Mini-project #1: Predicting the winner!

Jaspal Singh - 260727323 - jaspal.singh2@mail.mcgill.ca
Navin Mordani - 260744902 - navin.mordani@mail.mcgill.ca
Jeremy Georges-Filteau - 260713547 - jeremy.georges-filteau@mail.mcgill.ca

I. INTRODUCTION

The Montreal Oasis Marathon takes place every year in Montreal, Canada since 1979. Data representing participant's attendance and their performance at the the event from 2012 to 2015 along with the record of their performances at other similar events is available online. In hopes of predicting the attendance for the upcoming race in 2016 (hereafter referred to as problem Y1) and their time of completion (problem Y2), three machine learning algorithms were implemented to train models on the given data.

II. METHODS

Problem representation

The features selected to predict Y1 are presented in Table 1 and similarly for Y2 in Table 2. For both problems the GENDER was obtained by parsing the CATEGORY feature in the original data. Male represented as +1 and female as -1. Missing or conflicting values were assigned to 0.

For the first case, predicting Y1, the TOTAL_EVENTS feature was obtained by counting the total number of running events attended by the participant in the span of last three years. The YEAR-1, YEAR-2 and YEAR-3 are binary features representing the attendance of the participant at the 3 Montreal Marathons preceding the current prediction year. In the training phase of the model, YEAR-1 to 3 are considered to be 2012 to 2014 (training for predicting attendance in 2015). Accordingly, in the prediction phase YEAR-1 to 3 are considered to be 2013 to 2015 (for predicting attendance in 2016).

For the second problem at hand, predicting the time of completion at Montreal Oasis Marathon in 2016, the following features were considered: age, gender and average time for completing a marathon in each of the three preceding years (AVG-TIME-YEAR-1, AVG-TIME-YEAR-2, AVG-TIME-YEAR-3). Since many of the events considered for the average time of completion were not full marathons and were different in terms of distance (primarily 1 km, 5km, 10km and half or demi marathon (21 km)), it was necessary to extrapolate the equivalent finishing time in marathon distance. Extrapolating these times linearly based on the speed would have been an inaccurate measure as this would assume the speed to remain constant even when the distance increases many-folds. To estimate this precisely, a formula was used (based on data from external sources) that measures the equivalent time for a marathon. [3] For detailed information about this method refer to Appendix A. For the participants whose average completion time for a certain year was not available, the mean of their average completion times in other years along with some random noise was used to compensate. The noise of ± 1200 sec was added randomly to them. If a participant's time of completion was -1 then this was replaced with a time of 6 hours which is the maximum time to complete a marathon.

To train the model, the average time of each participant (accumulated over all running events) in each of the 3 preceding years, i.e., 2012, 2013 and 2014 was used to predict their time in Montreal Marathon 2015. To predict the times in 2016, the years 2013, 2014, 2015 were used. The age of a participant was calculated by taking the average of the categorical age over all the events. To explain this better an example is present in Appendix A.

Table 1 - Features selected to predict Y1

Feature Name	Type	Domain	Normalized
GENDER	Categorical	{-1, 0, 1}	-
TOTAL_EVENTS	Continuous	$\mathbb{R}+$	Yes
YEAR-1	Categorical	{0, 1}	-
YEAR-2	Categorical	{0, 1}	-
YEAR-3	Categorical	{0, 1}	-

Table 2 - Features selected to predict Y2

Feature Name	Units	Type	Domain	Normalized
AVG-TIME-YEAR-1	hours	Continuous	$\mathbb{R}+$	Yes
AVG-TIME-YEAR-2	hours	Continuous	$\mathbb{R}+$	Yes
AVG-TIME-YEAR-3	hours	Continuous	$\mathbb{R}+$	Yes
GENDER	-	Categorical	{-1,0,1}	Yes
AGE	years	Continuous	$\mathbb{R}+$	Yes
	-			

Training methods

For every model the true prediction error was estimated using k-fold cross validation. The number of folds was chosen individually for every task as a compromise between reducing bias and computational cost of the tasks. Before running the k-fold cross validation, the order of the instances is randomized to reduce bias. A Naive Bayes and a Logistic regression model were trained on the features presented in Table 1. Linear Regression was used for the features present in Table 2.

Linear regression

After extracting the required features from the raw dataset and normalizing them, both closed form matrix method and gradient descent were applied to train the model on the dataset. To reduce bias, a vector of 1's was added as the intercept. The learning rate α for gradient descent was decided upon after observing the error for different values of α. It was observed that for larger values of α the error would increase along with the number of iterations and thus a small value (α=0.001) was chosen. K fold cross validation was applied with k=10. The model was trained for all the participants who took part in the 2015 Montreal Marathon, i.e., 2831 participants. Based on the weights returned on the 2831 participants the time of completion of all the remaining participants was predicted. The loss function of least mean square errors was employed.

Naive Bayes

The Naive Bayes algorithm was implemented according to course documentation. The distribution of discrete features was assumed to be Multinomial. Laplace smoothing was incorporated to account for values not present in the training set. [4] The distribution of continuous features was assumed to be Bayesian and the probabilities estimated with the scipy.stats.norm package. [2]

Logistic regression

The logistic regression model was implemented according to course documentation with gradient descent optimization. The stopping criteria was chosen as a threshold on the gradient, up to a maximum number of iterations. In other words, the learning is stopped when the average of the gradient vector Δw goes under the specified threshold value. The stopping criteria and the learning rate were implemented to be automatically adjusted according to dataset size, multiplying and dividing them by n, respectively. A limit on the number of iterations during the training phase was also added.

III. RESULT

Linear regression

After applying linear regression the following hypothesis was generated:

$$Y2_{Precited} = 0.40706839 * AVG\text{-}TIME\text{-}YEAR\text{-}3 + 0.15478422 * AVG\text{-}TIME\text{-}YEAR\text{-}2 + 0.21527892 * AVG\text{-}TIME\text{-}YEAR\text{-}1 + 0.03778091 * AGE - 0.06489948 * GENDER + 4.69862935$$

We see that the features GENDER and AGE have less weights compared to the weights of the average time in different years thus proving that these have less impact for predicting the running time of the participant as compared to the participant's running time in previous years. Different performance measures obtained using Linear Regression are present in Table 3. The correlation between predicted and actual values for Y2 are plotted in Figure 1.

Table 3 - Performance Measures for Y2

Mean Squared Error	0.33
Mean Absolute Error	0.30

Correlation	0.83
Mean Absolute Error Percentage	6.68%

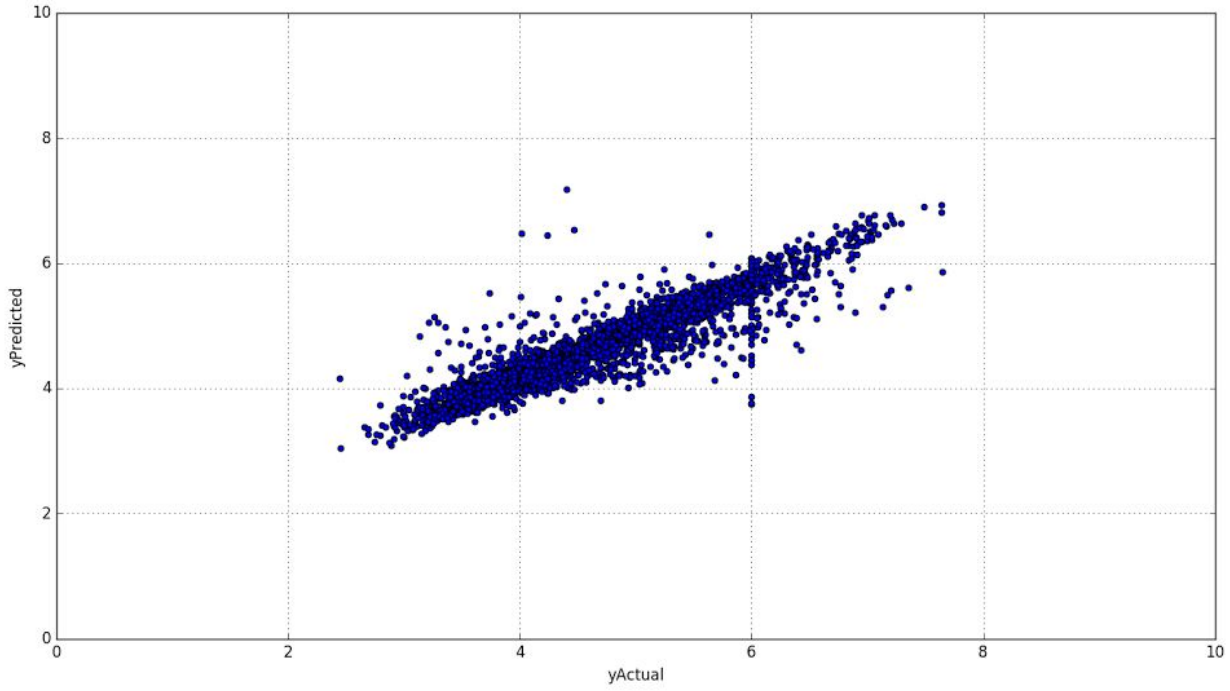


Figure 1 - Graph showing correlation between $Y2_{actual}$ and $Y2_{predicted.}$ predicted using linear regression.

In order to check if the loss function used was converging to a minima and also to be sure that the model was not overfitting a graph (*Figure 2*) of training error and testing error was plotted against the number of iterations used in gradient descent. For this purpose the dataset was divided into training set and test set in the ratio of 7:3. Based on this the number of iterations of gradient descent was set to 2000 as the stopping point.

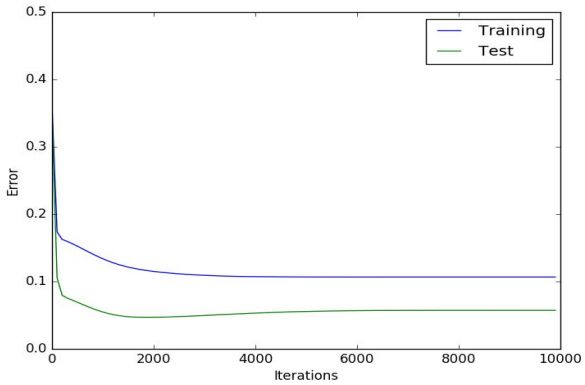


Figure 2 - Training and validation mean squared error rate versus number of iterations using the linear regression model and $\alpha = 0.001$

To see the relation between the feature to be predicted (i.e., Y2,the time of completion) and the features selected, multiple graphs were plotted between them. These could be referred in Appendix C

Naive Bayes

Prediction results for training and validation of the Naive Bayes model with 10-fold cross validation are presented in Table 4. The Bayes error rate is the probability of an instance being misclassified knowing the true class probability. [1] The error rates are nearly identical in the training and validation phase.

Table 4 - Mean Error rate and Mean True Error rate for training and validation of the Naive Bayes model with 10-fold cross validation.

Phase	Mean Error Rate	Mean Bayes Error Rate
Training	0.160	0.303
Validation	0.160	0.304

Logistic regression

The iteration limit for the algorithm was intuitively set to 400, as higher values did not yield improvement in any test runs. To select an appropriate

value for the learning rate, multiple values were tested. Very large values ($\alpha \gg 50$) caused the gradient descent algorithm to oscillate, while very small values ($\alpha \ll 0.01$) made the runs attain the iteration limit set to 400. In any case, the effect of the learning rate on the error rate with 5-fold cross validation was insignificant. The value was thus set at $\alpha = 1$ for subsequent runs.

To select an appropriate threshold for the stopping criteria, multiple values of the parameter were tested with 3-fold cross validation. The training and validation error rate for this test are presented in Appendix B-2. Improvement of the error rates can be obtained by lowering the stopping criteria, but values under *0.04* do not continue to provide significant improvement. A value of $\Delta w <= 0.005$ was chosen for the threshold in subsequent runs as to reduce running time without affecting results significantly.

To select the optimal decision boundary threshold, an ROC curve was produced by training on part of the dataset [:-2000] and predicting on the rest for attendance in 2015 for different values of the threshold (Appendix B-2). Any value between 0.15 and 0.55 inclusively is an acceptable compromise between FPR (False Positive Rate) and TPR (True Positive Rate) (0.21 and 0.93 respectively). A value 0.6 reduces slightly the number of false positives but has a greater negative effect on recall. Values between 0.05 and 0.10 inclusively slightly enhance the TPR but bring the FPR to unacceptable levels For our prediction purposes, reducing the FPR is as important as enhancing the TPR.

To better understand the prediction behavior of the model, a confusion matrix was also produced by training the logistic regression model on part of the data [:2000] and predicting the whole for attendance in 2015. The number of false-positives is much higher than the number of false-negatives, as previously seen in the ROC false-positive rate. This probably means that a better balance between both could be achieved with different features or models.

Table 6 - Confusion matrix for the logistic regression model trained on a subset (n=2000) predicting attendance for all participants in 2015

Predicted class	Actual class	
	True	**False**
True	2608	1183
False	222	4698

Error rates for training and validation of the Logistic regression model with 5-fold cross validation are presented in Table 5. At the end of each k validation phase, weights obtained in the model were such as Equation 1.

Equation 1 - Relation between the weights obtained in the Logistic regression model after each k validation phase.

$$w_{YEAR-1} > w_{YEAR-2} > w_{YEAR-3}$$

$$w_{YEARS-1-2-3} \gg w_{GENDER} \mid w_{TOTAL-EVENTS}$$

Table 5 - Mean Error rate for training and validation of the Logistic regression model with 5-fold cross validation.

Phase	Mean Error Rate
Training	0.16
Validation	0.16

Predictions

The predictions made for attendance at the Montreal Marathon are presented in Table 7. The results are very similar for both models used.

Table 7 - Attendance predictions for 2016 Montreal Marathon according to the different models tested.

Model	Participants predicted			
	Attending	**%**	**Not attending**	**%**
Naive Bayes	2218	25.5	6493	74.5
Logistic regression	2263	26.0	6448	74.0

IV. DISCUSSION

Both models used to predict Y1 produce very similar error rates and agree on attendance for the vast majority of participants. We are thus moderately confident of our prediction.

The weights obtained for every feature in the Logistic regression model seem intuitively correct. It is logical that the year preceding the prediction has more correlation with the attendance in prediction year than earlier years in descending order. It is also not surprising that gender is not giving much weight in predicting attendance. It would be interesting to test the effect of removing this feature on the error rate.

The Naive Bayes model depends on the assumption that all features are conditionally independent. However, we predict attendance in one year from attendance in multiple previous years. By claiming correlation in one year from previous ones for prediction we contradict that our features are conditionally independent. It would thus be necessary to use a model that takes into account this dependence between our features, such as LDA.

Overall, a 84% percent accuracy seems acceptable for the goal at hand and considering the data provided. Obviously, investing more time in fine tuning the features could yield improvements.

To solve Y2, linear regression predicted an absolute mean error percentage of about 6.7% which though acceptable could have been reduced, by accumulating more data about the participants' workout regime, the elevation of the marathon track and weather conditions during the previous year marathons. It would have also been good to observe the error percentage after considering higher degrees of certain features.

V. STATEMENT OF CONTRIBUTION

Coding the script for preparing the data for Y1 used with the Naive Bayes and the Logistic Regression algorithm, coding the the Naive Bayes and the Logistic Regression algorithm, testing and producing the predictions from these algorithms and writing the report parts concerning these algorithms was done by Jeremy.

Describing the problem, cleaning the data, extracting the features from the dataset, writing parts of the report for Y2 using linear regression was done by Jaspal.

Implementation of the linear regression algorithm for Y2 along with the testing of it and literature survey for the extra insights on the problem was done by Navin.

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

VI. REFERENCES

[1] Wikipedia contributors. 2016. "Bayes Error Rate." *Wikipedia, The Free Encyclopedia*. August 2. https://en.wikipedia.org/w/index.php?title=Bayes_error_rate&oldid=732668070.

[2] "Scipy.stats.norm — SciPy v0.16.1 Reference Guide." 2016. Accessed September 23.http://docs.scipy.org/doc/scipy-0.16.0/reference/generated/scipy.stats.norm.html.

[3] Jordan, Kevin. 2016. "Coaches Education - Running Training: Determining Your Current Level of Fitness." Accessed September 23. http://www.coacheseducation.com/endur/jack-daniels-nov-00.php

[4] Mitchell, Tom M., 2016, Machine Learning, McGraw Hill

APPENDIX

Appendix A

1. Calculating equivalent time for full marathon

There were many cases in the dataset where the participant's completion time of non marathon events are mentioned. Deleting all such data from the dataset would have reduced the data by an appreciable margin. Thus, to prevent this loss in data a method was devised to predict a participant's equivalent time in marathon compared to his time in smaller running events.

To estimate this a table was referred from the internet[3] and based on this table the below formulae were derived using linear regression:

1. $\text{Time}_{\text{Full Marathon}} = 1.9755 * \text{Time}_{\text{Half Marathon}} + 637.3307$
2. $\text{Time}_{\text{Full Marathon}} = 2.8096 * \text{Time}_{\text{15kms}} + 800.1356$
3. $\text{Time}_{\text{Full Marathon}} = 4.3268 * \text{Time}_{\text{10kms}} + 807.3818$
4. $\text{Time}_{\text{Full Marathon}} = 46.8293 * \text{Time}_{\text{1kms}} + 1598.3947$

The unit of time in the above is in seconds.

2. Extracting features for Y2: An example

Consider the below data for a given participant is retrieved from the raw dataset.

Event Year	Event Distance (km)	Time of Completion (in sec)	Equivalent time in Marathon (in sec) (Calculated as explained above)	Category	Gender (extracted From Category)	Age (Extracted from category)
2012	42.2	14264	14264	M35-39	M	37
2012	21.1	5400	11305.0307	M30-34	M	32
2013	10	3000	13787.7818	M35-39	M	37

Using the data of the table above, the features for Y2 i.e., feature matrix X and output vector y is shown below. These features were grouped by year.

Feature Matrix X (not normalized)					**Output Vector Y**
AVG-TIME 2012 (in hours)	**AVG-TIME 2013 (in hours)**	**AVG-TIME 2014 (in hours)**	**AGE**	**GENDER**	**TIME IN 2015 MONTREAL MARATHON (retrieved from raw dataset) (in hours)**
3.55 = Mean (14264,11305.0307)/ 3600	3.83 =(13787.7818/3600)	3.69 + randomNoise =Mean(AVG-TIME 2012, AVG-TIME 2013)	35.3 =Mean(37,32,37)	M	3.9

The feature matrix X was normalized and then used in the linear regression algorithm.

Appendix B - Additional figures

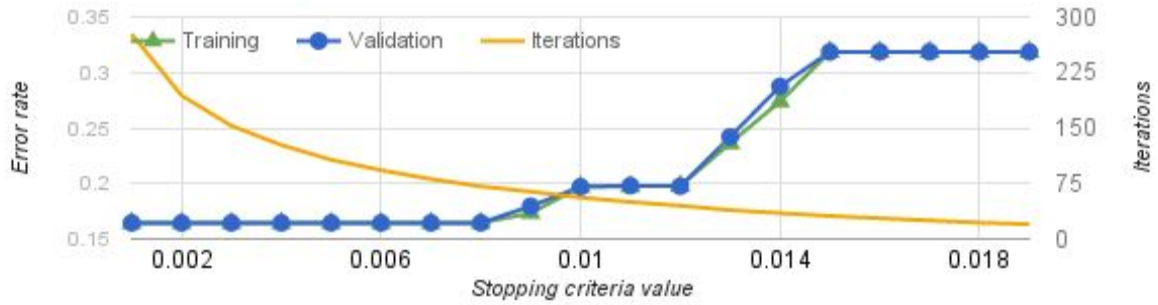
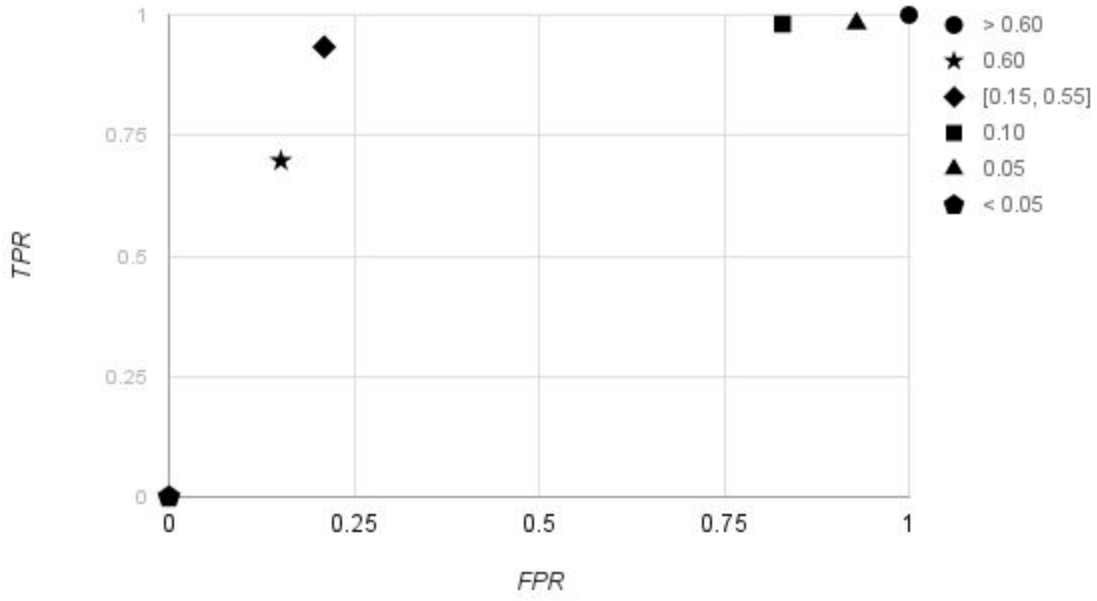


Figure B-1 - Average training and validation error rate and average number of iterations with 3-fold cross validation versus stopping criteria.

Figure B-2 - Receiver-Operator characteristic curve (ROC) for different values of the decision boundary threshold in the



Logistic Regression model. True Postitive Rate (TPR) versus False Positive Rate (FPR)

Appendix C

Variation of output Y2 vs all the features used for Linear Regression

Figure C-1: TimeOfCompletion vs Age

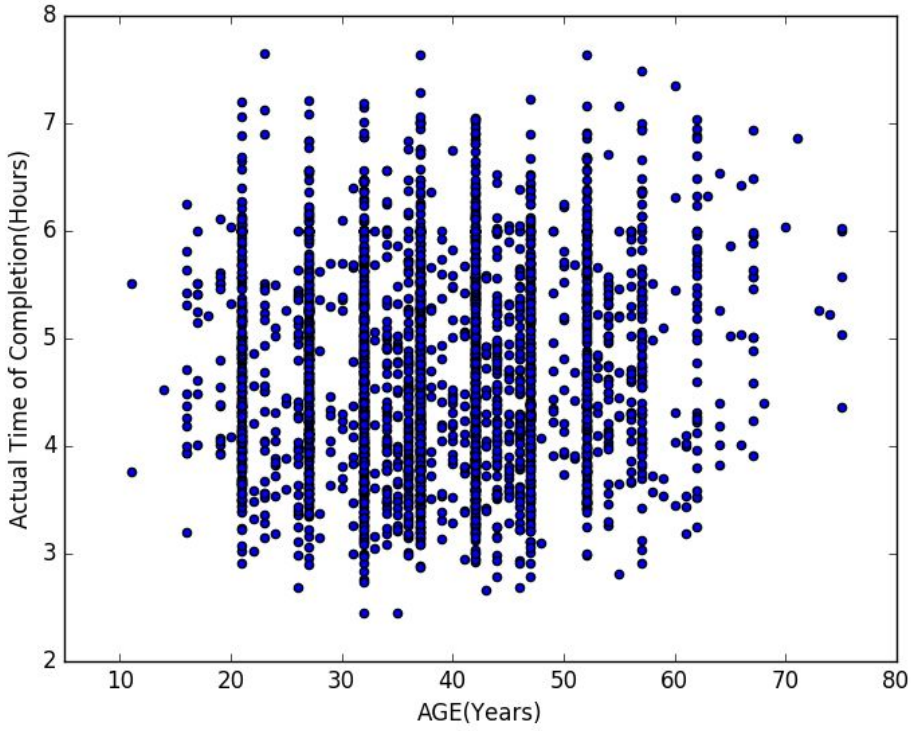


Figure C-2: TimeOfCompletion vs Gender

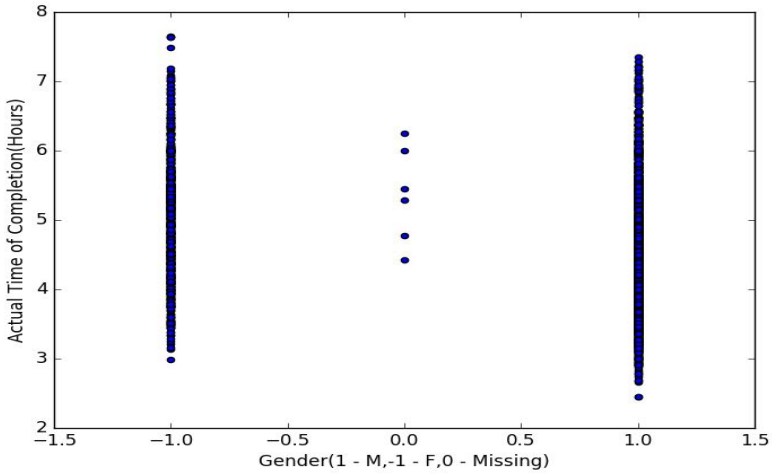


Figure C-3: TimeOfCompletion vs AVG-TIME-YEAR-1(Time 2014)

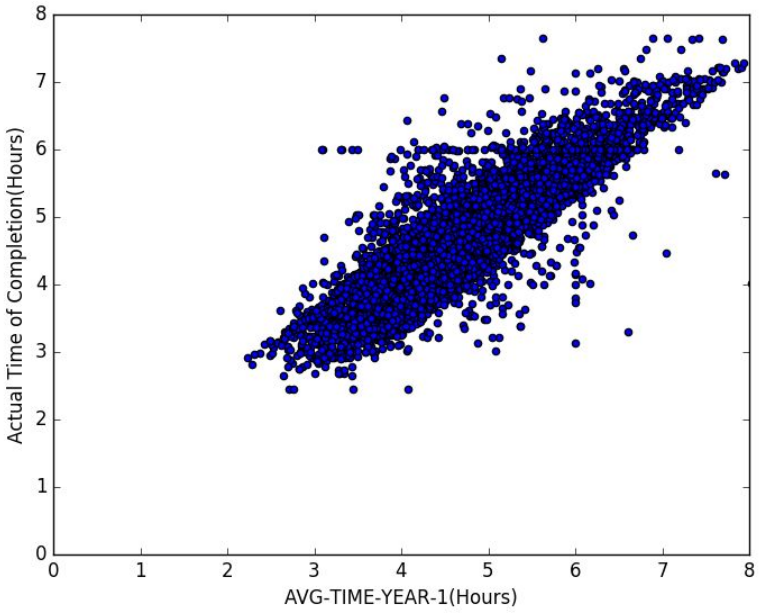


Figure C-4: TimeOfCompletion vs AVG-TIME-YEAR-2(Time 2013)

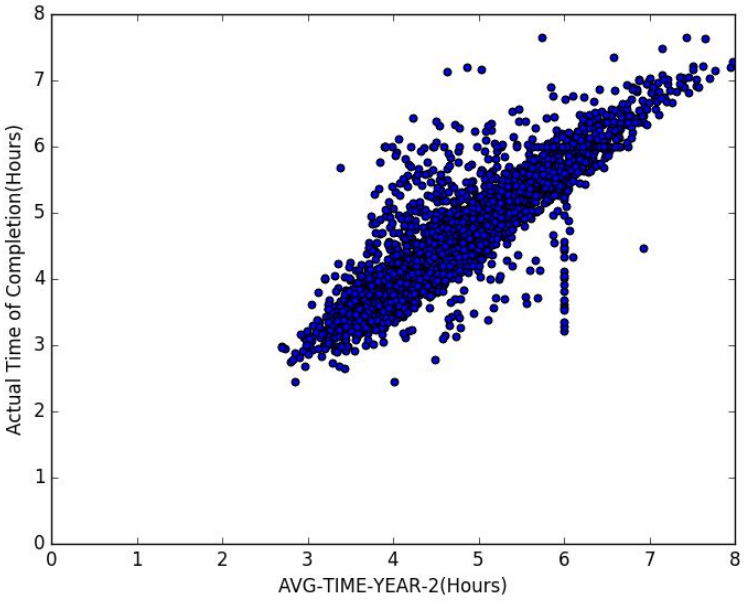


Figure C-5: TimeOfCompletion vs AVG-TIME-YEAR-3(Time 2012)