Horse Racing Data Analysis Project - ML1 HS24

[sindhu.chinnamsetti@stud.hslu.ch](mailto:sindhu.chinnamsetti@stud.hslu.ch), [ronald.schwalb@stud.hslu.ch](mailto:ronald.schwalb@stud.hslu.ch), [eniko.simo@stud.hslu.ch](mailto:eniko.simo@stud.hslu.ch)

October 13th, 2024

# 1) Data Cleaning

WIP

# Import the Data from CSV Files  
df1 <- read.csv("Lingfield\_AW\_2014\_2024\_flat.csv")  
df2 <- read.csv("Lingfield\_AW\_2014\_2024\_jumps.csv")  
df3 <- read.csv("Lingfield\_turf\_2014\_2024\_flat.csv")  
df4 <- read.csv("Lingfield\_turf\_2014\_2024\_jumps.csv")  
  
df.horse <- rbind(df1, df2, df3, df4)

* gotta diminish dataset size to:
* max 10^5 = 100,000 rows
* 10-20 predictors

Keep the following columns based on domain knowledge and careful inspection of the dataset:

df.horse <- subset(df.horse, select=c(date, race\_name,type, class, dist\_m, going, ran, pos, draw, horse, age, sex, lbs, hg, secs, jockey, trainer, prize))

Create a new column ‘won’, assigning 1 if ‘pos’ is 1, otherwise 0

df.horse$won <- ifelse(df.horse$pos == 1, 1, 0)  
# Check the first few rows to ensure the new column has been added  
head(df.horse[c("pos","won")],50)

## pos won  
## 1 1 1  
## 2 2 0  
## 3 3 0  
## 4 4 0  
## 5 5 0  
## 6 6 0  
## 7 7 0  
## 8 8 0  
## 9 9 0  
## 10 10 0  
## 11 11 0  
## 12 1 1  
## 13 2 0  
## 14 3 0  
## 15 4 0  
## 16 5 0  
## 17 6 0  
## 18 7 0  
## 19 8 0  
## 20 1 1  
## 21 2 0  
## 22 3 0  
## 23 4 0  
## 24 5 0  
## 25 6 0  
## 26 7 0  
## 27 8 0  
## 28 9 0  
## 29 10 0  
## 30 11 0  
## 31 12 0  
## 32 13 0  
## 33 1 1  
## 34 2 0  
## 35 3 0  
## 36 4 0  
## 37 5 0  
## 38 6 0  
## 39 7 0  
## 40 8 0  
## 41 9 0  
## 42 10 0  
## 43 11 0  
## 44 1 1  
## 45 2 0  
## 46 3 0  
## 47 4 0  
## 48 5 0  
## 49 6 0  
## 50 7 0

# => looks good

Check for missing values. NAs and empty strings

colSums(is.na(df.horse) | df.horse == "")

## date race\_name type class dist\_m going ran pos   
## 0 0 0 0 0 0 0 0   
## draw horse age sex lbs hg secs jockey   
## 3950 0 0 0 0 34045 0 0   
## trainer prize won   
## 0 22402 0

# let's drop hg and prize because most of them are missing  
# but let's keep draw because AFAIK it might influence outcome  
# keep it in mind that it only exists for Flat course type  
  
df.horse <- subset(df.horse, select=-c(hg,prize))

Check datatypes.

str(df.horse)

## 'data.frame': 53368 obs. of 17 variables:  
## $ date : chr "2014-01-04" "2014-01-04" "2014-01-04" "2014-01-04" ...  
## $ race\_name: chr "Coral Mobile Just Three Clicks To Bet Classified Claiming Stakes" "Coral Mobile Just Three Clicks To Bet Classified Claiming Stakes" "Coral Mobile Just Three Clicks To Bet Classified Claiming Stakes" "Coral Mobile Just Three Clicks To Bet Classified Claiming Stakes" ...  
## $ type : chr "Flat" "Flat" "Flat" "Flat" ...  
## $ class : chr "Class 6" "Class 6" "Class 6" "Class 6" ...  
## $ dist\_m : int 2012 2012 2012 2012 2012 2012 2012 2012 2012 2012 ...  
## $ going : chr "Standard" "Standard" "Standard" "Standard" ...  
## $ ran : int 11 11 11 11 11 11 11 11 11 11 ...  
## $ pos : chr "1" "2" "3" "4" ...  
## $ draw : int 7 12 4 3 5 11 6 1 2 9 ...  
## $ horse : chr "Ocean Applause (GB)" "Copperwood (GB)" "Paddys Saltantes (IRE)" "Exclusive Waters (IRE)" ...  
## $ age : int 4 9 4 4 6 5 6 5 7 8 ...  
## $ sex : chr "G" "G" "G" "G" ...  
## $ lbs : int 113 115 118 118 116 128 115 116 118 115 ...  
## $ secs : chr "124.96" "125.11" "125.41" "125.66" ...  
## $ jockey : chr "Joe Doyle" "Jimmy Quinn" "Luke Morris" "Andrea Atzeni" ...  
## $ trainer : chr "John Ryan" "Lee Carter" "J S Moore" "Gary Moore" ...  
## $ won : num 1 0 0 0 0 0 0 0 0 0 ...

Some variable types should be modified for better analysis.

Convert ‘date’ column to date format

df.horse$date <- as.Date(as.character(df.horse$date), format = "%Y-%m-%d")

# Convert all character columns to factors  
df.horse[sapply(df.horse, is.character)] <- lapply(df.horse[sapply(df.horse, is.character)], as.factor)  
# df.horse$race\_name <- as.character(df.horse$race\_name)  
df.horse$pos <- as.integer(as.character(df.horse$pos))

## Warning: NAs introduced by coercion

# Convert secs to numeric after handling non-numeric values  
df.horse$secs <- as.numeric(as.character(df.horse$secs))

## Warning: NAs introduced by coercion

-> TOO MANY NAs INTRODUCED! -> COME BACK TO THIS LATER!

Log-transform “amounts” -> at prof’s recommendation In our dataset, these variables can be considered amounts: - dist\_m - lbs - secs

# Log-transforming 'dist\_m', 'lbs', and 'secs'  
df.horse$log\_dist\_m <- log(df.horse$dist\_m)  
df.horse$log\_lbs <- log(df.horse$lbs)  
df.horse$log\_secs <- log(df.horse$secs)

HOW TO VISUALIZE THE LOG AMOUNT TO COMPARE THEM WITH ORIGINAL??

Running the models take forever with our current dataset, so create a randomized sample with 10% of the data so that we can see results faster at this early stage of the project.

# Sample 10% of the dataset for testing  
df.horse\_sample <- df.horse[sample(nrow(df.horse), 0.1 \* nrow(df.horse)), ]  
  
dim(df.horse\_sample)

## [1] 5336 20

# 2) Linear Model

Placeholder

# 3) Generalised Linear Model - Poisson

Placeholder

# 4) Generalised Linear Model - Binomial

WIP

glm.horse\_sample <- glm(won ~ type + log\_dist\_m + going + age + sex + log\_lbs + log\_secs,   
 data = df.horse\_sample,  
 family = "binomial")  
summary(glm.horse\_sample)

##   
## Call:  
## glm(formula = won ~ type + log\_dist\_m + going + age + sex + log\_lbs +   
## log\_secs, family = "binomial", data = df.horse\_sample)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -149.85288 10.04269 -14.922 < 2e-16 \*\*\*  
## typeFlat -3.93412 0.60140 -6.542 6.09e-11 \*\*\*  
## typeHurdle -0.57501 0.40899 -1.406 0.15974   
## typeNH Flat -4.27781 0.71050 -6.021 1.74e-09 \*\*\*  
## log\_dist\_m 38.23285 2.69322 14.196 < 2e-16 \*\*\*  
## goingGood To Firm -0.13682 0.28000 -0.489 0.62510   
## goingGood To Soft 0.38314 0.43377 0.883 0.37708   
## goingHeavy 3.47607 0.50601 6.870 6.44e-12 \*\*\*  
## goingSoft 1.87947 0.36644 5.129 2.91e-07 \*\*\*  
## goingStandard -0.01919 0.23301 -0.082 0.93436   
## goingStandard To Slow 0.02399 0.30136 0.080 0.93656   
## age -0.09942 0.03085 -3.222 0.00127 \*\*   
## sexF -0.02159 0.16086 -0.134 0.89323   
## sexG -0.10986 0.16178 -0.679 0.49710   
## sexH 0.37217 0.51236 0.726 0.46760   
## sexM -0.45287 0.26039 -1.739 0.08200 .   
## log\_lbs 6.18213 1.08678 5.688 1.28e-08 \*\*\*  
## log\_secs -34.86958 2.44827 -14.243 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3740.4 on 5252 degrees of freedom  
## Residual deviance: 3417.3 on 5235 degrees of freedom  
## (83 observations deleted due to missingness)  
## AIC: 3453.3  
##   
## Number of Fisher Scoring iterations: 6

<span style=“color:red”;>HOW TO INTERPRET ALL THESE FACTOR VARIABLES? SHOULD THEY ALL BE FACTORS OR IS THERE A BETTER WAY? I DON’T REMEMBER SO MANY FACTORS FROM THE LECTURES.

# 5) Generalised Additive Model

Placeholder

# 6) Neural Network

Placeholder

# 7) Support Vector Machine

Placeholder

# 8) Use of Generative AI

Placeholder

# 9) Conclusion

Placeholder