Predicting Participation and Run Time of the “Marathon Oasis Rock ’n’ Roll de Montréal”

Xiru Zhu(xiru.zhu@mcgill.ca, 260498514),
Tianzi Yang(tianzi.yang@mail.mcgill.ca, 260599365),
Arnaud Y. Massenet(arnaud.massenet@mail.mcgill.ca, 260608613)

***Abstract*- This paper outlines the methodology utilized to predict the runners and run time of the “Marathon Oasis Rock ’n’ Roll de Montréal”. This paper also covers the evaluation and results.**

I. INTRODUCTION

The “Marathon Oasis Rock ’n’ Roll de Montréal” takes place every year in autumn in the city of Montreal from “Île Ste-Hélène” to “Le Plateau-Mont-Royal” through the “Vieux-Port”. It includes multiple running events from 1 km to a full Marathon of 42 km.

The goal of this paper is to analyze the data of the 2012-2015 Montreal Marathon runners and attempt to predict whether they will participate to this year’s edition of the marathon. The second goal of this paper is to predict the time each runner will achieve based on their previous marathon times. To fulfill the first goal, we have elected to use a Logistic Regression Model and a Naive Bayes Model to predict participation in the 2016 Montréal Marathon. To predict the 2016 Marathon runners time, we have chosen a Linear Regression Model . In both of our models, we have elected to be parsimonious in the selection of estimators. We believe a simpler model is not only more elegant but also reduces the risk of overfitting. Insignificant estimators will results in poor prediction. Hence, our methodology although not novel or groundbreaking are robust and consistent to produce high levels of prediction accuracy for both trained and untrained data sets.

II. ASSUMPTIONS

In this paper, we assume participation in the Montreal Marathon to be defined as a runner participating only in the 42 km event in the “Marathon Oasis Rock ’n’ Roll de Montréal” on September 25th 2016. Participation in other events such as ‘Le Ptit Marathon’ or even the ‘Demi-Marathon’ will not be counted as Marathon Participation. Furthermore, we assume that no participants outside the dataset will be participating. This is due to the lack of data for such prediction. Although we could account for unknown participants, the results becomes too generalized and imprecise to be useful. In addition, we assume that the time for all runners which could not finish to be 7 hours. Since the Montreal Marathon’s time limit is 6 hours, we consider all estimations with times greater than 6 hours to be failure to finish a marathon. Further on with the estimator for sex, we assume that this represent the biological sex of the participant. This is important for time prediction as there is a significant athletic performance difference between the two biological sex. Lastly, what we refer in this paper as estimator is synonymous with the term “feature” in machine.

III. MARATHON PARTICIPATION MODEL

To estimate Marathon participation, in both logistic and naive bayes method we only selected four estimators. They consist of Age, Gender, Weighted Montréal Marathon Participation and Weighted Running Events Participation excluding Montreal Marathon. We define x1, x2, x3, x4 as the above list in such order.

Age is simply the age of the participant. Younger participants such as those in the range of 30-50 are may be likely participate than other age groups such as 70. To properly process age data, we simply took the age range and averaged the result. Further rationalization was applied to fill missing data and smooth the data such that the age is increasing annually by one each year. This leads to more consistency in the dataset at the expense of some precision in age. Note that some data points for age could not be properly estimated. Entry such as 13+ is often utilized for adults with age of 30! Hence, all entries with such age will be eliminated from the model for potential misleading age.

Sex is another estimator for our model. Here we are assuming that by classifying by gender will give up

some more information on participation. One gender may be more inclined to participate than the other.

Weighted Montreal Marathon Participated is the weighted number of marathon a participant has participated in. This is computed by calculating the cumulative number of Montreal Marathon ran and then weighting the data sets to obtain the number for a specific year. We believe the participation in the Montreal Marathon in the past is a good estimator for further participation. However, had we simply used a cumulative number of Montreal Marathon participated, it leads to unbalanced and illogical weights. For example a marathon participation in 2012 is weighted the same as a 2015 participation. Because the 2015 data is closer to 2016, we can suppose that it is more likely a participant of 2015 will be in the 2016 marathon than say one which only participated in 2012. Therefore, we utilized a weight system where the value of a past participation is

Where weight(i,j) is simply the cumulative number of Montreal Marathons participated up to such year.

$$weight_{j,year} = \sum_{i=2012}^{year} weight_{j,i} * k^{2016-i}$$

where k is a constant we change to get a greater weight and j is the jth participant.

In our model we utilized a k = 1.5 as a way to greatly increase the weight value of the years closests to 2016. Overall, the value of all Weighted Montreal Marathon Participated is increased by the relative difference between a value from 2012 and 2016 is greatly increased. The value k here is adjusted until the model gives a proper prediction number. A reduction in k will lead to less predicted participants and an increased will increase the predicted participants.

Weighted Running Events Participation is similarly the count of all events the runner participated in excluding the montreal marathon. This was done to reduce the correlation between the x3 and x4 since such information is already in the previous estimator. Of course, x4 should still have some correlation with x3 but such level is considered acceptable for our model. Here as was the case above we want to reduce the weight of years farther away from 2016.

Where value(year, 1:end) is the cumulative number of Running Events participated up until that year

where k is a constant we change to get a greater weight and j is the jth participant.

$$weight_{j,year} = \sum_{i=2012}^{year} weight_{j,i} * k^{2016-i}$$

Here, we utilized k = .67 mostly to reduce the significance of running events in comparison to number of Montreal Marathon participated. A higher level of q will lead to greater significance for when more events are participated excluding montreal marathon.

Previously, we have considered other estimators such as the year, event time and other derivatives of event participation rate. However, given the potential lack of independence between so many estimators and high likelihood of insignificant estimators, we elected not to include them in the model. Although more estimators can lead to better data fit, it can also lead to overfitting and poorer prediction for untrained data set. Therefore by the principle of parsimony, we elected to use as few estimators as reasonable given the dataset.

IV. CLASSIFIER DESIGN

A. Logistic Regression for Marathon Participation

Before doing logistic regression, we need to do some preprocessing on the matrix. We have three matrix X matrix, y matrix(used for both cross_val and training) and pred_X(for predict 2016 results). In X matrix, we have multiple features and most of them are not in the same scales, like gender(0 or 1) and age (13 ~ 60). If we use them directly, logistic regression will return NaN as results and it also slower the training speed. So we did mean normalization to both X and pred_X matrix. Also, we map features to higher order function to improve the results.

For the weight initialization, we use the function rand() due to it returns number in uniform distribution which is good for training.,

The first part of logistic Regression is using cross_cal to predict hyperparameters, we used function to cross_validation() do that. For the first one, we used it decide the learning rate value and lambda value(regularization weight). We randomly pick out 80% and 20 % data from X matrix and used them as training X matrix and validation X matrix. We repeat doing this for 5 times and take out the best value among them.

The second part is real training, we used logistic regression (R2 regularization)with gradient descent. When updating the new weights, we use the momentum update instead of the normal vanilla update.

The momentum update can converge much faster than normal update.

The third part is predict the results.The problem we met is both training part accuracy and validation part can go up to 80%, but most of our predictions are 0 (means no one will take the marathon). In other word, our function force all the prediction to 0 because there is more 0 in our y matrix(about 80% 0). Another reason causing this is we run the training in too many iterations, overfit in other words.

To prevent this problem, we add one more testing part using function cross_validation2(), this function decides when we stop training and the order we need for the features. To help us doing that, we used F1 score. F1 score = 2*(PR)/P+R where P is precision and R is recall. The true positive here is we predict 1 and it should be 1(in y matrix). We found that the F1 score starts from 0.4 at 0 iterations and it would drop to 0.2 when we got 75% accuracy. At the time we just got 79% accuracy, the F1 score is 0.1 which is still not that bad. But it dropped fast to 0.05 when we got 80% and kept dropping to 0 ~ 0.01 while the accuracy is still around 80%. So we decided to choose the time when between we just got 79% and F1 score is higher than 0.02. Also we compared the logistic regression prediction and naive bayes prediction and human prediction on each iterations. We prefered the iteration number which has the highest similarity on these 3 predictions.

B. *Naive Bayes for Marathon Participation*

The Naive Bayes classifier is often depicted as the one of the most popular classifiers. It is an elegant model for a more civilized age. Since the model is simpler, the choice of estimators is what defines the accuracy and precision of the classifier. In order to truly understand how Naive Bayes works, we decided to begin with models that used only binary estimators for the estimators mentioned above. Although the accuracy we computed during testing phase was acceptable at around 80~90%, the prediction results the model predicted was quite surprising. The model was predicting that none of the runners in our data will be participating to the 2016 Montreal Marathon. Of course, we know that such answer will contain a high degree of type 2 error.

We decided to change the estimators, but the model was not improving. Although simple Naive Bayes with binary feature is straightforward to implement, we decided to utilize Gaussian Naive Bayes for some estimators. Estimators for Age and Weighted Number of Marathons and Events were changed to gaussian while the Estimator for Sex remained binomial. The accuracy of the training and test set suffered heavily.

However the prediction for 2016 seemed better with higher predicted participation rate. In order to improve both the training accuracy and the prediction accuracy of our model we decided to make modification on our datasets such that we could give a higher weight to the data related to newer events. When we finalized our model we ended up with 4 estimators, the age of the runners, the gender of the runners, a weighted number of event participation and a weighted number of participation to the “Marathon Oasis Rock’n’Roll de Montréal”. As stated above we picked these parameters based on their significance and relative independence from each other.

The final Naive Bayes Model simply computes the probabilities of each estimator given montreal participation or not. Then, with the average and variance table, we simply computed the probabilities for both probability of participating and not participating. Whichever is higher in the comparison will be selected as the result. No training was utilized for the Naive Bayes Model.

C. Linear Regression for Marathon Speed

In this model, we have elected to utilize 6 estimators. They are Sex, Age, Average Number of Half Marathons, Time in Half Marathon, Average Number of Full Marathons and Time in Full Marathons. We respectively defined each as x1, x2, x3, x4, x5 and x6.

For both the age and sex, we utilized the same dataset as computed for solving Marathon Participation and thus each estimators follows its respective idiosyncrasies.

To obtain the data for x3,x4,x5,x6, we had severe difficulty securing complete data for the training set. While scraping information from the 2012-2016 dataset, we noticed a high number of varying events in the dataset.There were simply far too many events in the dataset where we simply did not have enough data to compare with. To make it worse, the same events were often times spelt differently, french or english, without or without ‘-’s. Furthermore, the events themselves were often of different type, triathlon, 10km run, cross country and even cycling! Of course utilizing such dataset to estimate marathon speed would be questionable at best, completely overfitting at worse.

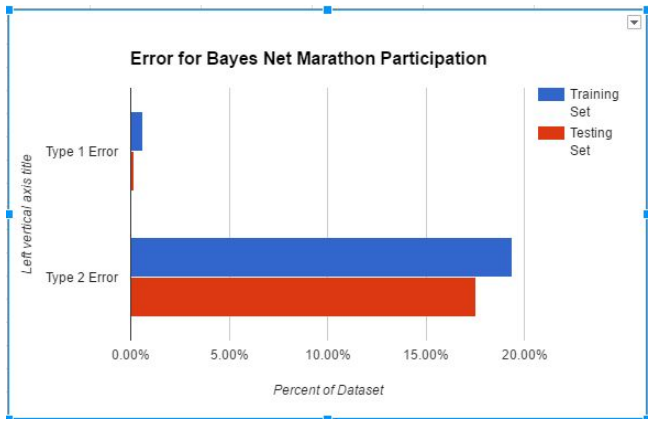
For example, the most egregious example would be “Le P’tit Marathon” which only runs for 1 km. Of course, someone running 1 km will run faster than someone running a Marathon. The two events are completely unrelatable in terms of time. To make matters worse, we often do not have the full Montreal Marathon time for everyone. Many participants do not participate in the full Marathon instead opting out for demi marathons or even 10 km. Hence, we would have difficulty training from the dataset. Thus, with the given dataset, we had exceptional difficulty scrapping important into usable data. Many types of events were simply forcibly removed with no good way to use it.

Given the overwhelming variance of dataset, we have elected to only utilize demi-marathon and full marathon data and ignore the remaining. For example, although a triathlon is certainly a good show of athleticism, the time given is exceptionally poor way to estimate running a marathon. One could easily be good at running but awful at swimming for example.

To resolve the problem of missing data, we had no choice but to fill in the dataset with available partial data. Otherwise without data filling we end with merely 1027 data entries in order to predict 8710 participants speed. The first step of data filling was making all estimation with time as -1, meaning failure to finish were set to 4.5 hours for demi-marathons and 7 hours for full marathons. This is because the time limit is 3.5 hours for half marathon and 6 hours for full marathons in the Montreal marathon. The second data fill in was to simply fill in missing marathon or demi marathon times with the average time for the individual. This of course has exceptional drawback in terms of accuracy of the prediction. However, given the lack of data available, any more data outweighs the cost of bias. Finally, and probably worse of all, we filled in any missing demi marathon times with half of the speed of full marathon. Of course, this has exceptional bias, given that half marathon are usually ran faster than full marathons but the lack of data forced our hands.

V. EVALUATION

In this section we will present the performance of our 3 models depicted above. The first two models were built to predict participation in the “Marathon Oasis Rock’n’Roll de Montréal”. With binary classification, an error can be one of the following: Type I Error (False Positive) or Type II Error (False Negative). Since neither is worse than the other we will evaluate both without giving more weight to one or the other. We decided to present in terms of percentage over all the dataset.



We also calculated the accuracy which was 79.9% on the training set and 83.2% on the testing set using .

Given these result we can obviously state that our false negative negative rate is much higher than our false positive rate. We can conclude that our classifiers are failing to predict whether or not a runner is participating or not. This could be caused by the fact we are using data from 2012 and 2013 where data are much more limited and makes it difficult to predict participation. Second, our weight system is more specifically to predict 2016, not 2012 or 2013.

In order to evaluate the performance of our linear regression model we decided to picked squared-errors. However due to the way time is often represented, we had to decided on a model that was more fitted for evaluating the error using squared-errors. This meant that we had to “reduce” the value of the representation of time, hence we decided to represent time using hours as a reference. So a time of HH:MM:SS was represented as HH+MM/60+SS/3600. In order to choose between the different possible order-d functions we computed (order-1 to order-7), we decided to use the Least-square method between the average running time of each runners and their predicted time. This lead us to pick the order-3 fit function.
Using an 5-fold cross-validation we got an approximate square-error of 1154 across all 5 sets. This leads us to think that the error is in the acceptable range since we are using data over 4 different years.

In logistic regression, the first cross_val function returns that lambda = 0.01 and learning rate = 1e-1 can return the best result after the first 500 iterations. However,

our final decision is lambda = 5, learning rate = 1e-2 due to 1e-1 is good in the first few hundred iterations but it will be too high at the last part the training and all the testing lambdas return the same result. So to prevent overfit we use large a lambda value. By cross_val2 function results, we decided to stop the training when the F1 score is around 0.02 and we predict 600+ 1 in 2016. The Order number we choose is 2.

The final performance of logistic regression is training accuracy around 79.8% and F1 score 0.023.
We believe the prediction success for the bayes model and logistic regression model should be 80%. At least our model should reach such level of accuracy.

VI. FUTURE IMPROVEMENTS

For further improvement to the models, a better scraping strategy could have been utilized to pull more data for predicting the marathon time. Perhaps some kind of gaussian distribution for each specific event to get the participant's percentile. Such percentile value could be used as predictor for time performance. As for the Logistic Regression Model, perhaps removing the 2012 and 2013 data as part of the training set could have improved the prediction accuracy. Many of the data in those two years were severely lacking.

VII. CONTRIBUTION

Data Scraping: Xiru
Logistic Regression Model: Tianzi
Linear Regression Model: Tianzi, Arnaud
Bayes Net Model: Xiru, Arnaud
Report: Xiru, Arnaud, Tianzi