CS 780 MMS Class Project

Final Project Report

Vega Group: Daroc Alden, Samantha Piatt, and Jeremy Walker

May 4, 2018

Motivation

This project is intended to explore opitons for automating a key part of the ongoing Magnetospheric Multiscale Mission - the selection of which detailed datapoints ought to be downloaded from the sattelite. This job is crrently done by a Scientist in the Loop (SITL), who must spend time each day evaluating the data observed by the satelite to decide how to spend bandwidth resources.

Automating the selection of interesting data would free up valuable human time on the project. To do this, we have explored several methods for determining what data points might be interesting, as detailed below.

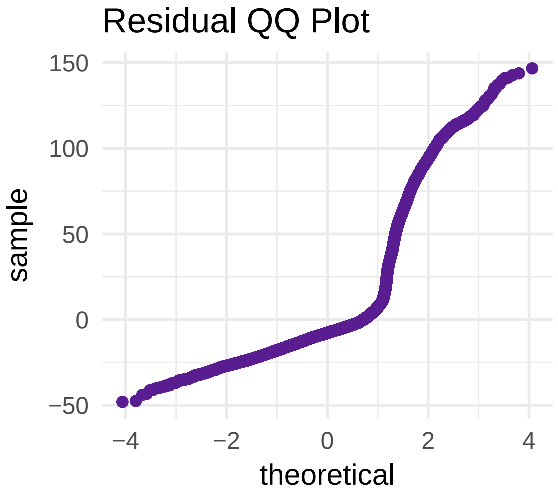
Evaluation Criteria

We are using the evaluation method outlined in Project Evaluation: Leaderboard. The method orders the predictions and then picks the top n elements where n = true number of sitl selections of the data set. Based on this, the number of found and missed selections are counted, and a weighted error is calculated (using the priority of the true SITL selections that were missed). The evaluation criteria also includes the overall classification error between the predicted selections and the true selections.

For easier interpretation, we will list only the performance on the test data in this report.

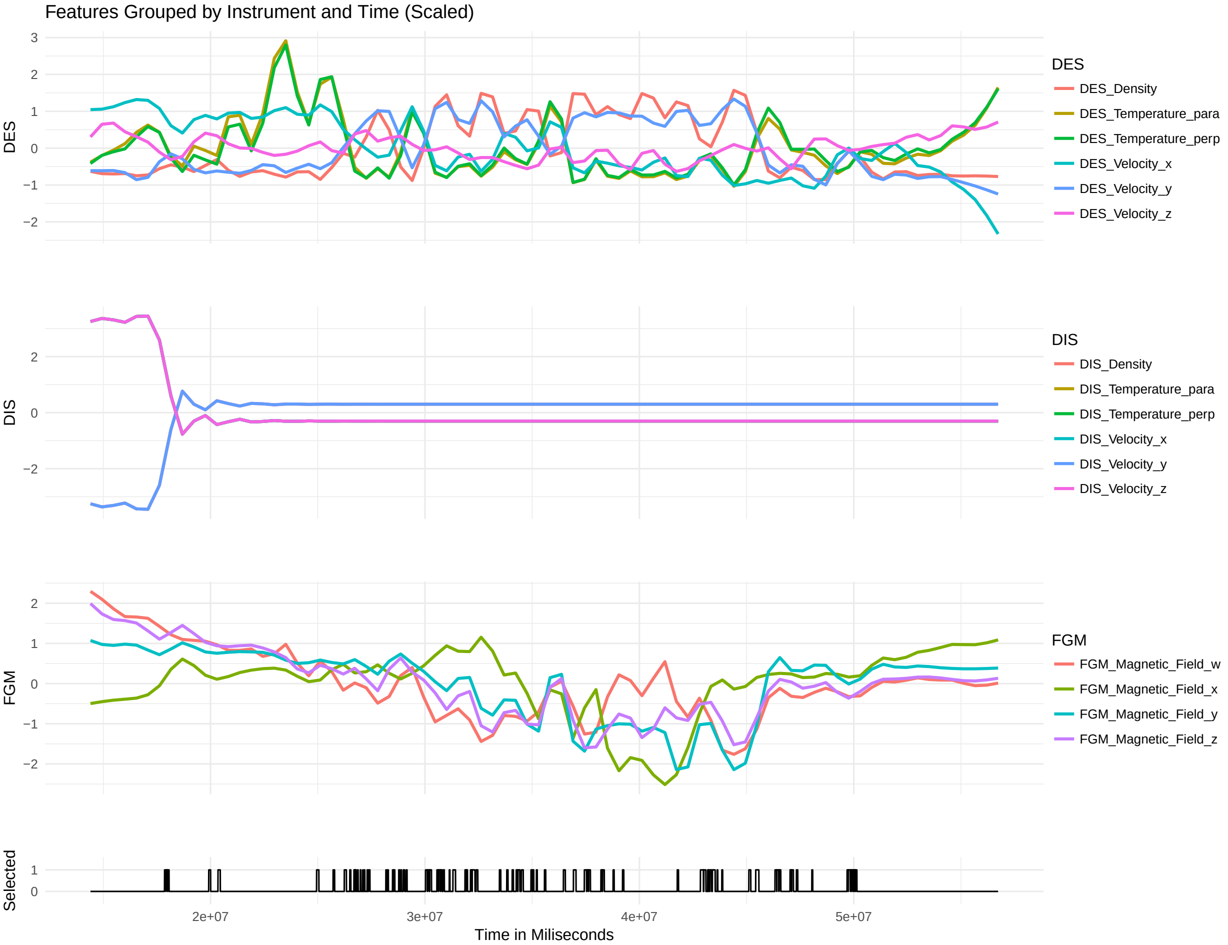
Data Exploration

We found that the data was not identically and independantly distributed, as demonstrated by the exemplary QQ plot below.



In fact, the data is a time series. We hypothesise that adding time-delayed factors (i.e. factors which contain values from previous samples) will significantly improve the performance of many machine learning models on the test and validation set.

In addition, we saw that the DIS instrument data did not seem to impact the predictions, as shown on the plot below. Because of this, we did not include it in our models, which also helps to reduce possible overfitting and computation times.

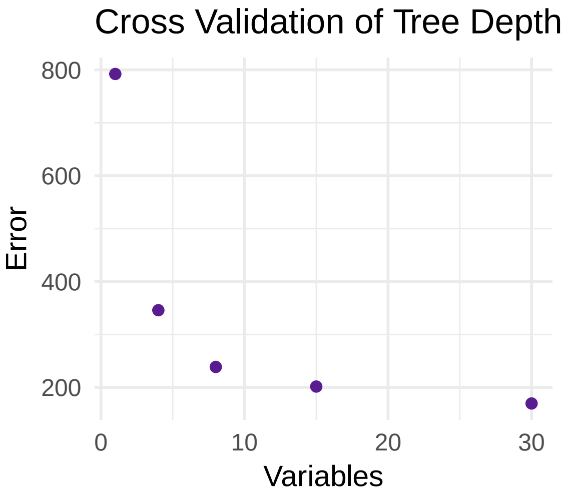


Methods

We compared our time series feature sets against the basic feature set using multiple methods. We chose to use the Priorities that the SITL used when selecting time points as our prediction taarget. Doing this, the resulting predictions will be that of a priority, and not a probability.

Our first time series feature set, which we will call time shifting, involves adding n = 5 columns to the data that contain each feature shifted n points in the past. The second time series feature set, which we will call time difference, involves adding two columns which take the average differece between the current data point x and the past and future n = 30 data points.

When we used the random forest method, we used the default tree depth setting except for our time difference model. For this, we used cross validation to pick the tree depth of 15. To save computation time, we only used 50 trees in our random forest models.



The table below shows the test performance results for Linear Regression, Radial SVM, and Random Forests using all feature sets.

	Total	Found	**Missed**	**Classification Error**	Weighted Error
Linear Regression					
Basic Data	426	186	**315**	**0.1460517**	6100.4859
Time Shift	438	200	**306**	**0.1443801**	5307.1438
Time Difference	436	261	**243**	**0.1265440**	5311.7890
Radial SVM					
Basic Data	426	272	**229**	**0.1043559**	4468.3803
Time Shift	438	294	**212**	**0.0987254**	3699.4909
Time Difference	436	423	**81**	**0.0418452**	1181.6560
Random Forest					
Basic Data	426	323	**178**	**0.0805963**	3140.3850
Time Shift	438	325	**181**	**0.0852839**	2948.5776
Time Difference	436	466	**38**	**0.0191581**	529.1858

Analysis

The time difference feature set does significantly better than the others when using any of the three methods we tried (Linear Regression, Radial SVM, or Random Forests).

Validation

Because we compared our feature sets using multiple methods (using the the same test and training data), we held out a small validation set to reduce the chances of overfitting. Since the time difference features using random forests did the best, we will further evaluate it's performance on the validation set we withheld. This model performs well on our validation set, indicating that it should do fairly well on unseen data.

	Total	Found	**Missed**	**Classification Error**	Weighted Error
Time Difference	256	267	**19**	**0.0202117**	370.5141

Reccomendations

It is likely that training our model on a wider variety of data, from different days, would give better prediction performance. Additionally, our model would benefit from cross validating the width of the average difference window. At the moment, we are picking a ‘reasonable’ static window size of 30, which may not be the optimum choice. The only hyperparameter we were able to tune is the tree depth. However, the number of trees used in the random forest model should also be tuned using cross validation for better prediction accuracy.

Related work

- Accuracy of Extrapolation (Time Series) Methods: Results of a Forecasting Competition by S. Makridakit et al. is a definitive comparison of different forecasting methods on different types of and horizons of data. While this is not a forecasting problem per se, the conclusion that they draw is likely appropriate to our problem because of the complex and varied nature of the S.I.T.Ls who decide what data ought to be acquired from the satelite: “[S]tatistically sophisticated methods do not do better than simple methods . . . when there is considerable randomness in the data.” (pg. 142 in the Journal of Forecasting, Vol 1, Iss. No. 2)
- Rolling Window Selection for Out-of-Sample Forecasting with Time-Varying Parameters by Atsushi Inoue, Lu Jin, and Barbara Rossi reccomends limiting the size of the window of data used for forecasting so that regime changes in the input data don’t disrupt the fit of one’s model. Unfortunatly, this is not directly applicable to our project either, since we do not have access to all the data up to the present day, but instead only have selected samples.
- Time Series Prediction Using Support Vector Machines: A Survey by Jicholas Sapaankevych and Ravi Sankar points out that in the most common application for predicting time series data - the financial sector - a weighted variant of SVMs is often used called C-ascending SVMs, and that these have proven more adaptable, since they emphasize recent or current data over older data.