Mini-project #1: Predicting the Marathon Winner

COMP-551: Applied Machine Learning

Koustuv Sinha
McGill ID : 260721248
koustuv.sinha@mail.mcgill.ca

Ramchalam Kinttinkara Ramakrishnan
McGill ID : 260711189
ramchalam.kinattinkararamakrishn@mail.mcgill.ca

Xiaoqing Ma
McGill ID : 260668927
xiaoqing.ma@mail.mcgill.ca

***Abstract*—This paper presents the basic methodology and results obtained for prediction for participation and race times in Montreal Marathon 2016, using Logistic Regression and Naive Bayes classifiers to predict participation and Linear Regression to predict race times. Our algorithm provides 85% accuracy and 720 MSE respectively in classification and regression tasks in validation test set, and we hope it draws nearer to the actual results which will be posted after the Marathon completion.**

I. Introduction

This report is mainly concerned with our working process and the outcome on analyzing original marathon data, extracting necessary features, training the model and evaluating the errors. Finally, a prediction on whether an athlete in the dataset will attend MARATHON OASIS ROCK 'N' ROLL DE MONTREAL 2016 is presented.

II. Problem Representation

The original data downloaded from the course website is in a format such that the events information for one single participant ID is in one single row. At first, we split the rows respectively and obtain a new dataset in which each row represents one certain participants one event record. In this case, rows are used to save the whole information of one athlete. The resulting dataset contains 26961 rows in total. However, during our initial analysis, we realized that this strategy of splitting rows would result in a huge computation load in the stage of classification and perform adversely in linear regression as well. Thus we shifted our strategy to another format with one single row for each participant ID with aggregated data. This section primarily deals with the method we came up with for selecting and generating useful features. We also scraped and downloaded the Location data for the events as it was not provided, which was later useful to determine the number of Marathons a particular user had participated in a particular year. The modified dataset is available in this public url [2].

In order to make the information in dataset understandable for the algorithm to predict, some of the contents had to be transformed or encoded into a convenient format. The following features were added/modified and utilized for obtaining other features:

- **Event Type**: As we are going to predict whether one certain participant will show up in MARATHON OASIS ROCK 'N' ROLL DE MONTREAL 2016 in addition to the finishing time, we needed to look up into his previous attendances and results in several Marathon-related events. Nevertheless, one athletes performance in a cycling race or an Ironman grand Prix does not necessarily correlate with that in a running event, so it is reasonable for us not to consider a number of event types other than those relate to running, which would make the heterogeneous records much easier to be handled. For the simplicity of narration, the phrase "all event types" are used to denote all the running event types in the following report.
- **Event Date**: Year is separated from the original data
- **Categories**: Originally CATEGORIES contained 2 parts of information: gender and age. Here we separate it into 2 features. The gender can be encoded easily as shown in Table.1. Age in the dataset is shown in a format of a range. The mean of upper and lower bounds is taken to encode the age range. For example, "M40-44" can be divided into "Gender = 1" and "Age = 42". For the age coding, another concern is that for athletes who have taken part in multiple years events, their age information no doubt varies between different event records. To deal with this variation, we just take the average of all his age records as a feature value. Participants without a valid age record have 0 assigned to the AGE feature. (which in this case is about 30 records).
- **Time**: Times are all transformed into seconds to represent ones performance. Whats more, the average finish time for different Marathon types may significantly vary. For example, most participants finished their Marathon races in approximately 4 hours or above, but for Half-Marathon, most records are distributed between 2 and 3 hours. According to the project instruction, we should predict the Full Marathon result for every participant ID in the list, even though many of them have never had experience in completing a Full Marathon race. In the case of non-uniform data, we use the feature named normalized time instead of time from original data to provide a more accurate prediction. To obtain the normalized time, original time value (in second) is divided by the race distance of this event type, which is taken from the event type name and the event website
- **Location**: In the original dataset, there was no Location information given for each event. We modified

the provided python scraper and scraped Sportstats.ca [1] website for those event names and generated the Locations. Even after scraping we found several events were not provided any Location maybe because the Event name had already contained the locations or our scraper was not foolproof. In those minority cases, we filled the Location data from the Event Names as well as some manual scavenging of the data

TABLE I
GENDER CODING

Gender	Encoded Value
Male	1
Female	0
Invalid / Not Provided	-1

In this project, we are going to perform 2 kinds of prediction: classification and regression. Therefore, we are using 2 different training data in different formats for classification and regression respectively.
For the classification part, we construct a dataset in which each row represents one participants attendance in recent years along with information of gender and age. Table. 2 shows our selection of features and the description for each. Whether a feature is binomial or continuous is also important as a probability distribution should be introduced if one certain feature is not in a multinomial format.

During data processing, we discovered that participant ID 7959 has 2 entries on Montreal Marathon but in different categories in 2012. The same thing happens with participant ID 6497 in 2015. This problem is solved by discarding one row from each. All data manipulation was done in python Pandas library and SQL Server.

For the regression part, the emphasis is not given to the features with the event counts for a certain athlete, but the time he finished a Marathon competition. In regression, we have normalized the time to time per km. All the data manipulation for regression has been done in SQL Server by importing the data. Averages are computed from the time per km records on each year. Table.3 describes what features are selected for regression.

In this case for the linear regression problem, selecting the features are not very straightforward. Our team used multiple data analysis techniques to finally zero in on the most efficient method. From the existing data, many extra features were scraped or derived. We took the distinct event types and assigned them appropriate weights (For e.g., Montreal marathon was assigned a value 6, Normal marathon 5 and Half marathon 4 etc.) The feature, location, was scraped from the event name and the event website and then Montreal Marathon was given a special weight. (ParticipatedinMontMara = 1 else 0). We

TABLE II
A LIST OF FEATURE SETS IN THE DATASET FOR CLASSIFICATION

Feature Name	Description	Feature Category
2012	Participation in MARATHON OASIS DE MONTREAL 2012 (yes = 1, no = 0)	binomial
2013	Participation in MARATHON OASIS DE MONTREAL 2013 (yes = 1, no = 0)	binomial
2014	Participation in MARATHON OASIS DE MONTREAL 2014 (yes = 1, no = 0)	binomial
2015	Participation in MARATHON OASIS DE MONTREAL 2015 (yes = 1, no = 0)	binomial
NUM_MARA	Number of Marathons participated in total	continuous variable
NUM_MON_MARA	Number of Marathons participated in Montreal	continuous variable
NUM_TOTAL	Total number of events participated	continuous variable
TOTAL_2012	Total number of events participated in 2012	continuous variable
TOTAL_2013	Total number of events participated in 2013	continuous variable
TOTAL_2014	Total number of events participated in 2014	continuous variable
TOTAL_2015	Total number of events participated in 2015	continuous variable
GENDER	Encoded as per Table 1	binomial
AGE	Encoded from the mean of age range	continuous variable

TABLE III
A LIST OF FEATURE SETS IN THE DATASET FOR REGRESSION

Feature Name	Description
AGE	Encoded from the mean of age range
GENDER	Encoded as shown in TABLE 1
AvgTimeForAllMarathons 2012	Average finish time per km for all marathon events paticipated in 2012
AvgTimeForAllMarathons 2013	Average finish time per km for all marathon events paticipated in 2013
AvgTimeForAllMarathons 2014	Average finish time per km for all marathon events paticipated in 2014
AvgTimeForAllMarathons 2015	Average finish time per km for all marathon events paticipated in 2015
AvgTimeInAllMarathons	Average Finish time in all marathon events
TotalNoOfMarathonEvents	Total number of marathon events participated
AvgTimeInAllEvents	Average finish time in all events

also calculated the distance in km for all the events which had distance mentioned either by event name or event type and excluded non-running events. The key feature which was used to normalize all these events was time per km. This would help normalize all events for all participants to an extent and assist in the overall prediction of time taken.

Taking all these features into consideration, we finally sampled different feature sets. For each participant, we took the AvgTimePerMontrealMarathon for each year along with a count of events participated. This resulted in an average prediction error (MSE close to 6000 (seconds)). This would have worked perfectly in cases where the data was complete. But the original file had many missing data for each participant

and we had to assume the value 0 for a participant who hadnt participated in the event for the year. This was a major constraint in the final prediction and is a problem of missing data and hence we shifted our approach. We came to the conclusion that filling the rows where the participant has not participated, with the mean of the time taken for all participants for each year, would be a better solution. Considering the missing data, it was the only option with the least risk involved as far as prediction is concerned. Hence we used the mean time (in this case mean of both test and training set). This brought down the MSE close to 700 which was over 7 times better.

Finally, for linear regression, the results would be then multiplied by 42km and then converted into hh:m:ss format

III. Training Method

A. Logistic Regression

For Logistic Regression, we have taken the parameters alpha (step size) as 0.06, and lambda (regularization) as 0.01. We arrived at the hyper-parameter selection by Cross-Validating the data with several combinations of hyper-parameter values and checking which one causes the most accuracy.

For training and testing split for Logistic Regression, we have taken 6000 records for training and the remaining 2711 records for testing. All features were normalized before passing to the algorithm for fitting and prediction. As we found out after Cross Validation, regularization parameter 0.01 worked best for our data set, which indicates that the algorithm needs minimal regularization, as it has fewer features than training data. As per the features mentioned above, we trained the algorithm using generic features as well as the participation information of 2012, 2013 and 2014, and used 2015 year as the validation data. For final prediction, we use the generic features and 2013, 2014 and 2015 data to predict 2016 participation's. Thus, using historical data we arrive at our predictions for 2016.

B. Naive Bayes

As shown in Table. 2, among 13 features we have selected, more than half of them are continuous variables. Therefore basic Naive Bayes classifier which concentrates on the binary case would perform badly. In order to enhance the accuracy of Naive Bayes algorithm, the Gaussian distribution is adopted into the classifier to represent the cases where features are not binomial.

Gaussian Naive Bayes model assumes that given y continuous variables $x_0, x_1, x_2..., x_n$ are independent of each other and obey different normal distributions respectively. The probability density function (PDF) of normal distribution of a continuous variable x, or called Gaussian distribution, is shown as below:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{1}$$

where μ and denotes the mean and σ standard deviation of the distribution. The parameter μ and σ can be computed from the dataset likelihood. For samples in the whole dataset whose label is 1, we have the likelihood of this part of dataset as:

$$L_1 = \prod_{i=1}^{M'} [P(y_i = 1) \prod_{j=1}^{N} P(x_{j,1}|y_i = 1)] \tag{2}$$

$$\log L_1 = \sum_{i=1}^{M'} (\log P(y_i = 1) + \sum_{j=1}^{N} \log(P(x_{j,1}|y_i = 1))) \tag{3}$$

Assume that given $y = 1$, the feature x_c is continuous and obeys Gaussian distribution with parameters $\mu_{c,1}$ and $\sigma_{c,1}$. Take the partial derivative of (3), and use the PDF of Gaussian distribution to estimate the probability $P(x_c|y_i = 1)$, we have:

$$\frac{\partial \log L_1}{\partial \mu_{c,1}} = \frac{\partial}{\partial \mu_{c,1}} \sum_{i=1}^{M'} \log P(x_c|y = 1) \tag{4}$$

To maximize the likelihood, let the derivative equals zero:

$$\sum_{i=1}^{M'} (x_C - \mu_{C,1}) = 0 \tag{5}$$

$$\mu_{C,1} = \frac{1}{M'} \sum_{i=1}^{M'} x_C \tag{6}$$

Similarly, compute the equation

$$\frac{\partial \log L_1}{\partial \sigma_{C,1}} = 0 \tag{7}$$

we can get

$$\sigma^2_{C,1} = \frac{1}{M'} \sum_{i=1}^{M'} (x_C - \mu_{C,1})^2 \tag{8}$$

Therefore, the parameter $\mu_{c,1}$ and $\sigma_{c,1}$ can be learned by computing the mean and variance of the values of feature x_c with the samples whose label is 1. The distribution parameters x_c of with label 0 can be derived with the same computation process. By implementing the distribution parameters within the PDF function, Naive Bayes classifier can calculate the probabilities and make a prediction on given inputs. Also, we introduce a parameter α as a Laplace smoothing factor in binomial cases to improve the prediction performance. The probability computation can be represented as follow:

$$P(x_j = 1|y = 1) = \frac{\sum_{x_j=1, y=1} S + \alpha}{\sum_{y=1} S + 2 * \alpha} \tag{9}$$

C. Linear Regression

Algorithm strategies and parameter settings in linear regression stage are slightly different from that of logistic regression. We have taken out around 4 strategies in data analysis and we have chosen the best and logical one out of those. In the best case scenario, for cases where the participant has not participated in the event, we are updating that field value

as the average value for all the other participants for that feature. This gives a better average output than the case whereby if the Participant has not participated in the event and updating that value as 0. For the final prediction, we used the 2013,2014,2015 data along with the generic features to finally predict 2016 time. Since the time records in training set have been normalized and is a measurement of seconds per km, the formal prediction of time should be the multiplication of linear regression output with the distance of Marathon (42km).

Similar to logistic regression, the training and testing split are in the same ratio, 6000 and 2711. In linear regression, we have not used regularization for the reason that the number of features is much less compared to the total number of training sets. The parameter alpha is same as that with logistic regression. 0.06. We are iterating the Gradient Descent function around 500 times and have thus plotted the data for cost vs number of iterations.

We are using k folds cross-validation method to split the data into test and training data and then find the Mean Square Error. For our current implementation of linear regression, the MSE for training set is approximately equal to 705 (seconds) and the MSE for testing is approximately equal to 720. We compared to the scikit learn inbuilt functions and the MSE, in that case, was around 650.

IV. Results

To evaluate the two classifiers, we first separate approximately 70% samples from the original dataset to train the model and then use the rest as a validation set.

A. Logistic Regression

The Training Set Accuracy for Logistic Regression comes up to 84.33%, which increases slightly to 85.28% for Validation Set. The detailed metrics are as follows :

TABLE IV
Accuracy Precision and Recall of Logistic Regression

Logistic Regression	Accuracy	
	Implemented Algorithm	Scikit Learn
Training Stage	0.843166666667	0.841833333333
Validation Stage	0.854297307267	0.854297307267
	Precision	
Training Stage	0.703794369645	0.70183299389
Validation	0.723577235772	0.720391807658

B. Naive Bayes

The Training Set Accuracy for Naive Bayes comes to 82.61%, and the Validation Set Accuracy comes to 83.47%.

Table IV and Table V shows that logistic regression algorithm we implemented provides good performance in accuracy, precision and recall, which are very close with those from Logistic Regression module provided by machine learning package scikit-learn. We could achieve precision of 0.72 while the recall exceeds 0.9. The same thing also happens in Naive Bayes classifier. Though the prediction recall could achieve 0.9 in validation stage, the precision, which is 0.68, is significantly far less than the precision. This phenomenon can also be reflected from confusion matrices. Table VI and Table VII present the confusion matrix of Logistic Regression and Naive Bayes from both training and validation stage.

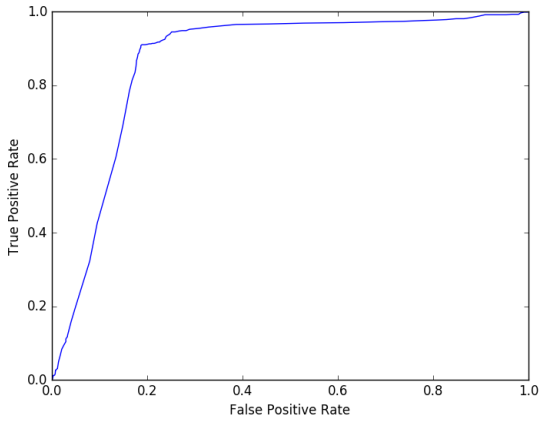

Fig. 1. ROC Curve for Logistic Regression

TABLE V
Accuracy, Precision and Recall for Naive Bayes

Naive Bayes	Accuracy	
	Implemented Algorithm	Scikit Learn
Training Stage	0.820833333333	0.831833333333
Validation Stage	0.831427517521	0.844706750277
	Precision	
	Implemented Algorithm	Scikit Learn
Training Stage	0.667961165049	0.687474828836
Validation Stage	0.683942225998	0.70480349345
	Recall	
	Implemented Algorithm	Scikit Learn
Training Stage	0.886597938144	0.879896907216
Validation Stage	0.904494382022	0.906741573034

TABLE VI
Confusion Matrix for Logistic Regression

Training Stage		Validation Stage	
TP = 1725	FP = 726	TP = 801	FP = 306
FN = 89	TN = 3334	FN = 215	TN = 1515

TABLE VII
Confusion Matrix for Naive Bayes

Training Stage		Validation Stage	
TP = 1720	FP = 855	TP = 805	FP = 372
FN = 220	TN = 3205	FN = 85	TN = 1449

From the confusion matrices, we could find out that the numbers of false positives generated by both classifiers are larger than those of false negatives, which means that classifiers are giving out more 1s than expected. This inaccuracy probably results from the imbalance between label 0 and label 1 in training sets. To evaluate the two classifiers, we also vary the decision boundaries and plot the ROC curves for Logistic Regression and Naive Bayes. The ROC curves is shown in Fig 1 and Fig 2 respectively.

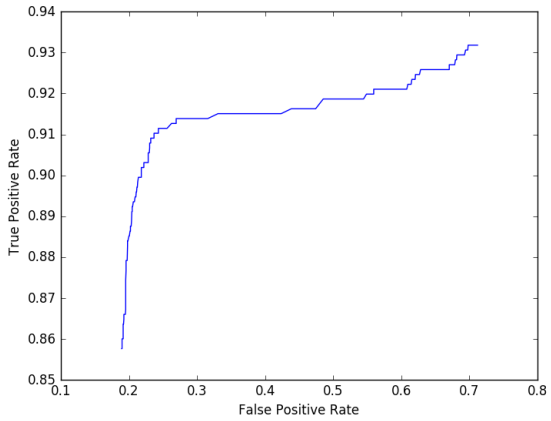

Fig. 2. ROC Curve for Naive Bayes

TABLE VIII
MSE FOR LINEAR REGRESSION

	Implemented Algorithm	Scikit Learn
Training Stage	705.221	650.444
Validation Stage	720.923	670.122

C. Linear Regression

The training set Mean squared error comes up to around 705 and the validation set Mean squared error comes to 720. The cost function along with the number of iterations has been plotted in Figure 3

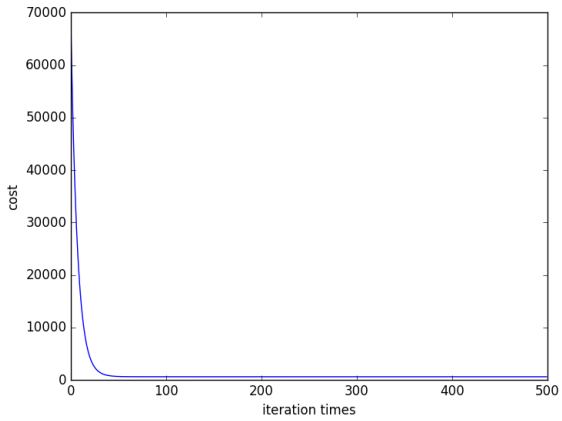

Fig. 3. Cost vs Iteration for Linear Regression

V. DISCUSSION

In the final prediction for participation in Montreal Marathon 2016, our Logistic Regression Algorithm predicted less number of participation than Naive Bayes. This might be due to the fact that Naive Bayes works taking counts and averages of all participation's, but going by the general trend of participants we feel that the actual number of participation's might be somewhere in between our predictions of Logistic and Naive Bayes Classifiers. In our approach, we didn't measure the performance on LDA, which can be considered as a future scope. Also, we didn't account for more realistic features like Weather or participant's personal bio, which can be gathered and processed to improve the accuracy. For linear regression specifically, we have overcome the missing data problem by taking the mean of the distribution, and hence the prediction comes in the optimum range without too many fluctuations. A future improvement may be to incorporate different techniques for the missing data problem which may result in better prediction.

VI. STATEMENT OF CONTRIBUTIONS

A. Koustuv Sinha

Primary Role was to implement the algorithm, data analysis and cleaning for Logistic Regression. The data cleaning was done in Python using Pandas library. Also involved in developing the cross-validation pipeline. Finally contributed to the writing of the report by describing the procedures, mostly for Logistic Regression.

B. Ramchalam Kinattinkara Ramakrishnan

Primary role was in Data Cleaning and data analysis, especially for Linear Regression. Part of the data manipulation for Classification was also done. Most of the coding has been done in SQL Server by importing the file and then writing SQL queries to group and modify data. The CSV output generated was tested in the algorithm and then tested in scikit learn to get a comparative result. Python coding for Linear Regression was also partly done. Testing of the final code as well as minor corrections and bug fixes to the codes. Finally, contributed to the completion of the report by elaborating all the procedures, especially for linear regression. To sum up, developing a methodology, performing data analysis, coding the solution(partly) and writing the report (mostly related to linear regression) were the contributions to the project.

C. Xiaoqing Ma

Python coding for Gaussian Naive Bayes, as well as generating evaluation parameters and plots for testing the performance of classification algorithms (confusion matrix, precision, recall, ROC curve, etc.). Partly contributed to data labeling and clearing-up. Also contributed to the completion of report in problem presentation section, Naive Bayes part of training methods section and classification part in the results section.

We hereby state that all the work presented in this report is that of the authors.

REFERENCES

[1] Sportstats.ca Location Data for Events, Accessed 17/09/2016, https://sportstats.ca
[2] https://github.com/koustuvsinha/data-adventures/tree/master/montreal2016