IML Final Report

KS, SL

2024-12-16

Introduction

This document is term project final report on course Introduction to Machine Learning Autumn 2024 at University of Helsinki.

Term project was based on GeckoQ data set that had roughly 32 000 atmospherically relevant molecules with 24 variables. Our task was simply to train best possible prediction model on this data set of atmospheric measurements for log_pSat_Pa(response). Our best solution would then be evaluated and compared and ranked against solutions by other groups in Kaggle Competition organised by the course personnel.

We went on to explore the data first, carry out feature engineering, try out different models that we learned during the course, evaluate our models performance on chosen metrics and finally perform model selection with what we would take part in the competition.

In this report we first go through our evaluation metric and data details. Then we explain how we inspected the data and what method we used. After that we explain our approach to feature selection. Finally we introduce our models, inference and conclusions. Our model performed rather close to winning team and we ended up somewhere in the middle of the pack in the final rankings. The model we ended up choosing was SVR.

Authors of this report are Simo Liimatta and Kim Ståhlberg. Our group number was 16.

Evaluation metric

Evaluation metric used in this task was dictated by course personnel and it was the variance explained, R-squared:

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

Where $RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ and $TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$ and y_i is observed value and $\hat{y}_i$ is predicted value. This metric provides a proportional mean squared error and it takes values between 1 (perfect fit) and 0 (worst fit). With this metric it is possible to compare models independent of the absolute scale these models operate on and it fits well to evaluate regression models. Later in this document when we mention “score”, we mean specifically this R-squared metric if not defined otherwise.

Data

Following is a direct quote from the task instructions concerning our data:

> The term project is based on the GeckoQ dataset with atomic structures of 31,637 atmospherically relevant molecules resulting from the oxidation of α-pinene, toluene and decane. The GeckoQ dataset is built to complement data-driven research in atmospheric science. It provides molecular data relevant to aerosol particle growth and new particle formation. A key molecular property related to aerosol particle growth is the saturation vapour pressure (pSat), a measure of a molecule's ability to condense to the liquid phase. Molecules with low pSat, low-volatile organic compounds (LVOC) are particularly interesting for NPF research.

Acknowledging our limited expertise in atmospheric science, we approached this task using techniques and methods learned in this course, complemented by additional study of linear models and statistics.

Response variable was logarithmic transformation of pSat mentioned above and it took values on numeric scale. Rest of the data set had 3 numeric variables and 20 variables with integer values (some categorical and some counts) and one character variable.

Data was already split into training set (size 26 637) and test set (size 5 000). Training set included response variable and test set did not.

This data set had no missing values in numeric variable columns. "Parentspecies" that was the character variable had 210 missing entries which we decided to remove from our data set.

Visual exploration

Visual exploration of the predictor variables, including examination of their distributions and scatter plots against the response variable (log(pSat)), did not reveal any patterns or relationships that we felt like would need further action.

When exploring linearity between response and each variable we fitted a linear model per each and looked at the residual plots. NumOfConf was the variable that seemed to have the least linear relationship between the response. You can see from Figure 1. (upper plot) that residuals has this funnel-like shape that indicates non-linearity and non-constant variance in relation.

Inspecting correlation matrix in Figure 2 we see that there is strong correlation between variables in top left corner. This is logical since all variables correlating heavily are related to number of atoms and molecular weight. We take this into consideration in feature engineering for our linear models.

We also inspected the variables using the Principal Component Analysis (PCA). Figure 3 shows the PCA plot. From it, we can inspect, that the first 2 principal components explain around 28% of the total variability of the response. We also see, that quite some number of the variables add little to none explaining power to the variance. We explored this further in subset selection for linear regression.

Feature engineering

Based on the residual plots we decided to try different transformations (exp, log, different order of polynomials) with NumOfConf. From the residual plot in Figure 1 (lower plot) we can see that this transformation truly affects the residuals to appear more randomly than without it. This indicates that this relation with log-transformation is more linear. In practice it increased our score a little bit with linear models so we decided to experiment this transformation with other models as well.

Modeling

We started modeling with a dummy model that was just an average of response and this model gave us a score of 0 with test set. This model and score served as a starting point for us. Next logical step to try to best this model was fitting a linear model with all features. This model gave us a score of 0.711 and acted now as our new reference model.

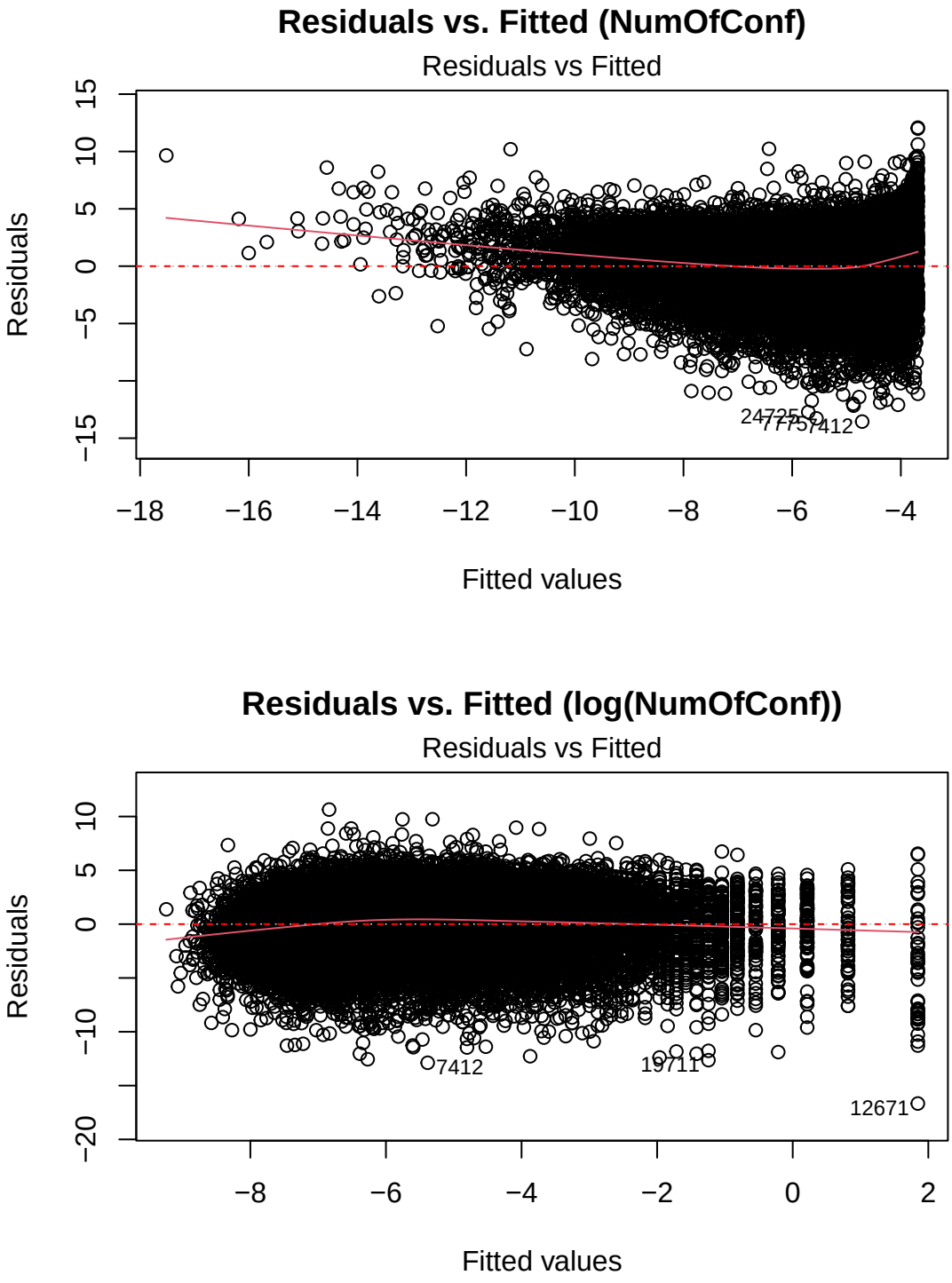

Figure 1: Comparison of Residual Plots; response vs NumOfConf (top) and response vs log(NumOfConf) (bottom)

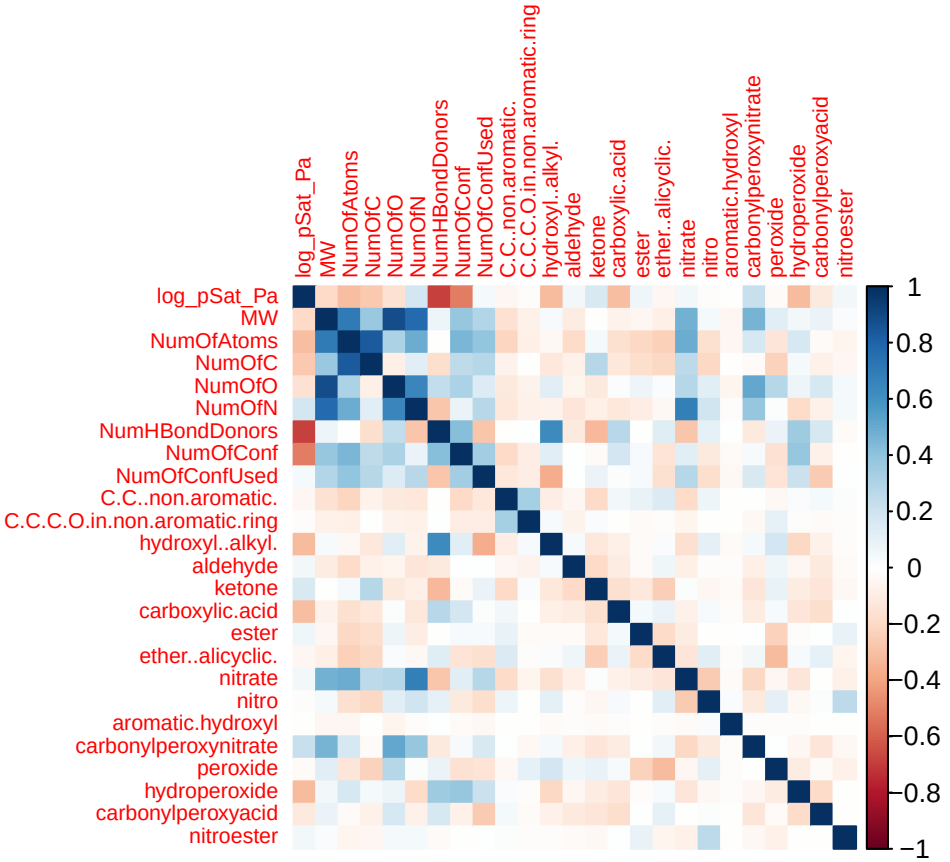

Figure 2: Correlation matrix with all features.

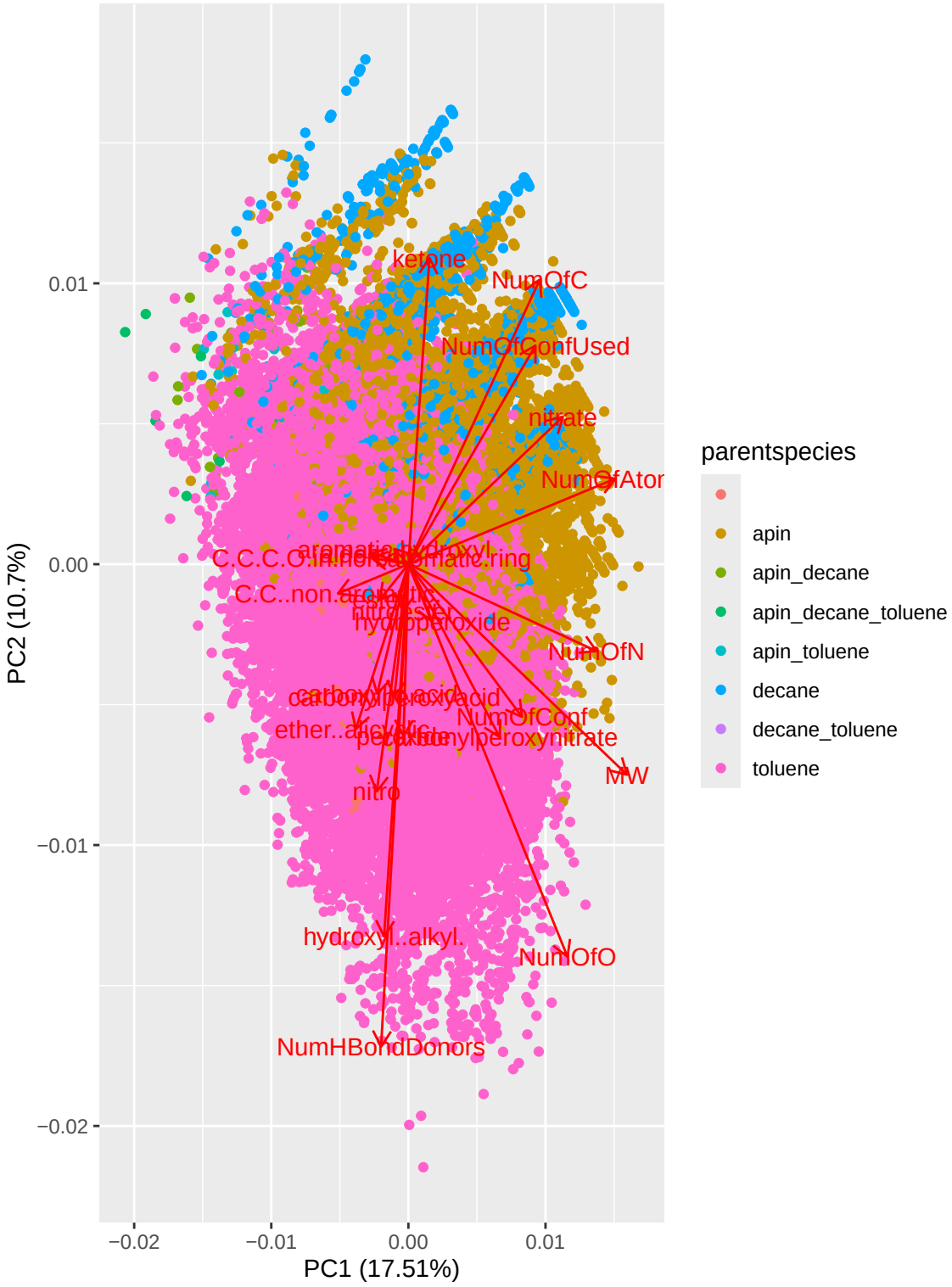

Figure 3: Principal component analysis of the variables

From this point on we decided to try following models and see how they perform against out baseline; linear model with feature selection and engineering, PCR, RF and tree based methods, and SVM. In the following sections we discuss more of these models and these performed.

When we had an option to use validation set with the models we used cross-validation (5 folds) as validation method to provide more robust approximation of our score in comparison to just splitting our training data into (new subsets of training data) fixed training and validation sets.

Linear models

Feature selection for linear models In order to improve our linear model with feature set we decided to reduce most of the highly correlated variables. We ended up removing following variables from our data set: MW, NumOfAtoms, NumHBondDonors and NumOfConf. Correlation matrix for this subset can be seen in Figure 4.

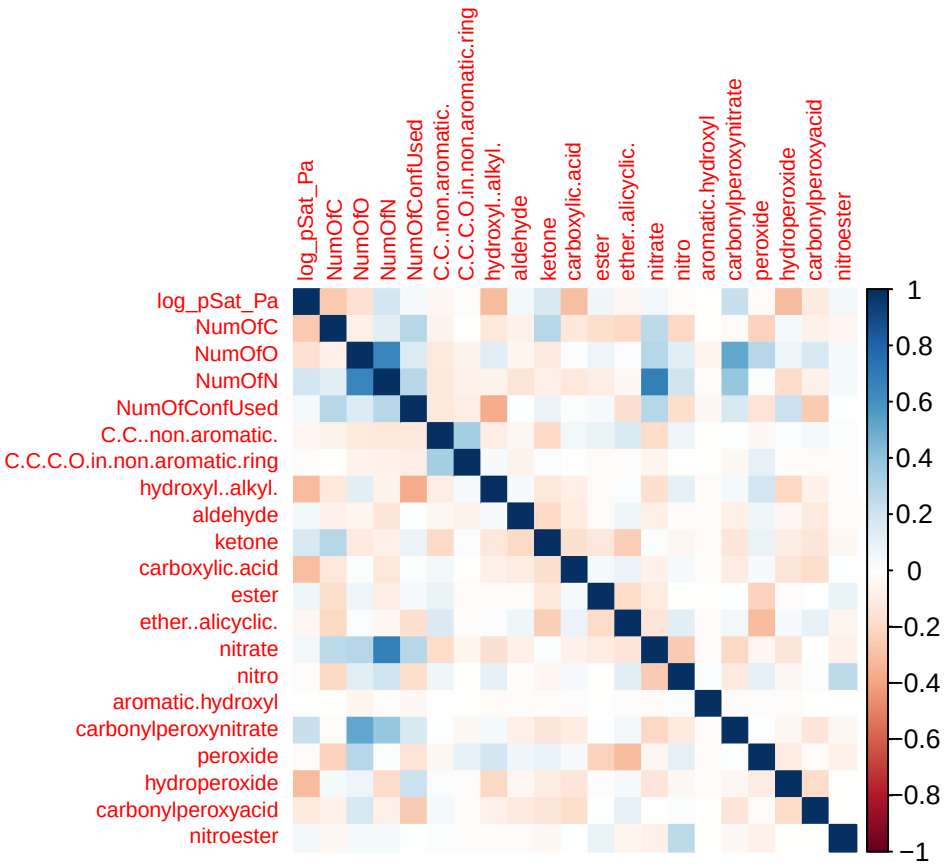

Figure 4: Corralation matrix with removed variables (MW, NumOfAtoms, NumHBondDonors, NumOfConf) to reduce correlation between variables.

As we can see this correlation matrix is more neutral in color that means there is mostly correlation closer to value 0 that the extreme points 0 or 1.

Next we inspected how does this removal of variables compare in exhaustive subset selection. Results can be seen in Figure 5. From this figure we can see that with subset that has less (mean closer to zero) correlation it reaches faster better score but eventually full data set overcomes it.

Because removing these variables does not improve adjusted R^2 significantly we decided to continue our feature selection with full data set just in case we lack some domain knowledge and accidentally remove some variables of whose effect on our model performance we can not guess.

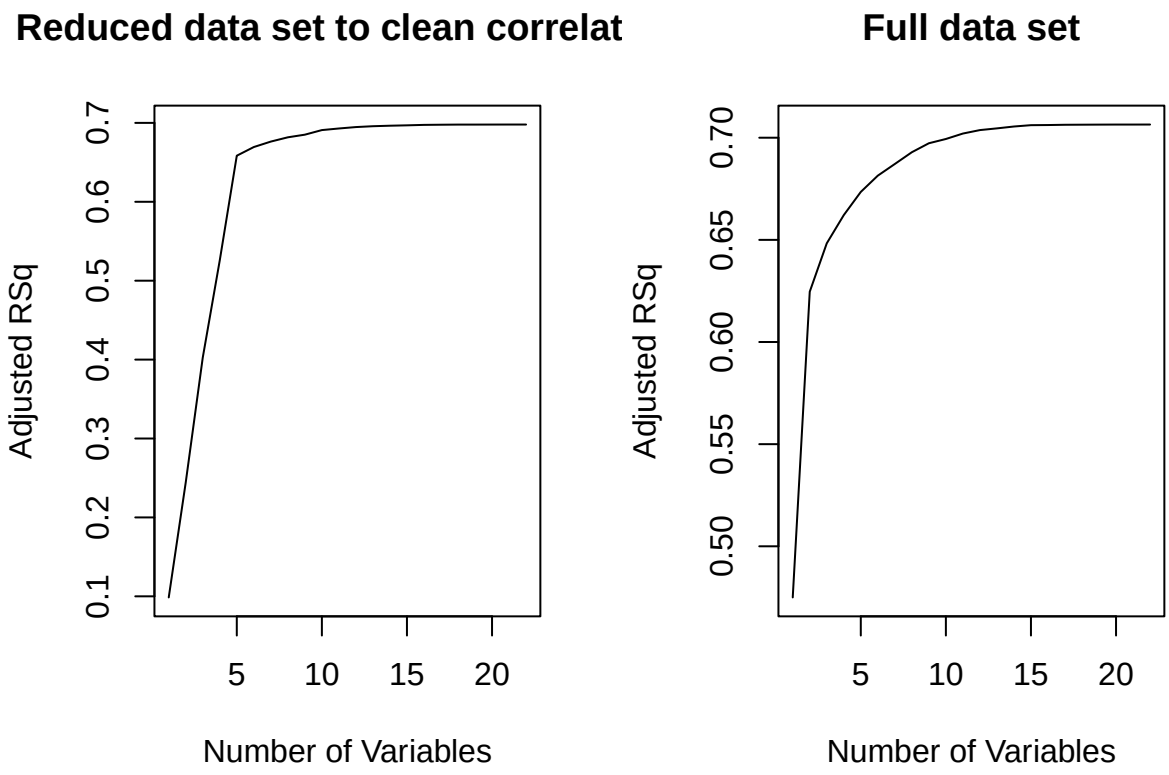

Figure 5: Left: Data set with reduced corralation. Right: full data set.

We wanted to also see what results forward and backward step-wise methods would provide. Results can be seen in Figure 6. From this figure we can see that forward selection looks a lot like exhaustive selection with full data set. Backward selection seems to perform worse in comparison to exhaustive and forward methods with full set.

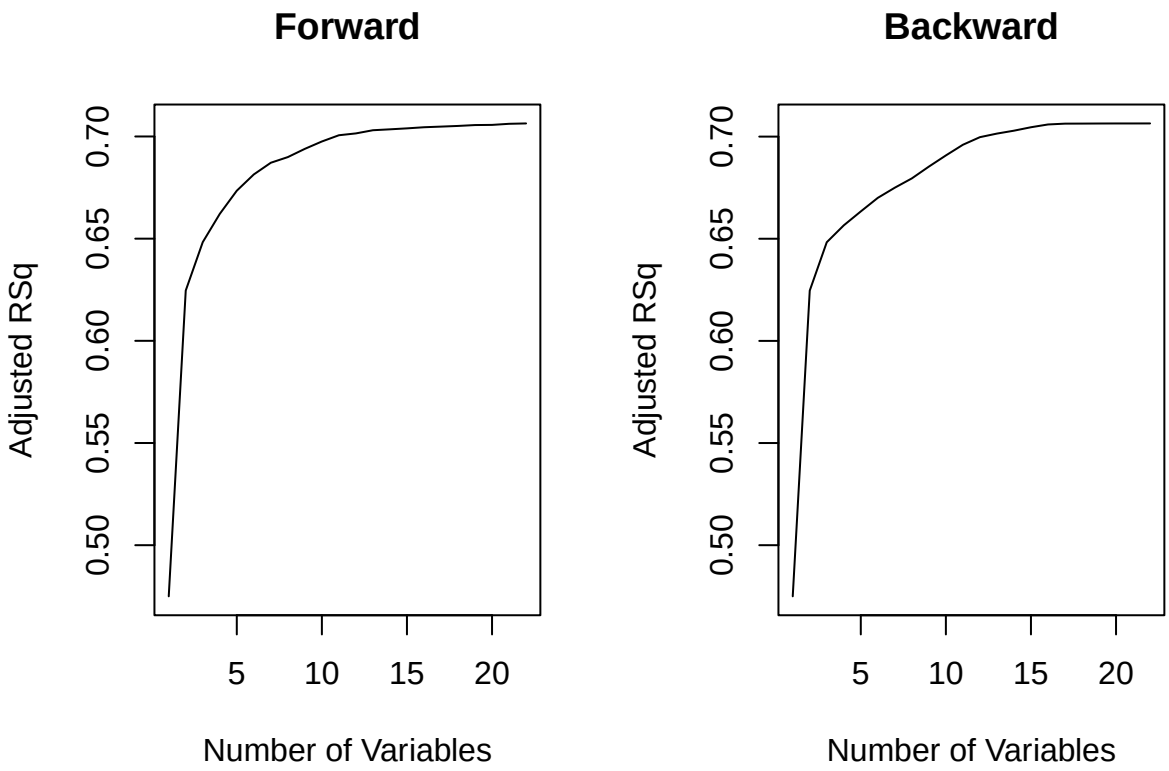

Figure 6: Forward and backwards feature selection with full data set.

Because we wanted to try linear regression with fewer variables than full data set we decided to try forward selection (Figure 6 left plot) with 13 variables. It seems to plateu at 13 variables and there was no great difference with exhaustive method. Since one variable was a sub category of this character variable "parentspecies" we decided to include this full variable into our sun selection of variables. Features in this subset can be seen in Table 1.

Table 1: Variablse included after forward step-wise feature selection.

Variable
NumOfC
NumOfO
NumHBondDonors
NumOfConf
NumOfConfUsed
parentspecies
C.C.C.O.in.non.aromatic.ring
carboxylic.acid
ester

Variable
ether..alicyclic.
aromatic.hydroxyl
hydroperoxide
carbonylperoxyacid
log_pSat_Pa

Linear model with feature selection and engineering When we fit linear model with this above mentioned sub set without log-transformation of NumOfConf (see Figure 1.) we got following results:

```
## Linear Regression
##
## 26427 samples
##    13 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 21141, 21141, 21142, 21143, 21141
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1.771605  0.6782671  1.329856
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

We got following results with the same sub set but this time with log transformation on NumOfConf:

```
## Linear Regression
##
## 26427 samples
##    13 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 21142, 21142, 21143, 21141, 21140
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1.761321  0.6819619  1.319318
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

As we can see linear model with feature engineered subset and with log transformation on NumOfConf we achieved almost as good of a score as with full data set. Even thou we did not manage to best our linear model with full data set we were able to cut dimensions almost to half.

Principal Component Regression (PCR)

The results with PCR with full data set excluding ID can be seen in Figure 7. PCR model reaches score 70.61 with 24 components and with 27 components it reaches it's max score 70.77.

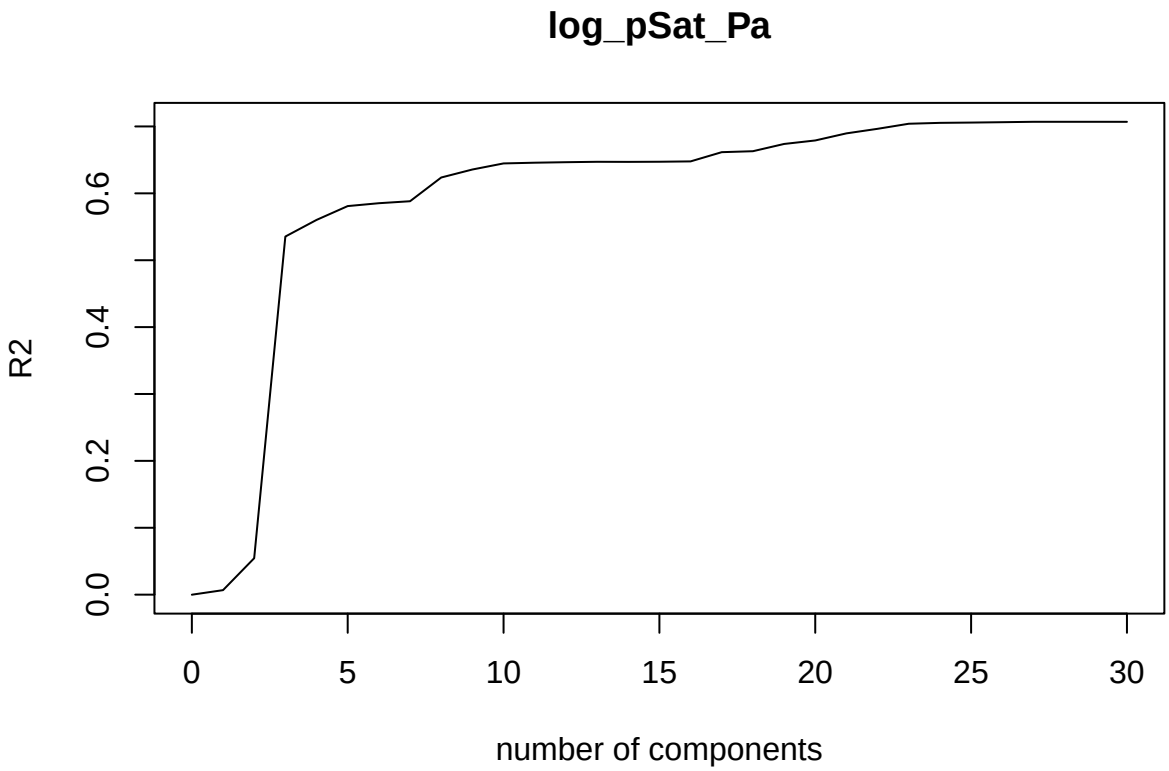

Figure 7: R2 scores with PRC on full data set (excluding ID).

Tree based methods

We implemented a selection of tree-based methods in order to better our prediction score. These included basic regression tree, Bagging, Boosting, Random Forest, and Bayesian Additive Regression Tree.

Interestingly, the basic regression tree used only 3 variables to train the regression tree. We implemented the trees to the full training data set, the feature engineered subset and both versions with and without the transformation of the NumOfConf variable. In the end, none of the regression trees with these data sets ended up giving much bigger score than the reference full linear regression. The largest score we got was from Random Forest, which was 0.73.

SVR

We also implemented Support Vector regression (SVR). Similarly, as in the tree based methods, we tried the SVR on the full data set, the selected subset, and both with the log transformation of NumOfConf. Using the full data set, the model yielded the private score of 0.739, which ended up being the highest score from all our models.

Best solution

In Table 2 we can see private scores for our models worth mentioning. Public scores were in the same order but differed just a little bit. So the best solution from our team was SVR. It is surprising to see how well linear model with no manipulation (except data cleaning) out into it performs.

Table 2: Private scores of the most important models

Model	Prive_score
Full linear regression	0.711
Subset linear regression	0.682
PCR	0.708
RF	0.728
SVR	0.739

Final thoughts

After seeing the great student presentations from the top scorers we started to reflect our approach to this task. First impression was that SVR appeared on a lot of lists, and it was our best model.

One difference in relation to ours that we saw up on the stage was that most of them seemed to try many models not mentioned in detail during this course. This involved neural networks and different boosting methods just to mention couple first ones that comes to mind. Our approach was to use models that we understood and that we experimented with during the exercise sets. We are not sure if this made a huge difference since we were quite close with our approach the top scorers.

Other thing that caught our attention was that while we focused on reducing the dimensions in our models others seemed to not give a damn about their dimensions. Maybe this is a good demonstration that dimension reduction is not so important while dealing with dimensions like in this task, say 30 times 30 000.

One could argue that with dimensions like we had, computation is not really a problem and algorithmic model and parameter brute forcing is viable option to get things going. Fine tuning and domain knowledge seems to come into play after this kind of fast random exploration of models and features.

Future exploration and improving the model would include one-hot-encoding the "parentspecies" variable, trying out the polynomial regression with different degree polynomials and what degree would yield best test score or MSE.

We found this course very interesting and while working this term project a lot of things which were difficult during the lectures came super nicely into practice.