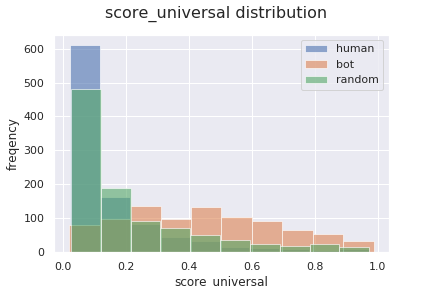
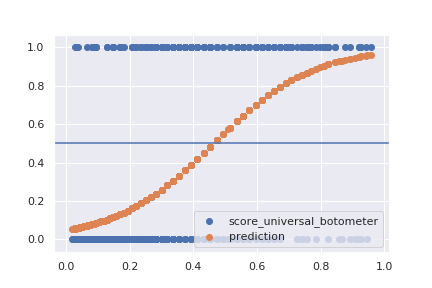
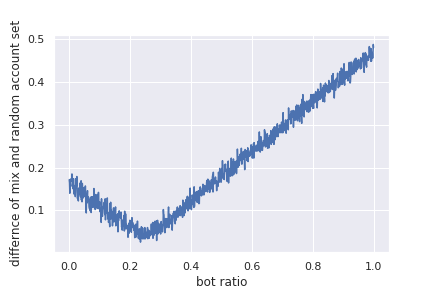
Models were built based on dataset of 1000 random twitter accounts (‘random set’). Those accounts are classified as ‘bot’ and ‘non-bot’ based on the “score\_universal” given by Botometer.

Since the division of ‘bot’ is essentially for this project. For better estimating, we access 936 self-identified bot accounts (‘bot sets’) from botwiki.org, and 1000 verified accounts (‘human set’). Kolmogorov-Smirnov statistic on 2 samples showed that, the distribution of “score\_universal” for ‘random set’ are significantly different with either ‘human set’ (p value = 4.2e-11) or ‘bot set’ (4.2e-92). This finding suggested that the ‘random set’ include both human and bot accounts.



Further analysis estimate that the ratio of ‘bot’ in the ‘random set’ is 23.4%. Using this ratio, we rebuilt a new set of account mixed with known ‘bot’ and ‘human’ accounts. Using Logistic regression modeling on this ‘mixed set’ indicated that a threshold at 0.47 give the best performance to divide bot accounts from non-bot accounts.



Consistently, according to the [authors](https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15587/14817) of Botometer, a threshold at or near 0.50 is appropriate for some automated accounts, but works more effectively on older, less-sophisticated bot programs. For more modern, sophisticated automated accounts, a threshold of 0.43 has been [shown](https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15587/14817) to maximize the accuracy of the system. This statement supported our find on threshold. Considering the motivation of this project, we use 0.43 in further study for including more modern bot accounts.

In our modeling, we using 1000 random twitter accounts, used a score of 0.43 or higher to declare an account is likely bot (130 accounts). Tweet metrics we will use as predictors for our modeling including: 'favourites\_count', 'followers\_count', 'friends\_count', 'listed\_count', 'statuses\_count', 'tweets\_per\_hour’, 'mean\_links\_per\_tweet', 'mean\_words\_per\_tweet', 'mean\_hashtags\_per\_tweet', 'mean\_user\_mentions\_per\_tweet', 'mean\_favourites\_per\_tweet', 'mean\_media\_per\_tweet', 'mean\_user\_symbols\_per\_tweet', 'mean\_retweets\_per\_tweet', 'mean\_truncations\_per\_tweet', 'mean\_links\_to\_twitter', 'mean\_links\_to\_top\_social\_media', 'mean\_links\_to\_top\_digital\_media', 'mean\_links\_to\_top\_news', 'mean\_links\_to\_top\_products\_services', 'mean\_links\_to\_top\_celebrities', 'mean\_links\_to\_top\_organizations', 'mean\_links\_to\_top\_sports', 'mean\_links\_to\_top\_adult', 'retweet\_ratio', 'mean\_ref\_to\_person', 'mean\_ref\_to\_norp', 'mean\_ref\_to\_org', 'mean\_ref\_to\_gpe', 'mean\_ref\_to\_product', 'mean\_ref\_to\_law', 'mean\_ref\_to\_money', 'default\_profile', 'default\_profile\_image', 'geo\_enabled', 'has\_extended\_profile', 'is\_translation\_enabled', 'profile\_background\_tile', 'profile\_use\_background\_image', 'verified'

All accounts are randomly split in to train and test data set (test ratio: 0.2). All following model are trained with train data set with 5-fold cross-validation, and then tested on test data set. Since we have highly imbalanced account number of bot and nun-bot (~ 1: 6.5), we increase the class weight to achieve the comparable accuracy of prediction on bot and non-bot.

With logistic regression, after tuning the weight of bot, we achieve a model with performance as:

Training Set total accuracy: 68.375%

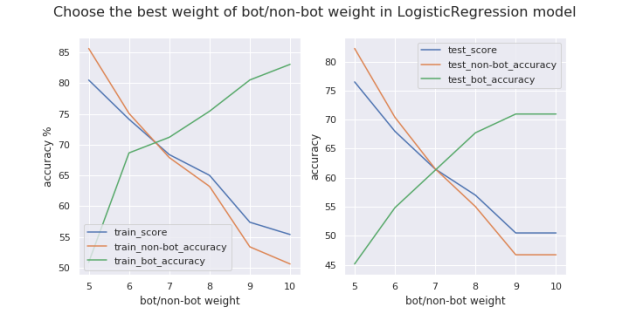
Training Set non-bot accuracy: 67.88856304985337%

Training Set bot accuracy: 71.1864406779661%

Test Set accuracy: 66.0%

Test Set non-bot accuracy: 65.89595375722543%

Test Set bot accuracy: 66.66666666666666%



With random forest, after tuning the weight of bot and tree depth, we achieve a model with performance as:

Training Set total accuracy: 91.5%

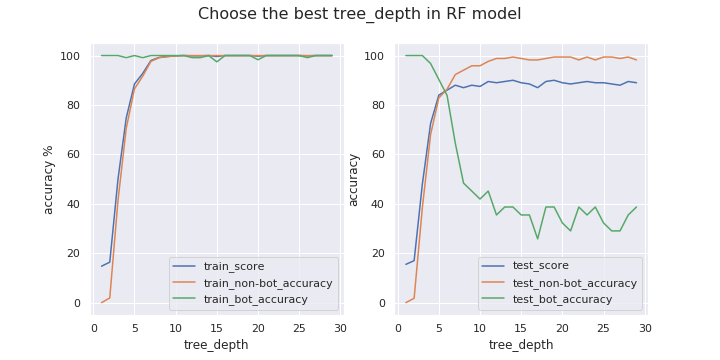
Training Set non-bot accuracy: 90.2439024390244%

Training Set bot accuracy: 100.0%

Test Set accuracy: 85.0%

Test Set non-bot accuracy: 84.97109826589595%

Test Set bot accuracy: 85.18518518518519%



With bagging decision tree, increading bot weight did not improve the bot accuracy. After tuning the cut\_off and n\_tree, we achieve a model with performance as:

Training Set total accuracy: 97.5%

Training Set non-bot accuracy: 97.21407624633432%

Training Set bot accuracy: 99.15254237288136%

Test Set accuracy: 85.5%

Test Set non-bot accuracy: 85.79881656804734%

Test Set bot accuracy: 83.87096774193549%

