- We can do a lot more with deriving fields from existing fields (eg. regex on race_name for races)
- Second, useful binning (eg. distance, going, season)
- Last but not least, any number of variables based on representing entity history (up to - but not including - each running date) in one row
  - Horses, Sires, Trainers, Jockeys, Owners, Stall Number
  - Dimensions on dimensions– eg. Trainer/ Jockey combos

– Data prep can be a computational nightmare

Models with horseracing data

- In all cases, predicting a winner is not actually the point, or even doing it accurately
- We are simply trying to beat the market

A couple of demo model approaches

- Pick one of our favourite features and look for profitable angles (old fashioned punters approach)
- A more comprehensive statistical model trained on past data and applied to new data

Model #1– finding a profitable angle

- Some trainers have good strike rates (#winners / #losers), some not so much
- All have differing specialities, but the market tends to focus on average trainer strike (if using it at all)
- We can use data analysis in R to find niches (using the new features we created for seasonal performance) where trainers outperform their average form
- Next, and more important, we can assess whether the market generally factors in their chances – or not
- Automating this process throws up many candidates (script)

Betting on the model

- This is simple enough to be good for showing off the live selection process and using *abettor*
  - https://github.com/phillc73/abettor/blob/master/vignettes/abettor-placeBet.Rmd
- Import all races and runners from daily races and daily runners in Smartform
- Find all horses running today by merging against our trainer hotlist
- Use the convenience daily betfair tables supplied in Smartform to look up betfair references
- Loop through each selection, and invoke abettor to bet at current market prices

Model #2– a real model!

- Many approaches to predicting outcome
  - Binomial (winner or bust)
  - Multinomial (win, show, place, lose)
  - Continuous regression (transform finish position to a point along an interval)
- Model types: R has a few - take your pick!
- With this model, Jay applies classic regression, based on continuous response variable

About the (Linear) Model

- Response: a transformation of the finishing position (1, 2, …, k for a k-horse race) to the unit interval [0,1].
- Predictors (examples):
  - The horse's past win percentage
  - The jockey's win percentage in the last month and over the last year (i.e. is the jockey "in form")
  - The trainer's win percentage in the last month…
  - The stall position, or "effect of the draw"
  - …

Disclaimers (from Jay)!

- We violate the independence assumption
- We violate the “normal errors” assumption
- … but …
- In a practical sense, it doesn’t matter much -- Yes, a little secret of real-world data analysis!
- We could spend a lot of time worrying about this and trying to improve it… and the payoff would be negligible.

Notes on Prediction (from Jay)

- Predicting win probabilities done via simulation. Why?
  - Because the model predicted response is like an unobserved "strength of performance" (latent variable)… while we're interested in rankings: Who wins? Who comes in second, etc…
  - … and of course there is a huge amount of variability (the "errors") in real-world problems like this one!
  - And it's really easy to pull off without making serious mistakes.

Predictions!? January 15, 2019: Upcoming Kempton Race – 7:45 pm

Name	Predicted Finish	Predicted Prob(Win)	Predicted Prob(2nd)	Predicted Prob(3rd)	Predicted Dec Odds
Six Strings	1	21.9	17.6	14.8	4.6
Mr Gent	2	17.4	15.7	14.4	5.7
In The Red	3	16.7	15.3	14.3	6
Storm Melody	4	11.6	12.3	12.7	8.6
Mount Wellington	5	10.7	11.9	12.4	9.4
Pearl Spectre	6	9.5	11	12	10.5
Wilson	7	6.7	8.6	10.2	14.9
Blue Candy	8	5.4	7.6	9.2	18.4

What next? Gambler's Ruin

- In betting, most wagers have an overall losing expectation – even if we are beating the market per se
- A player with finite resources will inevitably go broke unless there is careful:
  - Risk management
  - Bank management
- The objective is to optimise the size of the stake relative to the winning expectation in the bet and the size of the bank.
- What's needed? Copious backtesting and a staking strategy
- Plenty of strategies such as Kelly Criterion, and packages from our friends in Rfinance that can help , eg
  - *PerformanceAnalytics*

Thanks!

- Look out for ‘R is for Racing’ – sometime in 2019!
  - Blog: https://blog.betwise.net/
  - Q&A: https://answers.betwise.net/
  - Contact: https://www.betwise.co.uk/contact