Management of Boston Marathon Finish Line Spectator Density via Machine Learning

James DeLuca

11/02/2020

Executive Summary, Introduction and Data Overview

Executive Summary

Each year the Boston Marathon draws some 30,000 runners and an estimated 500,000 spectators, injecting an estimated $200 million into the Boston economy each year. In 2013, the spectator density at the finish line made the race a target for a heinous terrorist attack. In 2020 the marathon had to be entirely canceled (for the first time since its 1897 inception) due to the disease transmission risks associated with high population densities and the COVID-19 epidemic. The 2021 Boston Marathon has already been postponed for similar COVID transmission fears. For the Boston Marathon and other Marathon Majors to survive and chart a course forward it will become necessary to simultaneously manage spectator density at key viewing locations while still providing opportunities for family and friends to support the runners on the course, especially at the finish line.

For a race organizer to restrict access to an area of the race course, allowing in only family and friends of runners who are expected to finish within a certain window, the organizers need a good estimate of each runner's finish time and a relatively narrow confidence interval within which they are expected to finish. That confidence interval is derived from the RMSE of the model being used to predict the runner's finish time.

This paper describes the development of a machine learning algorithm which improves upon the existing method used by the Boston Athletic Association (B.A.A.) to project runner finish times. The existing method is to place check-point mats at regular intervals throughout the 42.2km race course. As each runner passes a check-point the runner's average pace from the start to that check-point is calculated and the Final Time is projected by presumption that the runner will complete the entire race at that pace.

The output of this work provides improved finish time predictions that race organizes can use to gate spectator access to restricted areas such as the finish line. The performance of the model is judged by the percent reduction in the RMSE (Root Mean Square Error) calculated at each of the race check-points for the optimized model compared to the B.A.A. baseline projections.

The model described in this paper is trained and tested using data from the 2015 and 2016 Boston Marathons. The performance of this model is validated on the entire list of 2017 Boston Marathon results. In validation this model provides an average improvement in RMSE of **40%** with a peak improvement of ~**47.5%** near the Half Marathon check-point.

Introduction to the Dataset

Each year the B.A.A. publishes the results of the Boston Marathon to their website. The results from 2015, 2016 and 2017 were scraped from the official website by rojour at Kaggle. Many thanks to rojour for his work preparing this data set and providing the Python notebooks to replicate this scraping for other years. The web-scraping notebook prepares csv format file which are used in this project. The features available in this data set are:

- **Bib:** The bib is the number worn by the runner. For most runners, this gives an indication of the runner's qualification time. Some runners enter the marathon via charity or club entry so their bib numbers do not hold any information about the runner's typical speed.
- **Name:** This is the name of the runner; while useful for runners and their friends but is not of interest for this project.
- **Age:** This is the age of the runner on the day of the race in years.
- **M/F:** The B.A.A. classified every runner as either male "M" or female "F". This selection matches the gender selected by the runner when submitting a qualifying race performance for entry.
- **City:** This is the city of the runner's mailing address.
- **State:** For runners from the United States, this feature is the State from the runner's mailing address
- **Country:** This is the Country from the runner's mailing address.
- **Citizen:** If a runner is a citizen of different country than the mailing address then this feature lists the runner's country of citizenship.
- **5K, 10K, 15K, 20K, 25K, 30K, 35K, 40K:** These are the elapsed times for each runner from the start to each of the timing mats every 5km through the course
- **Half, Official Time:** These are the elapsed times when each runner passes the half marathon line and the finish line. The **Official Time** is the dependent variable which we will predict at each Check-Point.
- **Pace:** The pace is average pace in time per mile of each runner up to the latest timing mat which has been crossed. In the scraped data this is always the average pace for the entire race but if the data are sampled through the B.A.A. website during the race this feature will be the average pace up to each runner's most recent update.
- **Proj Time:** In the scraped data this is empty because an Official Time has been registered. During the race this projected time will be the most recent average **Pace** times the total distance (26.2188 miles or 42.195 km).
- **Overall:** This is the placement of each runner relative to all other runners, ordered by Official Time
- **Gender:** This is the placement of each runner relative to all other runners with the same M/F designation.
- **Division:** This is the placement of each runner relative to all other runners with the same M/F designation and the same Age Group.

The definition of the age groups and the qualification standards for each group were collected from the B.A.A.'s history of qualification standards. Data was also collected from the B.A.A.'s history of the official cut-off for entry relative to the qualification standard. For use in this project, this data is downloaded from my GitHub Boston Marathon repository.

Summary of Key Preparation and Analysis Steps

The implementation of these models involves pre-processing the B.A.A. data into a more convenient format. Age Group and Wave information from the B.A.A. website is then joined into the scraped results data. Professional runners are removed from the set and the data sets are cleaned to remove any runners with missing or non-numeric time data. After cleaning the data, exploratory data analysis is performed on the 2015 and 2016 data to understand the key features of the marathon results in order to select the types of models used in this study (linear regression and regression tree random forests) and the form of the final model. The EDA is also used to identify a series of new features which can be calculated from existing features.

The entirety of the 2017 data set is completely withheld from this process for validation of the final. This decision of how to partition training and validation data is made because in production a predictive model will need to be prepared well in advance of race day (when we make a prediction for one runner at the 5km mark the other runners are also still on the course so we cannot have any finish time data to train on from this race). This inherently makes the task more challenging because the environmental conditions (temperature, precipitation, etc) will change from year to year and are not available as predictor variables in the training data.

To train the models, 67% of the data from each of the 2015 and 2016 races are combined into a single data set. The remaining 33% of the data is set aside as a testing set. The decision to train on 67% of the data is to mimic the training and validation split of 2 training races to predict one validation race. This decision ensures a large amount of testing data to help avoid over-fitting. Since there are about 52,000 runners between the two training races the training data set will remain large even with 33% of the data reserved for testing. After the models are trained and tested using this data the training and testing data sets are recombined to retrain the models before validation on the 2017 race.

Three types of models are developed in this project:

- **Final Time Linear Regression** models are trained to make predictions at each check-point of when the runner will finish
  - These models are trained to reduce the biases observed in the Exploratory Data Analysis
  - These follow the average slow-down of runners from each check-point to the finish line observed in 2015 and 2016
- **Next Check-Point Linear Regression Models** are used to predict the time that each runner will reach the next check-point
  - These models are intended to find signs of abnormal slow-down or speed-up seen in the EDA
  - When each runner gets to a check-point their time for the next check-point is predicted
  - When that runner reaches the next check-point the deviation between that time when a similar runner is expected to reach that check-point becomes a feature for use in the regression tree models
- **Final Time Modification Regression Tree Random Forest Models** are developed to predict runner specific deviations from the predictions of the linear regression model
  - These models are intended to reduce the large variances observed in the EDA
  - Runners could have either a smooth slow-down type race where the final time will be best modeled as multiple of what would be naively expected or runners could have step function changes in their finish time associated with bathroom breaks, injuries or the dreaded “wall”
  - For step function type changes an additive error term makes more physical sense than a multiplier
  - At each check-point separate regression tree random forests are trained to model each of these types of behavior

The final model for each check-point is an average of modifications to the linear model from each of the two regression tree models.

Exploratiry Data Analysis and Modeling Methods

Data Preparation

The Boston Marathon results include both professional runners and amateur runners. The performance of the professional runners on the course are not at all representative of how the vast majority of runners will run. The professional runners (any runner with a "F" bib or a bib number <100) are not running to finish a marathon, they are racing for prize money so they are likely to run relatively steady paces or drop out to compete another day. The first few results from the 2015 Boston Marathon illustrate that these runners are not representative of the three hour and forty-seven minute average time run by predominantly USA based runners who average about 40 years old:

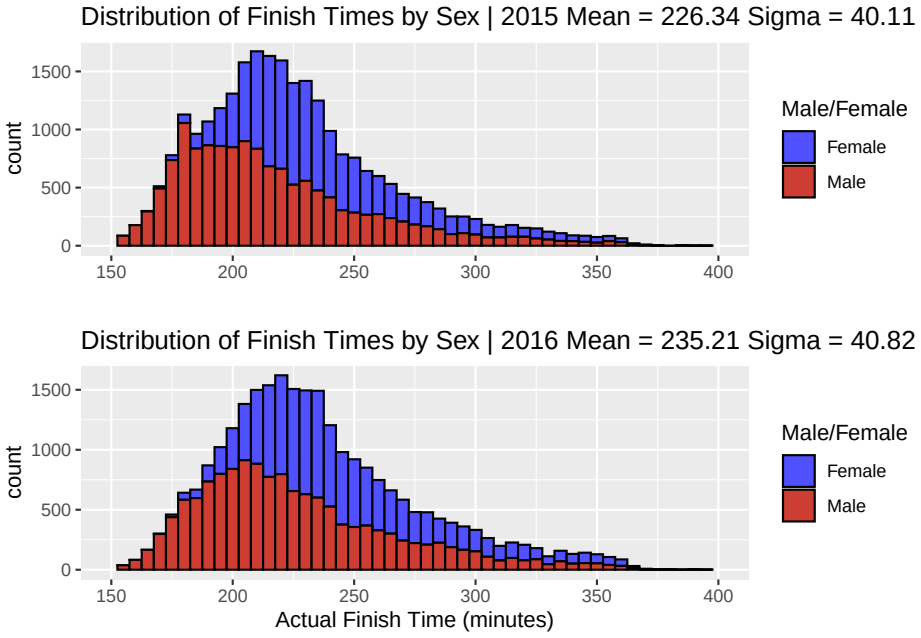
Bib	Name	Age	M.F	Country	X5K	Half	X35K	Official.Time	Pace
3	Desisa, Lelisa	25	M	ETH	0:14:43	1:04:02	1:47:59	2:09:17	0:04:56
4	Tsegay, Yemane Adhane	30	M	ETH	0:14:43	1:04:01	1:47:59	2:09:48	0:04:58
8	Chebet, Wilson	29	M	KEN	0:14:43	1:04:02	1:47:59	2:10:22	0:04:59
11	Kipyego, Bernard	28	M	KEN	0:14:43	1:04:02	1:48:03	2:10:47	0:05:00
10	Korir, Wesley	32	M	KEN	0:14:43	1:04:01	1:47:59	2:10:49	0:05:00
9	Chepkwony, Frankline	30	M	KEN	0:14:44	1:04:02	1:47:59	2:10:52	0:05:00

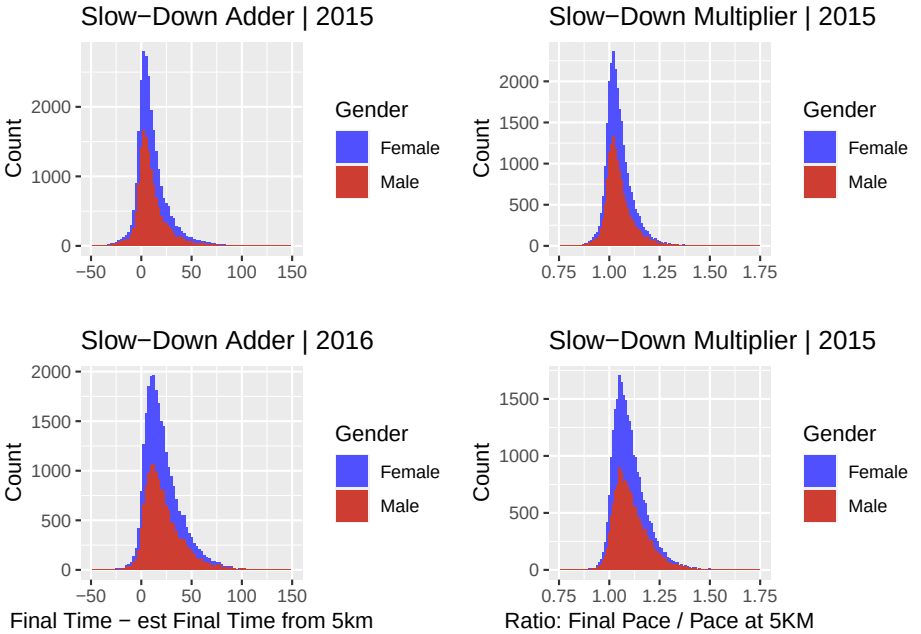
Country	Runners	Mean.Finish	Mean.Age
USA	21567	3:47:34	41.3
CAN	2154	3:37:17	46.1
GBR	277	3:35:03	43.8
MEX	224	3:36:18	42.8
GER	174	3:58:11	48.0
ITA	148	3:49:07	49.3
AUS	130	3:40:57	44.6
JPN	130	4:08:33	54.7

The data is cleaned by removing any runner with an elite bib and removing any runner with missing checkpoint or finish times (this project is not attempting to model the behavior of runners such as **Rosie Ruiz**). The time data are then converted from a h:m:s format into minutes. Bib numbers are used assign each runner to a Wave. The B.A.A. projected finish times at each split are added as well as a few other calculated features (such as pace from checkpoint to checkpoint both in units of minutes and normalized to the runner's pace over the first 10km of the course).

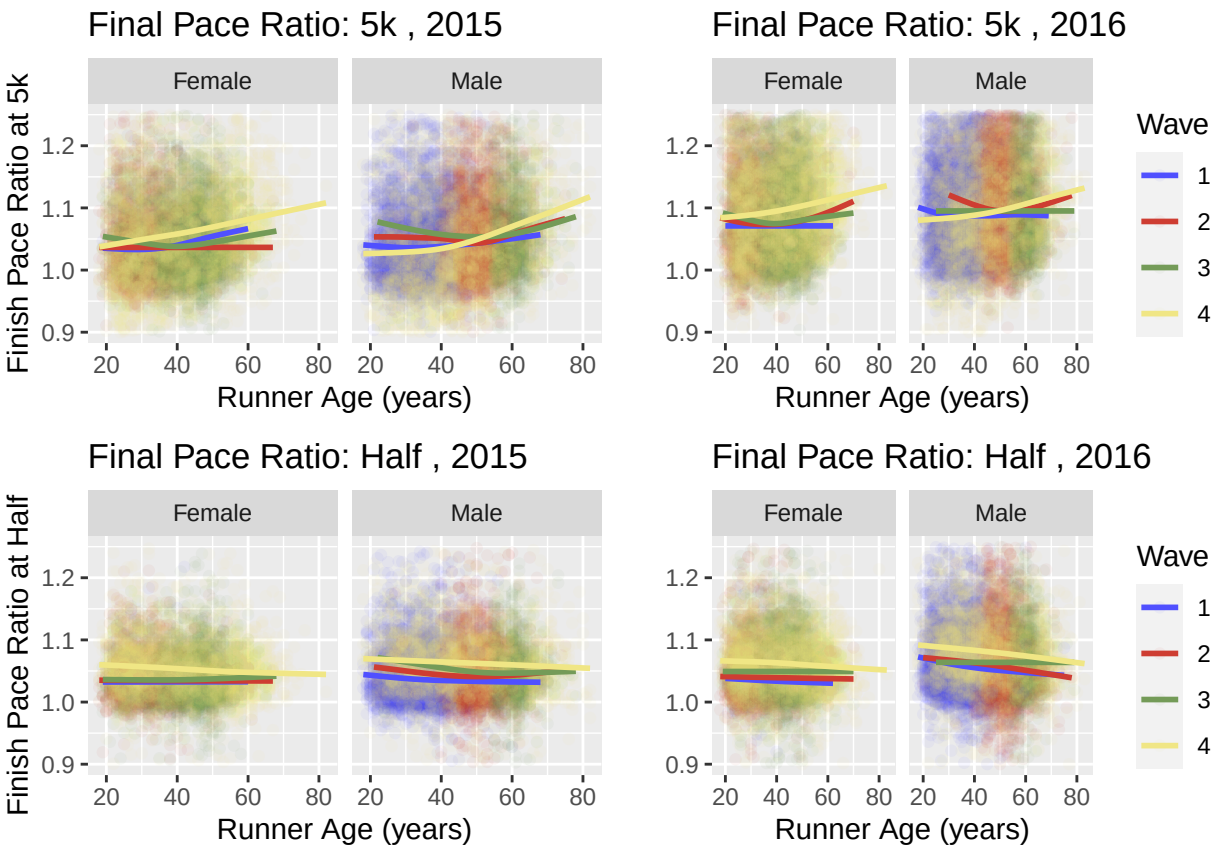
Data Visualization

Although the standard deviation of the finish time is similar in 2015 and 2016, the average is almost 10 minutes faster in 2015 and the width of the distribution of how much runners slow down from their 5km time is broader in 2016. These observations illustrate that the location of the distribution of finish times is not well known for an arbitrary race and the shape and location of the slow-down distributions will also be variable. This means that while a linear regression model may remove some of the general slow-down bias a more flexible model is necessary to discover the factors which predict this variation.

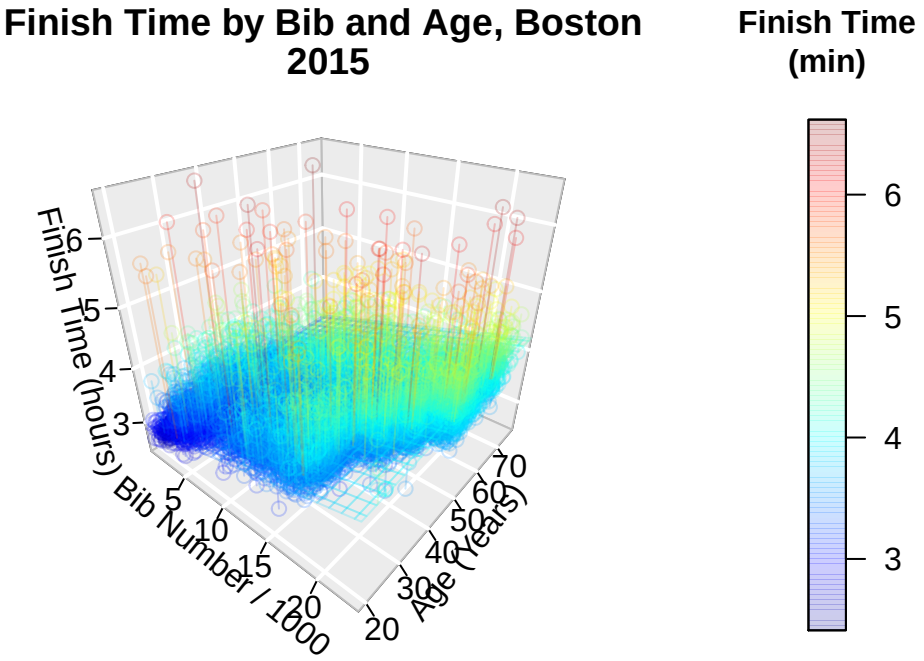




The comparison of the degree to which runners slow down from their 5km pace and half-marathon pace as a function of age, shows that there are qualitatively similar trends from year to year split by gender and wave. This means that Wave, Gender and Age could be included in the Linear Regression Models which will be intended to reduce the biases in the predictions. We can see from this same plot that although these trends are qualitatively similar and biased away from the nominal value of 1 (the B.A.A. baseline model always predicts a Pace Ratio of 1 at all splits) the variance around each trend is very large: a variance reduction model will need to be combined with a bias reduction model to optimize the RMSE.



A series of 3D surface plots were created to visualize effects such as age on the final time at various checkpoints in combination with other time-based factors. An example is shown below looking for whether or not Age modified the relationship between Bib number and Finish Time. In this example there are clusters or waves in the data which will not be well captured by linear regression models. Furthermore, in this example it appears that most of the information about Finish Time from Age is already captured by Bib Number. In each case, the Age feature does not provide a strong effect which is not already captured in some other metric. For this reason Age is excluded from the bias reduction linear regression models.



Since there are signs of clusters or non-linear relationships between Age and Bib number both features are included in the initial regression tree models. Due to the high degree of unexplained variance in the simple visualizations, new calculated features must be added to the Boston Marathon results data to extend the flexibility of the random forest regression tree models. These features will be intended to look for runner speed changes along the course so that runners who have similar velocity derivatives but dissimilar velocities can be clustered and analyzed using the regression tree models responsible for reducing the final variance. The calculations involved in exposing these features are described in the **Model Development** section of this report.

Model Development

The first step in the development of this project was to build two sets of 9 linear regression models. The first set of models are the *bias reduction models* which are based on the EDA observations that runners generally have slower finish times than predicted by assumption of a constant pace. At each check-point along the course a linear regression model is trained to predict the Final Time from the runner's check-point time, Gender and Wave. These models are selected to be highly simplistic to train rapidly and remove the gross biases to simplify the task of the more complex regression tree models.

Since the linear regression models are not tuned they are trained directly on the combination of all of the 2015 and 2016 results after spot-checking the significance of the feature coefficients on the training data subset and that the predictions on the test set have lower RMSE than the B.A.A. baseline model.

Using the same features, a second linear regression model was trained for each check-point to predict the time at which each runner would reach the next check-point. At each check-point after the first (5km), the actual check-point time is compared to the time predicted from the previous check point to get a residual in minutes (actual - predicted) and a ratio (actual / predicted); both of these features are added to the dataframe for use in the regression tree random forest variance reduction models.

For example, let us assume a runner arrived at the 10km point at 0:44:21 and the linear model predicted that this runner would arrive at the 15km check-point at 1:07:22 but the runner actually arrived at 1:06:45.

- In this example this runner would have an additive residual of **1:06:45 - 1:07:22 = -0.62 minutes**
- This runner would have a ratio of **1:06:45 / 1:07:22 = 0.991**
- This runner's pace was **0:07:09 per mile** for the first 10km
- The runner's pace from the 10km mark to the 15km mark was **(1:06:45 - 0:44:21) / 3.11 = 0:07:12**
- This means that this runner would also have a normalized pace at 15km of **0:07:12 / 0:07:09 = 1.009**

At each check-point two regression tree random forest models are trained using the training data subset of 2015 and 2016 results. Early in the race all available features are provided to the model; as the race progresses the variable importance is evaluated at each check-point and the least important variables are excluded from the next check-point's model and new features that become available at the new check-point are added. Each Random Forest model has its node size tuned on the training subset to optimize for the lowest estimated Out-Of-Sample RMSE on the reserved testing subset of 2015 and 2016 data.

The method selected to optimize node size is a for loop which iterates over a range of node sizes trains a model on the training subset, makes predictions on the testing subset and appends the estimated Out-of-Sample (O.o.S.) RMSE to a results vector. Due the slow process of tuning a random forest model (particularly with many trees and/or with small node sizes) this optimization uses forests of 75 trees rather than the 500 tree default. After the 5km check-point a narrower, course mesh of node sizes are used with the optimized node size from the previous model used to estimate the range of node sizes over which the next model is optimized. The node size with the lowest estimated O.o.S. RMSE and its closest neighbors are passed with the RMSE values to a function which calculates the coefficients for a quadratic fit to the data. The function then solves for the nose size where the derivative of the quadratic fit is 0 and returns the closest integer node size. With this optimized node size a while loop is used to grow the forest in increments of 25 trees until the RMSE stops decreasing by >1% per iteration. After the node size and the number of trees are optimized on the training and testing data both data subsets are recombined and the finalized random forest of regression trees is trained on the complete 2015 and 2016 results.

Speaking from the personal experience of having marathon finish times range from 3:02:47 to 4:57:39, having dropped out once with a broken leg and spent 20 minutes in a medical tent the magnitude of the most extreme residuals will be inherently unpredictable (though the probability of an extreme residual may be predictable). For this reason, I never allowed the minimum node size to be below 10 runners despite the function's default node size of 5 observations. The node size calculation is bounded to force the calculation away from arbitrarily small or large node sizes: if the minimum is found at either the largest or smallest node size tested then that node size is used rather than extrapolating outside the known data.

At each check-point, two random forest models are optimized and trained using the method described above. One model predicts the additive residual (actual final time - predicted final time) to the linear regression prediction using such features as check-point time, Age, Wave, Gender, check-point pace normalized to 10km and the observed errors between predicted and actual arrival at various check points. The second model predicts the ratio (actual finish time / predicted finish time) using similar features as the additive model.

The additive model is intended to look for signs of injury, the "wall", walk-breaks or other such fixed time type delays while the multiplicative model is intended to mimic more gradual slowdown behaviors associated with exhaustion. The two model effects are averaged with equal weights.

The final models at each check-point take takes the form:

$$\widehat{Y} = \left(\frac{(\widehat{Y_{lm}} + \widehat{A_{rf}}) + (\widehat{Y_{lm}} * \widehat{R_{rf}})}{2} \right)$$

Where:

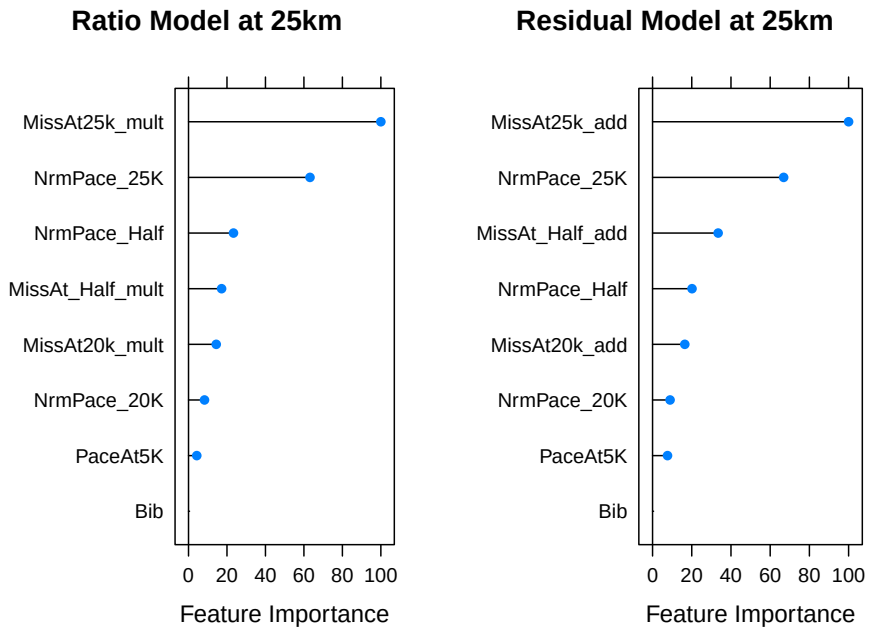
$$\widehat{Y} = Final\ Race\ Time\ Prediction$$

$$\widehat{Y_{lm}} = Linear\ Regression\ Model\ Prediction\ of\ Final\ Race\ Time$$

$$\widehat{A_{rf}} = Random\ Forest\ Model\ Prediction\ of\ Addition\ Residual\ to\ Linear\ Model\ Predicted\ Time$$

$$\widehat{R_{rf}} = Random\ Forest\ Model\ Prediction\ of\ Final\ Time\ Divided\ By\ Linear\ Model\ Predicted\ Time$$

The runner's pace normalized to their 10km pace and the degree to which they beat or missed the time for their recent check-points as predicted at the previous check-points are typically the most important features. An example below shows the relative feature importance analysis at 25km which resulted in the exclusion of Bib, Pace at 5k from the 30km models.

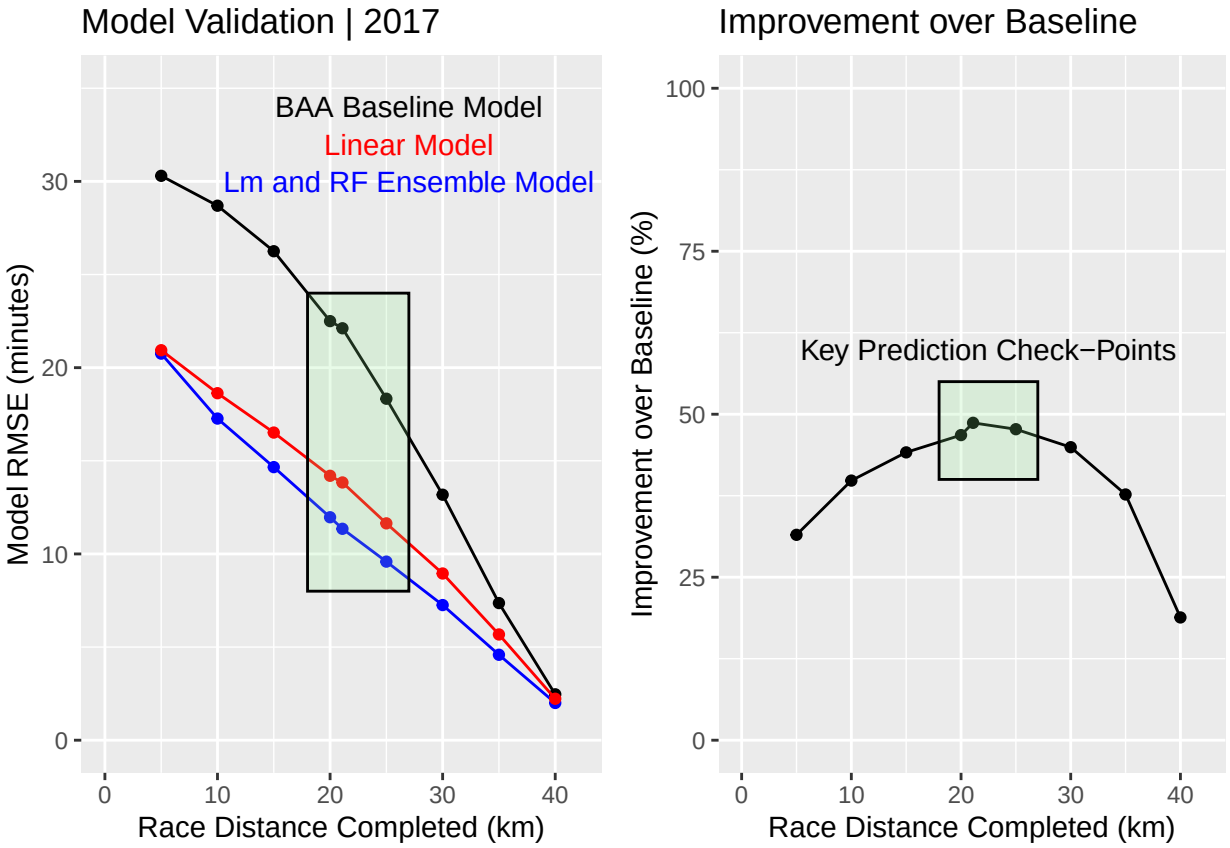


The "MissAt…" features are the residuals and ratios of the runner's actual arrival time at the listed check-point compared to their predicted arrival time. The "NrmPace_" features are the pace from the previous check-point to the listed check-point normalized to the runners pace over the first 10km of the course.

Results and Conclusions

Results

The model is validated using the entire 2017 Boston Marathon results dataset. The linear regression models show a significant improvement in RMSE over the entire race course. At the first and last check-points the Random Forest model only offers modest benefits over the linear regression model. This is because at 5km the data set does not have a sufficiently rich set of predictor features to improve the prediction and at 40km the race is almost over so there is little excess variation which can be modeled. Over the more important middle section of the race, the optimized ensemble model outperforms the Baseline model by approximately 47.5%. At the half marathon the RMSE for the ensemble model is approximately 18% lower than for the linear regression model.



Conclusions, Limitations and Future Work

This project details the development of a machine learning model which significantly improves the accuracy of finish time predictions at each of 9 check-points along the Boston Marathon course over the existing baseline model. An average improvement of **>40%** is demonstrated in out-of-sample validation using the 2017 marathon results which were completely withheld from the model training. At the most important splits at the center of the race (when spectators will be making their decisions about when to head to the finish line) the RMSE is improved by ~**47.5%**.

The model used in this project is an ensemble method which uses a linear regression model to reduce the baseline bias combined with a pair of Random Forest models which are used to reduce the variance left over after applying the linear regression model.

While the predictions from this model may be of interest to runners on the course for an improved understanding of their likelihood of meeting some personal finish time goal the true impact of this work is the improved knowledge given to race organizes about when runners are expected to finish and the improved RMSE of that prediction. By integrating these predictions and tightened confidence intervals with security at restricted areas spectators can be allowed into these prime viewing locations within a narrower window. By using this sort of spectator management the risks associated with highly congested areas can be reduced. This may potentially allow major marathon events like the Boston Marathon to resume in the age of Social Distancing.

A manageable but significant limitation to the existing model is that it sees only chip-time. This is the time at which runners cross timing mats along the course relative to the time when they crossed the start line. In future work, if the B.A.A. makes gun-time available (the absolute time from when the race first started, independent of when the runner crosses the start line) to the model then further separation of spectators can be achieved. For example, a runner who starts at the back of Wave 4 may cross the start line 90 minutes or more after a runner who starts at the front of Wave 1 so even if their projected finish times are the same, the actual projected time of day when the two runners are expected to reach the finish line could be quite different.

The inclusion of gun-time or actual time of day in the historical data could also allow for temperature and precipitation data to be merged into the training data allowing for weather forecasts to improve predictions at early-race check points and actual measured temperatures and precipitation levels to be used to improve predictions at later check points.