Heroes of the Storm Balance Analysis

Gavin Moss

March 4, 2017

Introduction

Is *Heroes of the Storm* balanced? Are there some characters that win significantly more often at all skill levels? Does Skill level or hero selection matter more in determening who wins a game? These are questions that many compedative players ask. In this document I will take a numerical approach to answer those questions using machine learning.

What is *Heroes of the Storm*

Heroes of the storm is a Multiplayer Online Battle Arena (MOBA) where two teams of five players take control of diffrent characters in order to destroy the oposing team's base. Each character has diffrent abilities and characteristics. Such as how much damage they can receive before dying and having to wait to join the game again, and how much damage they can deal out. The game tries to help players have competative games and shoots for any given player to win 50% of their games.

The Data

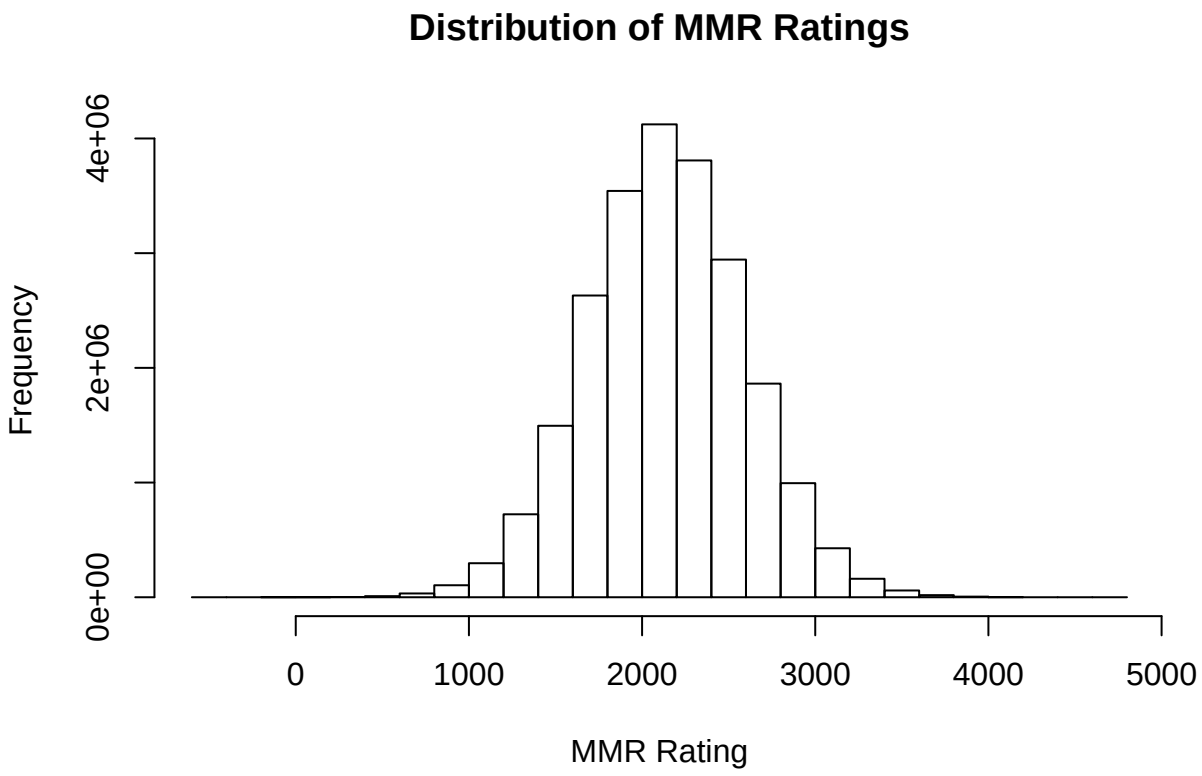
The data used in this analysis comes from HotsLogs.com and can be found at https://www.hotslogs.com/Info/API. The data used in the analysis was downloaded on 3/2/2017 and has data up through 2/26/2017.

Hots logs gathers this data through voluntary submissions of replay files. Each replay file has data for all 10 players in that game. This includes What hero people played, what level that hero is for the player (an indication of overall time spent playing a specific hero),and an estimated MMR ranking for an individual and more.

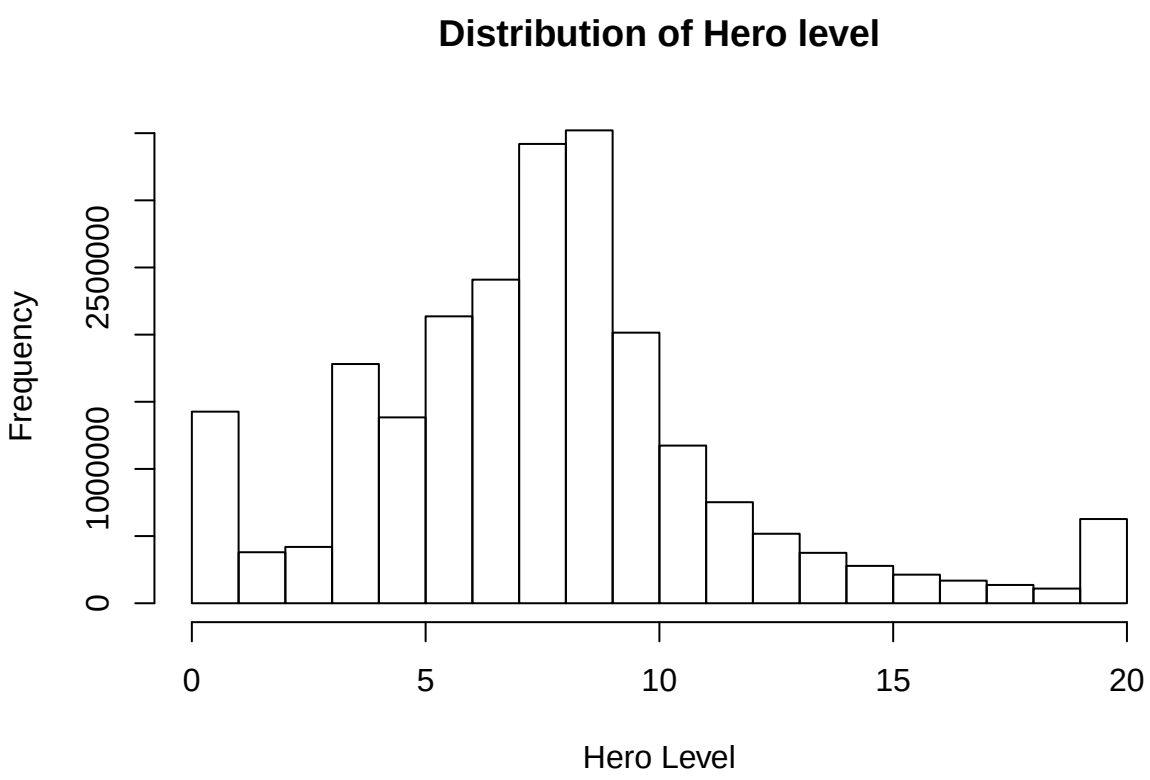
The data used in the analysis was downloaded on 3/2/2017 and has data up though 2/26/2017, and has 23,242,890 records. Of particular note the data has an even split of 11,621,445 records of wins, and 11,621,445 of losses, with all characters being played between 70,736 and 928,234 times.

There is a nice distribution of players at diffrent skill ratings (MMR), and uising heroes at diffrent varying levels.

```
hist(graphs$MMR.Before, main = "Distribution of MMR Ratings", xlab = "MMR Rating")
```




```
hist(graphs$Hero.Level, main = "Distribution of Hero level", xlab = "Hero Level")
```



The analysis

To answer the questions of is the game balanced, and what matters more for determining winning: Skill level or hero selection, We need to see what effect each Character has on a player's chance of winning a game, as well as what effect their skill has.

To measure a player's skill, I am going to use two variables. The first is the player's MMR before the game started. MMR is a skill rating given to a player by the game to assist in the match making process, and as such is a good approximator for the player's over all skill at the game. As the exact MMR is a number on an abitrary scale I standardize this data point using Min Max standardization. In some cases the MMR value is blank; this is due to a lack of games played. As there is no way of knowing how skilled that player is at competative MOBA Games I have set all null MMR Values to the median MMR Value.

As Players may or may not be equaly skilled with every character in the game I will be using Hero Level to measure how skilled a player is with a given character. This is not a perfect measure however as players gain levels with a character based of the number of games they have played with that character, not on how good they are with them. However I belive that it is reasonable to assume as a player spends more time playing a character they will also become more skilled with that character.

I use a logistig regression algorithem to isolate the effect of these variables from one another and to build a predictive model for who will win and lose any given game. It is important to note that if any sort of accurate prediction can be made as to who will win a game then the game is not balanced properly.

Create base logistic model

```
Char_v_skill <- h2o.glm(
  x=c("Name","Hero Level", "MMR standardized"),
  y="Is Winner",
  training_frame = train_char,
  validation_frame = test_char,
  family="binomial"
)
```

```
#summarize model and its preformance
summary(Char_v_skill)
```

```
## Model Details:
## ==============
##
## H2OBinomialModel: glm
## Model Key:  GLM_model_R_1488850374422_1
## GLM Model: summary
##     family  link                                regularization
## 1 binomial logit Elastic Net (alpha = 0.5, lambda = 1.731E-5 )
##   number_of_predictors_total number_of_active_predictors
## 1                         64                          60
##   number_of_iterations training_frame
## 1                    2     train_char
##
## H2OBinomialMetrics: glm
## ** Reported on training data. **
##
## MSE:  0.2489901
## RMSE:  0.4989891
## LogLoss:  0.6911222
## Mean Per-Class Error:  0.5
## AUC:  0.5354687
## Gini:  0.07093733
## R^2:  0.004039514
## Null Deviance:  27388126
## Residual Deviance:  27308113
## AIC:  27308235
##
## Confusion Matrix for F1-optimal threshold:
##        0        1    Error               Rate
## 0      0  9877547 1.000000   =9877547/9877547
## 1      0  9878810 0.000000         =0/9878810
## Totals 0 19756357 0.499968  =9877547/19756357
##
## Maximum Metrics: Maximum metrics at their respective thresholds
##                         metric threshold    value idx
## 1                       max f1  0.365527 0.666695 399
## 2                       max f2  0.365527 0.833351 399
## 3                 max f0point5  0.446687 0.556841 326
## 4                 max accuracy  0.498407 0.525577 191
## 5                max precision  0.604114 0.697368   0
## 6                   max recall  0.365527 1.000000 399
```

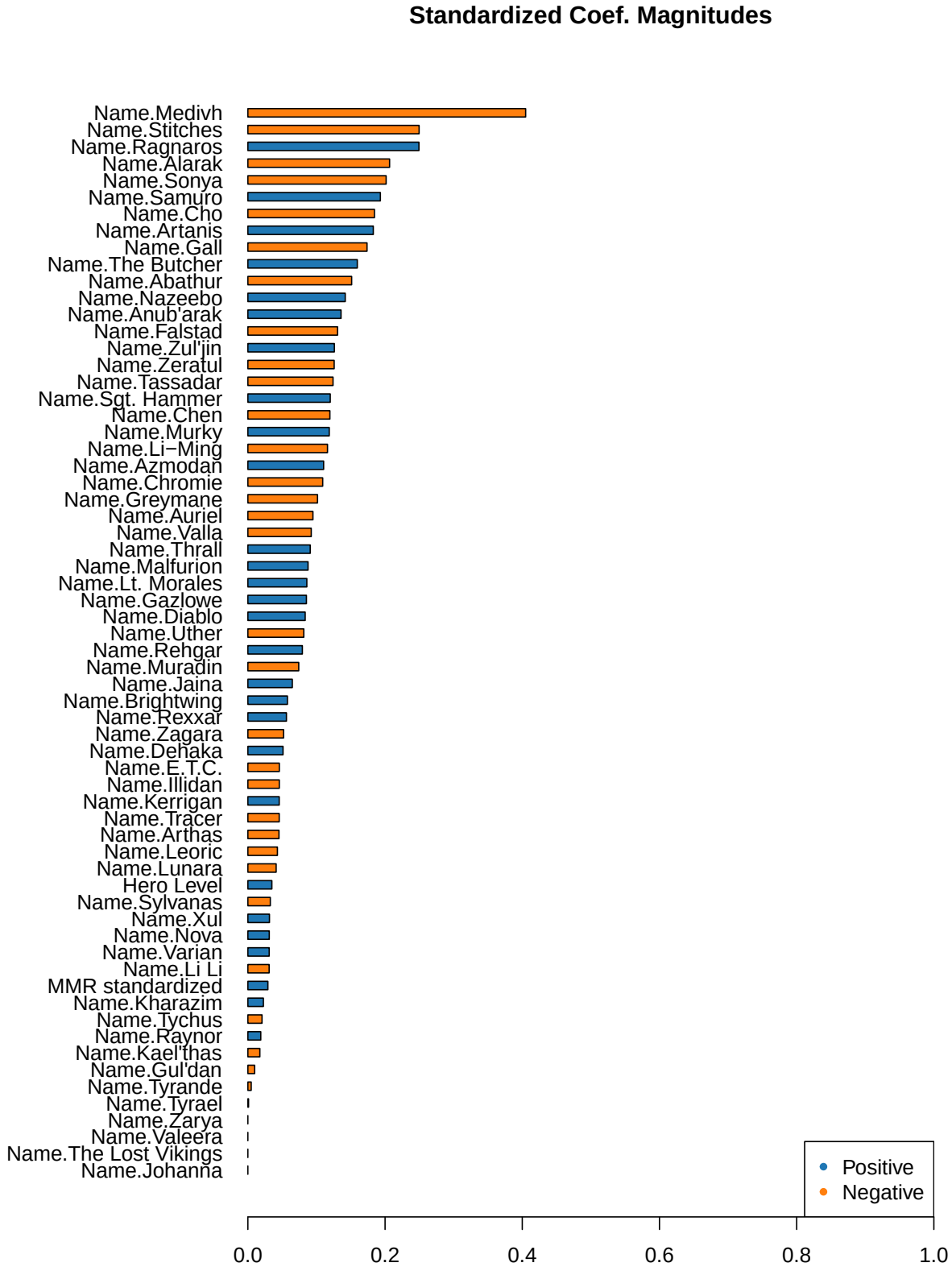
```
## 7              max specificity  0.604114 0.999998   0
## 8             max absolute_mcc  0.496034 0.051434 199
## 9   max min_per_class_accuracy  0.500985 0.524443 183
## 10 max mean_per_class_accuracy  0.498407 0.525575 191
##
## Gains/Lift Table: Extract with `h2o.gainsLift(<model>, <data>)` or `h2o.gainsLift(<model>, valid=<T/
## H2OBinomialMetrics: glm
## ** Reported on validation data. **
##
## MSE:  0.2489897
## RMSE:  0.4989887
## LogLoss:  0.6911213
## Mean Per-Class Error:  0.4999888
## AUC:  0.5354678
## Gini:  0.07093557
## R^2:  0.004041114
## Null Deviance:  4833361
## Residual Deviance:  4819234
## AIC:  4819356
##
## Confusion Matrix for F1-optimal threshold:
##         0       1    Error              Rate
## 0      69 1743829 0.999960  =1743829/1743898
## 1      30 1742605 0.000017       =30/1742635
## Totals 99 3486434 0.500170  =1743859/3486533
##
## Maximum Metrics: Maximum metrics at their respective thresholds
##                         metric threshold    value idx
## 1                        max f1  0.371927 0.666507 397
## 2                        max f2  0.356043 0.833233 399
## 3                  max f0point5  0.449254 0.556602 317
## 4                  max accuracy  0.499303 0.525489 183
## 5                 max precision  0.582017 0.632385   5
## 6                    max recall  0.356043 1.000000 399
## 7              max specificity  0.601287 0.999995   0
## 8             max absolute_mcc  0.492095 0.051709 206
## 9   max min_per_class_accuracy  0.501157 0.524143 177
## 10 max mean_per_class_accuracy  0.499303 0.525497 183
##
## Gains/Lift Table: Extract with `h2o.gainsLift(<model>, <data>)` or `h2o.gainsLift(<model>, valid=<T/
##
##
## Scoring History:
##             timestamp   duration iteration negative_log_likelihood
## 1 2017-03-06 18:34:57  0.000 sec         0          13694063.11217
## 2 2017-03-06 18:34:59  2.091 sec         1          13654064.58247
## 3 2017-03-06 18:35:00  3.078 sec         2          13654056.26456
##   objective
## 1   0.69315
## 2   0.69118
## 3   0.69118
##
## Variable Importances: (Extract with `h2o.varimp`)
## =================================================
```

```
##
## Standardized Coefficient Magnitudes: standardized coefficient magnitudes
##           names coefficients sign
## 1   Name.Medivh     0.404929  NEG
## 2 Name.Stitches     0.249640  NEG
## 3 Name.Ragnaros     0.249367  POS
## 4   Name.Alarak     0.206601  NEG
## 5    Name.Sonya     0.201561  NEG
##
## ---
##                    names coefficients sign
## 59          Name.Tyrande     0.004853  NEG
## 60           Name.Tyrael     0.001002  NEG
## 61          Name.Johanna     0.000000  POS
## 62 Name.The Lost Vikings     0.000000  POS
## 63          Name.Valeera     0.000000  POS
## 64            Name.Zarya     0.000000  POS
```

Looking at the summary of our model we can see that this model has almost no predictive power. With an "Area under the Curve" rating of just .53 and a mean error rate of.499 we see that this model can predict a win no better than a coin flip.

However, the variable importance and predected win probability for each data point can still give us insights into the overall game balance of Heros of the Storm. Below we see a ranking of what data points effect our prediction the most. Orannge bars indicate that the data point reduces a playrs change of wining, and a blue bar indicates that it increases a playrs chance to win.

```
h2o.varimp_plot(Char_v_skill)
```


Standardized Coef. Magnitudes
Name.Medivh
Name.Stitches
Name.Ragnaros
Name.Alarak
Name.Sonya
Name.Samuro
Name.Cho
Name.Artanis
Name.Gall
Name.The Butcher
Name.Abathur
Name.Nazeebo
Name.Anub'arak
Name.Falstad
Name.Zul'jin
Name.Zeratul
Name.Tassadar
Name.Sgt. Hammer
Name.Chen
Name.Murky
Name.Li−Ming
Name.Azmodan
Name.Chromie
Name.Greymane
Name.Auriel
Name.Valla
Name.Thrall
Name.Malfurion
Name.Lt. Morales
Name.Gazlowe
Name.Diablo
Name.Uther
Name.Rehgar
Name.Muradin
Name.Jaina
Name.Brightwing
Name.Rexxar
Name.Zagara
Name.Dehaka
Name.E.T.C.
Name.Illidan
Name.Kerrigan
Name.Tracer
Name.Arthas
Name.Leoric
Name.Lunara
Hero Level
Name.Sylvanas
Name.Xul
Name.Nova
Name.Varian
Name.Li Li
MMR standardized
Name.Kharazim
Name.Tychus
Name.Raynor
Name.Kael'thas
Name.Gul'dan
Name.Tyrande
Name.Tyrael
Name.Zarya
Name.Valeera
Name.The Lost Vikings
Name.Johanna
Positive
Negative
0.0
0.2
0.4
0.6
0.8
1.0

Below is a list of data points and how much they affect a player's chance of wining in descending order.

```
char_v_skill_prob <- exp(h2o.coef(Char_v_skill))/(exp(h2o.coef(Char_v_skill))+1)-.5
sort(char_v_skill_prob,decreasing = TRUE)
```

```
##      MMR standardized         Name.Ragnaros           Name.Samuro
##           0.0816466550           0.0620206187           0.0481412434
##           Name.Artanis      Name.The Butcher           Name.Nazeebo
##           0.0455885482           0.0397965805           0.0354277071
##         Name.Anub'arak           Name.Zul'jin      Name.Sgt. Hammer
##           0.0339047448           0.0314771500           0.0299805882
##             Name.Murky           Name.Azmodan            Name.Thrall
##           0.0296202658           0.0275766867           0.0227051748
##         Name.Malfurion      Name.Lt. Morales           Name.Gazlowe
##           0.0219172115           0.0214761507           0.0213267136
##            Name.Diablo            Name.Rehgar              Name.Jaina
##           0.0208726875           0.0198526657           0.0161944013
##       Name.Brightwing            Name.Rexxar            Name.Dehaka
##           0.0144095007           0.0140491460           0.0128142437
##          Name.Kerrigan               Name.Xul              Name.Nova
##           0.0114186417           0.0078684256           0.0077963208
##            Name.Varian          Name.Kharazim            Name.Raynor
##           0.0077689898           0.0056561825           0.0047009412
##             Hero Level           Name.Johanna Name.The Lost Vikings
##           0.0021890927           0.0000000000           0.0000000000
##           Name.Valeera             Name.Zarya            Name.Tyrael
##           0.0000000000           0.0000000000          -0.0002505943
##           Name.Tyrande           Name.Gul'dan         Name.Kael'thas
##          -0.0012133570          -0.0024756501          -0.0043665187
##            Name.Tychus             Name.Li Li          Name.Sylvanas
##          -0.0051605891          -0.0077518195          -0.0081789429
##            Name.Lunara            Name.Leoric            Name.Arthas
##          -0.0103017188          -0.0107631069          -0.0113330081
##            Name.Tracer           Name.Illidan            Name.E.T.C.
##          -0.0114185197          -0.0114532257          -0.0114584412
##            Name.Zagara           Name.Muradin             Name.Uther
##          -0.0130070625          -0.0185327431          -0.0203730500
##             Name.Valla            Name.Auriel          Name.Greymane
##          -0.0230851620          -0.0236919868          -0.0253627496
##           Name.Chromie           Name.Li-Ming              Name.Chen
##          -0.0272422315          -0.0289850981          -0.0298549795
##          Name.Tassadar           Name.Zeratul           Name.Falstad
##          -0.0309984551          -0.0314068006          -0.0326425576
##           Name.Abathur              Name.Gall               Name.Cho
##          -0.0377516701          -0.0433154483          -0.0460029208
##             Name.Sonya            Name.Alarak          Name.Stitches
##          -0.0502203041          -0.0514673933          -0.0620879687
##              Intercept            Name.Medivh
##          -0.0627499126          -0.0998713388
```

Adjust model to improve predictive power

As the base model had little predictive power lets try to tune it to increase its power. To do this, I employ a grid serch testing different coeffeicent penalty methods ranging from a pure ridge regression to a pure lasso regression.

```
hyper_parameters <- list(alpha = c(0,.2,.4,.6,.8,1))
#Creat lasso logit regression for all games
char_v_skill_grid <- h2o.grid(
    algorithm = "glm",
    grid_id = "char_v_skill",
    hyper_params= hyper_parameters,
    training_frame = train_char,
    validation_frame = test_char,
    x=c("Name","Hero Level", "MMR standardized"),
    y="Is Winner",
    lambda_search = TRUE,
    family = "binomial"
)
```

```
glm.sorted.grid <- h2o.getGrid("char_v_skill","auc")
best_model <- h2o.getModel(glm.sorted.grid@model_ids[[1]])
best_model
```

```
## Model Details:
## ==============
##
## H2OBinomialModel: glm
## Model ID:  char_v_skill_model_4
## GLM Model: summary
##     family  link                                regularization
## 1 binomial logit Elastic Net (alpha = 0.8, lambda = 0.007456 )
##                                                                lambda_search
## 1 nlambda = 100, lambda.max = 0.01082, lambda.min = 0.007456, lambda.1se = -1.0
##   number_of_predictors_total number_of_active_predictors
## 1                         64                           2
##   number_of_iterations training_frame
## 1                    5     train_char
##
## Coefficients: glm coefficients
##            names coefficients standardized_coefficients
## 1      Intercept    -0.071506                  0.000128
## 2   Name.Abathur     0.000000                  0.000000
## 3    Name.Alarak     0.000000                  0.000000
## 4 Name.Anub'arak     0.000000                  0.000000
## 5   Name.Artanis     0.000000                  0.000000
##
## ---
##               names coefficients standardized_coefficients
## 60      Name.Zagara     0.000000                  0.000000
## 61       Name.Zarya     0.000000                  0.000000
## 62     Name.Zeratul     0.000000                  0.000000
## 63     Name.Zul'jin     0.000000                  0.000000
## 64       Hero Level     0.001757                  0.007021
## 65 MMR standardized     0.111095                  0.009833
```

```
##
## H2OBinomialMetrics: glm
## ** Reported on training data. **
##
## MSE:  0.2499394
## RMSE:  0.4999394
## LogLoss:  0.6930261
## Mean Per-Class Error:  0.5
## AUC:  0.5118171
## Gini:  0.0236342
## R^2:  0.0002422363
## Null Deviance:  27388126
## Residual Deviance:  27383340
## AIC:  27383346
##
## Confusion Matrix for F1-optimal threshold:
##        0        1    Error               Rate
## 0      0  9877547 1.000000   =9877547/9877547
## 1      0  9878810 0.000000         =0/9878810
## Totals 0 19756357 0.499968  =9877547/19756357
##
## Maximum Metrics: Maximum metrics at their respective thresholds
##                         metric threshold    value idx
## 1                       max f1  0.485436 0.666695 399
## 2                       max f2  0.485436 0.833351 399
## 3                 max f0point5  0.490214 0.555591 391
## 4                 max accuracy  0.499875 0.507869 217
## 5                max precision  0.515656 0.659091   0
## 6                   max recall  0.485436 1.000000 399
## 7              max specificity  0.515656 0.999998   0
## 8             max absolute_mcc  0.498836 0.015828 248
## 9   max min_per_class_accuracy  0.499953 0.507804 215
## 10 max mean_per_class_accuracy  0.499875 0.507868 217
##
## Gains/Lift Table: Extract with `h2o.gainsLift(<model>, <data>)` or `h2o.gainsLift(<model>, valid=<T/
## H2OBinomialMetrics: glm
## ** Reported on validation data. **
##
## MSE:  0.2499388
## RMSE:  0.4999388
## LogLoss:  0.6930248
## Mean Per-Class Error:  0.5
## AUC:  0.511817
## Gini:  0.02363404
## R^2:  0.0002446734
## Null Deviance:  4833361
## Residual Deviance:  4832507
## AIC:  4832513
##
## Confusion Matrix for F1-optimal threshold:
##        0       1    Error              Rate
## 0      0 1743898 1.000000  =1743898/1743898
## 1      0 1742635 0.000000        =0/1742635
## Totals 0 3486533 0.500181  =1743898/3486533
```

```
##
## Maximum Metrics: Maximum metrics at their respective thresholds
##                         metric threshold    value idx
## 1                       max f1  0.485474 0.666506 399
## 2                       max f2  0.485474 0.833233 399
## 3                 max f0point5  0.490973 0.555394 383
## 4                 max accuracy  0.499465 0.507825 222
## 5                max precision  0.515106 0.727273   0
## 6                   max recall  0.485474 1.000000 399
## 7              max specificity  0.515106 0.999995   0
## 8             max absolute_mcc  0.498530 0.016066 249
## 9   max min_per_class_accuracy  0.499963 0.506558 207
## 10 max mean_per_class_accuracy  0.499465 0.507847 222
##
## Gains/Lift Table: Extract with `h2o.gainsLift(<model>, <data>)` or `h2o.gainsLift(<model>, valid=<T/
```

Our best tuned model using a penatly factor turns out to have less predictive power than the base model, lending more credance to the idea that we cannot build an accurate predictive model. Of note however, is that when using a penalty factor on our variables to make sure the effects on a players win rate were significant, all but our MMR and Hero Level data points drop out of the model. This suggests that player skill, not certain characters, wins games.

```
h2o.varimp_plot(best_model)
```


Standardized Coef. Magnitudes
MMR standardized
Hero Level
Name.Zul'jin
Name.Zeratul
Name.Zarya
Name.Zagara
Name.Xul
Name.Varian
Name.Valla
Name.Valeera
Name.Uther
Name.Tyrande
Name.Tyrael
Name.Tychus
Name.Tracer
Name.Thrall
Name.The Lost Vikings
Name.The Butcher
Name.Tassadar
Name.Sylvanas
Name.Stitches
Name.Sonya
Name.Sgt. Hammer
Name.Samuro
Name.Rexxar
Name.Rehgar
Name.Raynor
Name.Ragnaros
Name.Nova
Name.Nazeebo
Name.Murky
Name.Muradin
Name.Medivh
Name.Malfurion
Name.Lunara
Name.Lt. Morales
Name.Li−Ming
Name.Li Li
Name.Leoric
Name.Kharazim
Name.Kerrigan
Name.Kael'thas
Name.Johanna
Name.Jaina
Name.Illidan
Name.Gul'dan
Name.Greymane
Name.Gazlowe
Name.Gall
Name.Falstad
Name.E.T.C.
Name.Diablo
Name.Dehaka
Name.Chromie
Name.Cho
Name.Chen
Name.Brightwing
Name.Azmodan
Name.Auriel
Name.Arthas
Name.Artanis
Name.Anub'arak
Name.Alarak
Name.Abathur
Positive
0.0
0.2
0.4
0.6
0.8
1.0

```
char_v_skill_grid_prob <- exp(h2o.coef(best_model))/(exp(h2o.coef(best_model))+1)-.5
sort(char_v_skill_grid_prob,decreasing = TRUE)
```

```
##      MMR standardized            Hero Level          Name.Abathur
##          0.0277451378          0.0004392831          0.0000000000
##           Name.Alarak        Name.Anub'arak          Name.Artanis
##          0.0000000000          0.0000000000          0.0000000000
##           Name.Arthas           Name.Auriel          Name.Azmodan
##          0.0000000000          0.0000000000          0.0000000000
##       Name.Brightwing             Name.Chen              Name.Cho
##          0.0000000000          0.0000000000          0.0000000000
##          Name.Chromie           Name.Dehaka           Name.Diablo
##          0.0000000000          0.0000000000          0.0000000000
##           Name.E.T.C.          Name.Falstad             Name.Gall
##          0.0000000000          0.0000000000          0.0000000000
##          Name.Gazlowe         Name.Greymane         Name.Gul'dan
##          0.0000000000          0.0000000000          0.0000000000
##          Name.Illidan             Name.Jaina          Name.Johanna
##          0.0000000000          0.0000000000          0.0000000000
##        Name.Kael'thas         Name.Kerrigan         Name.Kharazim
##          0.0000000000          0.0000000000          0.0000000000
##           Name.Leoric             Name.Li Li          Name.Li-Ming
##          0.0000000000          0.0000000000          0.0000000000
##      Name.Lt. Morales           Name.Lunara        Name.Malfurion
##          0.0000000000          0.0000000000          0.0000000000
##           Name.Medivh          Name.Muradin            Name.Murky
##          0.0000000000          0.0000000000          0.0000000000
##          Name.Nazeebo             Name.Nova         Name.Ragnaros
##          0.0000000000          0.0000000000          0.0000000000
##           Name.Raynor           Name.Rehgar           Name.Rexxar
##          0.0000000000          0.0000000000          0.0000000000
##           Name.Samuro      Name.Sgt. Hammer            Name.Sonya
##          0.0000000000          0.0000000000          0.0000000000
##         Name.Stitches         Name.Sylvanas         Name.Tassadar
##          0.0000000000          0.0000000000          0.0000000000
##      Name.The Butcher Name.The Lost Vikings           Name.Thrall
##          0.0000000000          0.0000000000          0.0000000000
##           Name.Tracer           Name.Tychus           Name.Tyrael
##          0.0000000000          0.0000000000          0.0000000000
##          Name.Tyrande             Name.Uther         Name.Valeera
##          0.0000000000          0.0000000000          0.0000000000
##            Name.Valla           Name.Varian              Name.Xul
##          0.0000000000          0.0000000000          0.0000000000
##           Name.Zagara             Name.Zarya         Name.Zeratul
##          0.0000000000          0.0000000000          0.0000000000
##          Name.Zul'jin             Intercept
##          0.0000000000         -0.0178688712
```

Do we get the same results when casual games are removed

As game balance is most important for competitive games I will recreate the same models as above but for only games played competitively.

```
Char_split_comp <- h2o.splitFrame(Characters[Characters[["GameMode(3=Quick Match 4=Hero League 5=Team L
ratios = .85,
destination_frames = c("train_char","test_char"),
seed = 444)

train_char <- Char_split_comp[[1]]
test_char <- Char_split_comp[[2]]

Char_v_skill_comp <- h2o.glm(
    x=c("Name","Hero Level", "MMR standardized"),
    y="Is Winner",
    training_frame = train_char,
    validation_frame = test_char,
    family="binomial"
)
```

```
Char_v_skill_comp
```

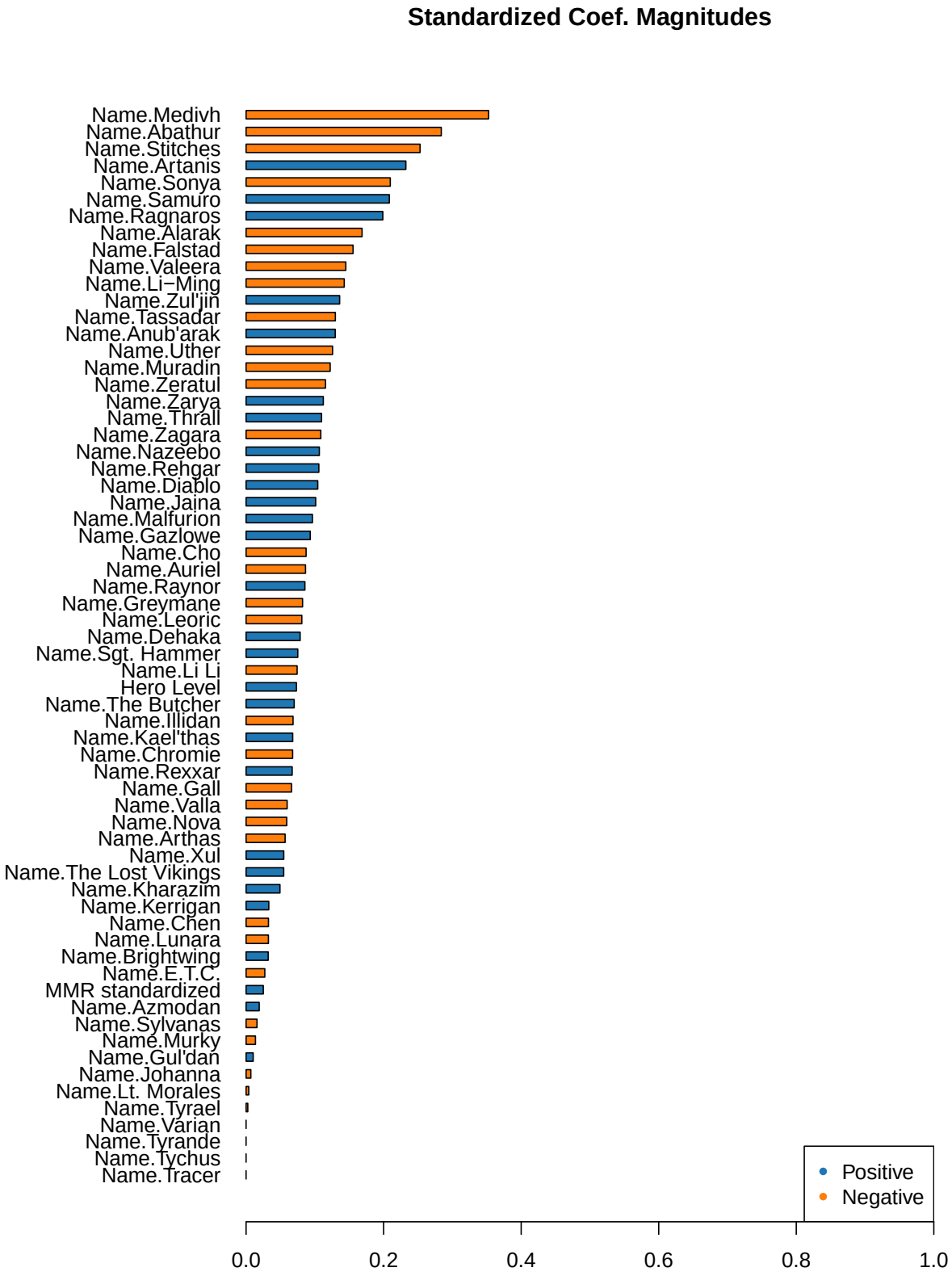
```
## Model Details:
## ==============
##
## H2OBinomialModel: glm
## Model ID:  GLM_model_R_1488850374422_18
## GLM Model: summary
##     family  link                                regularization
## 1 binomial logit Elastic Net (alpha = 0.5, lambda = 3.377E-5 )
##   number_of_predictors_total number_of_active_predictors
## 1                         64                         60
##   number_of_iterations training_frame
## 1                    2     train_char
##
## Coefficients: glm coefficients
##            names coefficients standardized_coefficients
## 1      Intercept    -0.359699                 -0.022025
## 2   Name.Abathur    -0.283849                 -0.283849
## 3    Name.Alarak    -0.168640                 -0.168640
## 4 Name.Anub'arak     0.129634                  0.129634
## 5   Name.Artanis     0.232347                  0.232347
##
## ---
##               names coefficients standardized_coefficients
## 60      Name.Zagara    -0.108651                 -0.108651
## 61       Name.Zarya     0.112318                  0.112318
## 62     Name.Zeratul    -0.115476                 -0.115476
## 63     Name.Zul'jin     0.136043                  0.136043
## 64       Hero Level     0.021277                  0.073232
## 65 MMR standardized     0.255320                  0.025055
##
## H2OBinomialMetrics: glm
## ** Reported on training data. **
##
## MSE:  0.2488968
## RMSE:  0.4988956
## LogLoss:  0.690936
```

```
## Mean Per-Class Error:  0.4997748
## AUC:  0.5377323
## Gini:  0.07546468
## R^2:  0.004412783
## Null Deviance:  8421823
## Residual Deviance:  8394957
## AIC:  8395079
##
## Confusion Matrix for F1-optimal threshold:
##           0       1    Error              Rate
## 0      2401 3035113 0.999210  =3035113/3037514
## 1      1033 3036514 0.000340     =1033/3037547
## Totals 3434 6071627 0.499772  =3036146/6075061
##
## Maximum Metrics: Maximum metrics at their respective thresholds
##                         metric threshold    value idx
## 1                       max f1  0.388636 0.666694 390
## 2                       max f2  0.356566 0.833335 399
## 3                 max f0point5  0.453140 0.556830 320
## 4                 max accuracy  0.498928 0.527074 200
## 5                max precision  0.616917 0.673469   0
## 6                   max recall  0.356566 1.000000 399
## 7              max specificity  0.616917 0.999989   0
## 8             max absolute_mcc  0.490665 0.054800 224
## 9   max min_per_class_accuracy  0.501064 0.526058 193
## 10 max mean_per_class_accuracy  0.498928 0.527074 200
##
## Gains/Lift Table: Extract with `h2o.gainsLift(<model>, <data>)` or `h2o.gainsLift(<model>, valid=<T/
## H2OBinomialMetrics: glm
## ** Reported on validation data. **
##
## MSE:  0.2489081
## RMSE:  0.4989069
## LogLoss:  0.6909584
## Mean Per-Class Error:  0.4997584
## AUC:  0.5373456
## Gini:  0.07469124
## R^2:  0.004367708
## Null Deviance:  1486065
## Residual Deviance:  1481372
## AIC:  1481494
##
## Confusion Matrix for F1-optimal threshold:
##          0       1    Error             Rate
## 0      501  535500 0.999065   =535500/536001
## 1      242  535726 0.000452      =242/535968
## Totals 743 1071226 0.499774  =535742/1071969
##
## Maximum Metrics: Maximum metrics at their respective thresholds
##                         metric threshold    value idx
## 1                       max f1  0.390570 0.666660 389
## 2                       max f2  0.356802 0.833325 399
## 3                 max f0point5  0.451750 0.556927 324
## 4                 max accuracy  0.495346 0.526841 210
```

```
## 5                max precision  0.615123 0.727273   1
## 6                   max recall  0.356802 1.000000 399
## 7              max specificity  0.618669 0.999993   0
## 8             max absolute_mcc  0.488356 0.054251 231
## 9   max min_per_class_accuracy  0.501064 0.526029 193
## 10 max mean_per_class_accuracy  0.495346 0.526843 210
##
## Gains/Lift Table: Extract with `h2o.gainsLift(<model>, <data>)` or `h2o.gainsLift(<model>, valid=<T/
```

Again, the base model gives us an AUC of just .53

```
h2o.varimp_plot(Char_v_skill_comp)
```


Standardized Coef. Magnitudes
Name.Medivh
Name.Abathur
Name.Stitches
Name.Artanis
Name.Sonya
Name.Samuro
Name.Ragnaros
Name.Alarak
Name.Falstad
Name.Valeera
Name.Li−Ming
Name.Zul'jin
Name.Tassadar
Name.Anub'arak
Name.Uther
Name.Muradin
Name.Zeratul
Name.Zarya
Name.Thrall
Name.Zagara
Name.Nazeebo
Name.Rehgar
Name.Diablo
Name.Jaina
Name.Malfurion
Name.Gazlowe
Name.Cho
Name.Auriel
Name.Raynor
Name.Greymane
Name.Leoric
Name.Dehaka
Name.Sgt. Hammer
Name.Li Li
Hero Level
Name.The Butcher
Name.Illidan
Name.Kael'thas
Name.Chromie
Name.Rexxar
Name.Gall
Name.Valla
Name.Nova
Name.Arthas
Name.Xul
Name.The Lost Vikings
Name.Kharazim
Name.Kerrigan
Name.Chen
Name.Lunara
Name.Brightwing
Name.E.T.C.
MMR standardized
Name.Azmodan
Name.Sylvanas
Name.Murky
Name.Gul'dan
Name.Johanna
Name.Lt. Morales
Name.Tyrael
Name.Varian
Name.Tyrande
Name.Tychus
Name.Tracer
Positive
Negative
0.0
0.2
0.4
0.6
0.8
1.0

```
comp_prob <- exp(h2o.coef(Char_v_skill))/(exp(h2o.coef(Char_v_skill))+1)-.5
sort(comp_prob,decreasing = TRUE)
```

```
##      MMR standardized         Name.Ragnaros           Name.Samuro
##          0.0816466550          0.0620206187          0.0481412434
##          Name.Artanis      Name.The Butcher          Name.Nazeebo
##          0.0455885482          0.0397965805          0.0354277071
##        Name.Anub'arak          Name.Zul'jin      Name.Sgt. Hammer
##          0.0339047448          0.0314771500          0.0299805882
##            Name.Murky          Name.Azmodan           Name.Thrall
##          0.0296202658          0.0275766867          0.0227051748
##        Name.Malfurion      Name.Lt. Morales          Name.Gazlowe
##          0.0219172115          0.0214761507          0.0213267136
##           Name.Diablo           Name.Rehgar            Name.Jaina
##          0.0208726875          0.0198526657          0.0161944013
##       Name.Brightwing           Name.Rexxar           Name.Dehaka
##          0.0144095007          0.0140491460          0.0128142437
##         Name.Kerrigan              Name.Xul             Name.Nova
##          0.0114186417          0.0078684256          0.0077963208
##           Name.Varian         Name.Kharazim           Name.Raynor
##          0.0077689898          0.0056561825          0.0047009412
##            Hero Level          Name.Johanna Name.The Lost Vikings
##          0.0021890927          0.0000000000          0.0000000000
##          Name.Valeera            Name.Zarya           Name.Tyrael
##          0.0000000000          0.0000000000         -0.0002505943
##          Name.Tyrande          Name.Gul'dan        Name.Kael'thas
##         -0.0012133570         -0.0024756501         -0.0043665187
##           Name.Tychus            Name.Li Li         Name.Sylvanas
##         -0.0051605891         -0.0077518195         -0.0081789429
##           Name.Lunara           Name.Leoric           Name.Arthas
##         -0.0103017188         -0.0107631069         -0.0113330081
##           Name.Tracer          Name.Illidan            Name.E.T.C.
##         -0.0114185197         -0.0114532257         -0.0114584412
##           Name.Zagara          Name.Muradin             Name.Uther
##         -0.0130070625         -0.0185327431         -0.0203730500
##            Name.Valla           Name.Auriel         Name.Greymane
##         -0.0230851620         -0.0236919868         -0.0253627496
##          Name.Chromie          Name.Li-Ming              Name.Chen
##         -0.0272422315         -0.0289850981         -0.0298549795
##         Name.Tassadar          Name.Zeratul          Name.Falstad
##         -0.0309984551         -0.0314068006         -0.0326425576
##          Name.Abathur             Name.Gall              Name.Cho
##         -0.0377516701         -0.0433154483         -0.0460029208
##            Name.Sonya           Name.Alarak         Name.Stitches
##         -0.0502203041         -0.0514673933         -0.0620879687
##             Intercept           Name.Medivh
##         -0.0627499126         -0.0998713388
```

The Variable importance and probabilities are about the same as for our model using all games played, with some differences in character ordering.

```
comp_grid <- h2o.grid(
    algorithm = "glm",
    grid_id = "comp_grid",
    hyper_params= hyper_parameters,
```

```
    training_frame = train_char,
    validation_frame = test_char,
    x=c("Name","Hero Level", "MMR standardized"),
    y="Is Winner",
    lambda_search = TRUE,
    family = "binomial"
)
```

```
glm.sorted.grid <- h2o.getGrid("comp_grid","auc")
best_model <- h2o.getModel(glm.sorted.grid@model_ids[[1]])
best_model
```

```
## Model Details:
## ==============
##
## H2OBinomialModel: glm
## Model ID:  comp_grid_model_2
## GLM Model: summary
##     family  link                                regularization
## 1 binomial logit Elastic Net (alpha = 0.4, lambda = 0.02201 )
##                                                                                                   lambda_search
## 1 nlambda = 100, lambda.max = 0.04221, lambda.min = 0.02201, lambda.1se = -1.0
##   number_of_predictors_total number_of_active_predictors
## 1                         64                           1
##   number_of_iterations training_frame
## 1                    8     train_char
##
## Coefficients: glm coefficients
##            names coefficients standardized_coefficients
## 1      Intercept    -0.086316                   0.000014
## 2   Name.Abathur     0.000000                   0.000000
## 3    Name.Alarak     0.000000                   0.000000
## 4 Name.Anub'arak     0.000000                   0.000000
## 5   Name.Artanis     0.000000                   0.000000
##
## ---
##               names coefficients standardized_coefficients
## 60      Name.Zagara     0.000000                   0.000000
## 61       Name.Zarya     0.000000                   0.000000
## 62     Name.Zeratul     0.000000                   0.000000
## 63     Name.Zul'jin     0.000000                   0.000000
## 64       Hero Level     0.008922                   0.030709
## 65 MMR standardized     0.000000                   0.000000
##
## H2OBinomialMetrics: glm
## ** Reported on training data. **
##
## MSE:  0.2497997
## RMSE:  0.4997997
## LogLoss:  0.6927466
## Mean Per-Class Error:  0.5
## AUC:  0.5205985
## Gini:  0.04119708
## R^2:  0.0008010076
```

```
## Null Deviance:  8421823
## Residual Deviance:  8416956
## AIC:  8416960
##
## Confusion Matrix for F1-optimal threshold:
##        0       1    Error              Rate
## 0      0 3037514 1.000000  =3037514/3037514
## 1      0 3037547 0.000000        =0/3037547
## Totals 0 6075061 0.499997  =3037514/6075061
##
## Maximum Metrics: Maximum metrics at their respective thresholds
##                          metric threshold    value idx
## 1                        max f1  0.480661 0.666669  19
## 2                        max f2  0.480661 0.833335  19
## 3                  max f0point5  0.482889 0.555563  18
## 4                  max accuracy  0.498496 0.514500  11
## 5                 max precision  0.523016 0.533240   0
## 6                    max recall  0.480661 1.000000  19
## 7               max specificity  0.523016 0.966129   0
## 8              max absolute_mcc  0.498496 0.029603  11
## 9   max min_per_class_accuracy  0.500727 0.430056  10
## 10 max mean_per_class_accuracy  0.498496 0.514499  11
##
## Gains/Lift Table: Extract with `h2o.gainsLift(<model>, <data>)` or `h2o.gainsLift(<model>, valid=<T/
## H2OBinomialMetrics: glm
## ** Reported on validation data. **
##
## MSE:  0.2497982
## RMSE:  0.4997981
## LogLoss:  0.6927434
## Mean Per-Class Error:  0.5
## AUC:  0.5208652
## Gini:  0.04173037
## R^2:  0.0008073219
## Null Deviance:  1486065
## Residual Deviance:  1485199
## AIC:  1485203
##
## Confusion Matrix for F1-optimal threshold:
##        0       1    Error             Rate
## 0      0  536001 1.000000   =536001/536001
## 1      0  535968 0.000000        =0/535968
## Totals 0 1071969 0.500015  =536001/1071969
##
## Maximum Metrics: Maximum metrics at their respective thresholds
##                          metric threshold    value idx
## 1                        max f1  0.480661 0.666653  19
## 2                        max f2  0.480661 0.833325  19
## 3                  max f0point5  0.485117 0.555561  17
## 4                  max accuracy  0.498496 0.515060  11
## 5                 max precision  0.523016 0.534807   0
## 6                    max recall  0.480661 1.000000  19
## 7               max specificity  0.523016 0.966127   0
## 8              max absolute_mcc  0.498496 0.030752  11
```

```
## 9   max min_per_class_accuracy  0.500727 0.430557  10
## 10 max mean_per_class_accuracy  0.498496 0.515063  11
##
## Gains/Lift Table: Extract with `h2o.gainsLift(<model>, <data>)` or `h2o.gainsLift(<model>, valid=<T/
```

```
h2o.varimp_plot(best_model)
```


Standardized Coef. Magnitudes
Hero Level
MMR standardized
Name.Zul'jin
Name.Zeratul
Name.Zarya
Name.Zagara
Name.Xul
Name.Varian
Name.Valla
Name.Valeera
Name.Uther
Name.Tyrande
Name.Tyrael
Name.Tychus
Name.Tracer
Name.Thrall
Name.The Lost Vikings
Name.The Butcher
Name.Tassadar
Name.Sylvanas
Name.Stitches
Name.Sonya
Name.Sgt. Hammer
Name.Samuro
Name.Rexxar
Name.Rehgar
Name.Raynor
Name.Ragnaros
Name.Nova
Name.Nazeebo
Name.Murky
Name.Muradin
Name.Medivh
Name.Malfurion
Name.Lunara
Name.Lt. Morales
Name.Li−Ming
Name.Li Li
Name.Leoric
Name.Kharazim
Name.Kerrigan
Name.Kael'thas
Name.Johanna
Name.Jaina
Name.Illidan
Name.Gul'dan
Name.Greymane
Name.Gazlowe
Name.Gall
Name.Falstad
Name.E.T.C.
Name.Diablo
Name.Dehaka
Name.Chromie
Name.Cho
Name.Chen
Name.Brightwing
Name.Azmodan
Name.Auriel
Name.Arthas
Name.Artanis
Name.Anub'arak
Name.Alarak
Name.Abathur
Positive
0.0
0.2
0.4
0.6
0.8
1.0

```
comp_grid_prob <- exp(h2o.coef(best_model))/(exp(h2o.coef(best_model))+1)-.5
sort(comp_grid_prob, decreasing = TRUE)
```

```
##            Hero Level          Name.Abathur           Name.Alarak
##           0.002230567           0.000000000           0.000000000
##        Name.Anub'arak          Name.Artanis           Name.Arthas
##           0.000000000           0.000000000           0.000000000
##           Name.Auriel          Name.Azmodan       Name.Brightwing
##           0.000000000           0.000000000           0.000000000
##             Name.Chen             Name.Cho          Name.Chromie
##           0.000000000           0.000000000           0.000000000
##           Name.Dehaka          Name.Diablo           Name.E.T.C.
##           0.000000000           0.000000000           0.000000000
##          Name.Falstad            Name.Gall          Name.Gazlowe
##           0.000000000           0.000000000           0.000000000
##         Name.Greymane         Name.Gul'dan          Name.Illidan
##           0.000000000           0.000000000           0.000000000
##            Name.Jaina         Name.Johanna        Name.Kael'thas
##           0.000000000           0.000000000           0.000000000
##         Name.Kerrigan        Name.Kharazim           Name.Leoric
##           0.000000000           0.000000000           0.000000000
##           Name.Li Li          Name.Li-Ming     Name.Lt. Morales
##           0.000000000           0.000000000           0.000000000
##           Name.Lunara       Name.Malfurion           Name.Medivh
##           0.000000000           0.000000000           0.000000000
##          Name.Muradin           Name.Murky          Name.Nazeebo
##           0.000000000           0.000000000           0.000000000
##             Name.Nova         Name.Ragnaros           Name.Raynor
##           0.000000000           0.000000000           0.000000000
##           Name.Rehgar           Name.Rexxar           Name.Samuro
##           0.000000000           0.000000000           0.000000000
##      Name.Sgt. Hammer            Name.Sonya         Name.Stitches
##           0.000000000           0.000000000           0.000000000
##         Name.Sylvanas         Name.Tassadar     Name.The Butcher
##           0.000000000           0.000000000           0.000000000
## Name.The Lost Vikings           Name.Thrall           Name.Tracer
##           0.000000000           0.000000000           0.000000000
##           Name.Tychus           Name.Tyrael          Name.Tyrande
##           0.000000000           0.000000000           0.000000000
##            Name.Uther          Name.Valeera           Name.Valla
##           0.000000000           0.000000000           0.000000000
##           Name.Varian             Name.Xul           Name.Zagara
##           0.000000000           0.000000000           0.000000000
##            Name.Zarya          Name.Zeratul          Name.Zul'jin
##           0.000000000           0.000000000           0.000000000
##      MMR standardized             Intercept
##           0.000000000          -0.021565623
```

The model with the penaltization factors for competitive games also performed about the same as the model with all the data. Interestingly when only using data for competitive games, the only data point that remains is Hero Level.

Conclusion

Given the extremely low predictive power of all the models created we should reject the idea that a prediction can be made for who will win a game based of individual characters played. This is not to say that there is not some combinations of characters that can be used to predict which players will win. This would require further testing, likely using association data techniques.

Furthermore given that the effect of each character on a player's win probability droped to 0 when penalized for significants, we can conclude that a players skill, not chosen characters, is what will carry them to victory.