Man v Machine: Greyhound Racing Predictions

MSc Research Project
Data Analytics

Alva Lyons
x15014274

School of Computing
National College of Ireland

Supervisor: Dr. Oisín Creaner

National College of Ireland
Project Submission Sheet – 2015/2017
School of Computing

Student Name:	Alva Lyons
Student ID:	x15014274
Programme:	Data Analytics
Year:	2016
Module:	MSc Research Project
Lecturer:	Dr. Oisín Creaner
Submission Due Date:	21/12/2016
Project Title:	Man v Machine: Greyhound Racing Predictions
Word Count:	5964

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	21st December 2016

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Man v Machine: Greyhound Racing Predictions

Alva Lyons
x15014274
MSc Research Project in Data Analytics

21st December 2016

Research Question

Can the implementation of machine learning techniques alone predict higher win percentages on greyhound races in Shelbourne Park than those of the expert employed on this race track?

Abstract

The purpose of this research is to ascertain whether greyhound racing results can be predicted with a high degree of certainty using machine learning techniques. The main focus of this research is in bridging the gap between existing sports prediction models which use manual feature selection to creating a model built from machine chosen subsets by algorithmically sub-setting the feature space. Feature selection is the process of sub-setting the feature space by analysing the relevance of features both to each other and to the predicted variable so that only the most relevant features are used within the modelling framework. The reason for introducing the greyhound expert is to test whether the model can outperform the average social gambler who tend to make their betting selection based on tips given to them by domain experts.

1 Introduction

The greyhound racing industry in Ireland is controlled by the Irish Greyhound Board (IGB). It is estimated that 720,000 visitors annually attend IGB controlled greyhound stadia. Shelbourne Park is the premier greyhound stadium in Ireland and hosts one of the world's richest greyhound races, The Irish Derby, every September. Racing takes place in Shelbourne Park every Wednesday, Thursday and Saturday.

This research will attempt to predict the finishing position of a greyhound in a given race. The data used in this research comprises of 64,908 observations of 10,986 races ran in Shelbourne Park between January 2009 and August 2016. The prediction rate of this model is bench-marked against that of the stadium's resident greyhound expert who is employed by IGB to predict the winning greyhound, the top 2 and the top 3 finishing greyhounds for the top of the race card for each race on a given race night.

The use of machine learning techniques in sports prediction is not a new phenomenon but rather it has gained many more practitioners since the spread of online gambling markets. In the sport of greyhound racing there have been 3 academic papers which have utilised varying machine learning techniques in order to predict racing results. The seminal paper in this field (Chen et al. (1994)) dates back to 1994 and uses the knowledge of a greyhound expert for feature selection in choosing which performance variables to use when running machine learning techniques on their dataset. Both of the follow up studies utilise a similar feature selection in their models (Schumaker and Johnson (2008), Johansson and Sönströd (2003)). This research uses feature selection algorithms to limit the problem space of the domain in order to avoid the subjectivity of human interactions within the modelling process.

Additionally, this paper combines various data mining techniques from sentiment analysis through to deep learning ensemble methods in its attempts to test if a machine learning model without human subjectivity can out-preform an expert in the area of greyhound racing predictions.

The rest of this document is laid out as follows:

- **Section 2** discusses the related work in the field of sports predictions and highlights the role this research plays within this field.
- **Section 3** discusses the methodology framework used in completing this research.
- **Section 4** reviews and justifies the implementation steps carried out in this research.
- **Section 5** evaluates the results of the prediction algorithms.
- **Section 6** concludes the research and discusses potential future work to be carried out.

2 Related Work

2.1 Sports Predictions

The literature and academic work produced in the field of sports predictions is far reaching. Using historical results data to predict the outcome of sporting events has gained exposure due to the growth of on-line betting markets and the large volumes of historical data which are easily accessible.

2.2 Greyhound Racing Prediction

While the use of machine learning techniques is prevalent in predicting horse racing (Butler et al. (1998), Silverman and Suchard (2013), Davoodi and Khanteymoori (2010), Williams and Li (2008)) results there have only three documented cases of utilising machine learning in predicting the outcome of greyhound races. While the two sports are often synonymous there are distinct differences between the two which ensures that modelling concepts need to be amended. A greyhound race result is the outcome of 6 greyhounds chasing a mechanical hare in their attempts to catch it; while a horse race result is the outcome of the interactions between a jockey and its mount as they traverse the race course. While this might seem trivial, the key difference is apparent when considering a model's attempts at predicting the finishing positions of competitors in a race. A greyhound is bred to chase the hare and will continue its mission even if the race has already been won. On the other hand, a jockey which surmises it has no chance of finishing in the first x poisitions, may choose to pull back so that the horse's handicap rating is not affected for its next race. This nuance is one of the factors that led this researcher to choose greyhound racing as the sport of choice for this research.

2.3 Problem Space and Feature Set

An important step in the data mining process is choosing which features to include in your model. Feature selection can either be done manually through the use of domain knowledge or algorithmically with the use of machine learning methods. The race card available on tracks includes 50 variables which could potentially affect the outcome of a race. Adding all of these variables into a model would increase it's complexity and be algorithmically inefficient. (Lyons (2016))

Many of the works done on predicting results of horse and greyhound races focus on the model used for prediction and it's tuning parameters rather than the selection of the feature subset (Pudaruth et al. (2013), Davoodi and Khanteymoori (2010), Williams and Li (2008)). Their feature subset are listed but the motivation behind choosing which features to include in their model is not elaborated on. One must assume that the features are chosen based on the subjective opinions held by the researchers on what performance variables affect the outcome of these sporting events.

McCabe and Trevathan (2008)'s paper focuses more on the feature set than the model used in sports prediction. This paper provides an interesting discussion on why variables were included in the model however they are very vague on the potential "subjective" variables not added. Similar to the papers listed above feature selection in the research by McCabe and Trevathan is a manual process and does not use machine learning to choose the optimal subset of features to include in the modelling.

2.3.1 Historical Feature Selection Techniques in Greyhound Racing Predictions

Chen et al. (1994) in their prediction of greyhound racing results chose their feature set following discussions with domain experts who informed them which 10 performance variables they believed were most important in predicting winners. They admit that while this is not optimal it is a consequence of their chosen algorithms being unable to handle noisy data. Remarkably neither Johansson and Sönströd (2003) nor Schumaker and Johnson (2008), in their follow up studies, chose to research further attempts at feature selection. Rather they used a similar feature subset to those used in the study by Chen et al.. (Lyons (2016))

2.3.2 Bridging The Research Gap

This research attempts to apply various feature selection algorithms to the transformed dataset in order to ascertain which features have a greater impact on influencing a greyhounds finishing position within a race. The features which are extracted as relevant to this domain problem are then chosen as the final dataset to be used in the model. The choice of using a neural network in the modelling phase of this research is to test if the use of machine based feature selection can outperform those as used by Chen et al. (1994) and Johansson and Sönströd (2003). As this research is focusing on classification rather than regression modelling the model choice of Schumaker and Johnson (2008) (Support Vector Regression) is discounted from the outset.

3 Methodology

The methodology used in this research is Knowledge Discovery in Databases (KDD). The KDD methodology allows for an iterative approach to the processes involved in extracting knowledge from raw data. Initial plans were to utilise the SEMMA notation, as developed by SAS, but the sequential nature of this methodology couldn't rival the flexibility and interactivity of KDD (Azevedo and Santos (2008)). KDD focuses on the entire process from data selection through pre-processing, extraction, data mining to interpretation (Fayyad et al. (1996)). An illustration of the KDD methodology as it pertains to this research is shown in Figure 1

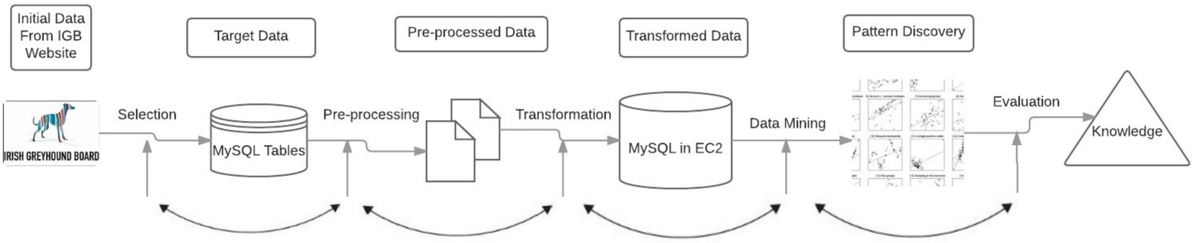


Figure 1: KDD Methodology

3.1 Selection

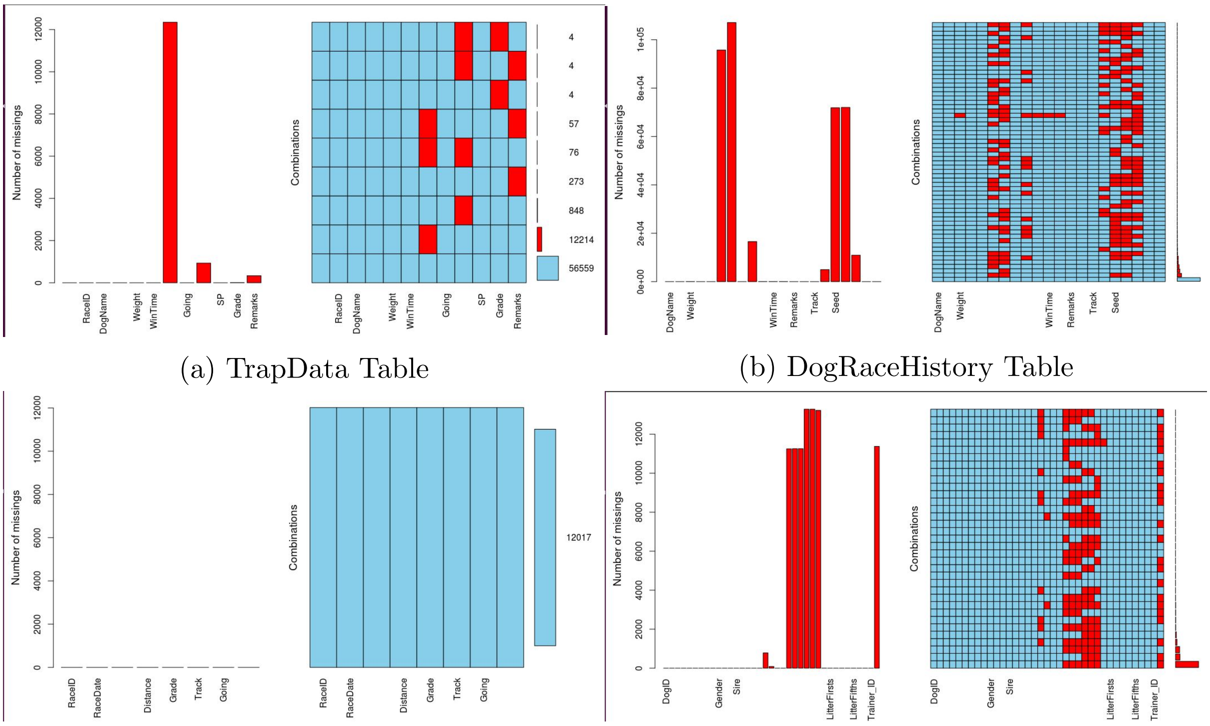
The raw data used in this research is extracted from the Irish Greyhound Board's (IGB) website [1] using a Python script. This data consists of 383,746 observations of 28,271 races ran throughout Ireland between 1st April 2007 and 3rd September 2016. This data is collected from multiple embedded pages and loaded into 6 tables in a MySQL database. The flow of this script is illustrated in the diagram in Appendix A.

3.2 Pre-Processing

The pre-processing phase covers the cleaning and preparing of data for modelling. The data from IGB's website contains numerous inaccuracies and missing data points ensuring the pre-processing phase of the KDD methodology plays an integral role in this research. Errors in the data were discovered when the data was examined using visualization and descriptive statistics.

[1] www.igb.ie/results

3.2.1 Dealing with Missing Values



(a) TrapData Table

(b) DogRaceHistory Table

(c) Races Table

(d) Dogs Table

Figure 2: Missing Values in Raw Data Tables

An important decision in any data mining task is to decide how to deal with missing and incorrect values. It is necessary to ascertain why a value is missing or incorrect and to decide if action is to be taken. (Witten et al. (2011))

As the raw data was scraped from tables on a web page missing values are stored in the database as empty strings. It was necessary to write a SQL script to convert these values to NULL before deciding what action to take. The proportion of missing values to actual data can be seen in Figure 2. In these diagrams the red squares represent missing values while the blue squares represent the presence of data. It is evident that the DogRaceHistory table, in particular, has a large proportion of missing values. As this table is integral to ascertaining the performance history of a greyhound it is important to ascertain the best method of handling this missing data.

Domain knowledge plays an important part in deciding what steps to take in handling missing values. For instance, a NULL value in the Seed column does not represent a true missing value. A greyhound's seed indicates their preferred running style; Inside (I) seeded greyhounds tend to run toward the bend; Middle(M) seeded greyhounds tend to run in the centre of the track; Wide(W) seeded greyhounds run toward the rails. The lack of one of these characters in the seed column is likely to indicate that a greyhound has no preferred running style. For this reason the missing data points in this column were replaced with "A" (Any).

Further exploratory analysis of this table showed up explanations for some of the high volume of missing data points within this dataset. Table 1 depicts the percentage of missing data in this table that can be explained by the inclusion of time trials.

Table 1: Time Trials v Missing Data

DogRaceHistory Table	
Field	**Responsible for % of Missing Data**
Weight	100
NumberOfDogs	100
WinTime	100
Going	100
PlacedDistance	99.9
RunnerGrade	86.90
RaceGrade	73.1
SP	70.5
Remarks	61.2
SectionalPosition	46.9
SectionalTime	44.5
EstimatedTime	8.4

Time trials take place on race nights before racing commences whereby 1 or more greyhounds run around the track in a non competitive setting to see how fast they can chase the mechanical hare. Due to the non competitive nature of these events it was deemed appropriate to remove these from the dataset.

3.2.2 Dealing With Incorrect Values

Outlier detection was performed on the dataset using visualisation and descriptive statistics methods. The advantage of outlier detection lies in the identification of errors within the raw data, removal of outliers allows for purification of the data before modelling commences. (Hodge and Austin (2004))

For instance in the Weight column of the DogRaceHistory table the minimum weight of a greyhound is listed as 0lbs while the maximum is listed as 677lb. The average weight of a greyhound is 65-70lbs. The outlier values were imputed by taking the mean value of a greyhounds weight in its preceding and succeeding 2 races and inputting this as the new value in the weight column. In the instances where a greyhound did not have any other races against it the mean value of the weights of its competitors in a given race is imputed as its new value. Similar imputations were performed on other outliers within the dataset.

3.2.3 Pruning The Dataset

The track ratings of greyhound stadia in Ireland differ depending on the ground conditions; as such the RunnerGrade variable will have varying significance depending on where a race took place. As this research attempts to predict results in Shelbourne Park it is deemed appropriate to limit the race history of greyhounds to their performance at Shelbourne Park. It is widely accepted in greyhound circles that the best greyhounds are raced in Shelbourne Park where the higher prize money is paid out. Additionally, only A grade (middle distance) races have been chosen for this research. The reason for omitting sprint races lies in the short distance between the first bend and the finish line. In sprint races if a runner gets knocked at the first bend their chances of recovery are limited. (Lyons (2016)) The remaining dataset consists of 64,908 observations of 10,986 races ran between January 2009 and August 2016.

3.2.4 Limitations Of The Dataset Discovered During Pre-Processing

It was during the pre-processing phase that limitations of the dataset were also exposed. These limitations lie in the scraped data being from a view of the IGB's database at the time of scraping. While this doesn't affect historical race content it does ensure that owner and trainer data is inadmissible for modelling as there is no way of ascertaining whether the greyhound was attached to its current owner or trainer at the time of each historical race. Including the data in these two tables in the modelling phase of this research could potentially lead to incorrect predictions based on corrupt data.

Litter distribution data in the Dogs table, which depicts the number of starts and race placings of a greyhound's siblings, is also inadmissible as it is that of a view of the to date total of the litter at the time of scraping rather than the time of racing.

While omitting this data does have its benefits in that it cuts down the processing time of feature selection in the data mining phase it restricts the model in that it does not have access to the same data available in real time to the greyhound expert the model will be bench marked against.

The benefits of a thorough pre-processing phase ensure a strong knowledge of the dataset is gained before transformations commence.

3.3 Transformation

3.3.1 Text Analysis

The remarks column in the DogRaceHistory table provides a shorthand of comments on how a greyhound ran in a given race eg. FAw (Fast Away), BBkd (Badly Baulked), TRec (Track Record) etc.

In order to analyse these remarks across the dataset the basic premise of sentiment analysis was performed such that the text is classified as expressing a positive or negative tenet. (Liu (2010)) While sentiment analysis deals with "the computational treatment of opinion, sentiment, and subjectivity in text" Pang and Lee (2008) and is considered to be more suitable for text mining of unstructured datasets; the simplicity and power of this method was deemed appropriate in analysing this variable.

The first step in utilising the premise of sentiment analysis is to create domain specific lexicons of the shorthands used in the remarks column. The author's domain knowledge was sufficient in their creation but confirmation was received from 2 experts working within the greyhound racing industry to ensure subjectivity was minimized during this phase. These dictionaries were created by assigning each shorthand comment into one of 5 categories; Very Positive, Positive, Neutral, Negative, Very Negative.

The motivation for utilising this variable and performing text analysis lies with the possibility that the greyhound's ability is not properly reflected by it's finishing position. For instance a greyhound may only finish 5th in a race despite being quick out of the traps due to being impeded by another greyhound. By the same respect the 1st placed finisher in this race might have missed the early fighting and received a clear run despite being slow away.

In order to run sentiment analysis on this column the 5 dictionaries were loaded into MySQL tables and the remarks column is scored using an SQL statement which scans each of these tables and matches the word in remarks to those in the dictionaries. A scoring is given to the words depending on which category they fall into:

- Very Positive = +2 points
- Positive = +1 point
- Neutral = 0 points
- Negative = -1 point
- Very Negative = -2points

The scoring for each remark is totalled and set as the remark score for each greyhound in a given race.

3.3.2 Feature Engineering

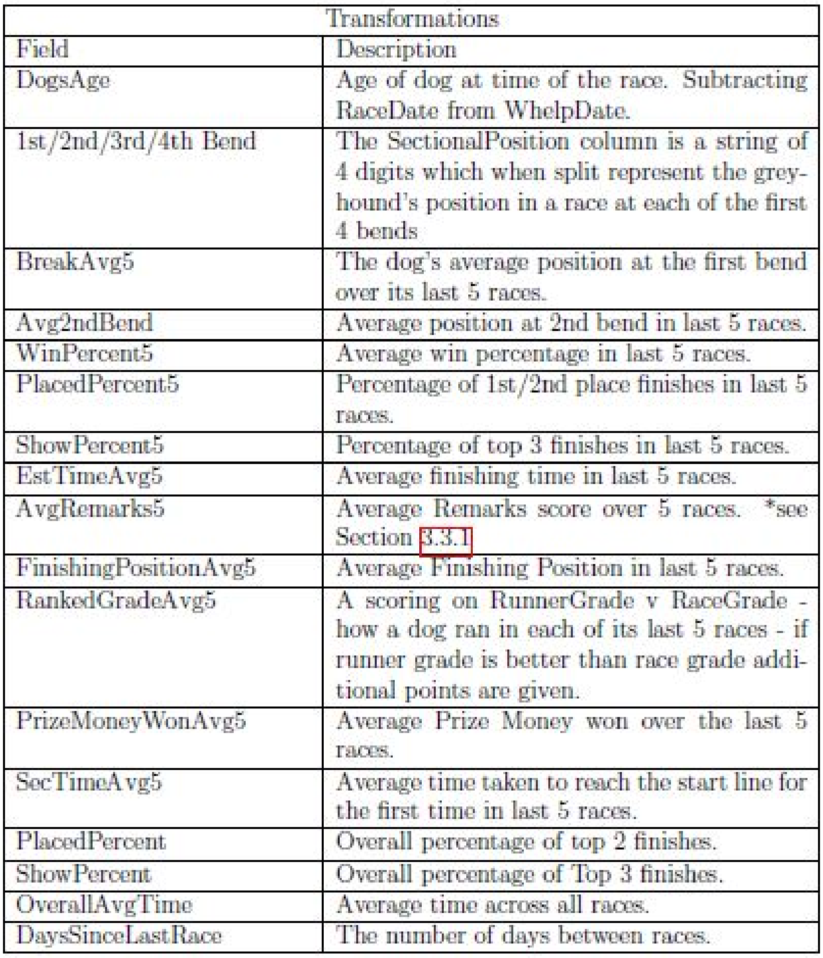
The importance of feature engineering lies in bridging the gap between the features in the initial problem domain to the structure of features needed for the solution architecture. (Dash and Liu (2003)) Feature engineering is used in order to find the best representation of the variables available within the dataset in the hopes of better being able to find a desirable solution to a problem. There are many elements to feature engineering from framing the problem domain to data cleansing and formatting. (Brownlee (2014)) The element discussed in this research pertains to the manual construction of new features from the raw dataset.

The race card provided on track for punters provides data on the last 5 races of a greyhound in a particular race. The initial data was transformed to create meaningful information from each dog's race history. While Chen et al. (1994) and Schumaker and Johnson (2008) average their variables over 7 races this research looks to emulate the on track race card by averaging the greyhound history statistics over 5 races. This ensures that the model has access to the same data as the ordinary punter. Where a greyhound has run less than 5 races the data is averaged over the number of runs of that greyhound up to a maximum of 5 races. The below formula shows how the rolling averages are calculated; this example calculates a greyhounds average position at the first bend in each of it's last n races.

$$\text{BreakAvg5} = \text{FirstBend-r1} + \text{FirstBend-r2} + ... + \text{FirstBend-rn}\frac{}{n}$$

Similar formulae were used to transform other variables in the raw data. Figure **??** depicts a table of the transformations that took place in this phase of the research methodology.

Figure 3: Variables Created From Raw Data



Transformations	
Field	Description
DogsAge	Age of dog at time of the race. Subtracting RaceDate from WhelpDate.
1st/2nd/3rd/4th Bend	The SectionalPosition column is a string of 4 digits which when split represent the greyhound's position in a race at each of the first 4 bends
BreakAvg5	The dog's average position at the first bend over its last 5 races.
Avg2ndBend	Average position at 2nd bend in last 5 races.
WinPercent5	Average win percentage in last 5 races.
PlacedPercent5	Percentage of 1st/2nd place finishes in last 5 races.
ShowPercent5	Percentage of top 3 finishes in last 5 races.
EstTimeAvg5	Average finishing time in last 5 races.
AvgRemarks5	Average Remarks score over 5 races. *see Section 3.3.1
FinishingPositionAvg5	Average Finishing Position in last 5 races.
RankedGradeAvg5	A scoring on RunnerGrade v RaceGrade - how a dog ran in each of its last 5 races - if runner grade is better than race grade additional points are given.
PrizeMoneyWonAvg5	Average Prize Money won over the last 5 races.
SecTimeAvg5	Average time taken to reach the start line for the first time in last 5 races.
PlacedPercent	Overall percentage of top 2 finishes.
ShowPercent	Overall percentage of Top 3 finishes.
OverallAvgTime	Average time across all races.
DaysSinceLastRace	The number of days between races.

3.4 Data Mining

Data mining is the process of analysing datasets to find unobserved and often unsuspected relationships within the data by combining statistics, artificial intelligence and machine learning features (Hand et al. (2001)). Fayyad et al. (1996) address the importance of understanding the data mining activity before including it in the KDD process. Similar to the KDD methodology, the choosing of an algorithm to use in tackling a prediction problem can involve many interactions and iterations before knowledge is gleaned. An important first step is to decide which data mining process of predictive analysis is required in ascertaining the value of the predictor variable. **Classification analysis** deals with predicting which category or class an object falls into. The required output is a discrete variable. **Regression analysis** is used to predict missing or unavailable numerical data values; the output variable is a continuous variable. Han (2005)

The application of predicting the outcome of a competitive event does not strictly fall into either a classification or a regression problem and as a result both regression and classification techniques are possible within the realms of this research domain. As a classification problem the output variable can be a matter of predicting the binary output of win or lose. As a regression problem it is possible to look at the finishing order of a race with the view to regressing on the FinishingPosition variable.

This research approaches the problem of predicting greyhound racing results as a classification problem. However, rather than choosing binary classification of ”win” or ”lose” it attempts are made to classify a greyhound’s Finishing Position. The reasoning for not choosing binary classification is partly due to class imbalance; for each race 6 greyhounds are entered and 5 greyhounds cannot win as such the number of observations in the lose class in the training set is larger than that of winners and random predictions could result in a higher rate of prediction due to chance alone. Additionally, by choosing to classify the problem using the Finishing Position as the predictor variable this allows for testing how wrong a predicted class is. For instance, incorrectly predicting a 1st place finisher will finish in 2nd place is ”less wrong” than predicting the same greyhound will finish in 6th place. The choice of algorithms and justification for their uses is discussed in the implementation section of this paper.

4 Implementation

4.1 Tools Used

The tools used in implementing this research are:

- Python (Version 2.7.12)
- MySQL
- R (Version 3.3.1)
- R Studio 64bit
- Amazon EC2

While python was used to scrape the raw data due to the power of its BeautifulSoup library; which provides an easy to use framework for parsing HTML into a tree representation; the data mining algorithms were running using R. R is a statistical programming language which is widely used for data analysis. (Lantz (2013))

4.2 Examining The Dataset

Once the feature engineering phase was complete the next step was to combine and explore the dataset. A flattened correlation matrix of the processed dataset is produced in R using the *corrplot* library (See Figure 4).

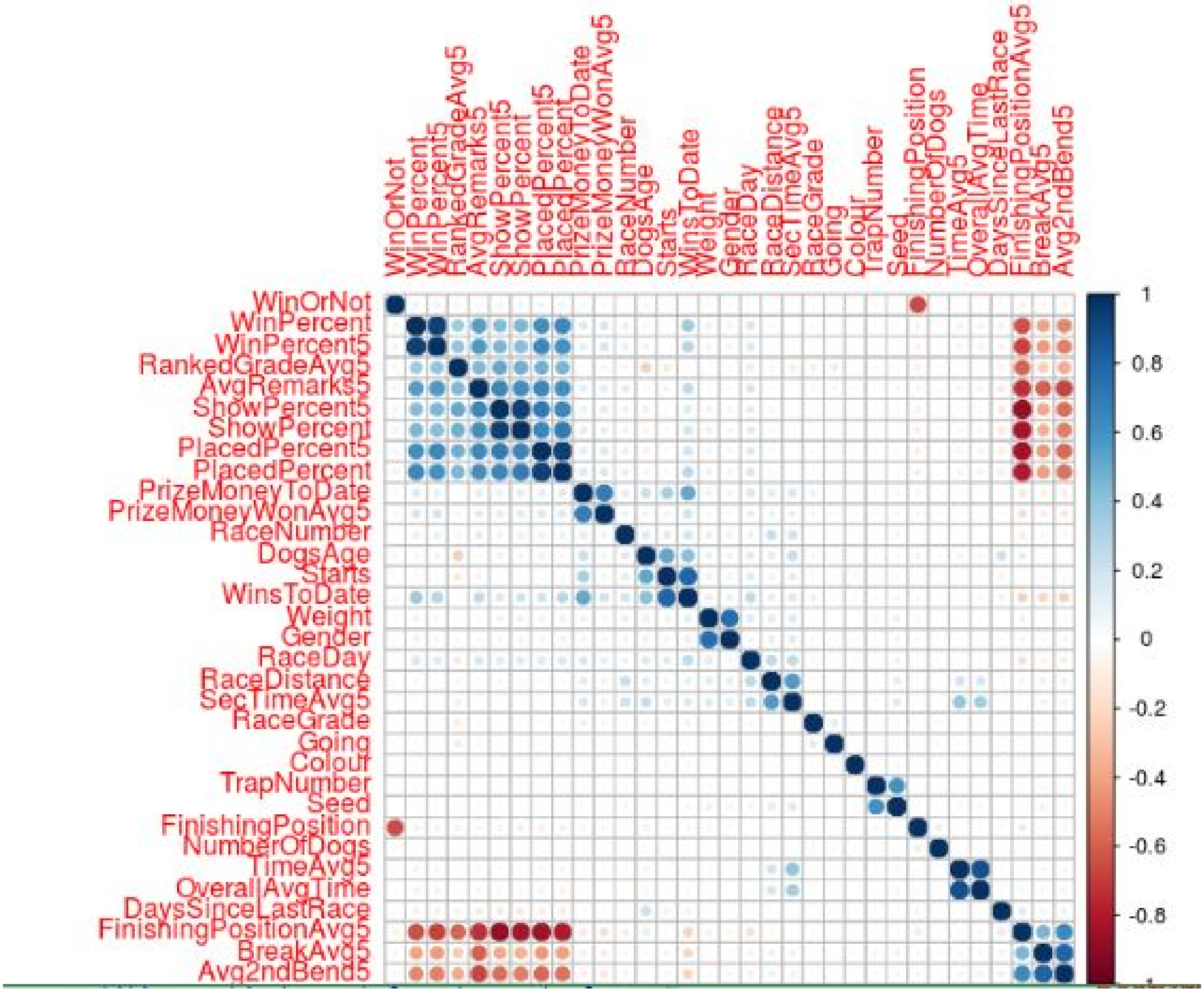


Figure 4: Flattened Correlation Matrix

As is evident in this visual representation of the correlation between features there are a number of strong correlations amongst variables in this dataset. While some of them are expected; such as the positive correlation between the number of starts of a greyhound and the number of wins; others are unexpected such as the negative correlation between a greyhound's win percentage (the percentage of wins over all races) and it's finishing position in it's last 5 races. A negative correlation depicts an inverse proportionality between variables in that as one increases the other will decrease. This would suggest that over time a greyhound's recent form has an obvious affect on its overall performance in that rather than remaining consistent it will either improve or deteriorate over time.

4.3 Feature Selection

Feature selection is the process of binning variables into subsets of relevant and irrelevant features such that only the most relevant features are used within the modelling framework (Dash and Liu (1997)). A feature is deemed to be relevant if it affects the target problem in any way. The benefits of feature selection lie in reducing the complexity and run-time of the machine learning algorithm. The reducing of complexity allows for better understanding of the patterns that arise in the data mining process. Additionally, feature selection when performed correctly, can improve model performance.

4.3.1 Methods of Feature Selection

There are three categories of feature selection methods; Wrapper, Filter and Embedded.

- **Filter Methods** - are concerned with exploring only the inherent features of a dataset. They are based on statistical tests and are independent of the variable to be predicted.

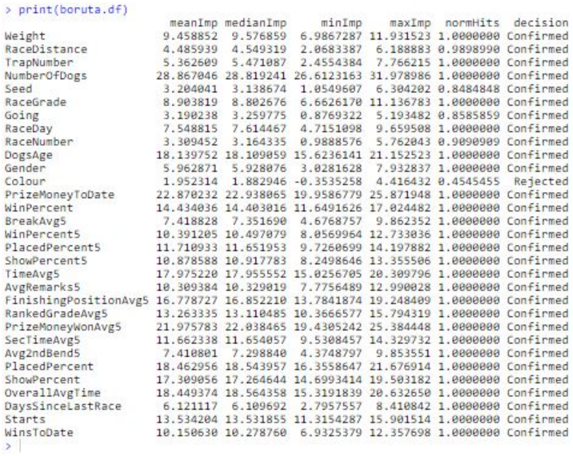
- **Wrapper Methods** - Unlike filter methods wrapper methods are used to find features subsets which interact with the variable to be predicted. In this way the choosing of a wrapper method is closely linked to the choosing of a modelling algorithm as the feature subset space is wrapped around the classifying model.(Saeys et al. (2007))

- **Embedded** - Embedded methods are an extension of the Wrapper Method framework and attempt to combine the best properties of the preceding two methods. The feature selection is 'embedded' in the modelling algorithm which runs feature selection and prediction concurrently.

This research focuses on wrapper and embedded methods as they interact with the variable to be predicted, are less likely to get stuck in a local optima and model feature dependencies. The limitations of these methods, however, lie in the increased risk of over-fitting. (Saeys et al. (2007))

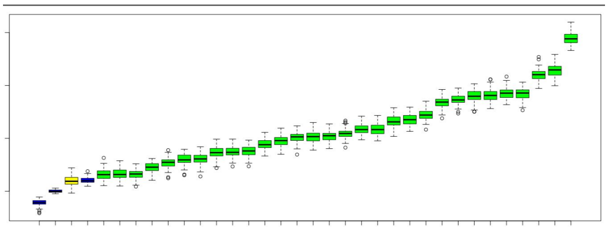
4.3.2 Wrapper with Boruta Package

Several packages are available in order to ascertain the importance of independent variables in predicting the dependent variable. The *Boruta* package in R comprises of a wrapper algorithm which utilises random forests in order to extract relevant features from a data set. This is achieved by comparing a variable's importance against importance that is achievable at random (Kursa and Rudnicki (2016)). The application of this package on the dataset did not dramatically reduce the feature space; removing only one variable from the original 30 inputted into the algorithm as can be seen in Figure 5.



```
> print(boruta.df)
                       meanImp medianImp     minImp    maxImp  normHits  decision
Weight                9.458852  9.576859  6.9867287 11.931523 1.0000000 Confirmed
RaceDistance          4.485939  4.549319  2.0683387  6.188883 0.9898990 Confirmed
TrapNumber            5.362609  5.471087  2.4554384  7.766215 1.0000000 Confirmed
NumberOfDogs         28.867046 28.819241 26.6123163 31.978986 1.0000000 Confirmed
Seed                  3.204041  3.138674  1.0549607  6.304202 0.8484848 Confirmed
RaceGrade             8.903819  8.802676  6.6626170 11.136783 1.0000000 Confirmed
Going                 3.190238  3.259775  0.8769322  5.193482 0.8585859 Confirmed
RaceDay               7.548815  7.614467  4.7151098  9.659508 1.0000000 Confirmed
RaceNumber            3.309452  3.164335  0.9888576  5.762043 0.9090909 Confirmed
DogsAge              18.139752 18.109059 15.6236141 21.152523 1.0000000 Confirmed
Gender                5.962871  5.928076  3.0281628  7.932837 1.0000000 Confirmed
Colour                1.952314  1.882946 -0.3535258  4.416432 0.4545455  Rejected
PrizeMoneyToDate     22.870232 22.938065 19.9586779 25.871948 1.0000000 Confirmed
WinPercent           14.434036 14.403016 11.6491626 17.024482 1.0000000 Confirmed
BreakAvg5             7.418828  7.351690  4.6768757  9.862352 1.0000000 Confirmed
WinPercent5          10.391205 10.497079  8.0569964 12.733036 1.0000000 Confirmed
PlacedPercent5       11.710933 11.651953  9.7260699 14.197882 1.0000000 Confirmed
ShowPercent5         10.878588 10.917783  8.2498646 13.355506 1.0000000 Confirmed
TimeAvg5             17.975220 17.955552 15.0256705 20.309796 1.0000000 Confirmed
AvgRemarks5          10.309384 10.329019  7.7756489 12.990028 1.0000000 Confirmed
FinishingPositionAvg5 16.778727 16.852210 13.7841874 19.248409 1.0000000 Confirmed
RankedGradeAvg5      13.263335 13.110485 10.3666577 15.794319 1.0000000 Confirmed
PrizeMoneyWonAvg5    21.975783 22.038465 19.4305242 25.384448 1.0000000 Confirmed
SecTimeAvg5          11.662338 11.654057  9.5308457 14.329732 1.0000000 Confirmed
Avg2ndBend5           7.410801  7.298840  4.3748797  9.853551 1.0000000 Confirmed
PlacedPercent        18.462956 18.543957 16.3558647 21.676914 1.0000000 Confirmed
ShowPercent          17.309056 17.264644 14.6993414 19.503182 1.0000000 Confirmed
OverallAvgTime       18.449374 18.564358 15.3191839 20.632650 1.0000000 Confirmed
DaysSinceLastRace     6.121117  6.109692  2.7957557  8.410842 1.0000000 Confirmed
Starts               13.534204 13.531855 11.3154287 15.901514 1.0000000 Confirmed
WinsToDate           10.150630 10.278760  6.9325379 12.357698 1.0000000 Confirmed
> |
```

(a) Output of Boruta Algorithm

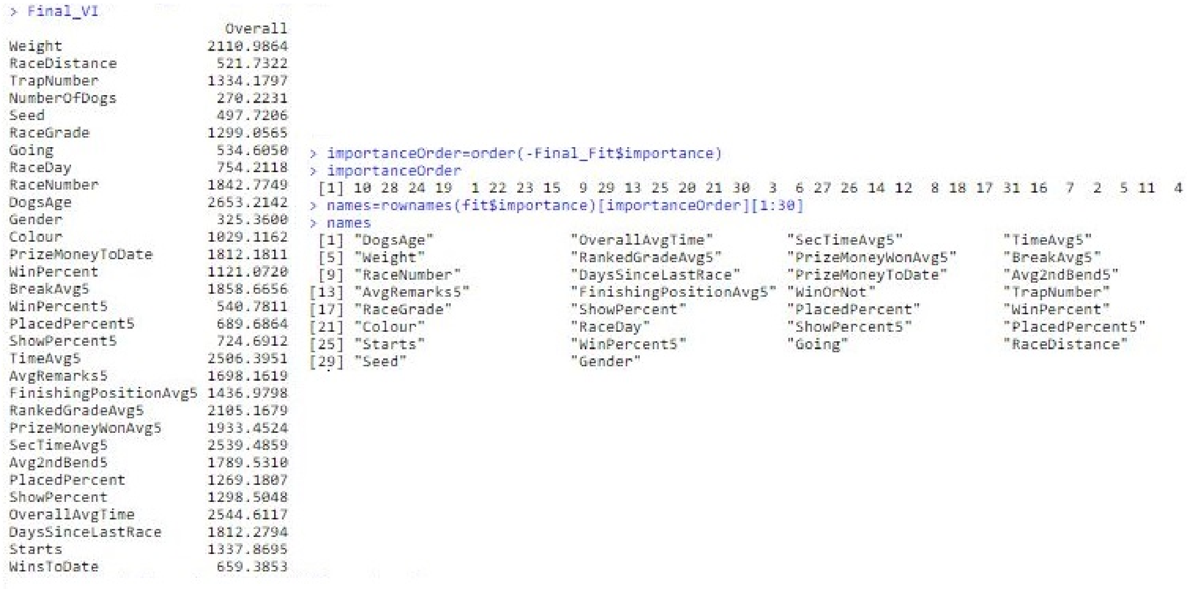


(b) Plot Of Boruta Output

Figure 5: Boruta Alogirthm for Variable Importance Detection

4.3.3 Wrapper With caret & randomForest Packages

The *caret*, an acronym for **C**lassification **a**nd **Re**gression **T**raining, package contains functions to organise a model's training approach(Kuhn (2016)) and utilises a number of other packages in r. In this example the caret package wraps around the *randomForest* package in order to rank the variable importance of the features in the dataset. Variable importance ratings are assigned to each feature and they are then ranked according to how important they are to the predictor variable, *FinishingPosition*. As can be see in Figures 6 and 7 the top 10 ranked features are *DogsAge, OverallAvgTime, SecTimeAvg5, TimeAvg5, Weight, RankedGradeAvg5, PrizeMoneyWonAvg5, BreakAvg5, RaceNumber and PrizeMoneyToDate.*



```
> Final_VI
                        Overall
Weight                2110.9864
RaceDistance           521.7322
TrapNumber            1334.1797
NumberOfDogs           270.2231
Seed                   497.7206
RaceGrade             1299.0565
Going                  534.6050
RaceDay                754.2118
RaceNumber            1842.7749
DogsAge               2653.2142
Gender                 325.3600
Colour                1029.1162
PrizeMoneyToDate      1812.1811
WinPercent            1121.0720
BreakAvg5             1858.6656
WinPercent5            540.7811
PlacedPercent5         689.6864
ShowPercent5           724.6912
TimeAvg5              2506.3951
AvgRemarks5           1698.1619
FinishingPositionAvg5 1436.9798
RankedGradeAvg5       2105.1679
PrizeMoneyWonAvg5     1933.4524
SecTimeAvg5           2539.4859
Avg2ndBend5           1789.5310
PlacedPercent         1269.1807
ShowPercent           1298.5048
OverallAvgTime        2544.6117
DaysSinceLastRace     1812.2794
Starts                1337.8695
WinsToDate             659.3853
```

```
> importanceOrder=order(-Final_Fit$importance)
> importanceOrder
 [1] 10 28 24 19  1 22 23 15  9 29 13 25 20 21 30  3  6 27 26 14 12  8 18 17 31 16  7  2  5 11  4
> names=rownames(fit$importance)[importanceOrder][1:30]
> names
 [1] "DogsAge"               "OverallAvgTime"        "SecTimeAvg5"           "TimeAvg5"
 [5] "Weight"                "RankedGradeAvg5"       "PrizeMoneyWonAvg5"     "BreakAvg5"
 [9] "RaceNumber"            "DaysSinceLastRace"     "PrizeMoneyToDate"      "Avg2ndBend5"
[13] "AvgRemarks5"           "FinishingPositionAvg5" "WinOrNot"              "TrapNumber"
[17] "RaceGrade"             "ShowPercent"           "PlacedPercent"         "WinPercent"
[21] "Colour"                "RaceDay"               "ShowPercent5"          "PlacedPercent5"
[25] "Starts"                "WinPercent5"           "Going"                 "RaceDistance"
[29] "Seed"                  "Gender"
```

Figure 6: Ranked Order of Variable Importance

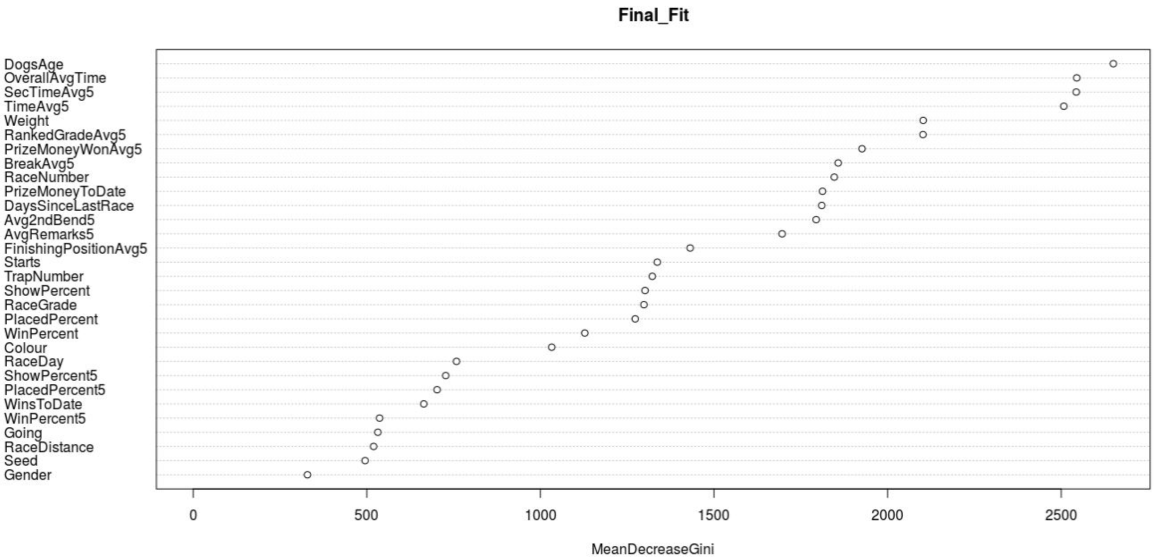


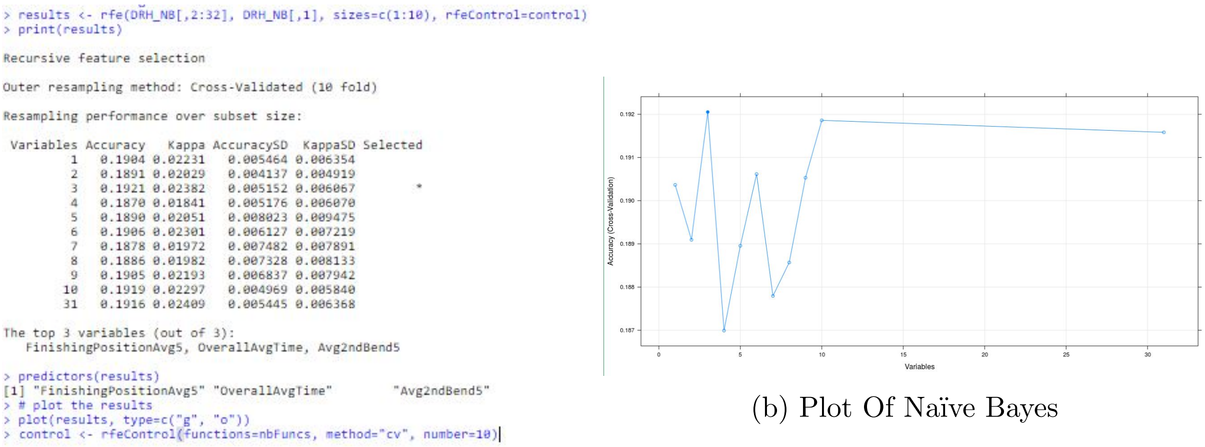
Figure 7: Plot Of Variable Importance With caret & randomForest packages

4.3.4 Embedded - Recursive Feature Elimination with caret

Recursive feature elimination (RFE) is an embedded method of feature selection. It attempts to find the optimal subset of features by iterating through all features, assigning weights to each feature dependent on their value to the dependent variable. Features are eliminated based on their ranked weighting, in that those with a smaller weighting are eliminated first. Once a feature is pruned the remaining features are reassigned weights and the process is iterated until a stopping criterion is reached whereby the optimal number of features is selected. (Guyon et al. (2002))

The caret packages provides a set of predefined functions to embed RFE with algorithmic functions; such as Naïve Bayes; Random Forests; and Bagged Trees. These 3 functions were modelled on the dataset in order to attempt to find the optimal subset of features to use in prediction.

1. **Naïve Bayes** is a classification algorithm which is based on Bayes Theorem and assumes independence amongst the feature space.



```
> results <- rfe(DRH_NB[,2:32], DRH_NB[,1], sizes=c(1:10), rfeControl=control)
> print(results)

Recursive feature selection

Outer resampling method: Cross-Validated (10 fold)

Resampling performance over subset size:

 Variables Accuracy   Kappa AccuracySD  KappaSD Selected
         1   0.1904 0.02231   0.005464 0.006354
         2   0.1891 0.02029   0.004137 0.004919
         3   0.1921 0.02382   0.005152 0.006067        *
         4   0.1870 0.01841   0.005176 0.006070
         5   0.1890 0.02051   0.008023 0.009475
         6   0.1906 0.02301   0.006127 0.007219
         7   0.1878 0.01972   0.007482 0.007891
         8   0.1886 0.01982   0.007328 0.008133
         9   0.1905 0.02193   0.006837 0.007942
        10   0.1919 0.02297   0.004969 0.005840
        31   0.1916 0.02409   0.005445 0.006368

The top 3 variables (out of 3):
   FinishingPositionAvg5, OverallAvgTime, Avg2ndBend5

> predictors(results)
[1] "FinishingPositionAvg5" "OverallAvgTime"        "Avg2ndBend5"
> # plot the results
> plot(results, type=c("g", "o"))
> control <- rfeControl(functions=nbFuncs, method="cv", number=10)
```

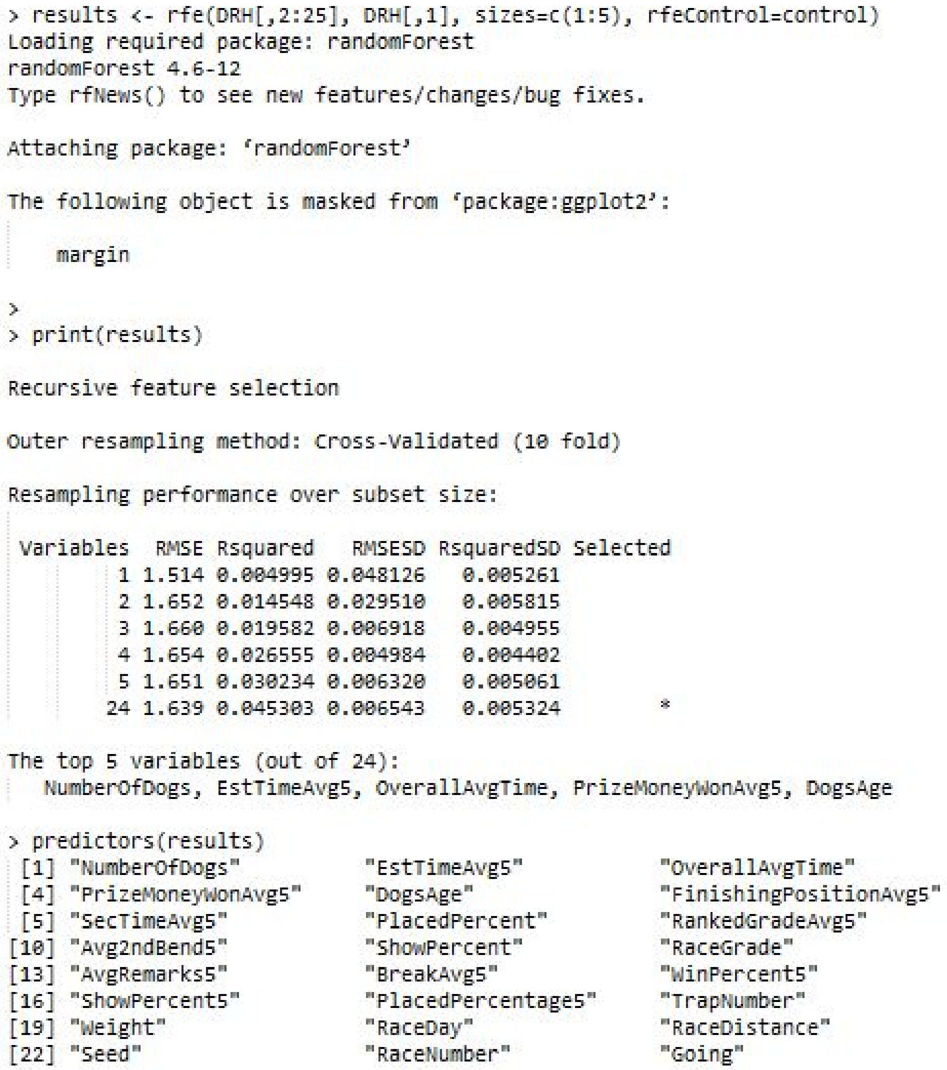
(b) Plot Of Naïve Bayes

(a) Output of Naïve Bayes

Figure 8: Naïve Bayes Recursive Feature Elimination

While running a recursive feature elimination using Naïve Bayes limits the feature subspace to 3 features examining the flattened correlation matrix in figure 4 tells us that the basic assumptions of Naïve Bayes are violated in the processed data in that the features are not independent.

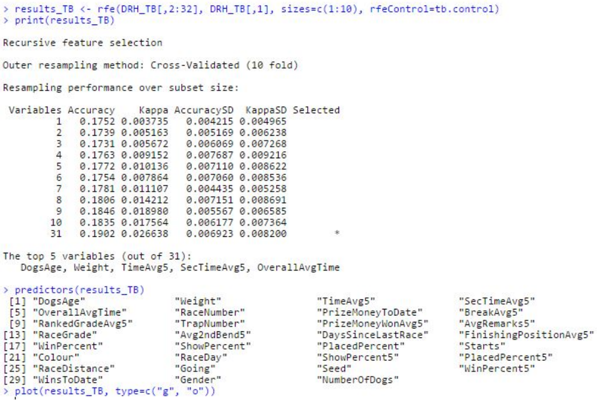
2. **Random Forests** - Random forests are the result of combining decision trees such that each tree depends on the values of a randomly sampled independent vector whereby the entire forest is distributed homogeneously (Breiman (2001)). The output of the random forest RFE is shown in Figure 9. This depicts the top 10 features selected to be *NumberOfDogs, EstTimeAvg5, OverallAvgTime, PrizeMoneyWonAvg5, DogsAge, FinishingPositionAvg5, SecTimeAvg5, PlacedPercent, RankedGradeAvg5 and Avg2ndBend5.*



```
> results <- rfe(DRH[,2:25], DRH[,1], sizes=c(1:5), rfeControl=control)
Loading required package: randomForest
randomForest 4.6-12
Type rfNews() to see new features/changes/bug fixes.

Attaching package: ‘randomForest’

The following object is masked from ‘package:ggplot2’:

    margin

>
> print(results)

Recursive feature selection

Outer resampling method: Cross-Validated (10 fold)

Resampling performance over subset size:

 Variables  RMSE Rsquared   RMSESD RsquaredSD Selected
         1 1.514 0.004995 0.048126   0.005261
         2 1.652 0.014548 0.029510   0.005815
         3 1.660 0.019582 0.006918   0.004955
         4 1.654 0.026555 0.004984   0.004402
         5 1.651 0.030234 0.006320   0.005061
        24 1.639 0.045303 0.006543   0.005324        *

The top 5 variables (out of 24):
   NumberOfDogs, EstTimeAvg5, OverallAvgTime, PrizeMoneyWonAvg5, DogsAge

> predictors(results)
 [1] "NumberOfDogs"         "EstTimeAvg5"          "OverallAvgTime"
 [4] "PrizeMoneyWonAvg5"    "DogsAge"              "FinishingPositionAvg5"
 [5] "SecTimeAvg5"          "PlacedPercent"        "RankedGradeAvg5"
[10] "Avg2ndBend5"          "ShowPercent"          "RaceGrade"
[13] "AvgRemarks5"          "BreakAvg5"            "WinPercent5"
[16] "ShowPercent5"         "PlacedPercentage5"    "TrapNumber"
[19] "Weight"               "RaceDay"              "RaceDistance"
[22] "Seed"                 "RaceNumber"           "Going"
```

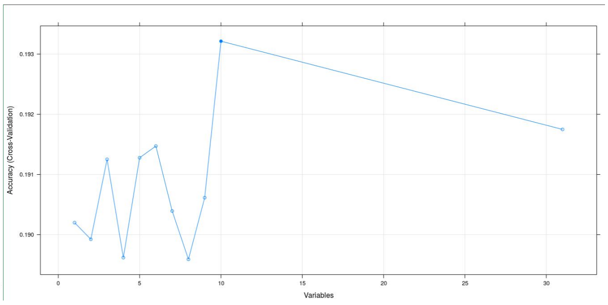
Figure 9: RFE with Random Forest Function

3. **Tree Bagging** - is an ensemble method which uses decision trees to generate multiple versions of a predictor and aggregate the result (Breiman (1996)). The top 10 features returned using RFE with Tree Bagging are *DogsAge, Weight, TimeAvg5, SecTimeAvg5, OverallAvgTime, RaceNumber, RankedGradeAvg5, PrizeMoneyToDate, BreakAvg5 and TrapNumber.*



```
> results_TB <- rfe(DRH_TB[,2:32], DRH_TB[,1], sizes=c(1:10), rfeControl=tb.control)
> print(results_TB)

Recursive feature selection

Outer resampling method: Cross-Validated (10 fold)

Resampling performance over subset size:

 Variables Accuracy    Kappa AccuracySD  KappaSD Selected
         1   0.1752 0.003735   0.004215 0.004965
         2   0.1739 0.005163   0.005169 0.006238
         3   0.1731 0.005672   0.006069 0.007268
         4   0.1763 0.009152   0.007687 0.009216
         5   0.1772 0.010136   0.007110 0.008622
         6   0.1754 0.007864   0.007060 0.008536
         7   0.1781 0.011107   0.004435 0.005258
         8   0.1806 0.014212   0.007151 0.008691
         9   0.1846 0.018980   0.005567 0.006585
        10   0.1835 0.017564   0.006177 0.007364
        31   0.1902 0.026638   0.006923 0.008200        *

The top 5 variables (out of 31):
   DogsAge, Weight, TimeAvg5, SecTimeAvg5, OverallAvgTime

> predictors(results_TB)
 [1] "DogsAge"           "Weight"            "TimeAvg5"           "SecTimeAvg5"
 [5] "OverallAvgTime"    "RaceNumber"        "PrizeMoneyToDate"   "BreakAvg5"
 [9] "RankedGradeAvg5"   "TrapNumber"        "PrizeMoneyWonAvg5"  "AvgRemarks5"
[13] "RaceGrade"         "Avg2ndBend5"       "DaysSinceLastRace"  "FinishingPositionAvg5"
[17] "WinPercent"        "ShowPercent"       "PlacedPercent"      "Starts"
[21] "Colour"            "RaceDay"           "ShowPercent5"       "PlacedPercent5"
[25] "RaceDistance"      "Going"             "Seed"               "WinPercent5"
[29] "WinsToDate"        "Gender"            "NumberOfDogs"
> plot(results_TB, type=c("g", "o"))
```

(a) Output of Tree Bagging RFE



(b) Plot Of Tree Bagging

Figure 10: Tree Bagging Recursive Feature Elimination

4.3.5 Evaluation of Feature Selection Results

The results of the caret & randomForest wrapper, embedded treebagging and embedded randomForest are combined to create a final subset of the data for use in the final stage of the modelling process. The top 10 of each of these tests are combined in order to ascertain if there are any features which are prevalent across all feature selection methods. The ranked table is shown in Table 2.

Table 2: Top Ten Features Selected

Features Selected		
caret & randomForest	**RFE with RandomForest**	**RFE with TreeBagging**
DogsAge*	NumberOfDogs	DogsAge*
OverallAvgTime*	TimeAvg5*	Weight
SecTimeAvg5*	OverallAvgTime*	TimeAvg5*
TimeAvg5*	PrizeMoneyWonAvg5	SecTimeAvg5*
Weight	DogsAge*	OverallAvgTime*
RankedGradeAvg5*	FinishingPositionAvg5	RaceNumber
PrizeMoneyWonAvg5	SecTimeAvg5*	PrizeMoneyToDate
BreakAvg5	PlacedPercent	BreakAvg5
RaceNumber	RankedGradeAvg5*	RankedGradeAvg5*
DaysSinceLastRace	Avg2ndBend5	TrapNumber

The asterisk beside a feature is to highlight that it was selected as a top 10 ranking feature in all 3 methods used. These 5 variables were selected for the final subset data. 5 more features for the prediction models were selected by choosing the highest ranked features amongst the 3 methods deployed. The final feature subset selected for inputting into the neural network is *DogsAge, OverallAvgTime, SecTimeAvg5, TimeAvg5, RankedGradeAvg5, Weight, RaceNumber, PrizeMoneyWonAvg5, BreakAvg5 and Avg2ndBend5* .

5 Evaluation

For the purpose of benchmarking this research against the predictions of a greyhound expert the dataset is split in a 60/20/20 ratio for training, validation and testing. The greyhound expert's predictions are scrapped from embedded pop-ups on the IGB's website and the percentage of first place finishers correctly predicted is derived. A limitation of this research lies therein. This research, similar to the greyhound expert, attempts to predict the finishing order of greyhounds in a race; however the expert makes 3 predictions per race (what greyhound will win the race, what two greyhounds will finish in the top 2 in any order and what 3 greyhounds will finish in the top 3 in any order). It is necessary to only choose the percentage of first place finishes correctly predicted as the other 2 predictions turn the problem from a 6 class multi-class problem to a binary classification problem. The greyhound expert correctly predicted 23.7% of first placed finishers in the time frame used in this research (Jan. 2009 - Sept. 2016).

5.1 Model Performance

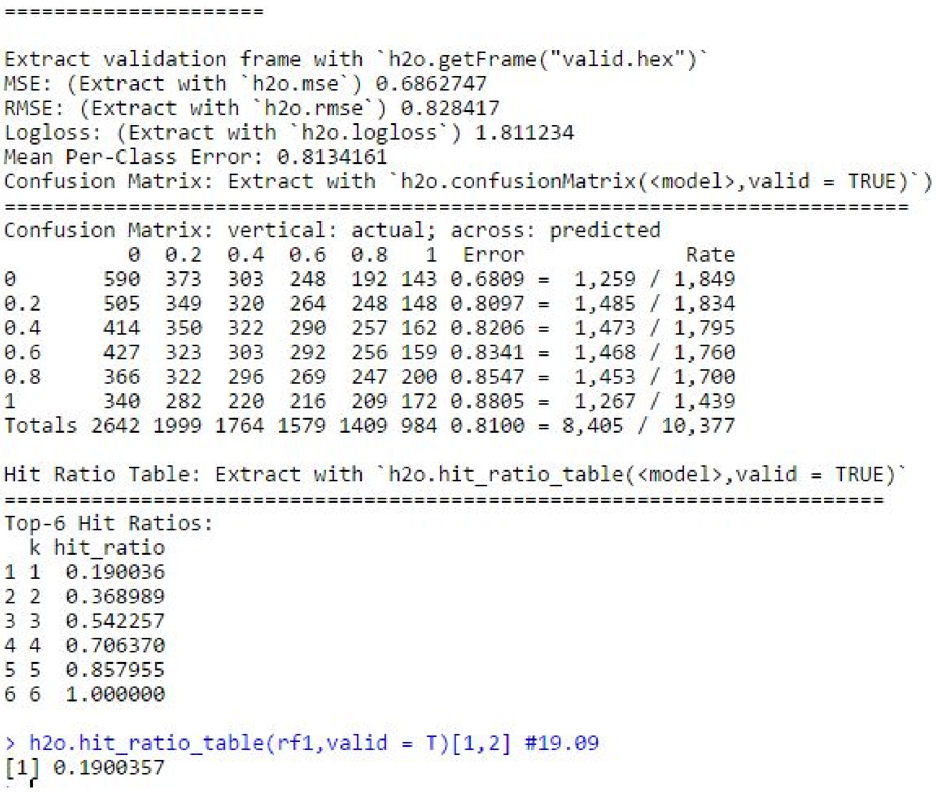
An important design criterion for model performance is choosing the correct parameters and while tuning these parameters the test set is not utilised so as to avoid the model learning from iterations over the test set.

5.1.1 Neural Network

In order to emulate the research in the field of greyhound racing by Chen et al. (1994) and Johansson and Sönströd (2003); who used shallow neural networks in their predictions; the optimal feature subset chosen following feature selection was inputted into a Deep Learning Neural Network using the *H2O* package in R. This research uses deep learning neural networks in its attempts at classifying the finishing position of a greyhound. Deep Learning reduces the complexity of an algorithm but is better suited to larger datasets. The neural network was ran several times across different subsets of the data and the average prediction performance was found to be 18.92%.

5.1.2 Random Forest

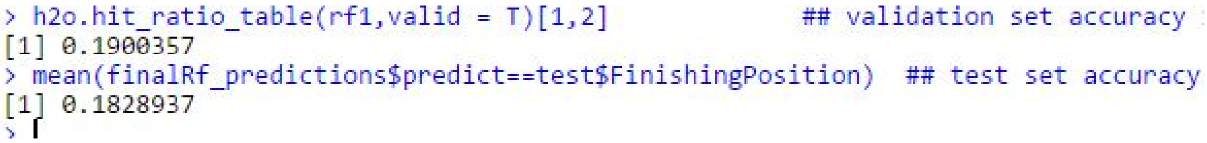
The image in figure 12 shows the output of running a random forest model on the feature subset using the *H2O* package. As can be seen this model accurately predicts 19% of finishing positions correctly when tested against the validation set.



```
=====================

Extract validation frame with `h2o.getFrame("valid.hex")`
MSE: (Extract with `h2o.mse`) 0.6862747
RMSE: (Extract with `h2o.rmse`) 0.828417
Logloss: (Extract with `h2o.logloss`) 1.811234
Mean Per-Class Error: 0.8134161
Confusion Matrix: Extract with `h2o.confusionMatrix(<model>,valid = TRUE)`)
=========================================================================
Confusion Matrix: vertical: actual; across: predicted
          0  0.2  0.4  0.6  0.8   1  Error             Rate
0       590  373  303  248  192 143 0.6809 =  1,259 / 1,849
0.2     505  349  320  264  248 148 0.8097 =  1,485 / 1,834
0.4     414  350  322  290  257 162 0.8206 =  1,473 / 1,795
0.6     427  323  303  292  256 159 0.8341 =  1,468 / 1,760
0.8     366  322  296  269  247 200 0.8547 =  1,453 / 1,700
1       340  282  220  216  209 172 0.8805 =  1,267 / 1,439
Totals 2642 1999 1764 1579 1409 984 0.8100 = 8,405 / 10,377

Hit Ratio Table: Extract with `h2o.hit_ratio_table(<model>,valid = TRUE)`
=======================================================================
Top-6 Hit Ratios:
  k hit_ratio
1 1  0.190036
2 2  0.368989
3 3  0.542257
4 4  0.706370
5 5  0.857955
6 6  1.000000

> h2o.hit_ratio_table(rf1,valid = T)[1,2] #19.09
[1] 0.1900357
```

Figure 11: Random Forest in H2O on Validation Set

When ran against the test set a similar prediction rate of 18.28% is recorded.



```
> h2o.hit_ratio_table(rf1,valid = T)[1,2]            ## validation set accuracy
[1] 0.1900357
> mean(finalRf_predictions$predict==test$FinishingPosition)  ## test set accuracy
[1] 0.1828937
>
```

Figure 12: Random Forest in H2O on Test Set

5.2 Model Evaluation

The reasoning behind the choice of deep learning methods for model building were a result of R Studio ”hanging” for long periods of time when attempting to run algorithms. This hanging state ensured it was not possible to ascertain if R was working in the background or if the instance had indeed hung. The research, while attempting to improve real time responsiveness and hanging R instances by using deep learning methods, failed to account for the basic premise of deep learning performance in that it requires relatively large datasets to work competently. As a result of this it is necessary for future research to utilise shallow machine learning techniques to ascertain whether the machine chosen optimal feature subset when combined with shallow machine learning techniques can better rival the 23.7% hit ratio of the resident greyhound expert in Shelbourne Park greyhound stadium.

6 Conclusion and Future Work

6.1 Conclusion

While the results of the prediction algorithms combined with the feature subset are less than adequate in predicting greyhound racing results better than the average gambler some interesting insights were discovered in the completion of this research. Machine learnt feature selection must at all times be accompanied by domain knowledge; it is in combining the two that an optimal feature set can be obtained.

The failure to adequately select an appropriate model to use in this research ensured that the answer to whether the non manual process of feature selection can improve on previous research in this domain remains inconclusive. Although feature selection plays an important role in data mining it is only 1 step within the iterative framework. It alone, cannot adequately account for a model's success or failure; rather the amalgamation of feature engineering, feature selection and model selection when combined optimally account for a model's success rate.

6.2 Future Works

This research focuses on feature selection in the domain of greyhound racing. Modelling is done on each individual greyhound separately in order to ascertain their individual probability of winning a race given their historical performance data. A possible future work would be to use the features subset developed in this research combined with conditional statistics to allow that the sum of probabilities for all greyhounds in a given race equal 1 so that within-race competition can be accounted for.

A deep learning algorithm improves with added data. A future focus to improve model performance could be to generalise the feature selection across all 28,271 races scraped from tracks throughout Ireland.

Acknowledgements

This research would not have been possible without the help of my supervisor **Dr. Oisín Creaner**, alongside **Mr. Michael Bradford** and **Dr. Simon Caton**, whose help and guidance throughout the completion of this Masters programme proved invaluable. I would like to take this opportunity to thank them sincerely for their generosity of time and spirit.

I would also like to thank the **Irish Greyhound Board** who generously allowed me to use their data in the completion of this Masters programme.

Without the continued support of my family and friends I would not have had the courage nor strength to be able to complete my Masters. Particularly, I would like to dedicate this thesis to my mam, **Eileen Lyons**, who sadly passed away before its completion. *Ar dheis Dé go raibh a h-anam.*

References

Azevedo, A. and Santos, M. F. (2008). Kdd, semma and crisp-dm: a parallel overview., *in* A. Abraham (ed.), *IADIS European Conf. Data Mining*, IADIS, pp. 182–185.
URL: *http://dblp.uni-trier.de/db/conf/iadis/dm2008.htmlAzevedoS08*

Breiman, L. (1996). Bagging predictors, *Machine Learning* **24**(2): 123–140.
URL: *http://dx.doi.org/10.1007/BF00058655*

Breiman, L. (2001). Random forests, *Machine Learning* **45**(1): 5–32.
URL: *http://dx.doi.org/10.1023/A:1010933404324*

Brownlee, J. (2014). Machine learning mastery.
URL: *http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/*

Butler, J., Tsang, E. P. K. and Sq, C. C. (1998). EDDIE beats the bookies.

Chen, H., Rinde, P. B., She, L., Sutjahjo, S., Sommer, C. and Neely, D. (1994). Expert prediction, symbolic learning, and neural networks: An experiment on greyhound racing., *IEEE Expert* **9**(6): 21–27.
URL: *http://dblp.uni-trier.de/db/journals/expert/expert9.htmlChenRSSSN94*

Dash, M. and Liu, H. (1997). Feature selection for classification, *Intelligent Data Analysis* **1**: 131–156.

Dash, M. and Liu, H. (2003). Consistency-based search in feature selection, *Artificial Intelligence* **151**(1–2): 155 – 176.
URL: *http://www.sciencedirect.com/science/article/pii/S0004370203000791*

Davoodi, E. and Khanteymoori, A. R. (2010). Horse racing prediction using artificial neural networks, *Proceedings of the 11th WSEAS International Conference on Nural Networks and 11th WSEAS International Conference on Evolutionary Computing and 11th WSEAS International Conference on Fuzzy Systems*, NN'10/EC'10/FS'10, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, pp. 155–160.
URL: *http://dl.acm.org/citation.cfm?id=1863431.1863457*

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996). The kdd process for extracting useful knowledge from volumes of data, *Commun. ACM* **39**(11): 27–34.
URL: *http://doi.acm.org/10.1145/240455.240464*

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines, *Machine Learning* **46**(1): 389–422.
URL: *http://dx.doi.org/10.1023/A:1012487302797*

Han, J. (2005). *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Hand, D., Mannila, H. and Smyth, P. (2001). *Principles of Data Mining*, A Bradford book, MIT Press.
URL: *https://books.google.ie/books?id=SdZ-bhVhZGYC*

Hodge, V. J. and Austin, J. (2004). A survey of outlier detection methodologies, *Artificial Intelligence Review* **22**(2): 85–126.
URL: *http://dx.doi.org/10.1007/s10462-004-4304-y*

Johansson, U. and Sönströd, C. (2003). Neural networks mine for gold at the greyhound racetrack, *Proceedings of the International Joint Conference on Neural Networks* pp. 1798 – 1801 vol.3.

Kuhn, M. (2016). A short introduction to the caret package.
URL: *https://cran.r-project.org/web/packages/caret/vignettes/caret.pdf*

Kursa, M. B. and Rudnicki, W. R. (2016). Boruta package in r - documentation.
URL: *https://cran.r-project.org/web/packages/Boruta/Boruta.pdf*

Lantz, B. (2013). *Machine Learning with R*, Packt Publishing.

Liu, B. (2010). Sentiment analysis and subjectivity, *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca.*

Lyons, A. (2016). RIC research proposal for greyhound racing predictions modelling.

McCabe, A. and Trevathan, J. (2008). Artificial intelligence in sports prediction, *Fifth International Conference on Information Technology: New Generations* .

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis, *Found. Trends Inf. Retr.* **2**(1-2): 1–135.
URL: *http://dx.doi.org/10.1561/1500000011*

Pudaruth, S., Medard, N. and Dookhun, Z. B. (2013). Article: Horse racing prediction at the champ de mars using a weighted probabilistic approach, *International Journal of Computer Applications* **72**(5): 37–42. Full text available.

Saeys, Y., Inza, I. n. and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics, *Bioinformatics* **23**(19): 2507–2517.
URL: *http://dx.doi.org/10.1093/bioinformatics/btm344*

Schumaker, R. P. and Johnson, J. W. (2008). An investigation of svm regression to predict longshot greyhound races, *Communications of the IIMA: Vol. 8: Iss. 2, Article 7* pp. 67–82.
URL: *"http://scholarworks.lib.csusb.edu/ciima/vol8/iss2/7"*

Silverman, N. and Suchard, M. (2013). Predicting horse race winners through a regularized conditional logistic regression with frailty, *Journal of Prediction Markets* **7**(1): 43–52.
URL: *http://EconPapers.repec.org/RePEc:buc:jpredm:v:7:y:2013:i:1:p:43-52*

Williams, J. and Li, Y. (2008). A case study using neural networks algorithms: Horse racing predictions in jamaica., *in* H. R. Arabnia and Y. Mun (eds), *IC-AI*, CSREA Press, pp. 16–22.
URL: *http://dblp.uni-trier.de/db/conf/icai/icai2008.htmlWilliamsL08*

Witten, I. H., Frank, E. and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd edn, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

A Python Script Flow Chart

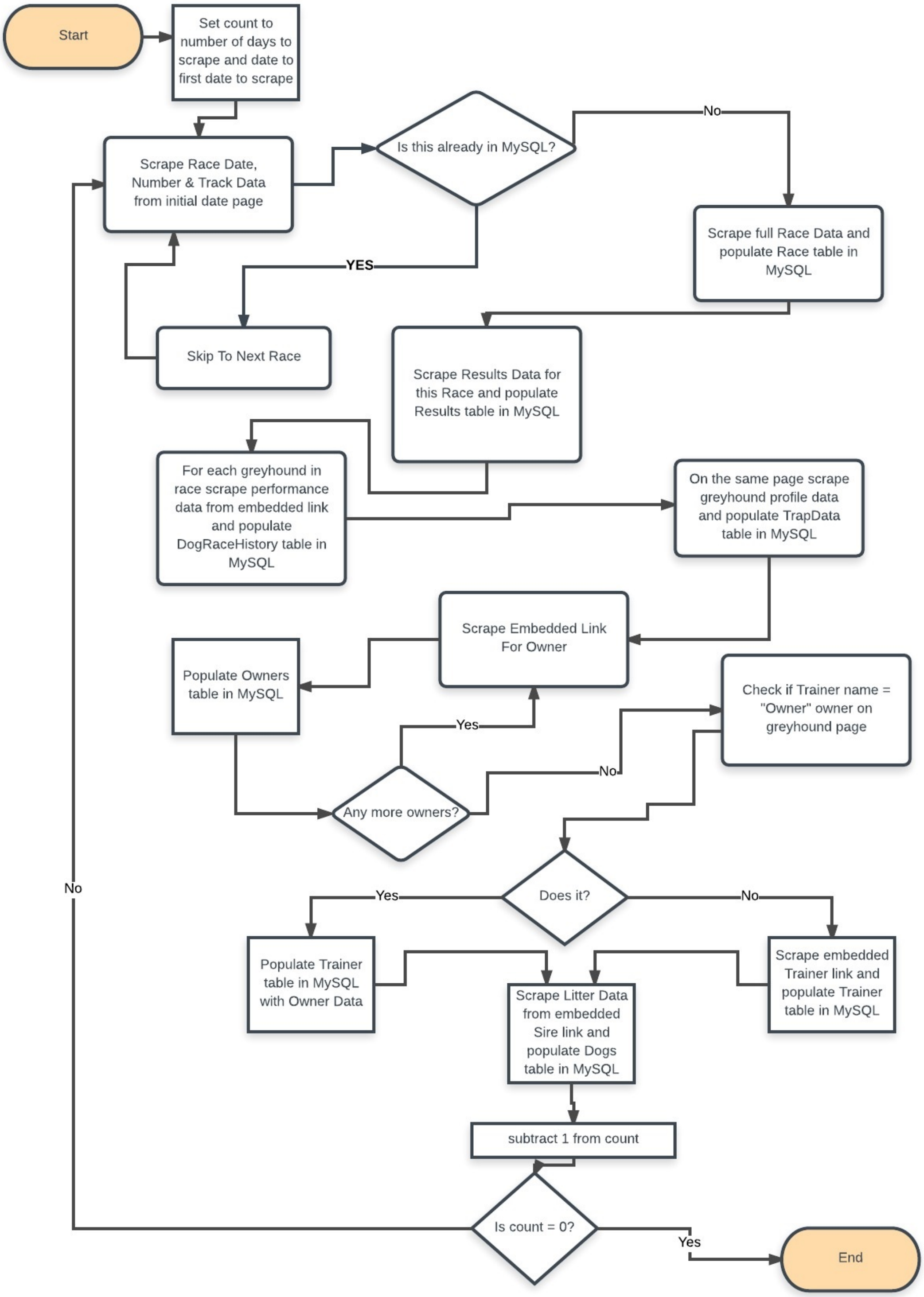

Start
Set count to number of days to scrape and date to first date to scrape
Scrape Race Date, Number & Track Data from initial date page
Is this already in MySQL?
No
YES
Scrape full Race Data and populate Race table in MySQL
Skip To Next Race
Scrape Results Data for this Race and populate Results table in MySQL
For each greyhound in race scrape performance data from embedded link and populate DogRaceHistory table in MySQL
On the same page scrape greyhound profile data and populate TrapData table in MySQL
Scrape Embedded Link For Owner
Populate Owners table in MySQL
Yes
Check if Trainer name = "Owner" owner on greyhound page
No
Any more owners?
No
Does it?
Yes
No
Populate Trainer table in MySQL with Owner Data
Scrape Litter Data from embedded Sire link and populate Dogs table in MySQL
Scrape embedded Trainer link and populate Trainer table in MySQL
subtract 1 from count
Is count = 0?
Yes
End

Figure 13: Python Script Flow Chart

. . .