DSCI6003: Ultimatum Game

A machine learning approach to maximizing your earnings

Jonathan Hilgart

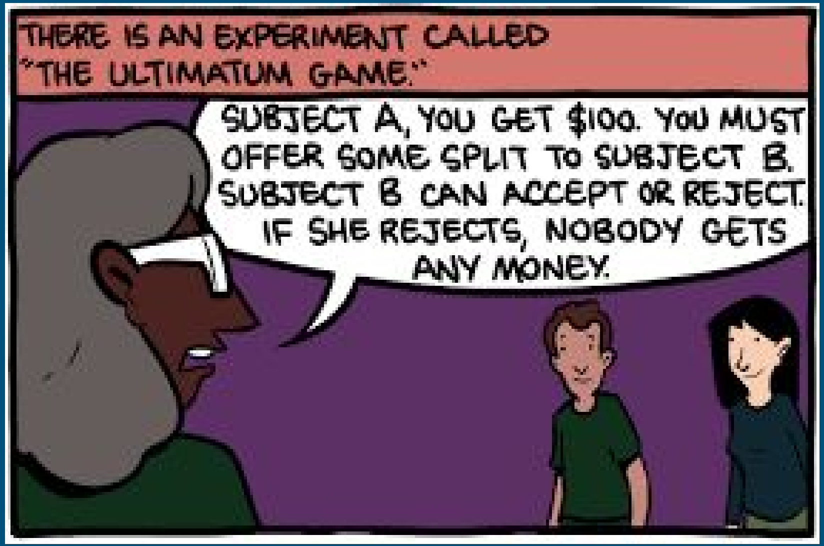
Overview

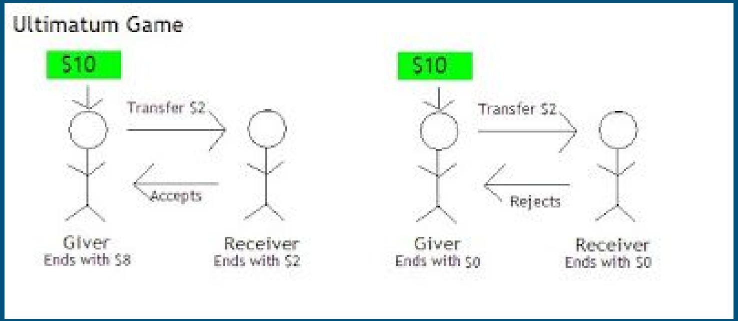
1) What is the ultimatum game?

2) What does the data look like?

3) How to maximize how much money we make?

Traditional Ultimatum Game





Empirical Results

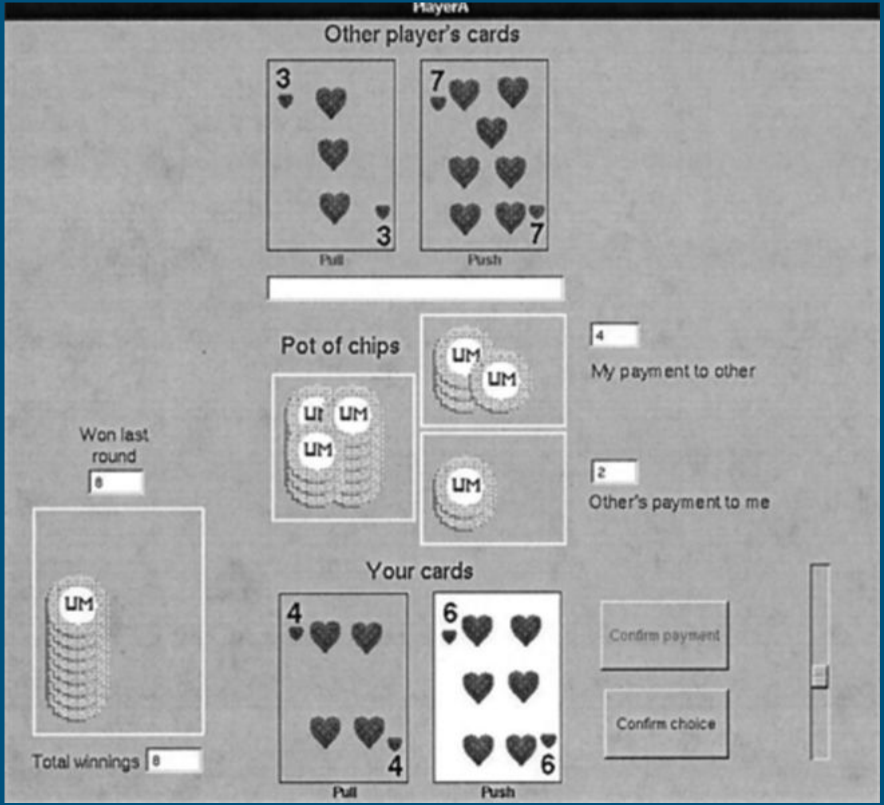
- Cooperation is very low (15-50%) even though nash equilibrium suggests responder should accept any offer.
- What is the accuracy for predicting our payoff playing this game?

Overview of game structure

1. Two rounds of experiments.
    - 1) First 15 rounds
        - Push : Give the other player money from the pot of money($6 or $7)
        - Pull : Take money from the pot ($3 or $4)
    - 2) Rounds past 15 allowed each player to make a side payment for the other to push
        - Typically offer between $1-$3 for other player to push
2. Three rounds of analysis
    - 1) Rounds without a side payment (rounds 1-15)
    - 2) Rounds with a side payment (rounds past 15)
    - 3) All rounds
3. Players played multiple rounds

Original Paper: http://www.pnas.org/content/96/19/10933.full

The experiment



The Data

- In total 1,920 rows. Each row was a full game experiment.
  - 720 without a side payment
  - 1200 with a side payment
- Features used (only use features available to you in the game).
  - Rd 1-15

	rd	myside	opptside	mypush	mypull	opptpush	opptpull	mychoice	mychoicecard	mypayoff	opptpayoff	totalpayoff
0	1	0	0	6	4	7	3	0	4	11	0	11
1	1	0	0	7	3	6	4	1	7	0	11	11

  - Past rd 15

	rd	myside	opptside	mypush	mypull	opptpush	opptpull	mychoice	mychoicecard	mypayoff	opptpayoff	totalpayoff
1915	40	4	3	6	4	7	3	0	4	7	4	11
1916	40	3	5	6	4	7	3	1	6	9	4	13

- Goal: Predict mypayoff

**Data was scaled for GLMnet due to binary choice for mychoice (0,1). All other features were in dollars.*

Algorithms Used

- Random Forest
- KNNregression
- Gradient Boosting
- Extreme Gradient Boosting
- GLM/Elastic Net

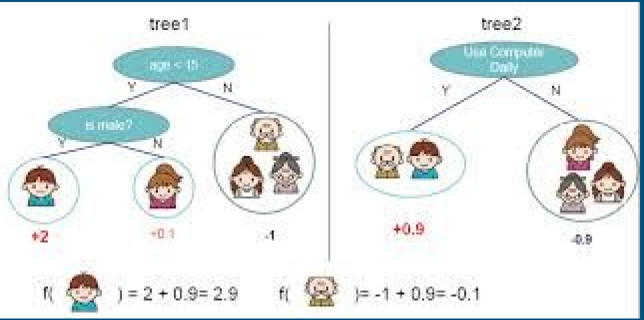
Ensembles

- Linear ensemble of all models
- GB ensemble of all models and all original features

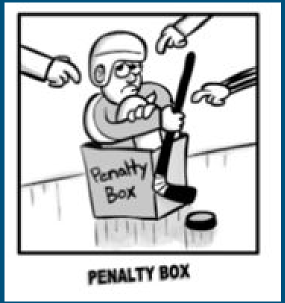
http://xgboost.readthedocs.io/en/latest/model.html

1) Random Forest

2) Boosting

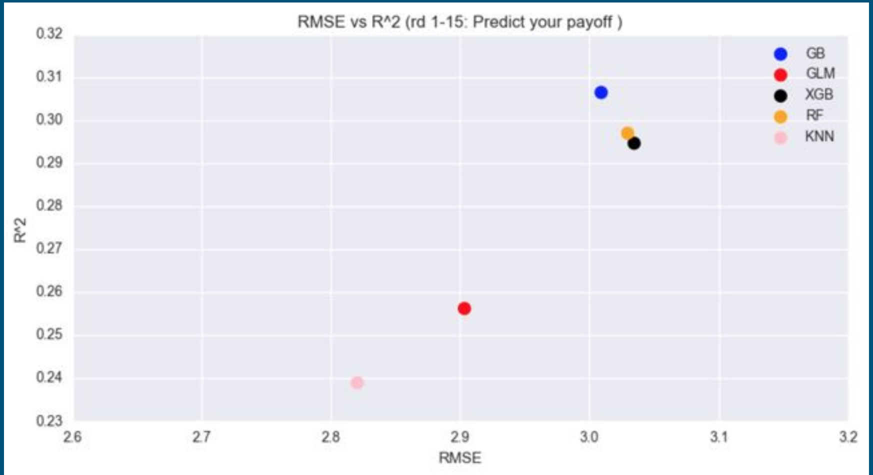


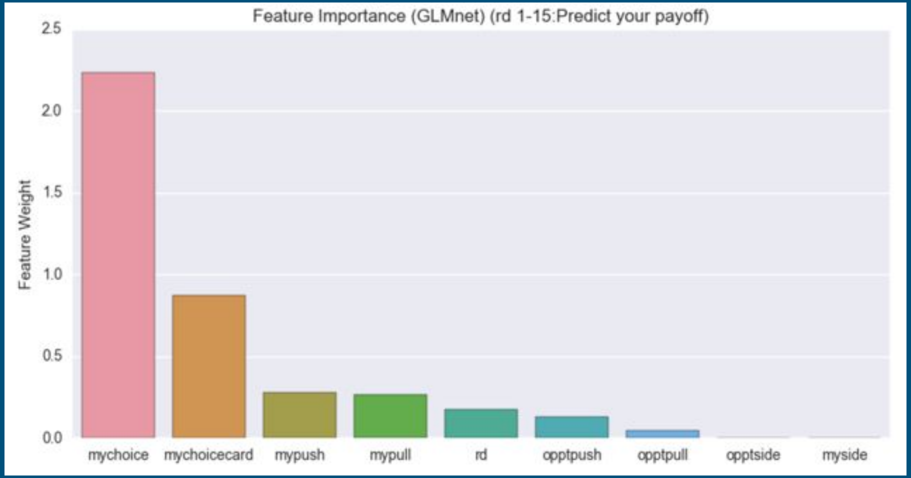
3) GLM/Elastic net



Rounds 1-15

- Results vary slightly based upon train test split

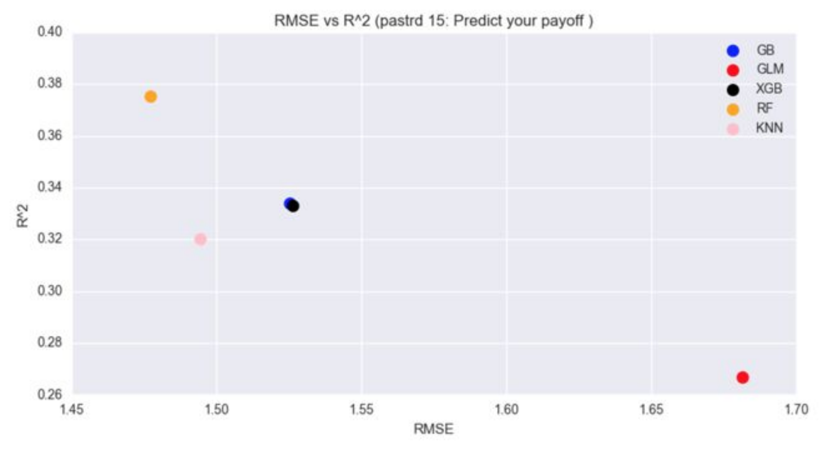


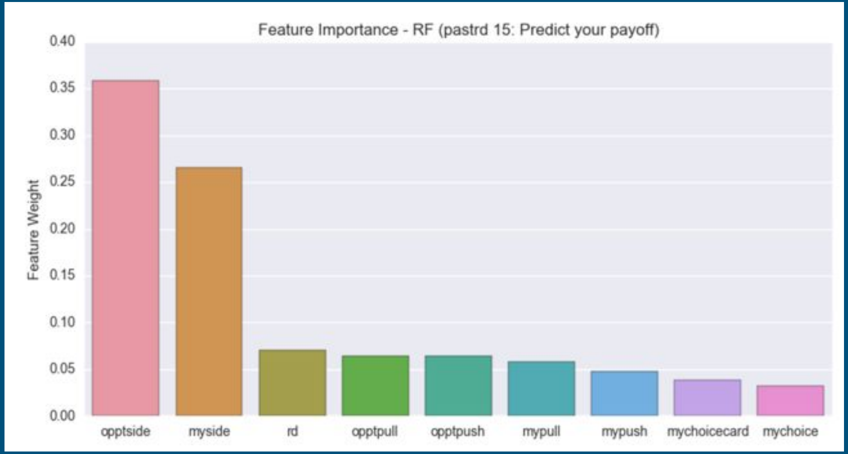


** Best parameters found from randomized gridsearch. Feature importance from glmnet.*

Rounds past 15

- Results vary slightly based upon train test split

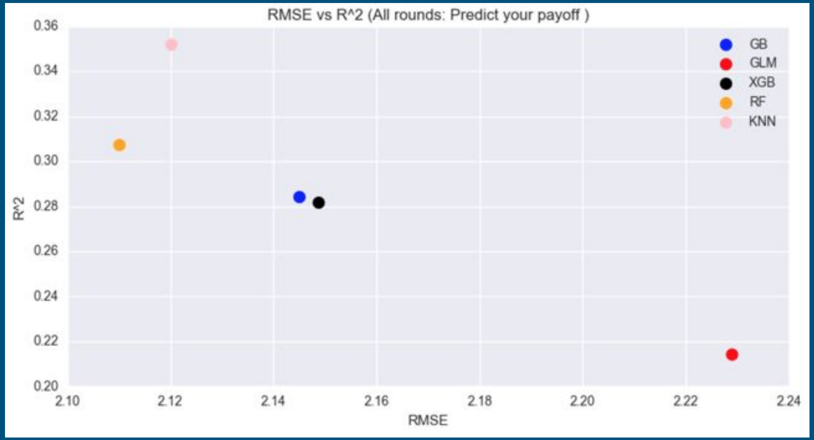


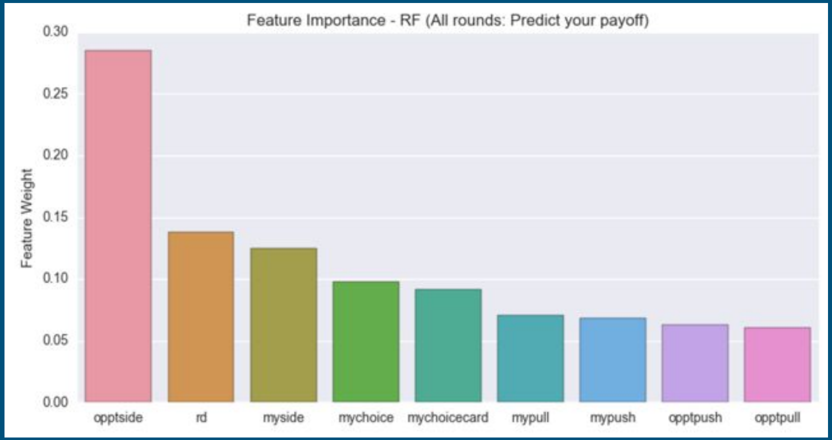


** Best parameters found from randomized gridsearch. Feature importance from RF.*

All rounds

- Results vary slightly based upon train test split





** Best parameters found from randomized gridsearch. Feature importance from RF.*

Ensemble

- Dataframe (linear):

	GLM_scaled_train	GB_train	RF_train	XGB_train	KNN_train	actual_training
0	5.342700	6.299544	6.238544	6.178087	5.53125	7
1	5.357743	6.536596	6.300855	6.752759	6.84375	7

- Dataframe (meta):

	rd	myside	opptside	mypush	mypull	opptpush	opptpull	mychoice	mychoicecard	GLM_scaled_test
count	384.000000	384.000000	384.000000	384.000000	384.000000	384.000000	384.000000	384.000000	384.000000	384.000000
mean	-0.092270	-0.141051	-0.078104	0.000000	0.000000	0.000000	0.000000	0.008464	0.027195	4.740063

- Weights per model (linear):
- Weights (meta):

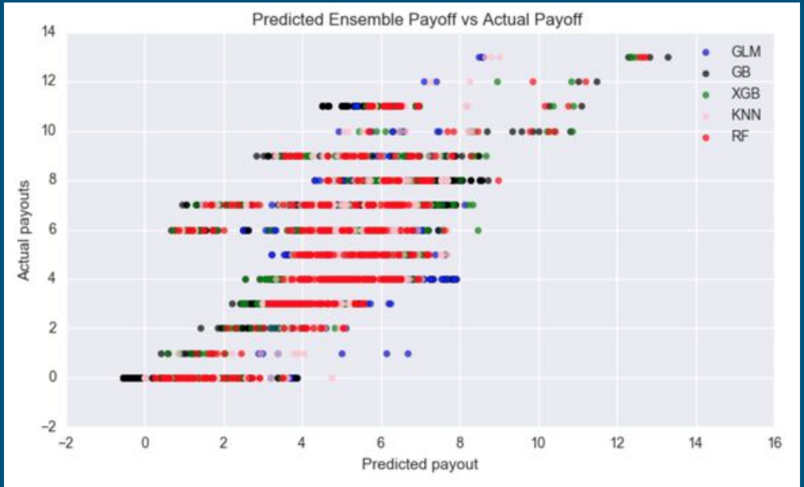
```
##Column order = GLM_scaled , GB, RF, XGB ,KNN
stacked_model.coef_
```

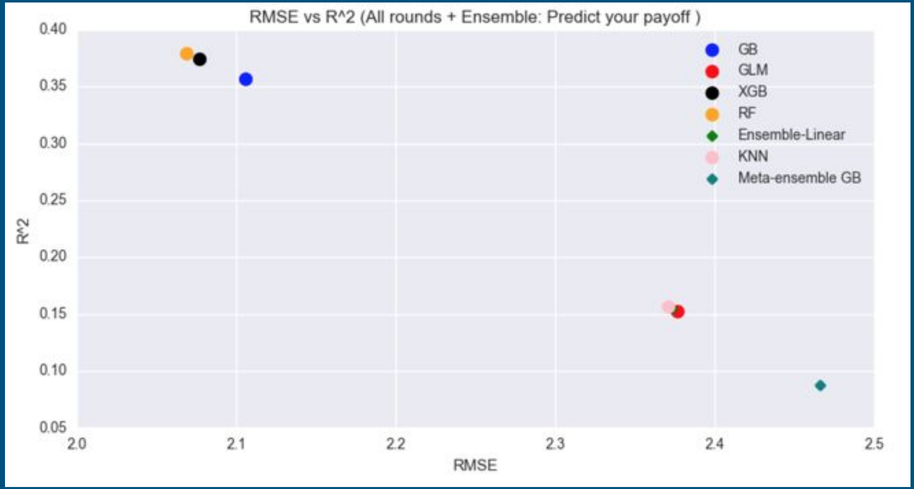
```
array([-0.12448028,  0.93010836,  0.28100216, -0.07034054, -0.05362199])
```

```
#Columns order - 'rd', 'myside', 'opptside', 'mypush', 'mypull', 'opptpush', 'opptpull',
       # 'mychoice', 'mychoicecard', 'GLM_scaled_train', 'GB_train', 'RF_train',
       # 'XGB_train', 'KNN_train'
stacked_model_gb.feature_importances_
```

```
array([ 0.38 ,  0.   ,  0.   ,  0.   ,  0.   ,  0.   ,  0.   ,  0.   ,
        0.004,  0.116,  0.204,  0.108,  0.064,  0.124])
```

Ensemble results






```
Linear Ensemble RMSE : 2.3735259857168507
Linear Ensemble R2: 15.46%
Meta-Ensemble GB RMSE : 2.4664525647799427
Meta-Ensemble GB R2: 8.71%
RF RMSE : 2.069208444284043
RF R2 : 37.89%
```

*Neither ensemble method could beat the final RF model.

Questions