Term Project - Group 80

Sari Ropponen, Outi Boman, Loic Dreano

2022-12-11

Description of data

In the project a data about new particle formation, NPF, is used. The data is divided into training data (npf_train.csv) and testing data (npf_test_hidden.csv) and it includes 104 variables relating to daily measurements taken in Hyytiälä forestry field station. The number of observations in training data is 464 and 965 in testing data.

Table 1: Summary of the dataset

	npf_train	npf_test
Measurements	464	965
Variables	104	104

Some of the variables like temperature T and CO2 are measured in different heights. The height is indicated in the name of the variable, for example, T84.mean is the mean temperature at 8.4 meters above the mast base. The data includes also a response variable indicating if a NPF event has happened during the day or not. In the project a binary classifier is build to predict if an NPF event will happen or not during the day according to the observed measurements.

Preprocessing data

The data includes also some variables that are not needed as variables. A summary of the the variables in training data is below:

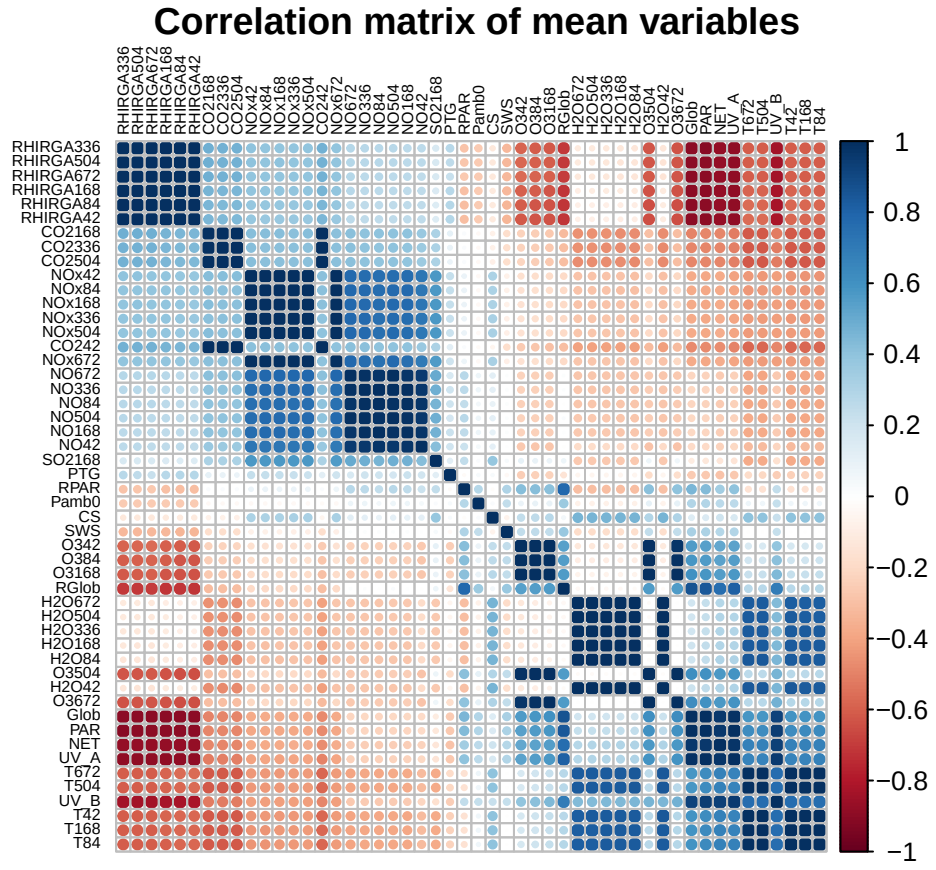
```
##        id            date              class4          partlybad
##  Min.   :  1.0   Length:464         Length:464         Mode :logical
##  1st Qu.:116.8   Class :character   Class :character   FALSE:464
##  Median :232.5   Mode  :character   Mode  :character
##  Mean   :232.5
##  3rd Qu.:348.2
##  Max.   :464.0
```

The column “date” was set to be the row names in the training data. Because the value of the logical variable “partlybad” is FALSE for all the observations, it doesn’t give any information. Columns “id”, “date” and “partlybad” were removed from the data.

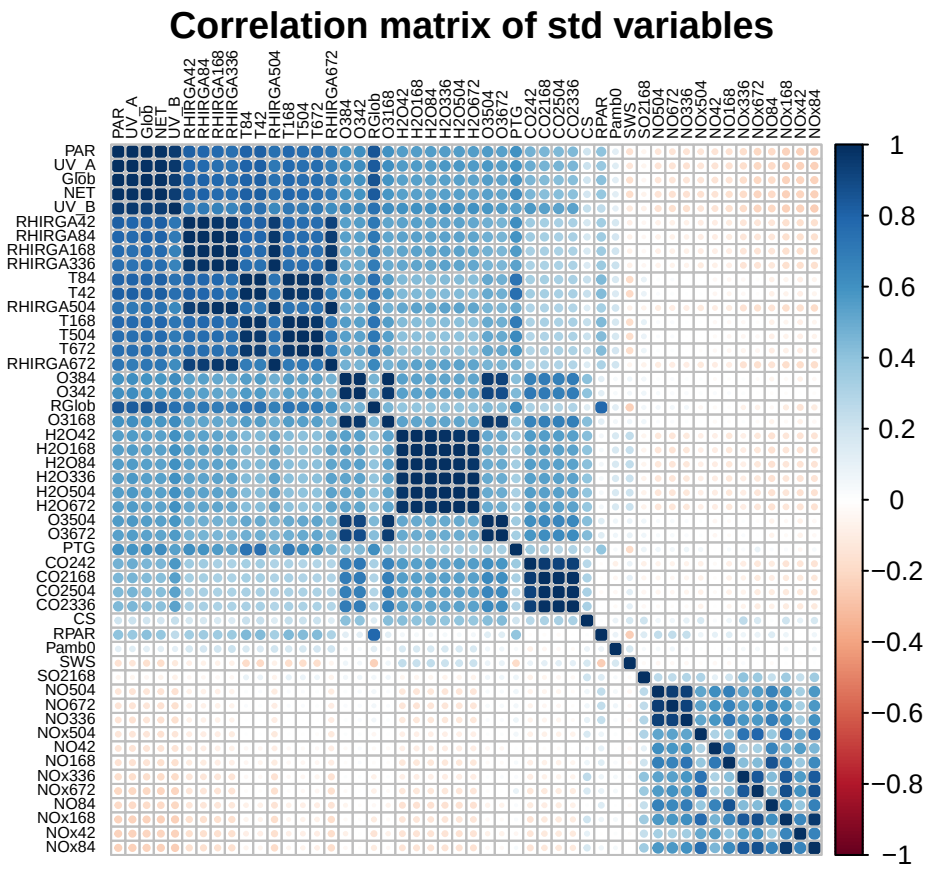
A qualitative variable “class2” is added to the training data. It gets either value “event” or “nonevent” according to “class4”. Variable “class4” indicates the type of the event if it has happened (“Ia”, “Ib” or “II”) or “nonevent” if no event has happened during the day.

Data exploration

Because the data includes same measurements at different heights it is expected that there are correlation between variables. Correlations between different mean values are shown in the matrix below:



The same matrix for different standard deviation values is



As we can see, there are a lot of highly correlated variables and since they carry the same information, keeping all of them would not improve the predictivity of our models. In order to increase the power of our models to identify independent variables that are statistically significant and to make them simpler to interpret (simpler model in general) we will remove the highly correlated variables.

To do so the variables are clustered together based on their correlation; every pairs of variables which have an absolute correlation greater than 0.8 are clustered together. Then a random variable from each cluster is kept for further analysis.

The correlations for mean and standard deviations after cleaning the data are shown in the pictures below as well as a table of variables left in the data (table 2). Table 3 shows that the data includes only 26 variables. Histograms of the variables left in the cleaned data are given in the annex. The histograms give us an idea of the values of the variables which we need, for example, in PCA to decide whether to normalize and scale the data or not.

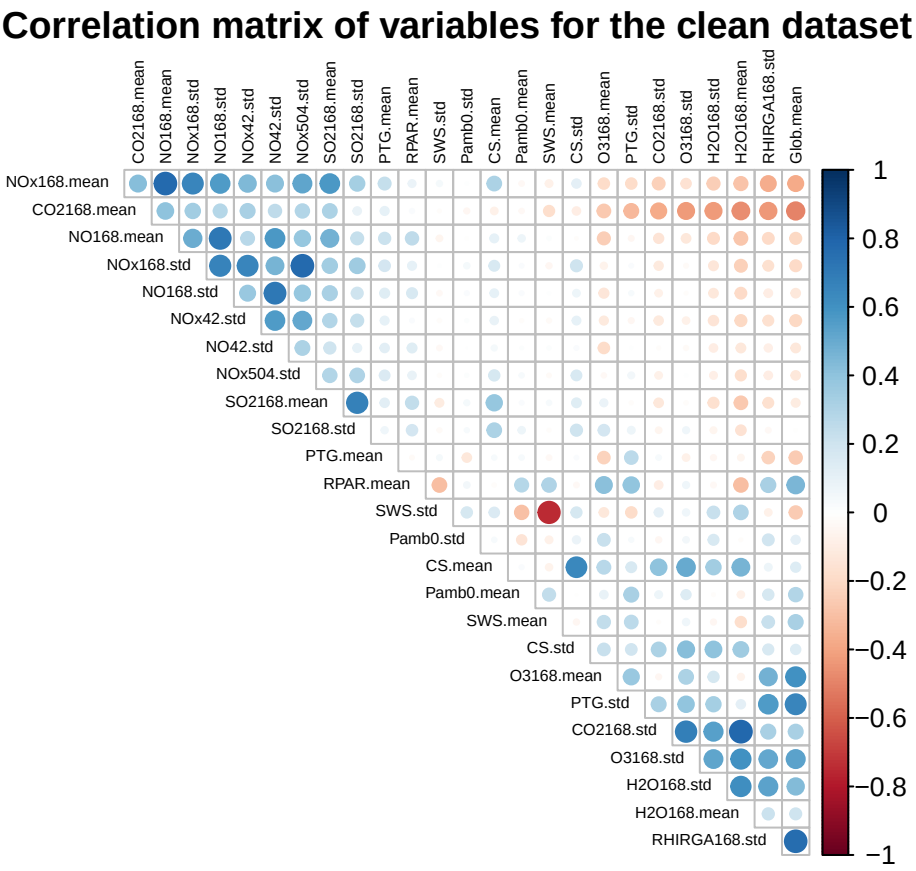


Table 2: List of kept variables

NO42.std	SO2168.mean	CO2168.std	NOx168.std
NOx42.std	SO2168.std	H2O168.mean	O3168.mean
NOx504.std	SWS.mean	H2O168.std	O3168.std
Pamb0.mean	SWS.std	NO168.mean	RHIRGA168.std
Pamb0.std	CS.mean	NO168.std	RPAR.mean
PTG.mean	CS.std	NOx168.mean	Glob.mean
PTG.std	CO2168.mean	NOx168.std	

Table 3: Summary of clean dataset

	npf_train	npf_test
Measurements	464	965
Variables	26	26

Performance measures

To compare different classifiers two measures are used: accuracy and perplexity. Accuracy is the proportion of the observations that has been classified correctly. Perplexity is a rescaled variant of log-likelihood. If perplexity equals to 1 the classifier predicts always the probability of an observation to an actual class. Perplexity of 2 corresponds to coin flipping.

If the method predicts a probability of an event, observation is classified as "event" if the estimated posterior probability is more than 0.5. Otherwise the observation is classified as "nonevent".

Performance measures are calculated using validation method and 10-fold Cross-Validation. For validation method the training data is randomly divided into 2 equally sized data sets: training data to fit the model and validation data to estimate the accuracy and perplexity. Accuracy and perplexity are also calculated in training data for comparison and to evaluate if there is overfitting.

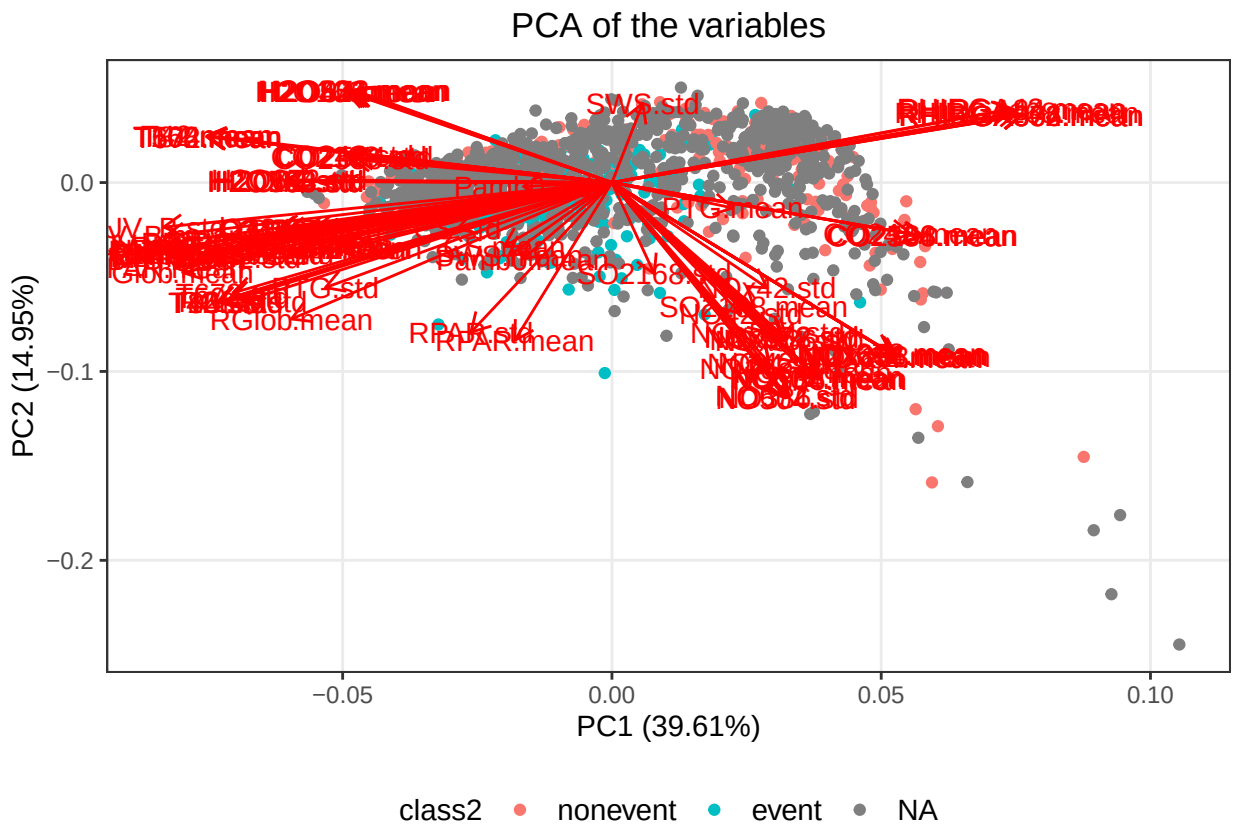
Generalization accuracy and perplexity is calculated using cross-validation. The data is randomly divided into 10 folds and performance measures are calculated using each fold as a validation data in turn. Since, a single run of the k-fold cross-validation procedure may result in a noisy estimate of model performance. In fact, different splits of the data may result in very different results. The cross-validation procedure is repeated 100 times and the mean result across all folds from all runs are reported.

The increase of repetition increase also the risk to get probabilities values of 1 or 0, in that case the calculation of the perplexity isn't possible. Thus, in order to get a finite number result, 1 values are transform to 0.999999999 and 0 values are transform to 0.000000001.

Investigation of the variables: PCA

Principal Component Analysis (PCA) is used to study how much we can reduce dimensionality of the data but to save as much of the variability (that is, information) of the data at the same time. We do the PCA using the original training data (npf_train.csv) combined with the original testing data (npf_hidden.csv). As we can see from the histograms of the variables (see the annex), the variables are measured in different units with different magnitudes of variances. Thus, the variables are centered to have zero mean and scaled to have standard deviation one. The responses variable are removed from the data because we are using unsupervised learning method.

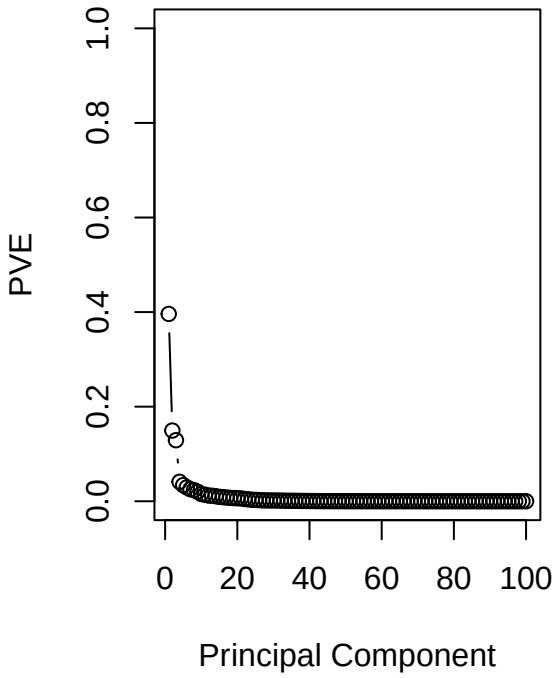
First, we do PCA to the original data before cleaning the correlated variables. The biplot of the analysis is below:

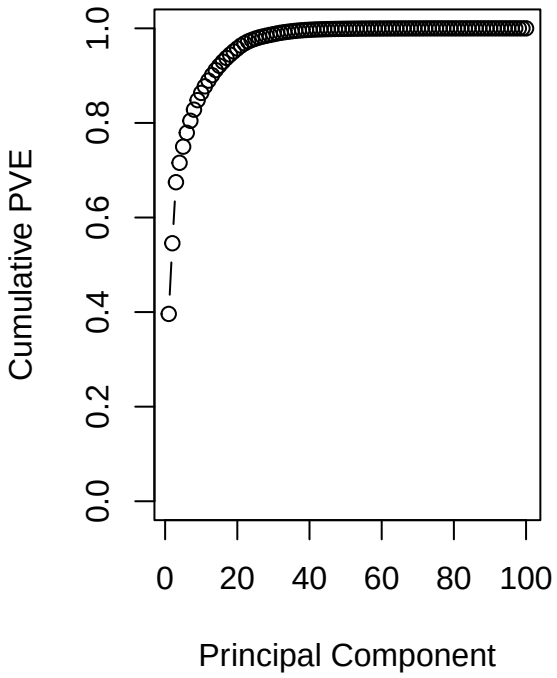


The biplot includes the first two principal components (PCs), both the scores and loading vectors. Also, the proportion of variance explained (PVE) by the PC is indicated. As we can see, many of the vectors are overlapping meaning that they are correlated. This gives us a further justification to remove the correlated variables and leaving only one of each in the data. We can also see that the first PC (PC1) explains only 39.61 % of the variance in the data and the second PC (PC2) 14,95 %. Together they explain only 54,56 % of the variance which is quite poor result the target being more than 80 %.

From the biplot we can also see that there are a couple of outliers in the bottom right corners. For now, we leave them to be and handle them during modelling if needed.

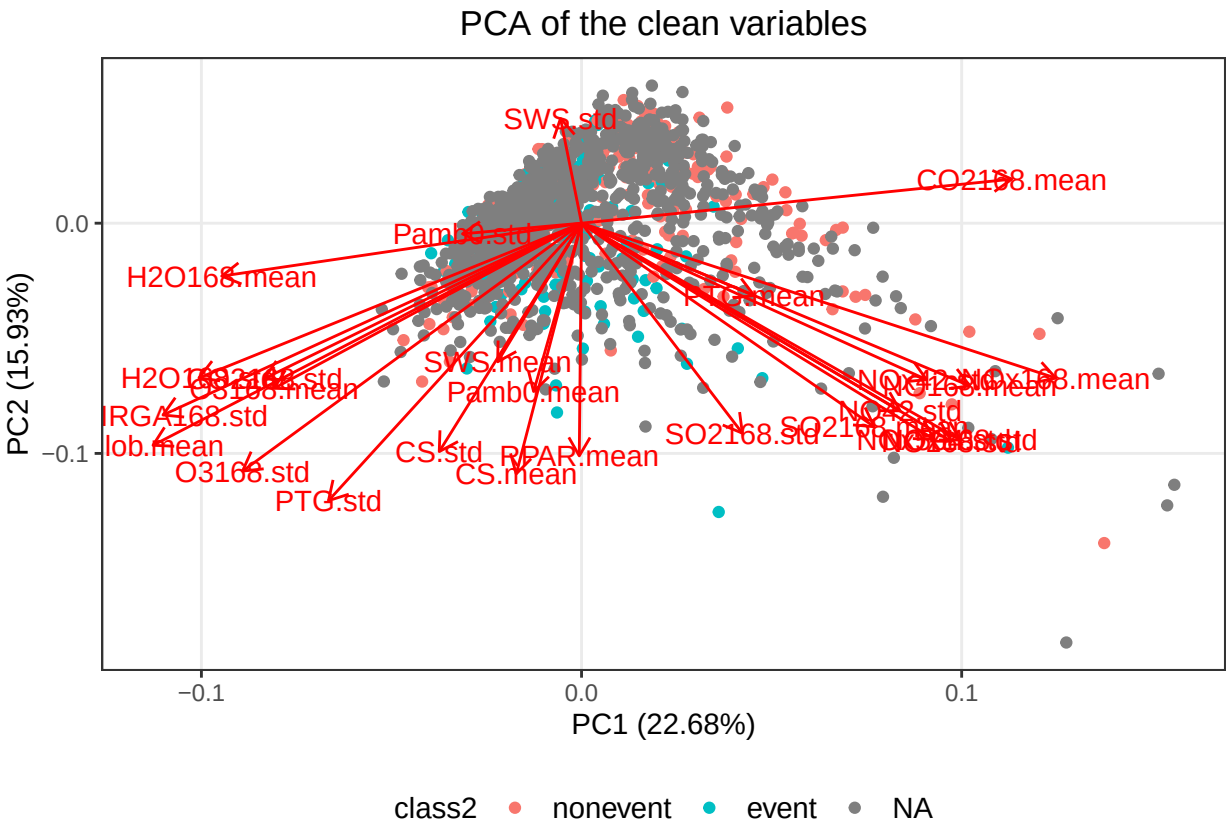
To see how many PCs are needed to explain more than 80 % of the variance the PVE by each principal component and the cumulative PVE is shown in the figures below:





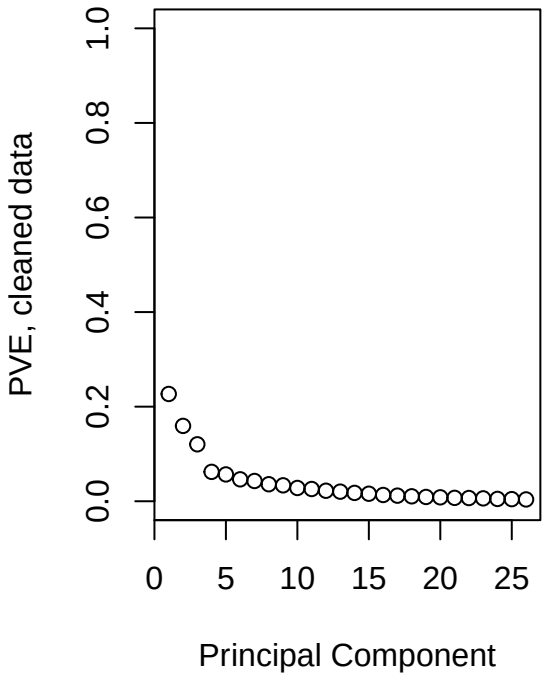
For the original data we would need 10-20 PCs to explain most of the variance.

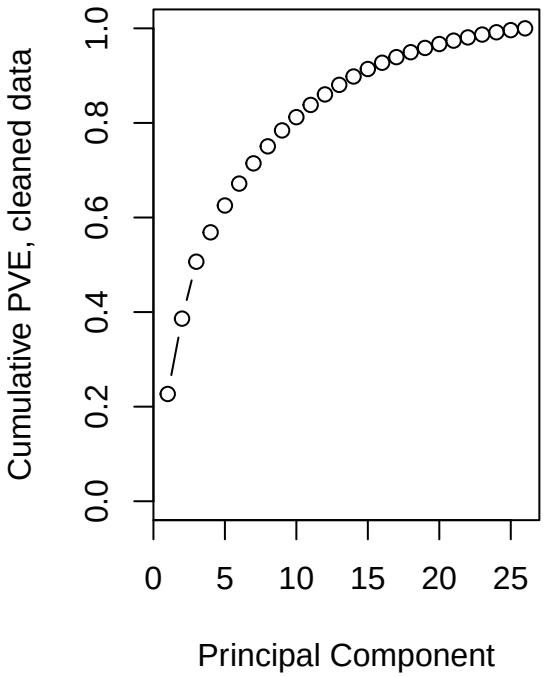
To see how the results are changed after cleaning the data from correlated variables PCA is also done for the cleaned data. The biplot is shown below:



The loading vectors are not as overlapped as previously but the PVE is still quite poor: the two first PCs explains only 38,61 % of the variance.

Again, PVE by each principal component and the cumulative PVE are investigated:





We can see that about 10 first principal components should be used to explain about 80 % of the variance.

The PCA shows that we could reduce the dimensionality of the data from 26 variables (in cleaned data) to some extend. We are going to do that later by using the results of the PCA together with Naive Bayes. In next section we first investigate some other methods which include also feature selections.

Models including feature selection

Some methods include variable selection in itself, like Lasso and decision trees. We investigate logistic regression with Lasso and random forest. The accuracy of the approaches are also calculated as well as perplexity of logistic regression with Lasso.

1) Logistic regression with Lasso

Logistic regression is a discriminant classifier which assumes that the log odds is linear in variables. When combined with Lasso a subset selection of variables are done by adding so called penalty term to residual sum of squares which is minimized in parameter estimation process. The bigger the estimated coefficients of the variables are the bigger the penalty. The penalty term forces some of the coefficients to zero. As we can see from the histograms in the annex the variables are measured in different units and scales. To make sure that the magnitude of a variable does not give too much weight we use data that is normalized to have zero mean and scaled to have unit variance.

The amount of penalty depends on a coefficient called lambda. The value of lambda is selected by cross-validation so that the value of the test error is minimized. For example, in validation approach the selected lambda is

```
## [1] 0.009
```

The estimated coefficients with the selected lambda in training data (with 232 observations) are

term	step	estimate	lambda	dev.ratio
(Intercept)	1	-0.050	0.009	0.69
NO42.std	1	0.009	0.009	0.69
Pamb0.std	1	0.199	0.009	0.69
PTG.std	1	0.263	0.009	0.69
SO2168.std	1	0.069	0.009	0.69
SWS.mean	1	0.582	0.009	0.69
CS.mean	1	-1.712	0.009	0.69
CS.std	1	0.267	0.009	0.69
CO2168.mean	1	-0.527	0.009	0.69
CO2168.std	1	0.250	0.009	0.69
H2O168.mean	1	-0.929	0.009	0.69
NO168.std	1	0.313	0.009	0.69
O3168.mean	1	1.443	0.009	0.69
O3168.std	1	0.230	0.009	0.69
RHIRGA168.std	1	0.379	0.009	0.69
Glob.mean	1	0.956	0.009	0.69

As we can see, a variable selection has been done since only 15 variables have a nonzero coefficient and some of them are still close to zero.

2) Random Forest

Decision trees are learning methods that segment observations into regions. The segmentation is done recursively using binary decision rules which minimize the selected measure like residual sum of squares (RSS) in case of regression tree or e.g. classification error rate or Gini index in case of classification tree. A prediction for an observation belonging to a certain region is given by the mean of responses of the training observations in the same region (regression tree) or by the label to which the majority of the training observations belong to in the same region (classification tree).

Random Forest builds a number of decision trees using each time a sample of training data, sampling done by bootstrapping. Predictions are averages of the resulting trees. During each split only one of the given number of randomly selected variables are considered. By reducing the number of variables the correlation between the trees is reduced.

To get predictions of probabilities of event given the observations when using random forest we change the response variable (class2) to a dummy variable: it gets a value 1 when an event has occurred and value 0 in case of nonevent. Nine variables are considered during each split. The selected number equals to the commonly used one third of the variables in the data in case of regression tree.

Confusion matrix for training data is

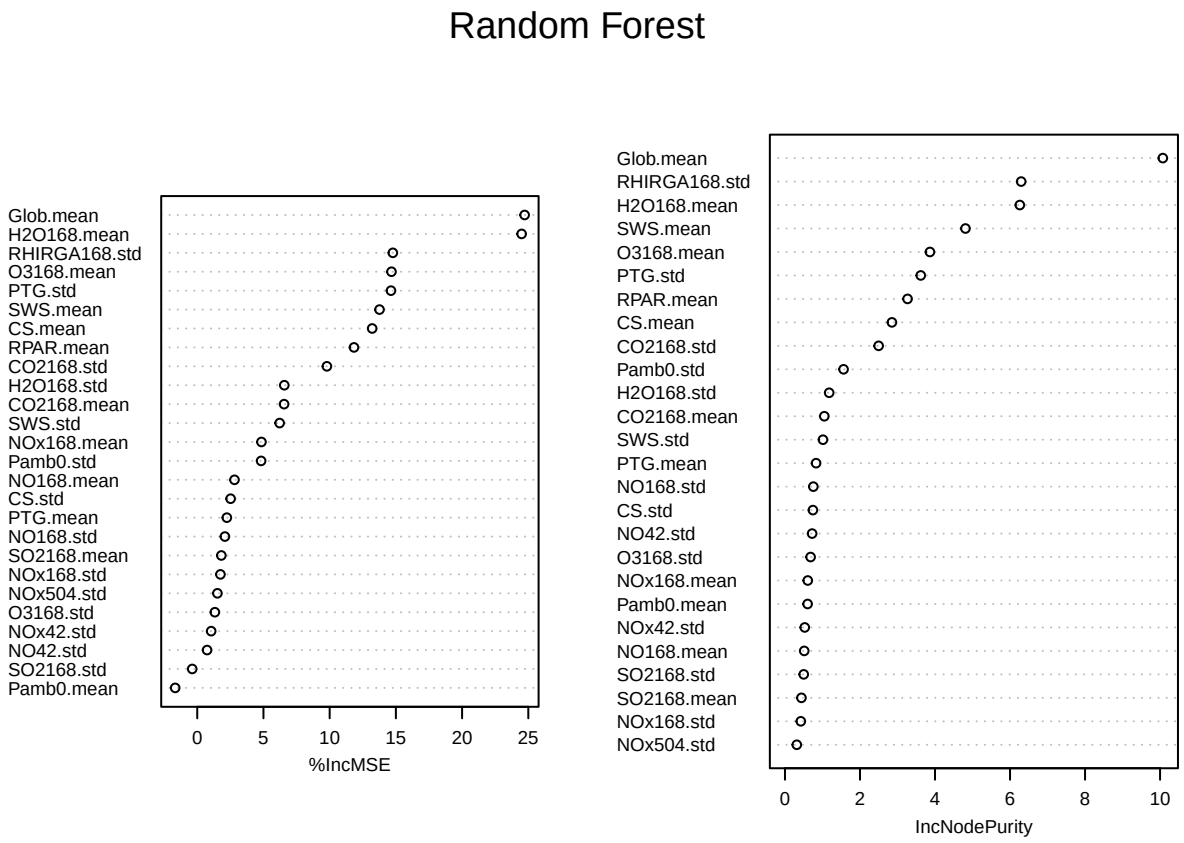
	nonevent	event
0	115	0
1	0	117

As we can see, no missclassification is done in training data. To see how well the model performs on validation set, we look at the confusion matrix for validation data set:

	nonevent	event
0	102	20
1	15	95

There are 20 observations predicted to be event even though no event has occurred and respectively 15 observations given the false prediction of nonevent. All together, the accuracy in validation set is 84,9 %.

The importance of variables are:



The picture above report the values of two variables: -Mean Decrease Accuracy (%IncMSE) that shows how much our model accuracy decreases if we leave out that variable. -Mean Decrease in MSE (IncNodePurity) that is a measure of variable importance based on the MSE.

The higher the value of mean decrease Accuracy or mean decrease Gini score, the higher the importance of the variable to our model. We can see that random forest do some kind of a variable selection since only some of the variables have a significant importance in forming the regions and thus, the predictions. However, the most important variables are not the same as with logistic regression with Lasso.

The performance of the classifiers investigated so far are:

Model	Train Accuracy	Validation Accuracy	CV Accuracy	Train Perplexity	Validation Perplexity	CV Perplexity
Log reg Lasso	0.918	0.828	0.873	1.24	1.343	1.337
Random Forest	1	0.849	NA	1.107	1.389	NaN

Even though the accuracy of random forest in validation set is better than logistic regression with Lasso the performance estimated by cross-validation is poor. Especially, cross-validated perplexity of logistic regression with Lasso seems to be very good.

Other models

The models tested in addition to the logistic regression with Lasso and random forest are a dummy model, Naive Bayes with reduced dimensionality, logistic regression using all variables in cleaned data (to compare the results with the one with Lasso) and k nearest neighbor.

1) Dummy

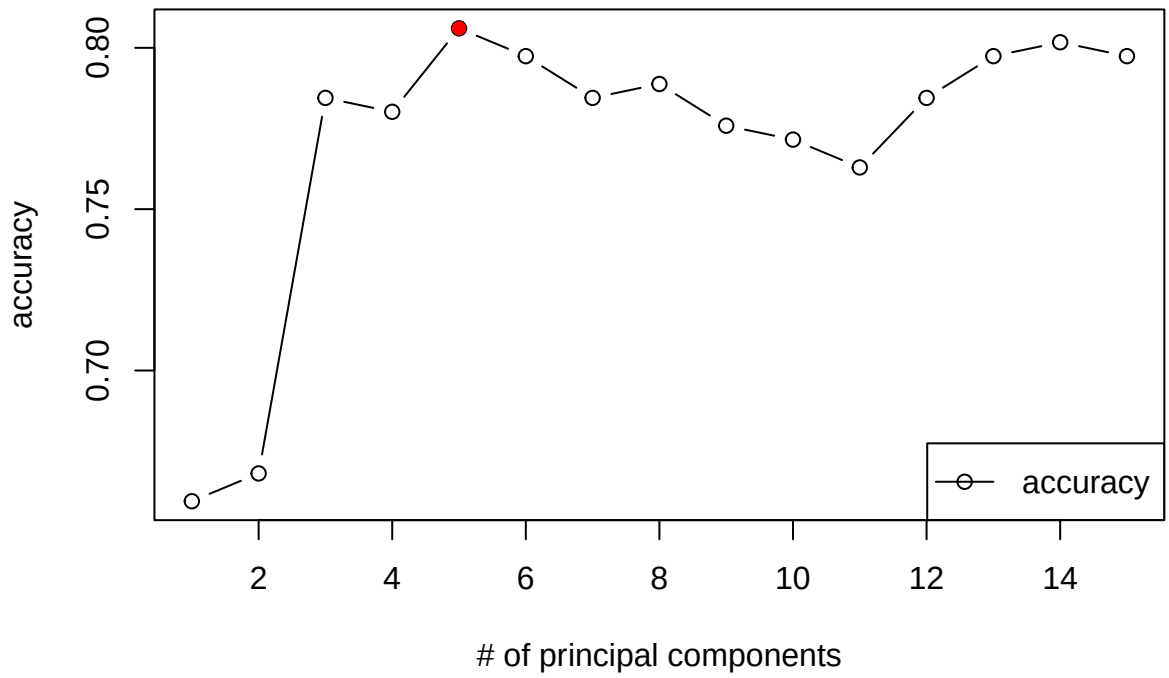
Dummy classifier makes predictions based only on classes or labels and it ignores all the input features of the data. Even though it is not highly performing classifier, it is still a valuable feature as it is usually used as a baseline to compare the performance of the other methods.

In this project the dummy classifier uses the most frequent class of the training set for the predictions. In another method the frequencies of the classes in a training data could be used as the probabilities for choosing between the classes. Some dummy classifiers does not even calculate the frequencies as they predict the classes randomly with a uniform distribution.

2) Naive Bayes

Naive Bayes is a generative classifier which assumes that the variables in each class (event or nonevent) are independent. It can be assumed that this is not the case in NPF data, but Naive Bayes can still produce quite decent results. Naive Bayes assumes that the variables are from Gaussian distribution. Thus, no scaling of the data is needed since it doesn't affect the predictions.

In addition to using Naive Bayes with all 26 variables in the cleaned data, we use Naive Bayes after reducing dimensionality by PCA. This means, that we use the selected number of the principal component score vectors as an input to the model. To decide the number of the PCs to use, we look at the accuracy of the model in validation data with different number of first PCs used (see the picture below). The highest accuracy is marked with red.



The highest accuracy is received when using the five first PCs so this model is added to the comparison of methods.

3) Logistic regression

To compare the results with logistic regression with Lasso we train also logistic regression on the training data with all the 26 variables. We use scaled data to avoid dominance of variables measured in high magnitudes.

4) k Nearest Neighbor (k-NN)

In k-NN the closest k training data points to the observation are selected and the observation is labeled with the class which occurs most often within the k closest training data points (that is, has the largest proportion). Because the distance is used the data needs to be scaled. Parameter k controls the flexibility of the classifier. The smaller the k the more flexible the decision boundary is. With a very small k the classifier can classify the training data more accurately and decrease the training error but this may end up to overfitting.

Different values of k are tried for choosing the optimal k. Accuracy on the validation set for each of the selected k are calculated and are respectively

k	accuracy
1	0.78
5	0.836
10	0.836
15	0.828

k	accuracy
20	0.823
50	0.78

When k is 5 or 10, the accuracy is highest. To avoid overfitting we select k = 10 to be one of the methods to compare with other methods.

Comparison of the methods

Performance measures for the methods are:

Model	Train Accuracy	Validation Accuracy	CV Accuracy	Train Perplexity	Validation Perplexity	CV Perplexity
Log reg Lasso	0.918	0.828	0.873	1.24	1.343	1.337
Random Forest	1	0.849	NA	1.107	1.389	NaN
Dummy	0.504	0.496	0.505	1.983	2.017	2
Naive Bayes	0.853	0.746	0.752	2.106	4.122	7.751
Naive Bayes with PCA	0.875	0.806	0.817	1.46	1.471	1.623
Logistic regression	0.94	0.862	0.886	1.184	1.439	1.369
10-NN	0.897	0.862	0.844	1.278	1.795	1.601

As we already concluded logistic regression with Lasso performs better than random forest. Logistic regression with all 26 variables performs surprisingly well. However, perplexity is better when variable selection is done with Lasso.

Naive Bayes with all 26 variables performs poorly producing high perplexities. When the dimensionality is reduced by using PCA the performance improves significantly but does not outperform logistic regression models. 10-NN performance near the performance of Naive Bayes with PCA.

As expected, the performance of dummy model is worst and is used only as a reference. — This text needs to be updated after the codes are ready!

According to the investigated performance measures logistic regression with Lasso is selected as a classifier.

Summary

We started by looking at the data and especially the correlations between variables. Correlated variables were removed from the data to improve the predictivity of the models. We investigated the importance of the variables left in the data by principal component analysis. The analysis showed that dimensionality could still be somewhat reduced. In stead of further selecting a subset of the 26 variables we used methods which included features of variable selection, like Lasso combined with logistic regression and dimensionality reduction through PCs combined with Naive Bayes. When appropriate we used scaled data to avoid the variables with high magnitudes to dominate the model fitting and predictions.

According to the performance comparison we select logistic regression with Lasso as our classifier. It is used to normalized and scaled data. We noticed that if the unscaled data is used the model will predict only events on testing data. The selected model gives the highest accuracy and smallest perplexity in cross-validation. Pros of this method are that the variable selection is done by Lasso and the fact that logistic regression doesn't make any assumptions of the distribution of the observations in the different classes. We also have enough data compared to the number of variables in cleaned data to train the model properly. By

using Lasso and reduced number of variables we also reduce the possibility of overfitting the model. The case of overfitting is investigated by using cross-validation to estimate the generalization error. One of the disadvantages of the model is the assumption of linear decision surface which is not usually valid. However, the method outperformed for example random forest which can handle more complex relationships. After cleaning the data from correlated variables and using Lasso together with logistic regression we can see from the previous result table that the performance measures of the model are pretty good.

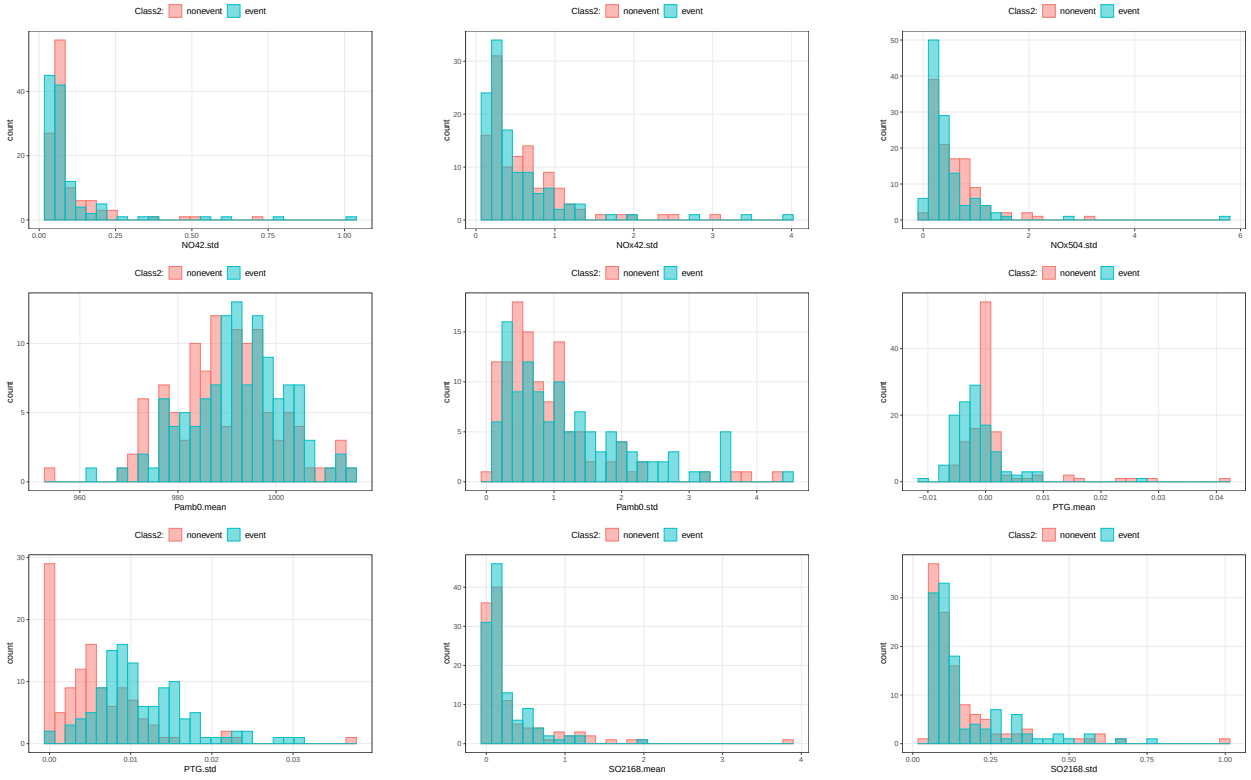
Challenge 11.12.2022

After selecting the model we trained it using the original, cleaned training data (npf_train.csv). The estimates of the probabilities of the observations in testing data (npf_hidden.csv) to be classified as "event" were predicted. If the probability is more than 0.5 the observation is labeled as "event" and otherwise "nonevent". The type of the event was set to be the same as the type that occure mostly in the training data. The table below shows the frequencies of different event types in the training data when the event has happened. The type "II" has the highest frequency of 48,7 %.

The accuracy is the cross-validatd accuracy calculated above and is added at the beginning of the answers.csv after exporting the file with predicted classes and probabilities.

Annex: Histograms of variables

After removing variables with high correlation to other variables 26 variables are left in the data. The histograms give an overview of the values of the variables:




Class2: nonevent event
count
SWS.mean
SWS.std
CS.mean
CS.std
CO2168.mean
CO2168.std
H2O168.mean
H2O168.std
NO168.mean
NO168.std
NOx168.mean
NOx168.std
O3168.mean
O3168.std
RHIRGA168.std
RPAR.mean
Glob.mean