kaggle.com

Completed • $20,000 • 478 teams

Melbourne University AES/MathWorks/NIH Seizure Prediction

Fri 2 Sep 2016 – Thu 1 Dec 2016 (13 days ago)

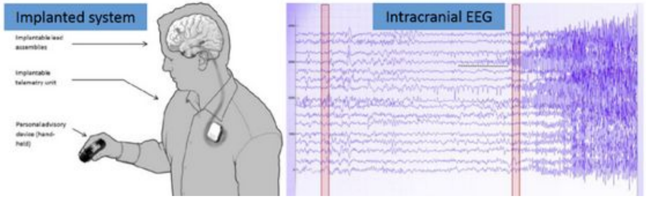
Private Leaderboard

1. Not-so-random-anymore
2. Areté Associates
3. GarethJones
4. QingnanTang
5. nullset
6. tralala boum boum pouêt pouêt
7. Medrr
8. michaln

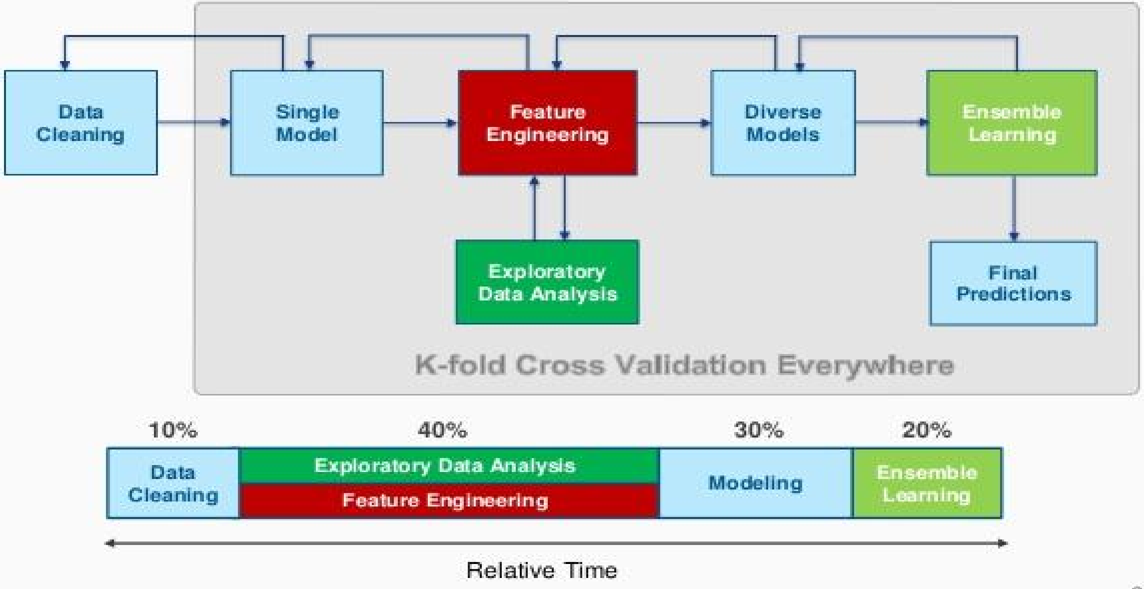
Predict seizures in long-term human intracranial EEG recordings

Epilepsy afflicts nearly 1% of the world's population, and is characterized by the occurrence of spontaneous seizures. For many patients, anticonvulsant medications can be given at sufficiently high doses to prevent seizures, but patients frequently suffer side effects. For 20-40% of patients with epilepsy, medications are not effective. Even after surgical removal of epilepsy, many patients continue to experience spontaneous seizures. Despite the fact that seizures occur infrequently, patients with epilepsy experience persistent anxiety due to the possibility of a seizure occurring.

Seizure forecasting systems have the potential to help patients with epilepsy lead more normal lives. In order for electrical brain activity (EEG) based seizure forecasting systems to work effectively, computational algorithms must reliably identify periods of increased probability of seizure occurrence. If these seizure-permissive brain states can be identified, devices designed to warn patients of impeding seizures would be possible. Patients could avoid potentially dangerous activities like driving or swimming, and medications could be administered only when needed to prevent impending seizures, reducing overall side effects.



Recommended Data Science Process (IMHO)



[goo.gl/2Sbh3f](goo.gl/2Sbh3f)

duration of competition: Fri 2 Sep 2016 – Thu 1 Dec 2016

we started : 11 Oct 2016

first submission : 25 Oct 2016

duration of competition: Fri 2 Sep 2016 – Thu 1 Dec 2016

we started : 11 Oct 2016

- data loading (~ 60 gb), cleaning
- feature extraction
- first model
- predictions

first submission : 25 Oct 2016

first submission: 25 Oct 2016

494	new	DeepakKarunakaran	0.53893	8	Sat, 22 Oct 2016 06:08:35 (-0.2h)
495	↓48	William Hau	0.53874	11	Wed, 12 Oct 2016 22:43:13 (-3.7d)
496	↓48	FeelTheLearn	0.53830	3	Mon, 10 Oct 2016 11:53:22
497	↓48	ManjunathMC	0.53824	3	Sat, 24 Sep 2016 19:59:46 (-0h)
498	↓48	zeon	0.53809	19	Sat, 10 Sep 2016 23:59:50 (-3.1d)
499	↓48	Team Jeff	0.53761	3	Sun, 09 Oct 2016 18:11:37 (-3.5h)
500	↓48	Jordan Gumm	0.53739	1	Mon, 19 Sep 2016 23:25:37
501	new	**nullset**	**0.53662**	**1**	**Tue, 25 Oct 2016 23:30:03**

Your Best Entry ↑
Congratulations on making your first submission!

Tweet this!

502	↓49	HarveyRichmond	0.53645	1	Mon, 10 Oct 2016 22:33:53
503	↓49	Mike G	0.53637	1	Wed, 21 Sep 2016 13:18:45
504	↓49	AlanDiego	0.53353	1	Sun, 25 Sep 2016 23:56:47
505	↓17	FutureAI	0.53342	4	Sun, 23 Oct 2016 06:29:49
506	↓50	BenGurion	0.53318	5	Tue, 13 Sep 2016 21:17:10
507	↓50	Leonardo Bonato	0.53300	13	Wed, 28 Sep 2016 16:59:44 (-26.6h)
508	↓50	Shecky & Stretchy	0.53213	5	Fri, 23 Sep 2016 21:32:58 (-46h)

HashtagWTT	0.56570	1	Fri, 14 Oct 2016 19:21:05
Sentdex	0.56569	9	Wed, 12 Oct 2016 23:50:09 (-2.2h)
pyramid222	0.56495	1	Wed, 07 Sep 2016 19:54:42
bob	0.56468	3	Sat, 03 Sep 2016 20:10:32
evil robots	0.56449	3	Wed, 14 Sep 2016 04:04:43 (-6.5d)
nullset	**0.56350**	**2**	**Wed, 26 Oct 2016 23:15:01**

Entry ↑
ved on your best score by 0.02688.

oved up 45 positions on the leaderboard. Tweet this!

Dustin Landers	0.56280	2	Sun, 11 Sep 2016 02:16:33 (-0.1h)
ISFArthur	0.56218	5	Mon, 24 Oct 2016 13:04:55 (-25h)
djbco	0.56195	4	Wed, 28 Sep 2016 01:25:28 (-10.7h)
Nicolae Chelea	0.55714	5	Mon, 10 Oct 2016 13:05:06 (-8.1d)
thatguy	0.55697	6	Sun, 18 Sep 2016 00:18:14 (-5.9d)
amdguru	0.55693	4	Tue, 27 Sep 2016 17:07:10
usama	0.55546	14	Tue, 04 Oct 2016 01:21:38 (-6d)
VT-CBIA	0.55528	5	Fri, 21 Oct 2016 00:22:54 (-0.1h)
NIAS Alpha Team	0.55525	13	Mon, 17 Oct 2016 11:18:03 (-4.2d)
Gal Eyal	0.55516	2	Wed, 14 Sep 2016 07:01:17
CarlosAsensioPizarro	0.55432	1	Sat, 17 Sep 2016 16:11:30
Stefano	0.55415	5	Wed, 26 Oct 2016 15:45:02 (-36.2d)
Anil	0.55310	9	Tue, 25 Oct 2016 18:15:45 (-0.6h)

duration of competition: Fri 2 Sep 2016 – Thu 1 Dec 2016

we started : 11 Oct 2016

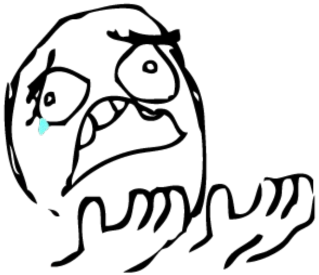
first submission : 25 Oct 2016

data leakage and new test set : 4 Nov 2016 duration of competition: Fri 2 Sep 2016 – **Thu 1 Dec 2016**

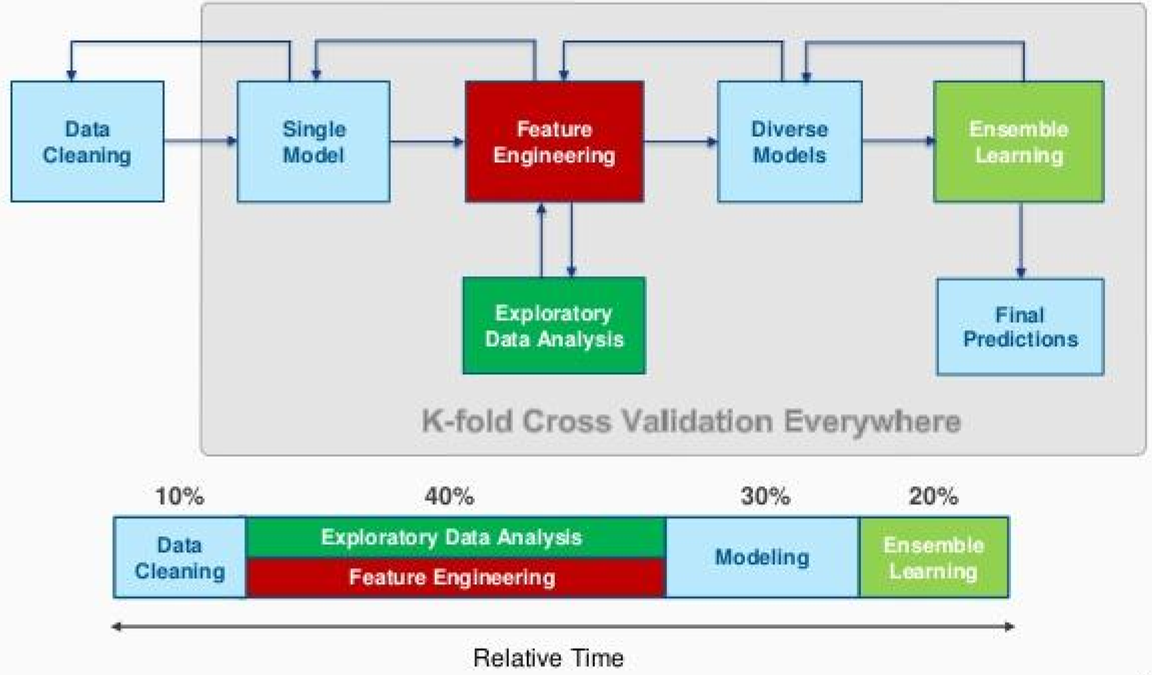
we started: 11 Oct 2016

first submission: 25 Oct 2016

data leakage and new test set: **4 Nov 2016**



Recommended Data Science Process (IMHO)



		All Zeros Benchmark	0.50000		
48	new	mindcool	0.50000	1	Fri, 04 Nov 2016 19:42:53
49	new	Andrey Larionov	0.50000	1	Sat, 05 Nov 2016 10:52:09
50	new	Kevin Diaz	0.50000	1	Sat, 05 Nov 2016 12:06:03
51	new	Kortex	0.50000	5	Sun, 06 Nov 2016 22:52:20 (-21.4h)
52	new	Vadim	0.50000	1	Sun, 06 Nov 2016 12:50:22
53	new	TZech	0.49887	1	Sun, 06 Nov 2016 21:00:17
54	new	nullset	0.46082	1	Mon, 07 Nov 2016 02:53:47

Download raw data

#	Δ1w	Team Name * in the money	Score	Entries
1	new	Chihiro Komaki *	0.79432	13
2	new	Joseph Chui *	0.77355	14
3	new	LabGOL *	0.76570	5
4	new	BR Unpredicted Predictions FR • Gilberto Titericz Junior • Alexandre Barachant	0.75620	20
5	new	**nullset** • irinaai • Oleg Panichev	**0.74431**	**7**

You

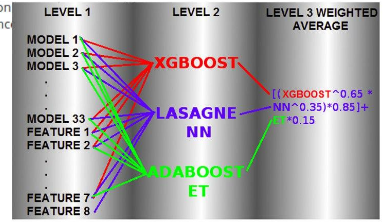
The guy she tells you not to worry about

```
from sklearn.neighbors import KNeighborsClassifier

neigh = KNeighborsClassifier(n_neighbors=3)
```

Models and features used for 2nd level training:
= Train and test sets

Model 1: RandomForest(R). Dataset: X
Model 2: Logistic Regression(scikit). Dataset: Log(X+1)
Model 3: Extra Trees Classifier(scikit). Dataset: Log(X+1) (but could be raw)
-Model 4: KNeighborsClassifier(scikit). Dataset: Scale(Log(X+1))
-Model 5: libfm. Dataset: Sparse(X). Each feature value is a unique level.
-Model 6: H2O NN. Bag of 10 runs. Dataset: sqrt(X + 3/8)
-Model 7: Multinomial Naive Bayes(scikit). Dataset: Log(X+1)
-Model 8: Lasagne NN(CPU). Bag of 2 NN runs. First with Dataset Scale(Log(X+1)) and s
)
-Model 9: Lasagne NN(CPU). Bag of 6 runs. Dataset: Scale(Log(X+1))
-Model 10: T-sne. Dimension reduction to 3 dimensions. Also stacked 2 kmeans feature
dimensions. Dataset: Log(X+1)
-Model 11: Sofia(R). Dataset: one against all with learner_type="logreg-pegasos" and lo
stochastic". Dataset: Scale(X)
-Model 12: Sofia(R). Trainned one against all with learner_type="logreg-pegasos" and lo
stochastic". Dataset: Scale(X, T-sne Dimension, some 3 level interactions between 13 m
based in randomForest importance)
-Model 13: Sofia(R). Trainned one against all with learner_type="logreg-pegasos" and lo
Dataset: Log(1+X, T-sne Dimension, some 3 level interactions between 13 most importa
randomForest importance)
-Model 14: Xgboost(R). Trainned one against all. Dataset: (X, feature sum(zeros) by row
-Model 15: Xgboost(R). Trainned Multiclass Soft-Prob. Dataset: (X, 7 Kmeans features w
clusters, rowSums(X==0), rowSums(Scale(X)>0.5), rowSums(Scale(X)< -0.5))
-Model 16: Xgboost(R). Trainned Multiclass Soft-Prob. Dataset: (X, T-sne features, Some
-Model 17: Xgboost(R): Trainned Multiclass Soft-Prob. Dataset: (X, T-sne features, Some
log(1+X))
-Model 18: Xgboost(R): Trainned Multiclass Soft-Prob. Dataset: (X, T-sne features, Some
)
-Model 19: Lasagne NN(GPU). 2-Layer. Bag of 120 NN runs with different number of ep
-Model 20: Lasagne NN(GPU). 3-Layer. Bag of 120 NN runs with different number of ep
-Model 21: XGboost. Trained on raw features. Extremely bagged (30 times averaged).
-Model 22: KNN on features X + int(X == 0)
-Model 23: KNN on features X + int(X == 0) + log(X + 1)
-Model 24: KNN on raw with 2 neighbours
-Model 25: KNN on raw with 4 neighbours
-Model 26: KNN on raw with 8 neighbours
-Model 27: KNN on raw with 16 neighbours
-Model 28: KNN on raw with 32 neighbours
-Model 29: KNN on raw with 64 neighbours
-Model 30: KNN on raw with 128 neighbours
-Model 31: KNN on raw with 256 neighbours
-Model 32: KNN on raw with 512 neighbours
-Model 33: KNN on
-Feature 1: Distanc



Software

All data analysis and models were built using Python. Libraries used: scikit-learn, pandas, xgboost.

Preprocessing

The signal from each file was divided on epochs 30 seconds length without any filtration. From each epoch features were extracted. We have tried also 15 and 60 seconds epoch length but the results were worse.

Feature extraction

We tried many features in different combinations during this competition, but not all of them were used in final models. **Feature sets** we've tried:

1. Deep's kernel for features extraction.
2. Tony Reina's kernel for features extraction.
3. Correlation between all channels (120 features).
4. Correlation between spectras of all channels (120 features).
5. Spectral features version 1: total energy (sum of all elements in range 0-30 Hz), energy in delta (0-3 Hz), theta (3-8 Hz), alpha (8-14 Hz) and beta (14-30 Hz) bands, energy in delta, theta, alpha and beta bands divided by total energy, ratios between energies of all bands.
6. Spectral features version 2: the same as Spectral features set 1 plus low and high gamma band were used in calculation of total energy, energy in bands and ratios between energies in bands. In addition, mean energy in bands was extracted.
7. Spectral features version 3: power spectral density was calculated for the whole epoch. Then it was divided on 1 Hz ranges and in each range energy was calculated (30 features).

Fitting and cross-validation

Dividing signals on epochs allowed to increase training dataset size, so total number of observations *No* was equal to

$$No = Nf * Ne,$$

where *Nf* - number of 10-minute signals, *Ne* - number of epochs per one 10-minute signal.

For cross-validation stratified K-folds with 6 folds was used. It was extremely important to use K-fold without shuffling the data, otherwise the leakage is very high and cross-validation performance estimations are much higher. The leakage during shuffling was present because two neighboring epochs with very similar parameters were often present both in train and test sets.

Each model predicted probability of epoch belongs to *preictal* class. The final probability for 10-minute signal was calculated as mean of all probabilities for epochs in this signal.

We tried both patient-specific and non-patient-specific approaches on the same model but performance was higher when patient-specific approach was used.

Models

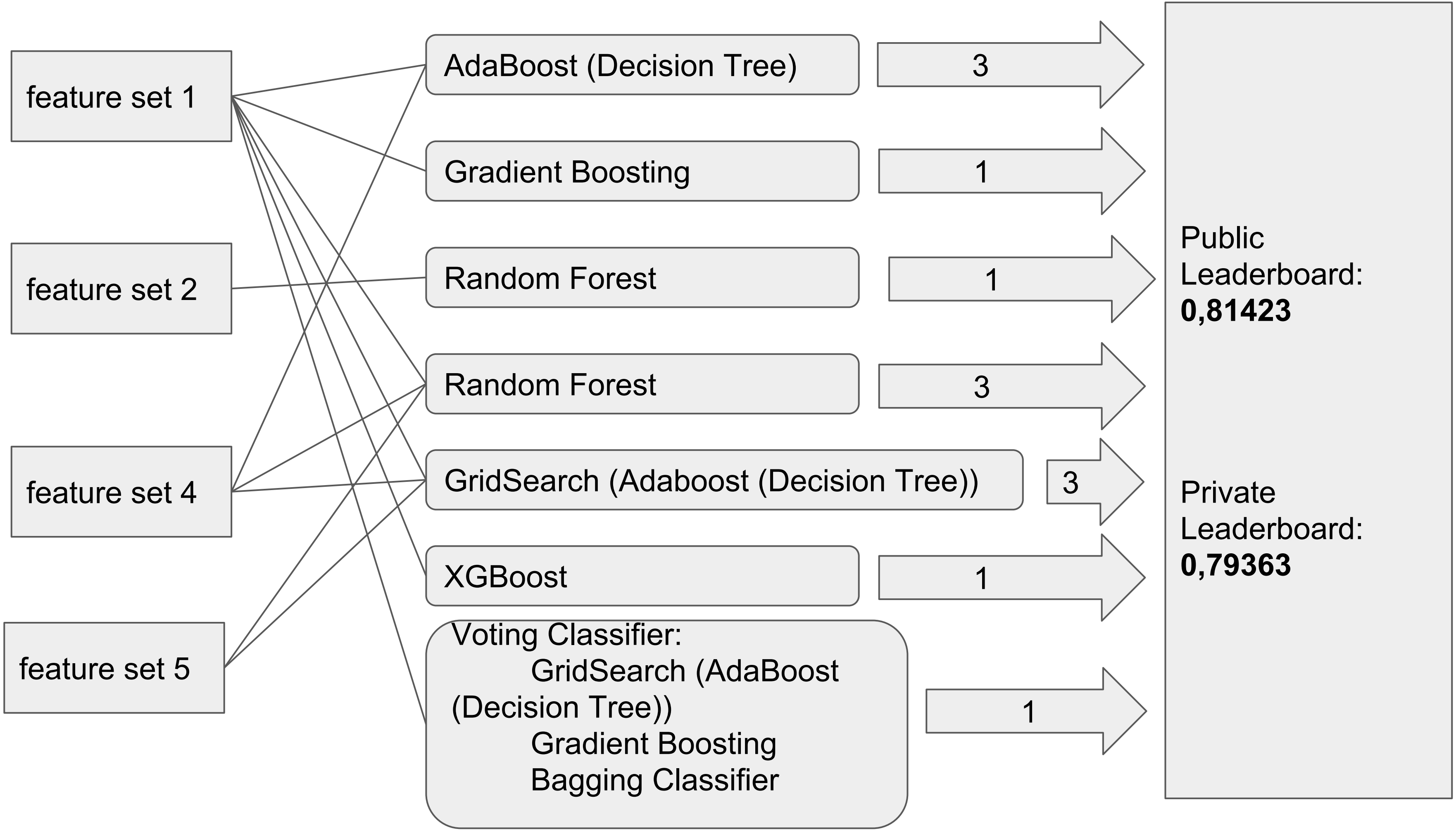
The final solution was an ensemble of best performing models (the first one is the best performing and the last one - is the worst):

1. AdaBoost with Decision Tree base estimator with combined feature sets 1, 4 and 5 .
2. Gradient Boosting Classifier with feature set 1.
3. Random Forest Classifier with feature set 2.
4. Random Forest Classifier with combined feature sets 1, 4 and 5.
5. GridSearch for “number of estimators” parameter for AdaBoost with Decision Tree base estimator with combined feature sets 1, 4 and 5.
6. Voting classifier with feature set 1. Voting was performed for 3 classifiers: GridSearch for “number of estimators” parameter for AdaBoost with Decision Tree base estimator; Gradient Boosting Classifier and Bagging Classifier.
7. XGBoost Classifier with feature set 1.

AdaBoost with Decision Tree base estimator with combined feature sets 1, 4 and 5 showed the highest performance among the models.

Final result *P* was calculated as follows:

P = 1/13 * (3*Model 1 + Model 2 + Model 3 + 3*Model 4 + 3*Model 5 + Model 6 + Model 7)


feature set 1
feature set 2
feature set 4
feature set 5
AdaBoost (Decision Tree)
Gradient Boosting
Random Forest
Random Forest
GridSearch (Adaboost (Decision Tree))
XGBoost
Voting Classifier:
GridSearch (AdaBoost (Decision Tree))
Gradient Boosting
Bagging Classifier
3
1
1
3
3
1
1
Public Leaderboard: **0,81423**
Private Leaderboard: **0,79363**

Dashboard

Public Leaderboard - Melbourne Prediction

This leaderboard is calculated on approximately 30% of the test data.
The final results will be based on the other 70%, so the final standings may be different.

#	Δ1w	Team Name * in the money	Score
1	↑16	DataSpring *	0.85457
2	↑1	Not-so-random-anymore *	0.84749
3	↓2	Komaki *	0.84443
4	↑51	Ehsan	0.83372
5	↑11	fugusuki	0.83306
6	↑3	Joseph Chui	0.82696
7	↓5	LabGOL	0.82659
8	↑23	rmldj	0.82114
9	↓1	Mehdi Pedram	0.82088
10	↓5	Kyle	0.82029
11	↓7	Claudia	0.81937
12	↑7	Medrr	0.81851
13	↑1	Alaa-Sean (UWaterloo)	0.81738
14	↓7	GarethJones	0.81524
15	**↓9**	**nullset**	**0.81423**
16	↑125	RNG	0.81216

Dashboard

Private Leaderboard - Melbourne Prediction

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

#	Δrank	Team Name ‡ model uploaded * in the money	Score
1	↑1	Not-so-random-anymore ‡ *	0.80701
2	↑35	Areté Associates ‡ *	0.79898
3	↑12	GarethJones ‡ *	0.79652
4	↑23	QingnanTang	0.79458
5	**↑11**	**nullset**	**0.79363**
6	↑14	tralala boum boum pouêt pouêt	0.79197
7	↑7	Medrr	0.79183
8	↑14	michaln	0.79074
9	↓8	DataSpring	0.79053
10	↓5	fugusuki	0.78773
11	↑21	tmunemot	0.78478
12	↓5	Joseph Chui	0.78468
13	↑12	cvanghel	0.78127
14	↓2	krischen	0.77870
15	↑14	QMRSD	0.77778
16	↑5	deepfit	0.77638