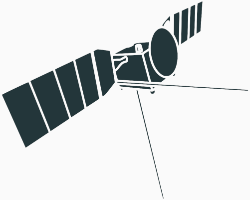


Mars Express Power Challenge

Runner-up Solution

Stephanos Stephani

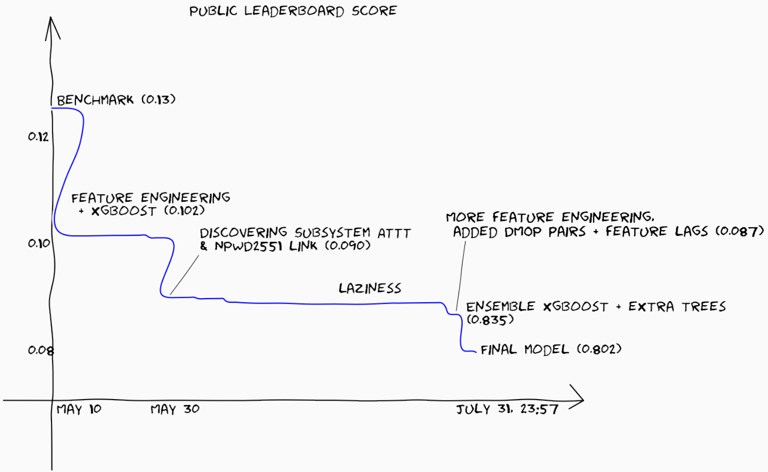
October 28, 2016

European Space Operations Centre, Darmstadt

Table of contents

Competition Recap

A rough timeline



Tools

- jupyter
- python
- scikit-learn
- xgboost
- pandas
- ~~TensorFlow~~

Feature Engineering

Long-term data (LTDATA)

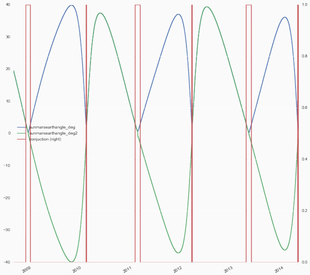


Figure 1: Added 2 features

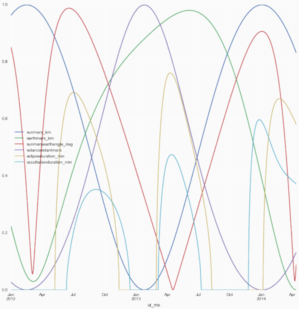


Figure 2: Removed proxies

Solar aspect angles files (SAAF)

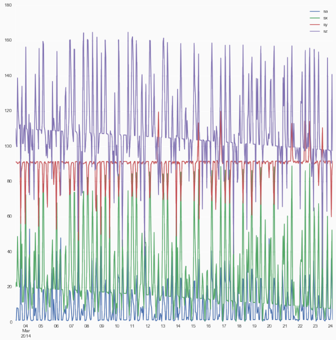


Figure 3: Sa, sx, sy, sz. These were used without modifications.

Events Files (EVTF)

- Time in Umbra/Penumbra
- Time to pericentre passage
- Height change

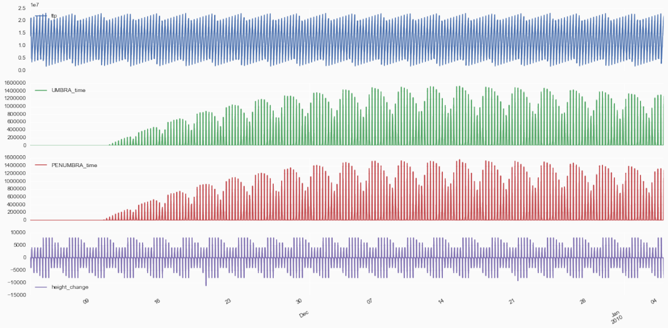


Figure 4: Time to pericentre, time in umbra/penumbra, height change.

Flight Dynamics Timeline (FTL)

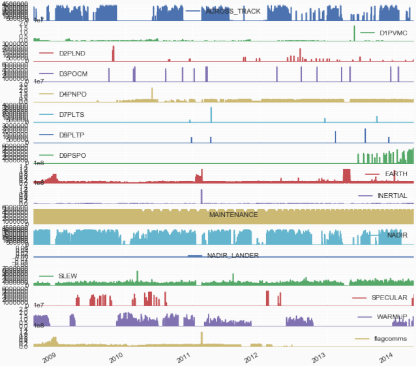


Figure 5: All FTL types in dummy form with a very bad shortcut that was forgotten.

Detailed Mission Operations Plan (DMOP)

- Command shorthands (first 4 chars)
- Pairs of subsystem commands suspected to be on-off toggles

Table 1: Examples of pairs

Command	Count
AACFM01A	7353
AACFM02A	7342
ATTTF310A	3601
ATTTF310B	3599

```
dmop.groupby(['group','subsystem'])['subsystem'].count().sort_values(ascending=False)
```

```
group  subsystem
AACF   AACFE91A         37317
       AACFE03A         36424
AAAA   AAAAF56A1        20035
APSF   APSF28A1         18312
       APSF38A1         16881
AACF   AACFE05A         16719
```

Detailed Mission Operations Plan (DMOP)

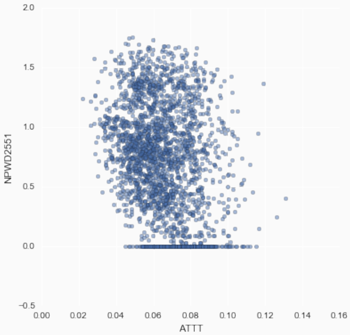


Figure 7: NPWD2551 against ATTT shorthand

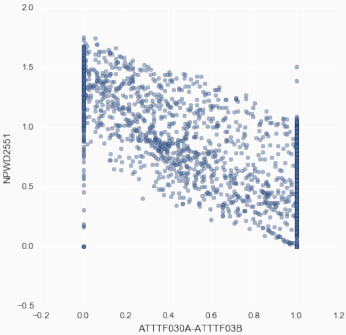


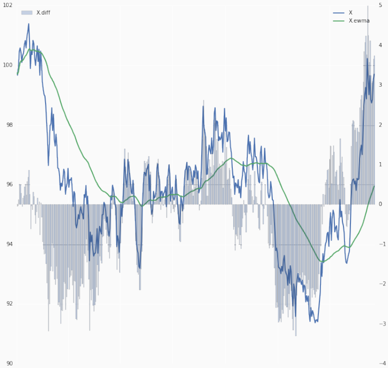
Figure 8: NPWD2551 vs ATTTTF310A-ATTTF310B

Adding lags and differences

All features exist in 3 incarnations.

Table 2: Lags and differences

Original	Lagged	Differenced
sa	sa.ewma	sa.diff
sx	sx.ewma	sx.diff
sy	sy.ewma	sy.diff
sz	sz.ewma	sz.diff



Putting it all together

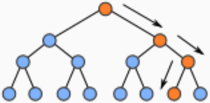
80 x 3 = 240 features available

```
'UMBRA_time,PENUMBRA_time,solarconstantmars,
sunmarsearthangle_deg,eclipseduration_min,
occultationduration_min,sa,sx,sy,sz,ut,
ttp,flagcomms,pair1,pair2,pair3,
pair4,pair5,pair6,pair7,pair8,pair9,
pair10,pair11,pair12,pair13,pair14,pair15,
pair16,pair17,PENE,PENS,AAAA,AACF,
ADMF,AHHH,AMMM,AOOO,APSF,ASSS,ASXX,ATMB,
ATTT,AVVV,AXXX,MAPO,MOCE,MOCS,MPER,PDNE,
PDNS,PPNE,PPNS,SCMN,UPBE,UPBS,ACROSS_TRACK,
D1PVMC,D2PLND,D3POCM,D4PNPO,D7PLTS,D8PLTP,
D9PSPO,EARTH,INERTIAL,MAINTENANCE,NADIR,
NADIR_LANDER,SLEW,SPECULAR,WARMUP,
UMBRA_season,conjuction,sunmarsearthangle_deg2,
des_len,asc_des,height_change'
```

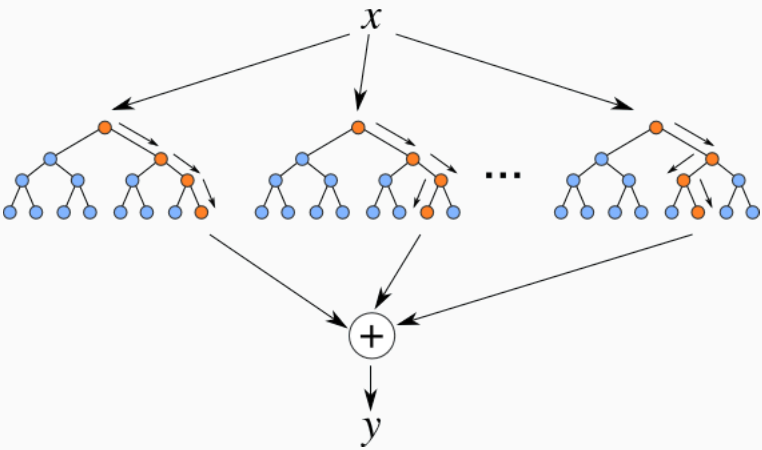
Models

The usual suspects

- Random Forests (scikit-learn)
- Extremely Randomized Trees (scikit-learn)
- Gradient Boosted Trees (xgboost)

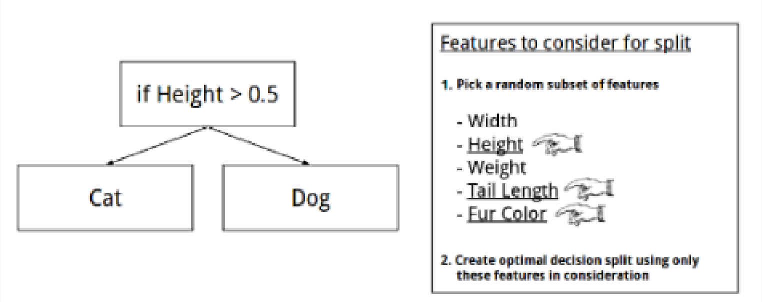


Bagging



Random Forests and Extra Trees

Random Subspace



A clumsy tour of Gradient Boosting

Boosting

In a nutshell...

Keep adding trees. Make sure each additional tree compensates for the weaknesses of the existing trees.

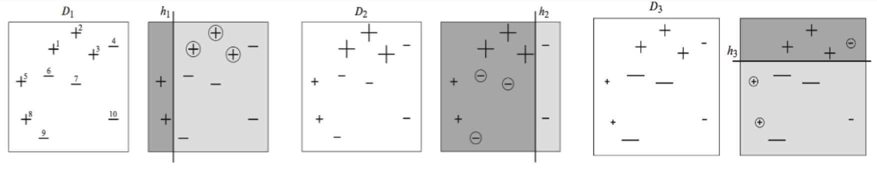
Boosting

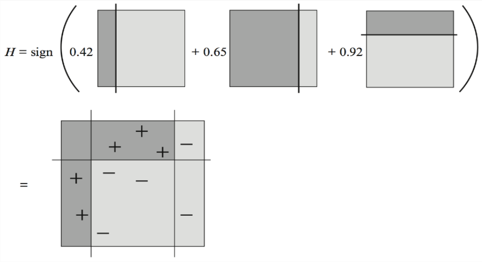
Kearns and Valiant, 1988 [5]: *"Can a set of weak learners be combined to create a stronger learner?"*

Schapire, 1990 [6]:

Schapire, 1990 [6]: *"Sure."*

Adaboost-Freund and Schapire, 1997 [2]





Gradient Boosting

Gradient Boosting [3]: AdaBoost + Gradient Descent

x	True Value (y)	Prediction ($F(x)$)	residual
x_1	102.5	101.0	1.5
x_2	45.3	40.3	5.0
x_3	231.2	230.0	1.2
x_4	149.8	149.7	0.1

Gradient Boosting

Simply add a new tree $h(x)$ which tries to predict all the residuals.

x	True Value (y)	Prediction ($F(x)$)	residual	$h(x)$
x_1	102.5	101.0	1.5	1.4
x_2	45.3	40.3	5.0	5.0
x_3	231.2	230.0	1.2	1.1
x_4	149.8	149.7	0.1	0.1

Gradient Boosting

New model = $F(x) + h(x)$

x	True Value (y)	Prediction ($F(x)$)	residual	$h(x)$	$F(x) + h(x)$	new residual
x_1	102.5	101.0	1.5	1.4	102.4	0.1
x_2	45.3	40.3	5.0	5.0	45.3	0.0
x_3	231.2	230.0	1.2	1.1	231.1	0.1
x_4	149.8	149.7	0.1	0.1	149.8	0.0

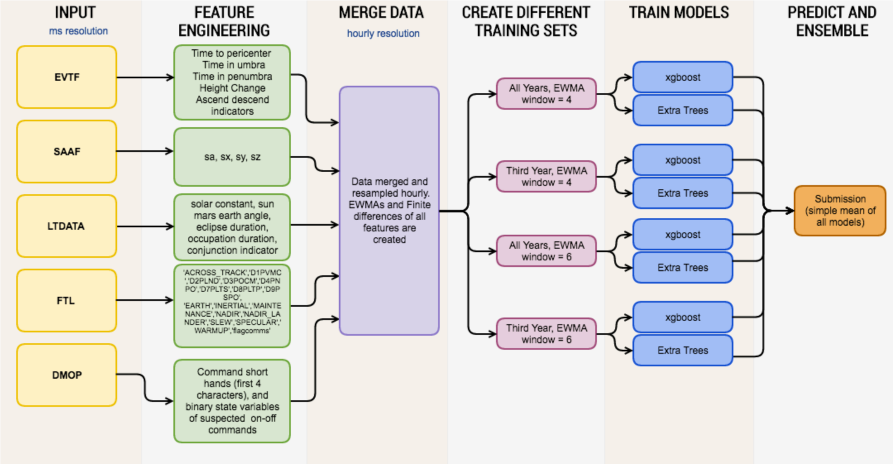
xgboost [1]

XGBoost

- Highly optimized implementation of gradient boosted trees
- State of the art results
- Fast and scalable
- Most successful algorithm on Kaggle

Final Product

Best submission



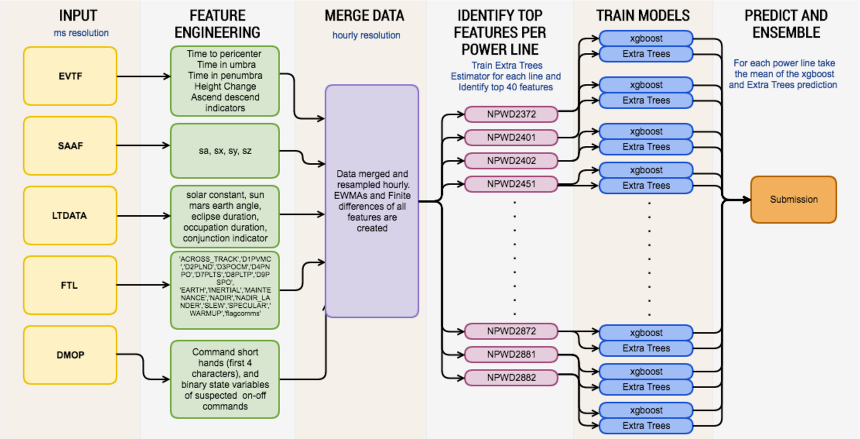
Final feature set

Table 3: Features subset for best submission, delays and lags not shown

Feature	Details
solarconstantmars	
sunmarsearthangle_deg	
sa sx sy sz	
ttp	(Time to pericenter)
pair3	['ASSSF01P0', 'ASSSF06P0'] used as on-off switches
pair9	['ATTTF030A', 'ATTTF030B'] used as on-off switches
pair10	['ATTTF321P', 'ATTTF321R'] used as on-off switches
pair11	['AACFM01A', 'AACFM02A'] used as on-off switches
UMBRA_time	Time since umbra event
PENUMBRA_time	Time since penumbra event
ATMB	Subsystem commands prefixed with ATMB
height_change	
asc_des	
SLEW	
SPECULAR	
MOCS MOCE	
WARMUP	
PENE	

Can it be simpler?

Skip manual feature selection. Skip multiple datasets.



Simpler models...

- Auto-feature selection, no data bagging: 0.082 (4th place)
- Single xgboost: 0.090 (7th place)

Get in touch

github.com/stephanos-stephani/
MarsExpressChallenge

ss2539@cornell.edu

Thank you!

Appendix - xgboost parameters

Hyperparameter tuning was performed with `sklearn.model_selection.GridSearchCV` and yielded the following:

```
xgboost.XGBModel(objective='reg:linear',
                 max_depth=11,
                 subsample=0.5,
                 colsample_bytree=0.5,
                 learning_rate=0.1,
                 n_estimators=500,
                 silent=1,
                 seed=42)
```

Appendix - Extremely Randomized Trees parameters

```
sklearn.ensemble.ExtraTreesRegressor(n_estimators=700,
                                     random_state=0,
                                     min_samples_leaf=20,
                                     n_jobs=-1)
```

Appendix - ExtraTrees-RF difference in sklearn

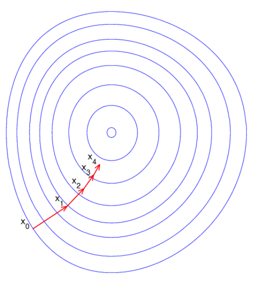
1.11.2.2. Extremely Randomized Trees

In extremely randomized trees (see ExtraTreesClassifier and ExtraTreesRegressor classes), randomness goes one step further in the way splits are computed. **As in random forests, a random subset of candidate features is used, but instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds is picked as the splitting rule.** This usually allows to reduce the variance of the model a bit more, at the expense of a slightly greater increase in bias.

Appendix - Gradient boosting

Why is it called gradient boosting?

- Mathematically related to gradient descent
- Residual = negative gradient
- When we fit new tree *h* to residual we are moving downhill towards the minimum error



Appendix - Gradient Boosting Intuition - For square error, residual happens to be the negative gradient

$h(x)$ compensates for the shortcomings of $F(x)$ by predicting $y - F(x)$

Conveniently, if we are minimizing square error, we want to minimize a loss function by changing $F(x)$. The loss function is:

$$L(y_i, F(x_i)) = \frac{(y_i - F(x_i))^2}{2}$$

$$-\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} = y_i - F(x_i)$$

Appendix - Extremely Randomized Trees [4]

Fast, good complement to xgboost

- Randomly splits (as opposed to selecting split with best information gain)
- Skipping optimization means speed
- Optimization may contribute to over-fitting or correlation of trees in ensemble. Maybe ERT adds robustness.

References I

T. Chen and C. Guestrin.
Xgboost: A scalable tree boosting system.
CoRR, abs/1603.02754, 2016.

Y. Freund and R. E. Schapire.
A decision-theoretic generalization of on-line learning and an application to boosting.
Journal of Computer and System Sciences, 55(1):119 – 139, 1997.

J. H. Friedman.
Greedy function approximation: A gradient boosting machine.
Ann. Statist., 29(5):1189–1232, 10 2001.

P. Geurts, D. Ernst, and L. Wehenkel.
Extremely randomized trees.
Mach. Learn., 63(1):3–42, Apr. 2006.

References II

M. Kearns.
Thoughts on hypothesis boosting.
1988.

R. E. Schapire.
The strength of weak learnability.
Mach. Learn., 5(2):197–227, July 1990.