CS 412 Final Project:
Human or Robot?

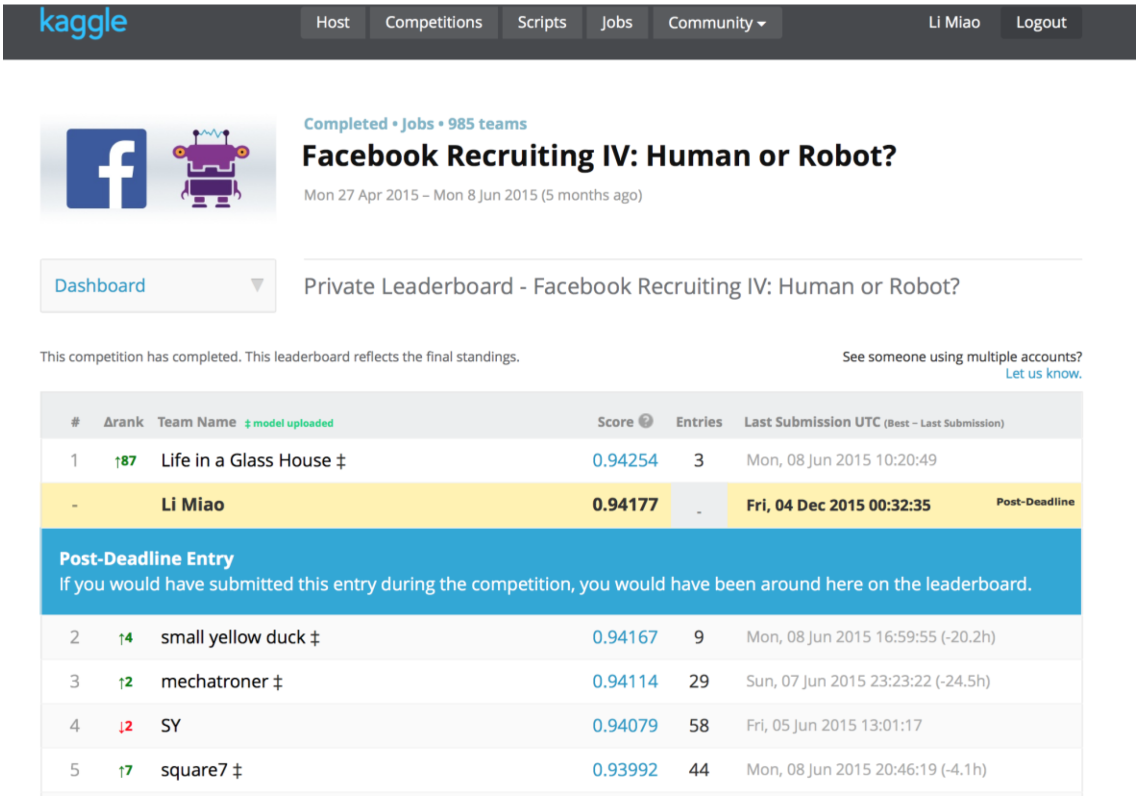
Group Member

Hongchuan Li(li193)
Hanqing Chen(hchen112)
Li Miao(limiao2)

Work Distribution

Hongchuan Li: randomForest Algorithm Implementation in Python
Hanqing Chen: Feature Engineering and Naive Bayes Algorithm Implementation in R
Li Miao: Data preprocessing, Bagging implementation, RandomForest revision in Python

Our Latest Score(Ranked 2nd, Score: 0.94177)



Completed • Jobs • 985 teams

Facebook Recruiting IV: Human or Robot?

Mon 27 Apr 2015 – Mon 8 Jun 2015 (5 months ago)

Private Leaderboard - Facebook Recruiting IV: Human or Robot?

This competition has completed. This leaderboard reflects the final standings.

See someone using multiple accounts?
Let us know.

#	Δrank	Team Name ‡ model uploaded	Score	Entries	Last Submission UTC (Best – Last Submission)
1	↑87	Life in a Glass House ‡	0.94254	3	Mon, 08 Jun 2015 10:20:49
-		**Li Miao**	**0.94177**	-	**Fri, 04 Dec 2015 00:32:35** Post-Deadline
2	↑4	small yellow duck ‡	0.94167	9	Mon, 08 Jun 2015 16:59:55 (-20.2h)
3	↑2	mechatroner ‡	0.94114	29	Sun, 07 Jun 2015 23:23:22 (-24.5h)
4	↓2	SY	0.94079	58	Fri, 05 Jun 2015 13:01:17
5	↑7	square7 ‡	0.93992	44	Mon, 08 Jun 2015 20:46:19 (-4.1h)

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

Introduction

This project requires coding for designing a classification algorithm to predict if an online bid is made by machine or a human. There are two datasets. One is the bidder dataset which includes information of all bidders, and the other is the bid dataset which includes 7.6 million bids. We started with Naive Bayes and then convert to Bagging(decision tree) and Randon Forest.

Feature Engineering

1.Missing Value Imputation

For missing vlaue, we firstly pinpointed them and then impute based on feature type. If the feature is “numerical”, we replace missing values with sample mean; otherwise, we use sample mode. Above are what we have tried in midterm check. In our final submission, we are intending to try one more advanced approach to deal with missing values issue, to use decision trees to predict missing values. Since missing values are inference-based, for each feature which contains missing values we built one decision tree by using the remaining features.

2. Factor and Character Features Handling

After checking all of the features, only one character feature has been found (Country). We remove that column because the library we used does not support. For the remaining features, all of them are numeric and binary, so they can be directly parsed to all of the algorithms we tried. We also removed features payment_account and address. They are useless predictors in our classification because they are unique ones to represent the bidder.

3. Features Selection

Based on our algorithm, we use one function in R called “importance( )” (http://www.inside-r.org/packages/cran/randomforest/docs/importance) to find out the contribution each feature has made in our randomForest, and we deleted those features which do not play significant role for our randomForest model.

Algorithm (randomForest,naive Bayes, Bagging)

Naive Bayes

Basic Theory

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong independence assumptions between the features.

Naive Bayes Classifier:

$$\hat{y} = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} \ p(C_k) \prod_{i=1}^{n} p(x_i | C_k).$$

Our purpose is to select the maximal posterior for each class.

Parameter Setting(Naive Bayes)

Having a look at our train and test data set, we found that the features are combined with both binary(0,1) and numeric type.

Based on the assumption, if features are categorical (in our case, they are binary), they can be regarded as Bernoulli Distribution;

$$p(\mathbf{x}|C_k) = \prod_{i=1}^{n} p_{ki}^{x_i}(1 - p_{ki})^{(1-x_i)}$$

if features are numerical, they can be regarded as Gaussian Distribution.

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

So we should separate the binary features from numerical features firstly and then compute their posterior independently. Considering the number of features are very large, the posterior could be pretty small since posterior is the product of each conditional probability. Our solution is to pick up the most important features (Top-10) through randomForest classifer by using funciton **importance( )** in R.

Packages: NA

Bagging

Each decision tree suffers from high variance. If we through a new dataset into the single decision tree model, the prediction accuracy could be quite different. To lower the variacne, we firstly decide to choose bagging procedure to gain higher prediction power.

In bagging, we bootstrap by taking repeated samples from the training date set. Then we have **N** different bootstrapped training set. For each new training set, we train a single decision tree model with all predictors. Therefore, we have **N** trees.

To solve our classification problem, we take a majority vote, which means the predicted class of a new data point is the most commonly occuring class in all **N** predictions.

In this project, we implemented Bagging with decision trees in Python. Totally 500 trees are generated based on bootstrapped training data set, and entropy is chosen as our criterion for each tree in bagging. This bagging algorithm gives us the Kaggle score 0.92641.

Packages used: pandas, numpy, random, sklearn.preprocessing, sklearn.DecisionTreeClassifier.

Random Forest

Random forest provides an improvement over bagging by decorrelating **N** trees. In random forest, we build a number of decision trees on bootstrapped training samples, each time a random sample of **m** predictors is chosen as a split candidates from the full set of **p** predicors. We decorrelates the trees and make the majority vote of the resulting trees less, hence less variable and more reliable than single decision tree. There are two basic rules as following:

Each classifier in the ensemble is a decision tree classifier and generated using a random selection of attributes at each node to determine the split

During classification, we take the majority vote from all the decision trees.

There are two main parameters require to be considered carefully. One is the number of features considered at each split, and another is the number of trees grown in our model. For the first one, we randomly selected $m = \sqrt{\square}$ predictors, where p is the full numder of features in the training dataset. In this project, we pick m = 17. As another one, we take 1000 as the default number of trees needed to grow in random forest according to the testing MSE.

In this project, we implemented our own Random Forest algorithm in Python, and similar to bagging, entropy is chosen as the criterion. Random Forest gives us the highest Kaggle score, 0.94177.

Packages: resample and DecisionTreeClassifier from sklearn

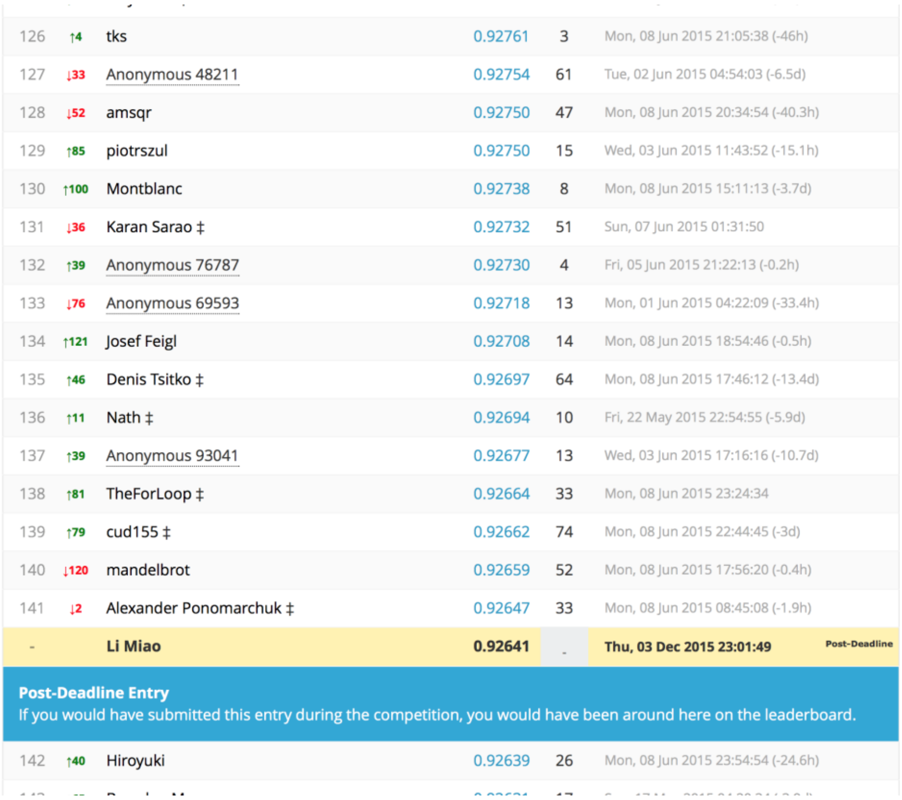
*Since we does not select a fix random seed, each time the algorithm could generate a slightly different result

Summary

In summary, Random Forest provides the best performance in our classification problem. Following is Bagging, and Naive Bayes peformed worst. So we may find that advanced algorithm like randomForest peforms pretty well for this classification problem but simple algorithm like Naive Bayes does not obtain a good result since many assumptions are not satisfied.

Evaluation

1. Bagging

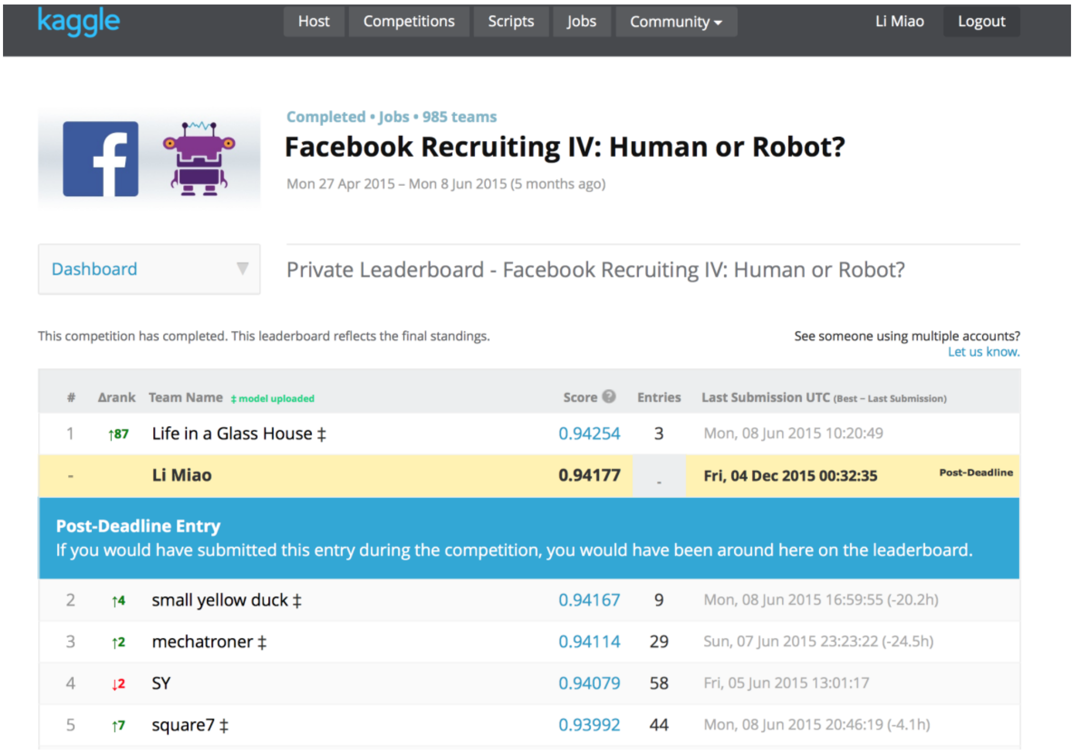


126	↑4	tks	0.92761	3	Mon, 08 Jun 2015 21:05:38 (-46h)
127	↓33	Anonymous 48211	0.92754	61	Tue, 02 Jun 2015 04:54:03 (-6.5d)
128	↓52	amsqr	0.92750	47	Mon, 08 Jun 2015 20:34:54 (-40.3h)
129	↑85	piotrszul	0.92750	15	Wed, 03 Jun 2015 11:43:52 (-15.1h)
130	↑100	Montblanc	0.92738	8	Mon, 08 Jun 2015 15:11:13 (-3.7d)
131	↓36	Karan Sarao ‡	0.92732	51	Sun, 07 Jun 2015 01:31:50
132	↑39	Anonymous 76787	0.92730	4	Fri, 05 Jun 2015 21:22:13 (-0.2h)
133	↓76	Anonymous 69593	0.92718	13	Mon, 01 Jun 2015 04:22:09 (-33.4h)
134	↑121	Josef Feigl	0.92708	14	Mon, 08 Jun 2015 18:54:46 (-0.5h)
135	↑46	Denis Tsitko ‡	0.92697	64	Mon, 08 Jun 2015 17:46:12 (-13.4d)
136	↑11	Nath ‡	0.92694	10	Fri, 22 May 2015 22:54:55 (-5.9d)
137	↑39	Anonymous 93041	0.92677	13	Wed, 03 Jun 2015 17:16:16 (-10.7d)
138	↑81	TheForLoop ‡	0.92664	33	Mon, 08 Jun 2015 23:24:34
139	↑79	cud155 ‡	0.92662	74	Mon, 08 Jun 2015 22:44:45 (-3d)
140	↓120	mandelbrot	0.92659	52	Mon, 08 Jun 2015 17:56:20 (-0.4h)
141	↓2	Alexander Ponomarchuk ‡	0.92647	33	Mon, 08 Jun 2015 08:45:08 (-1.9h)
-		**Li Miao**	**0.92641**	-	**Thu, 03 Dec 2015 23:01:49** Post-Deadline

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

142	↑40	Hiroyuki	0.92639	26	Mon, 08 Jun 2015 23:54:54 (-24.6h)

2. Random Forest



Completed • Jobs • 985 teams

Facebook Recruiting IV: Human or Robot?

Mon 27 Apr 2015 – Mon 8 Jun 2015 (5 months ago)

Private Leaderboard - Facebook Recruiting IV: Human or Robot?

This competition has completed. This leaderboard reflects the final standings.

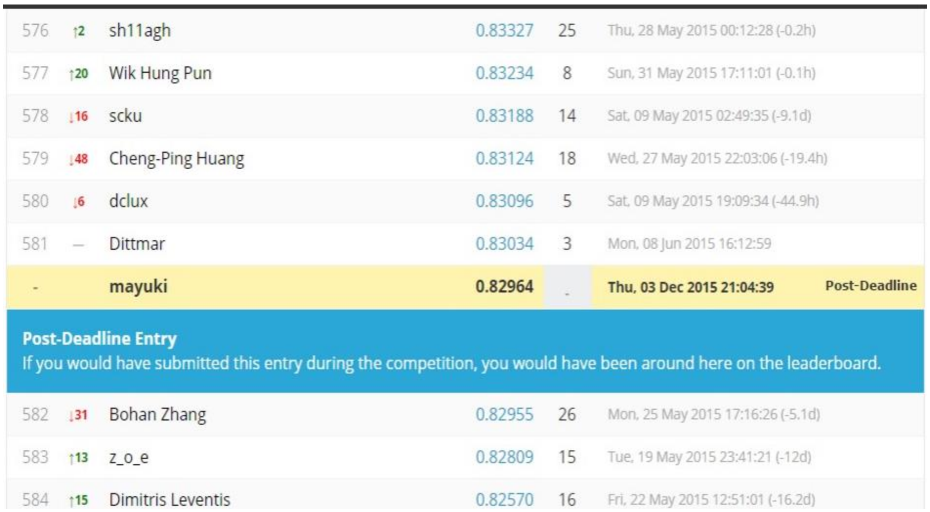
See someone using multiple accounts?
Let us know.

#	Δrank	Team Name ‡ model uploaded	Score	Entries	Last Submission UTC (Best – Last Submission)
1	↑87	Life in a Glass House ‡	0.94254	3	Mon, 08 Jun 2015 10:20:49
-		**Li Miao**	**0.94177**	-	**Fri, 04 Dec 2015 00:32:35** Post-Deadline

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

2	↑4	small yellow duck ‡	0.94167	9	Mon, 08 Jun 2015 16:59:55 (-20.2h)
3	↑2	mechatroner ‡	0.94114	29	Sun, 07 Jun 2015 23:23:22 (-24.5h)
4	↓2	SY	0.94079	58	Fri, 05 Jun 2015 13:01:17
5	↑7	square7 ‡	0.93992	44	Mon, 08 Jun 2015 20:46:19 (-4.1h)

3. Naive Bayes



576	↑2	sh11agh	0.83327	25	Thu, 28 May 2015 00:12:28 (-0.2h)	
577	↑20	Wik Hung Pun	0.83234	8	Sun, 31 May 2015 17:11:01 (-0.1h)	
578	↓16	scku	0.83188	14	Sat, 09 May 2015 02:49:35 (-9.1d)	
579	↓48	Cheng-Ping Huang	0.83124	18	Wed, 27 May 2015 22:03:06 (-19.4h)	
580	↓6	dclux	0.83096	5	Sat, 09 May 2015 19:09:34 (-44.9h)	
581	—	Dittmar	0.83034	3	Mon, 08 Jun 2015 16:12:59	
-		**mayuki**	**0.82964**	-	**Thu, 03 Dec 2015 21:04:39**	**Post-Deadline**

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

582	↓31	Bohan Zhang	0.82955	26	Mon, 25 May 2015 17:16:26 (-5.1d)
583	↑13	z_o_e	0.82809	15	Tue, 19 May 2015 23:41:21 (-12d)
584	↑15	Dimitris Leventis	0.82570	16	Fri, 22 May 2015 12:51:01 (-16.2d)