# Udacity Machine Learning Capstone Project: August 2020

# Greyhound Race Prediction

# Tony Ward

# Definition

## Project overview

*Greyhound racing is an organized, competitive sport in which greyhounds are raced around a track. As with horse racing, greyhound races often allow the public to bet on the outcome.1*

I was introduced to betting on greyhounds whilst at University. About once a week a group of us would head down to the local book maker and bet a few pounds, which provided entertainment and way of winding down between lecturers. I remember checking the form guide and comparing each’s dogs previous winning times which soon became overwhelming. I realised at that time there was an opportunity to use the past racing results to guide the betting decision in an intelligent way.

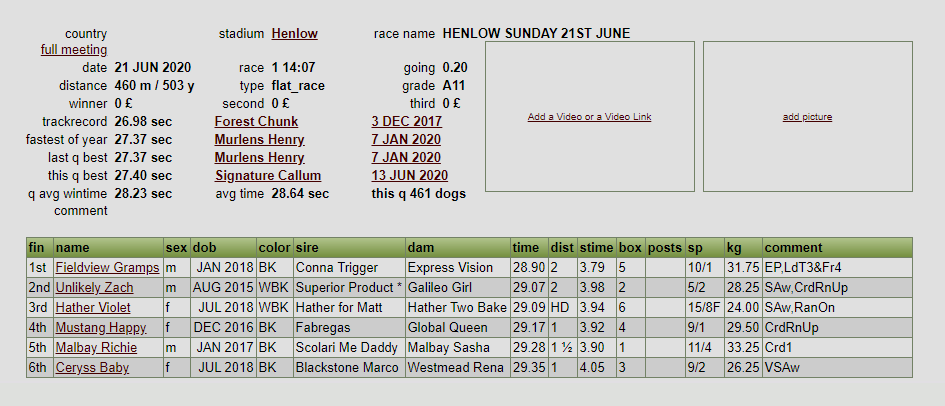
## Problem Statement

This project is focused on using historic data on greyhound races to investigate whether a machine learning system can accurately predict the results of the UK greyhound racing scene.

### Dataset and Inputs

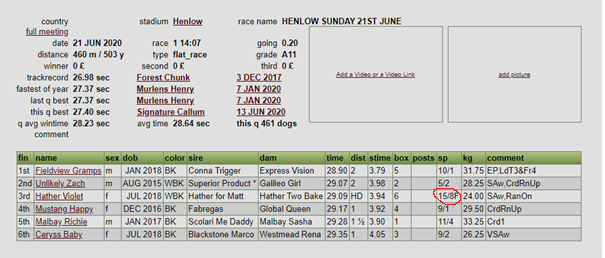
<http://www.greyhound-data.com/> is website which provides information about greyhounds from all over the world with pedigree information drawn from the last four centuries. Online are 4,549,034 race results and 2,333,577 greyhound pedigrees.

Here is an example of the data available for a given race



### Solution

As part of this project I scraped 18 years’ worth of race data for one stadium (Monmore), which constitutes approximately 55k races. Next, I designed a Postgres SQL database and inserted the data for later analysis. Using SQL queries I was able to construct several features which were later used to build and validate a machine learning model whose goal it was to predict the eventual race winner. This model has been evaluated against the benchmark given by the bookmakers favourite, which is indicated as either a F (favourite) or JF (joint favourite) in the sp (starting price) column



## Metrics

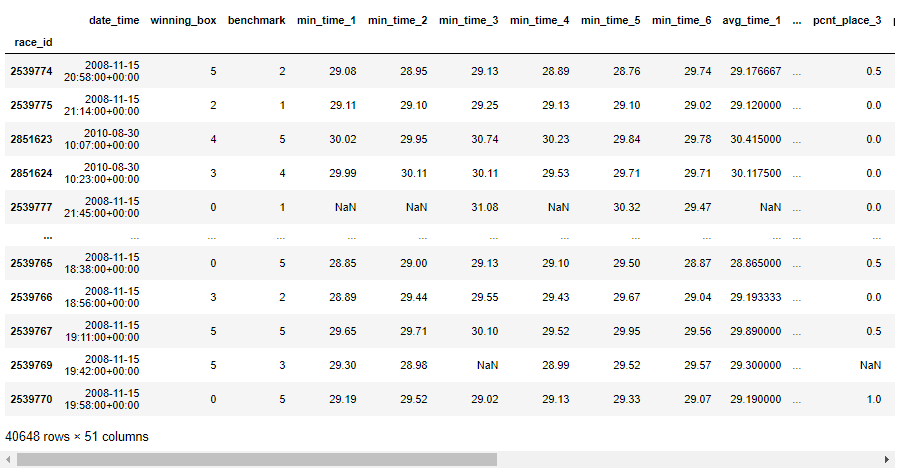
For each race we must predict a race winner. This prediction will be compared to the eventual race winner.

|  |  |  |
| --- | --- | --- |
| Greyhound | Predicted Winner | Actual Winner |
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| 3 | 0 | 1 |
| 4 | 1 | 0 |
| 5 | 0 | 0 |
| 6 | 0 | 0 |

To evaluate the model against the benchmark we will use the accuracy metric. This will tell us the proportion of races where we correctly forecasted the race winner.

## Framing the Problem

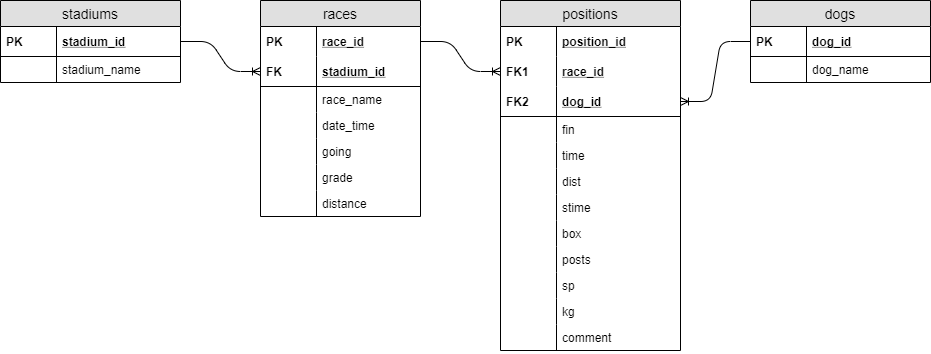
We treat this as a multi class classification problem where we must predict the box (1 to 6) that the winning dog started the race from. We will organise our data such that each row represents an individual race, with features relating to each of the 6 dogs running in the race. For example



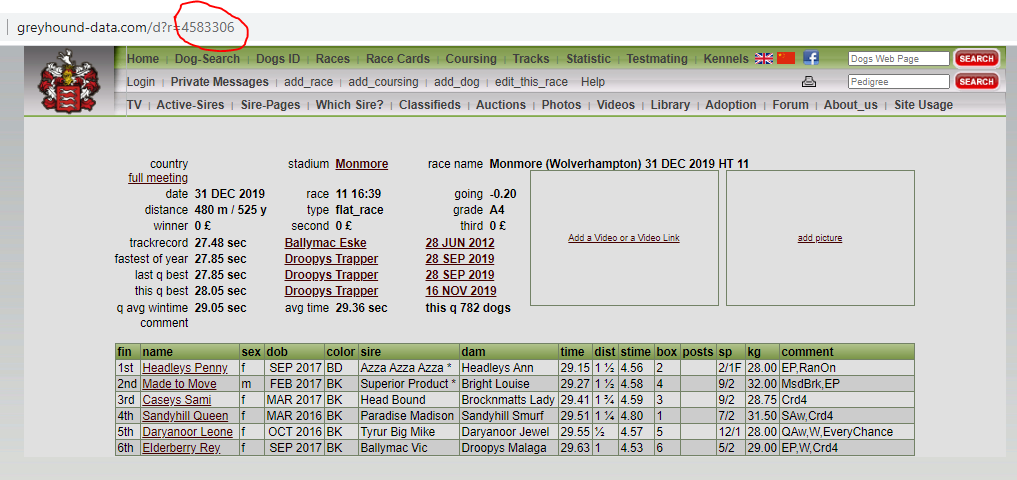
Rows are indexed by race\_id which is a unique identifier for each race. The winning\_box is the target variable and the benchmark column relates to the bookmakers favourite. Note how that both these columns are indexed to start at zero rather than one – this is purely because lightgbm requires the data this way. Min\_time\_1 is the fastest time run by the dog starting from box 1 in the last 25 days.

# Data Exploration and Visualisation

Data is stored in a Postgres database under the following schema

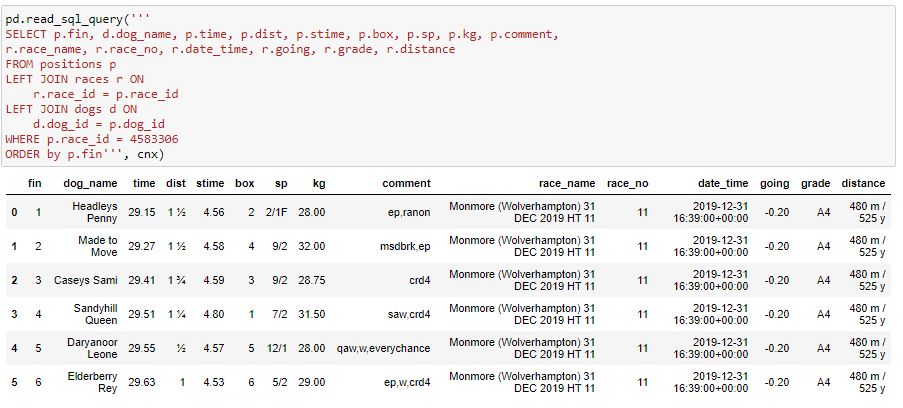


A typical race contains several pieces of information. An example race is provided below



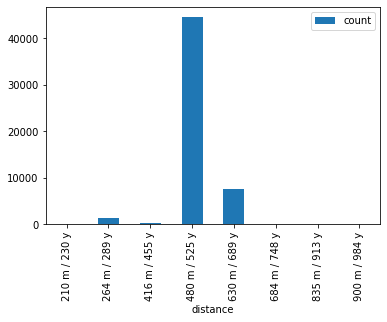
The URL contains a unique identifier which we call race\_id. The race\_id is the primary key in both the races and positions tables.

To avoid duplication and protect the integrity of the relationships we have chosen to store the data in third normal form. We can reconstruct the above table using the following query (note that due to time constraints we did not capture the sex, dob, colour, sire, dam information of the dog)



## Distance

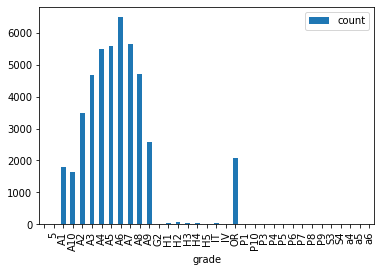
Dogs race over a range of distances



For simplicity we will restrict ourselves to the most frequently occurring race distance of 480m.

## Grade

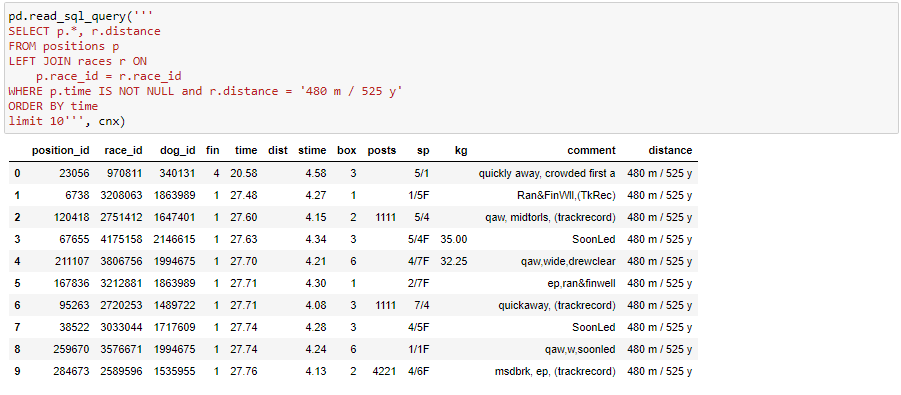
Races are organised in such a way so that dogs of similar ability race each other. The most competitive grade that has the fastest dogs are Open Races (OR). In these races’ dogs travel across the country and race against dogs from different stadiums. Grades A1 to A10 are local races where the dogs will race at the same stadium each week. A1 has the fastest dogs and A10 the slowest. Dogs are moved up and down a grade at the discretion of the race director based on recent form. A dog is automatically moved up a grade if it wins a race, and moves down a grade if it fails to place (finish in the top 3) for three races in a row



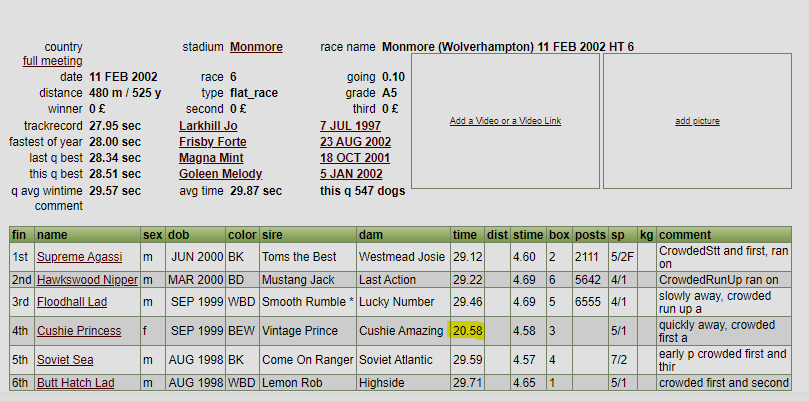
We will only include races of grade A1 – A10 in our modelling, and exclude Open Races and everything else. It is hoped that there is more chance of successfully predicting races where the dogs run on the same track.

## Time

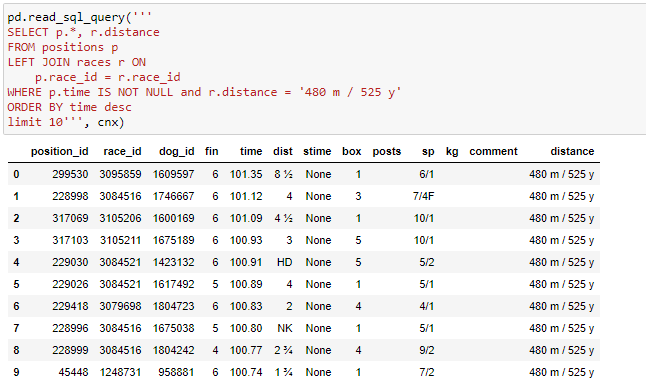
Lets have a look at the fastest race times for the chosen race distance of 480m



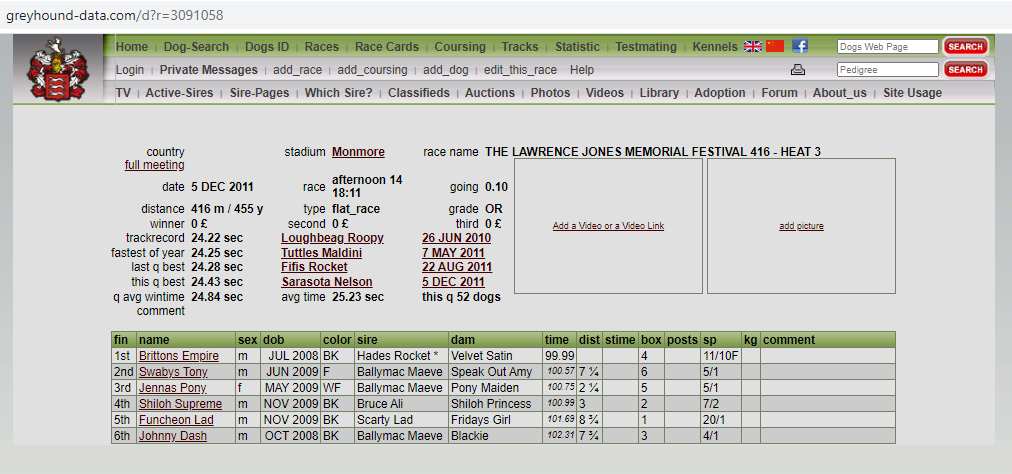
Looking at the race with this quickest time (race\_id = 970811) at the following url <http://www.greyhound-data.com/d?r=970811> We can see that the dog finished first in 29.12 however the dog that finished fourth finished in 20.58 which is clearly an error.



Now lets have a look at the races with the slowest times



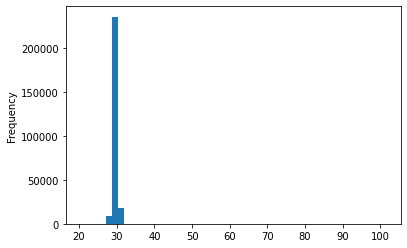
There looks to be something strange going on here. If we inspect the race with the slowest times 3091058 <http://www.greyhound-data.com/d?r=3091058> we can see that the first dog recorded a time of 99.99 and the rest of the times are in italics. These times are unusually slow.

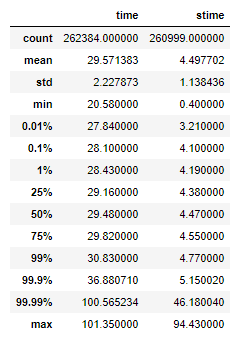


Clicking on the dog "Brittons Empire" that won the race we can see the history of its past races. This dog typically completes the race in under 30 seconds, and the 99.99 must be a data quality issue



A histogram of race times and summary statistics confirms that those race times of 99+ must be an error.





we decide to remove any race from the data the contains times outside the following ranges

* 26 < time < 40
* 3 < stime < 6

# Methodology

Ours is a supervised learning task with structured tabular data, therefore a natural choice is to use gradient boosting.

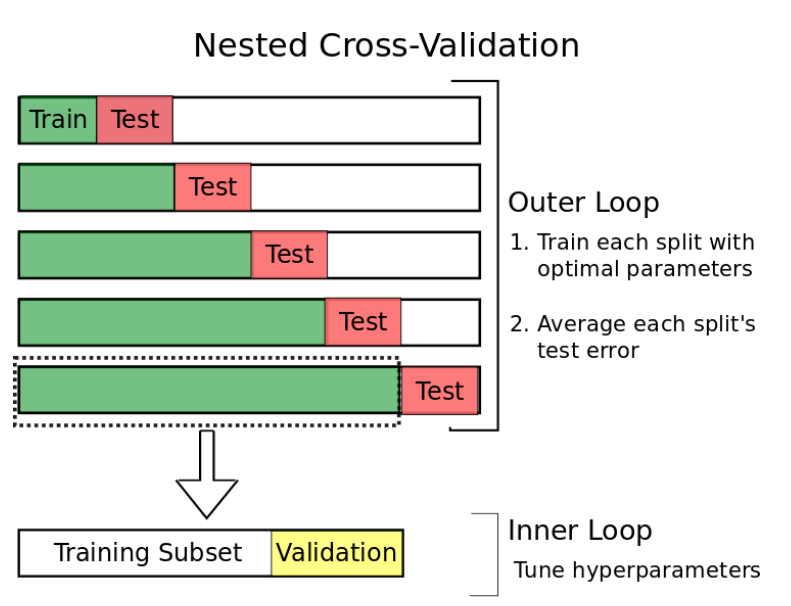
## Features

The following features were constructed for each dog in the race. Summary statistics are calculated based on the last 25 days.

* Minimum time
* Average time
* Minimum stime (stime is time to the first bend)
* Average stime
* Average finish position
* Win percentage
* Place percentage (a place is coming in the top 2)
* Show percentage (a show is coming in the top 3)

## Model Assessment

I decided to keep 2019 as a final test set, and used 2018 as a validation sample to test various modelling strategies (different features/varying amounts of training data etc). Since there is a time component to our problem I selected to use nested cross validation as follows



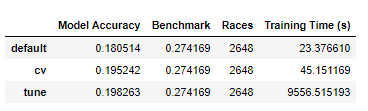
Source : <https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9>

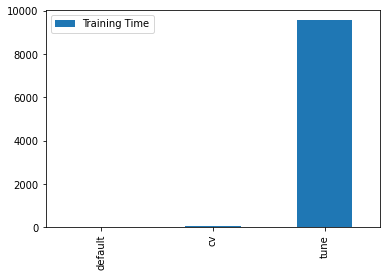
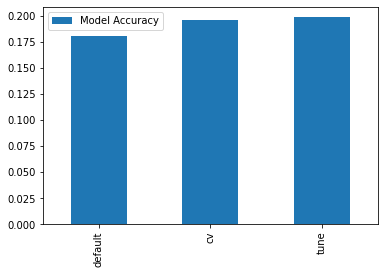
### Training the Algorithm

I have implemented three methods of training the model

1. **Default** - using the default parameters with no validation/early stopping
2. **CV** - Using the default parameters but tuning the number of boosting iterations using cross validation/early stopping to maximise accuracy.
3. **Tune** Using cross validation/early stopping to find the optimal hyperparameters and boosting iterations using Bayesian optimisation to maximise accuracy.

The default method produced the quickest result but also the lowest accuracy. CV took slightly longer but provided a noticeable jump in accuracy. Tuning the hyperparameters provided a modest improvement in accuracy for a very large increase in training time. It is noted that model accuracy is somewhat behind the benchmark, although it is ahead of random guessing (which is 0.167).



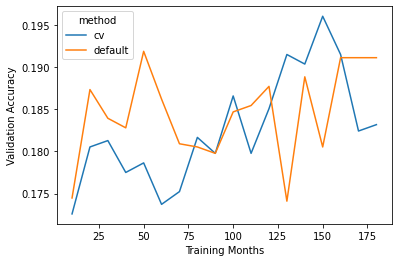


## Complications

For a while I was getting terrible performance, worse than random change. I discovered this was because the algorithm requires the labels to be integers starting from 0, where as I had supplied them starting from 1! Subtracting one from both the winning box and benchmark resolved this.

## Refinement

Learning curves were developed to understand the impact of providing the algorithm with more data. Two method of training the models (default, cv) were provided with increasing amounts of training data and the average accuracy across the 12 holdout months of 2018 are calculated.

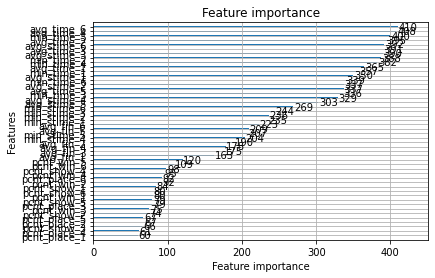


It appears that the cross validation method benefits more from receiving more data than the default parameters. This is most likely due to the fact that the default method trains for a fixed number of iterations, where as the cross validation method will keep training until the validation loss stops decreasing. It is also noted that the learning curve are quite noisy. This is due to the nature of the accuracy metric which is binary in nature. A small change in the predicted probabilities can suddenly change the predicted classification especially in a multi-class setting.

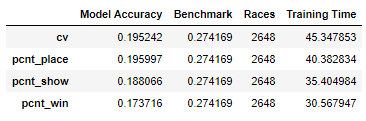
In also subsequent investigations a training method of CV and training months equal to 150 were used.

## Feature Importance

For each of the 12 model runs used in the model assessment we produce a feature importance plot, an example of which is shown below. Average time appears at the top and the percent variables appear at the bottom.

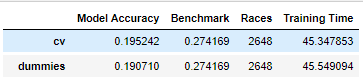


c. The results of which are as follows



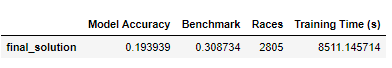
We can see that dropping pcnt\_place has a marginal improvement relative to baseline, but taking out pcnt\_show and pcnt\_win has a detrimental effect on performance.

Lastly I constructued some dummy variables that indicated which of the dogs had the fastest average and minimum times. Unfortunately this reduced performance relative to the baseline



# Results

To provide an honest assessment of model performance we evaluated the final model parameters/process on the previously unseen 2019 data.



Although it is pleasing that our model did not degrade in the test set and it is still ahead of random chance (0.167) we are way off the benchmark model.

### Potential Improvements

Framing the problem as a regression problem where we try and predict the finish position for each dog would potentially make better use of the available information. In addition external data such as weather could be added, and other features based on a dogs past grade could be investigated further.