Term Project - Group 80

Sari Ropponen, Outi Boman, Loic Dreano

2022-12-10

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:

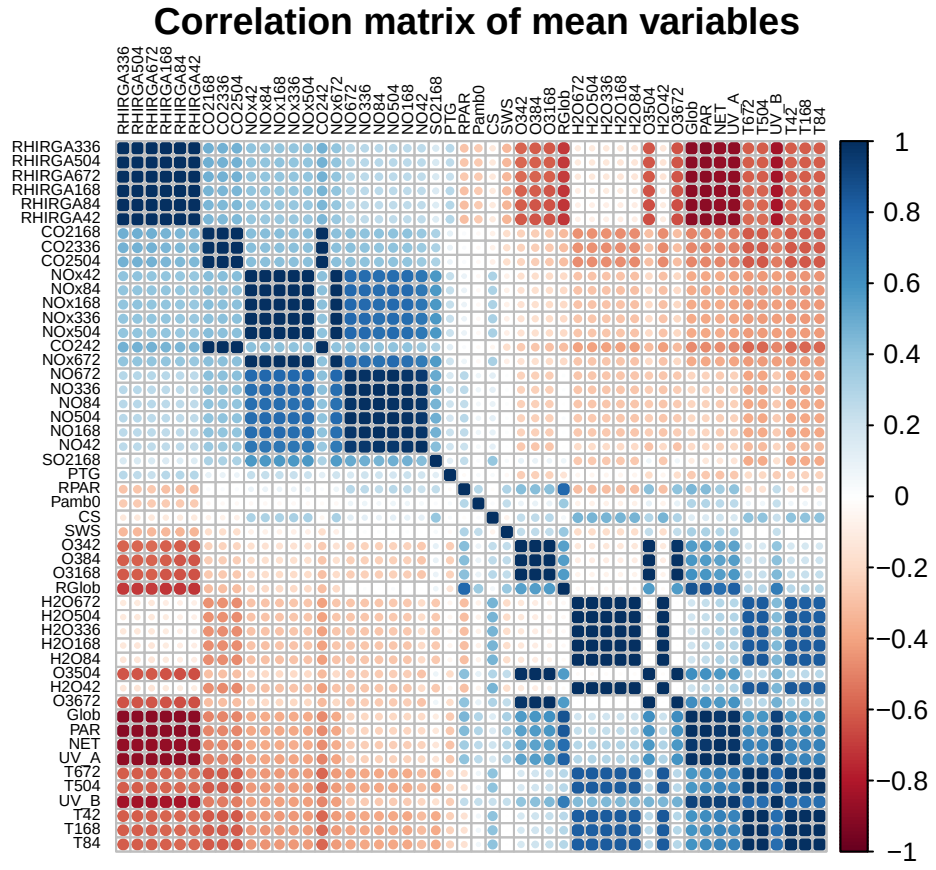
```
##        id            date              class4          partlybad
##  Min.   :  1.0   Length:464         Length:464         Mode :logical
##  1st Qu.:116.8   Class :character   Class :character   FALSE:464
##  Median :232.5   Mode  :character   Mode  :character
##  Mean   :232.5
##  3rd Qu.:348.2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

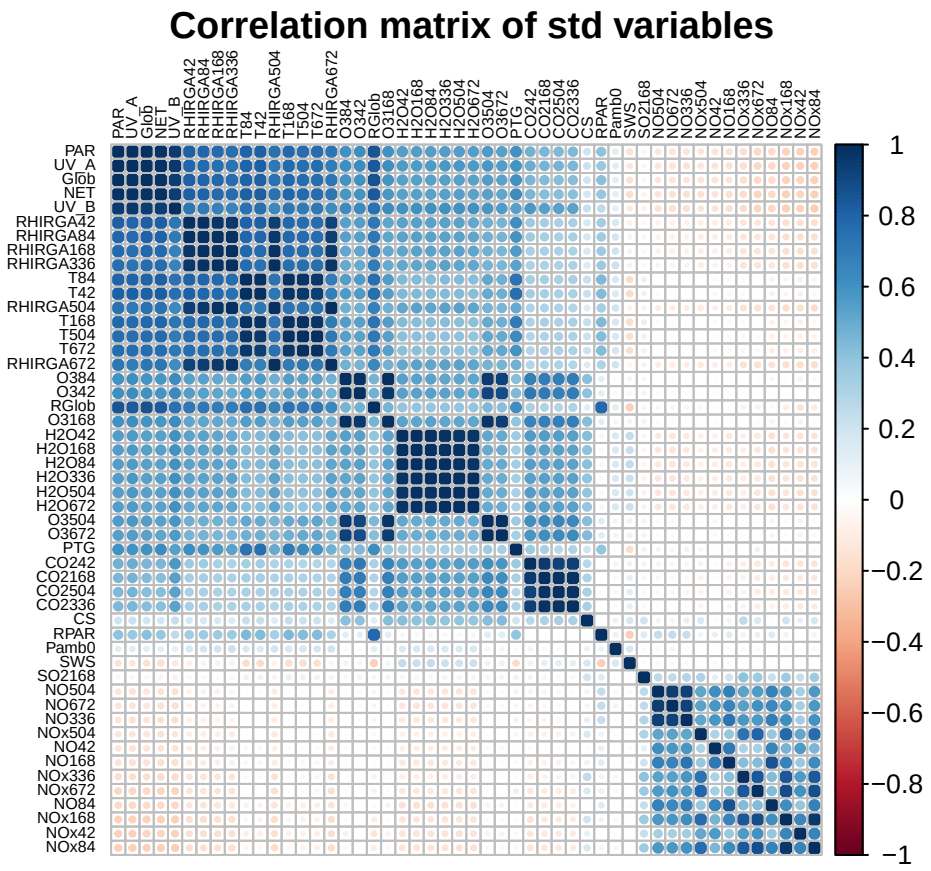
A qualitative variable “class2” is added to the training data. It gets either value “event” or “nonevent” according to “class4”. Variable “class4” indicates the type of the event if it has happened (“Ia”, “Ib” or “II”) or “nonevent” if no event has happened during the day.

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.

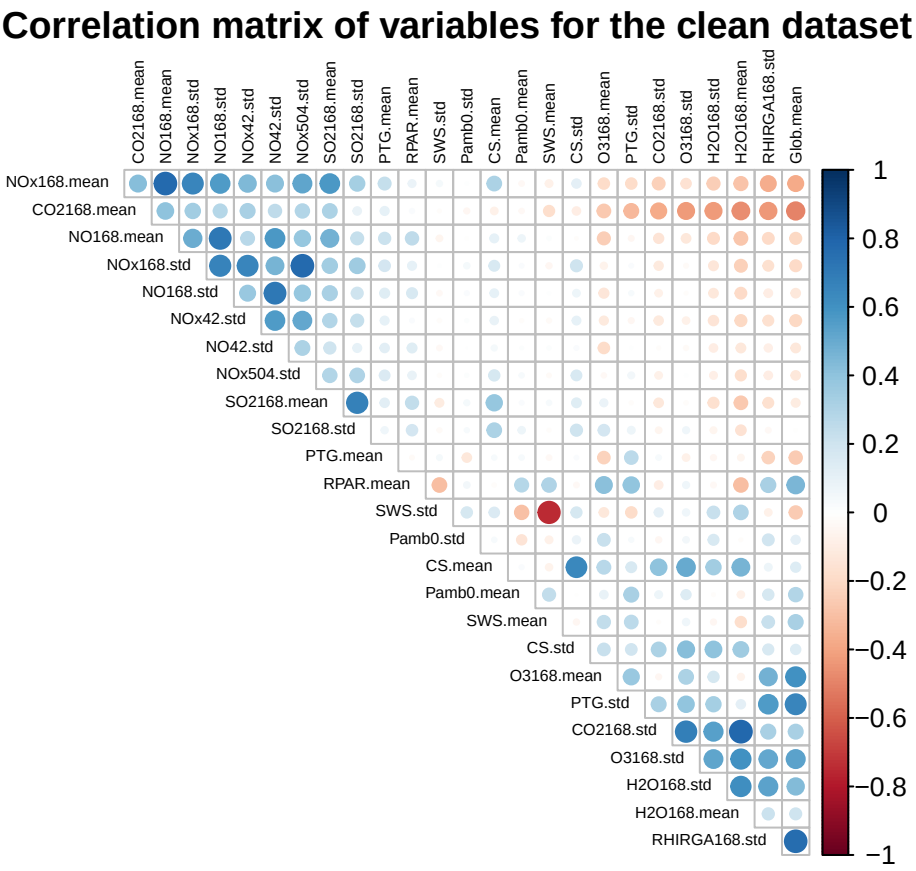


Table 2: List of kept variables

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".

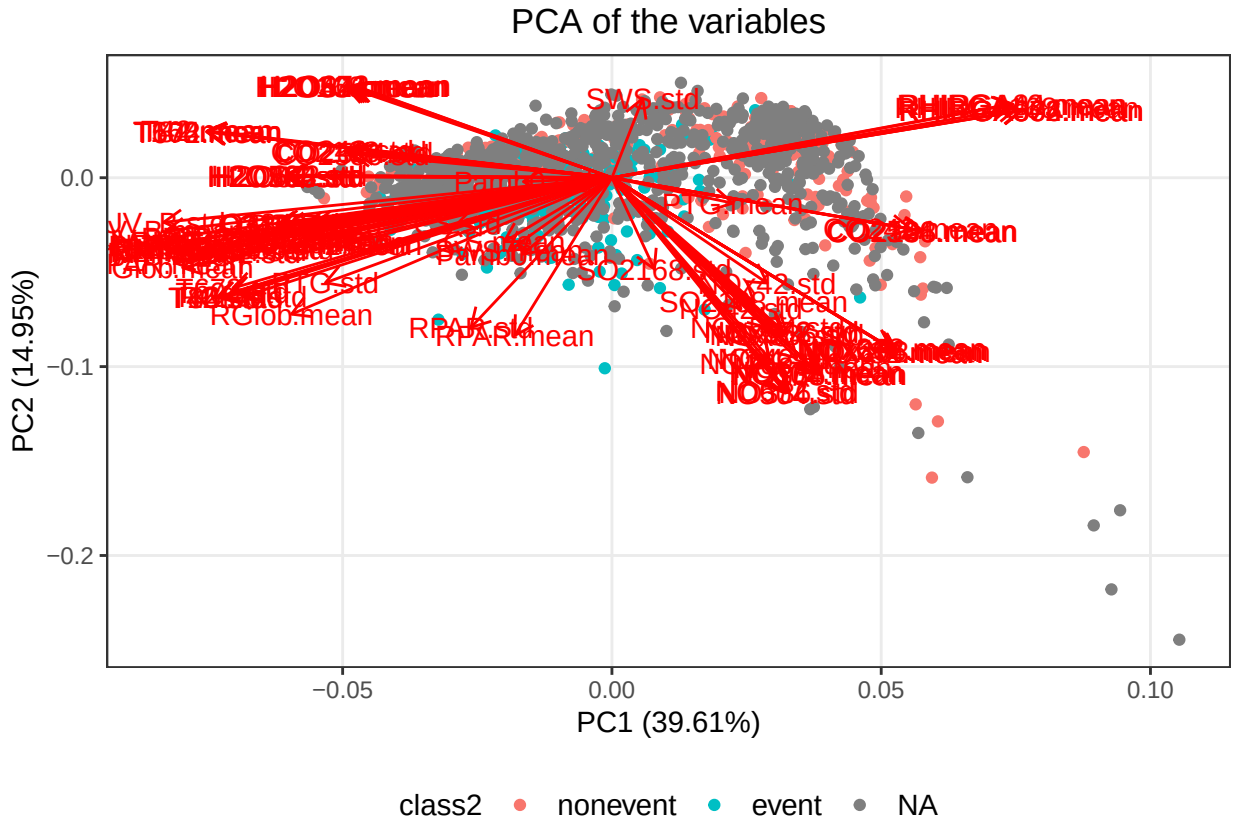
Performance measures are calculated using validation method and 10-fold Cross-Validation. For validation method the training data is randomly divided into 2 equally sized data sets: training data to fit the model and validation data to estimate the accuracy and perplexity. Accuracy and perplexity are 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. The performance measures are the means of the 10 validation results.

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.

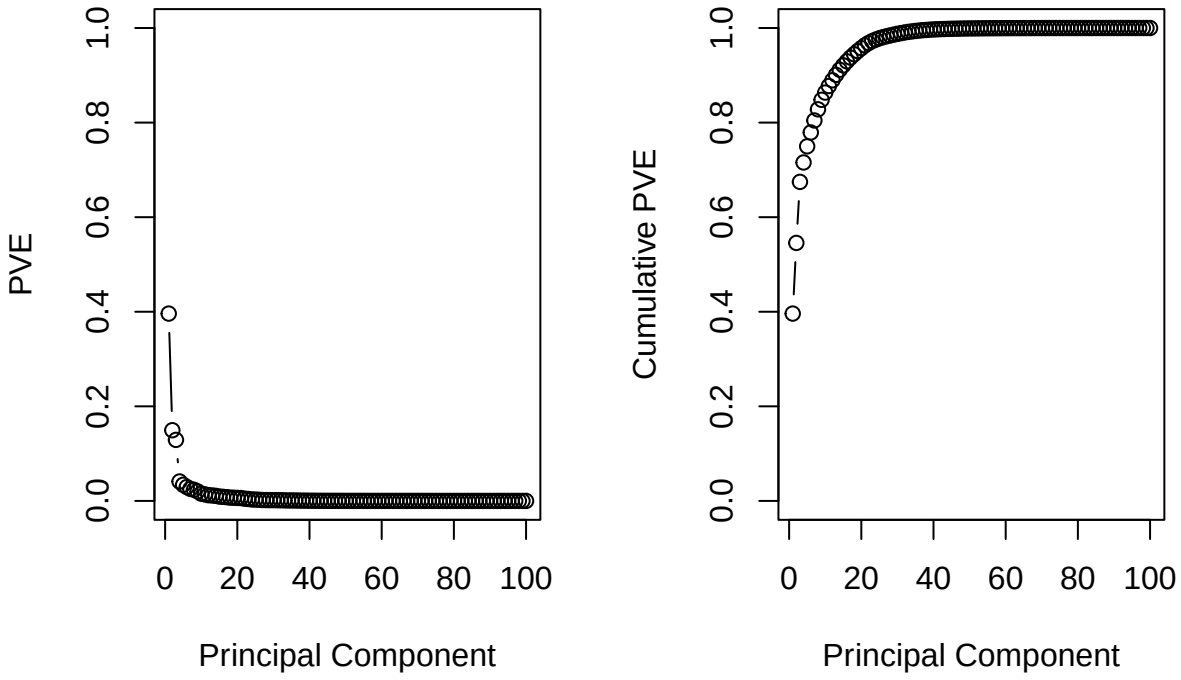
First, we do PCA to the original data before cleaning the correlated variables. The biplot of the analysis is below:



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

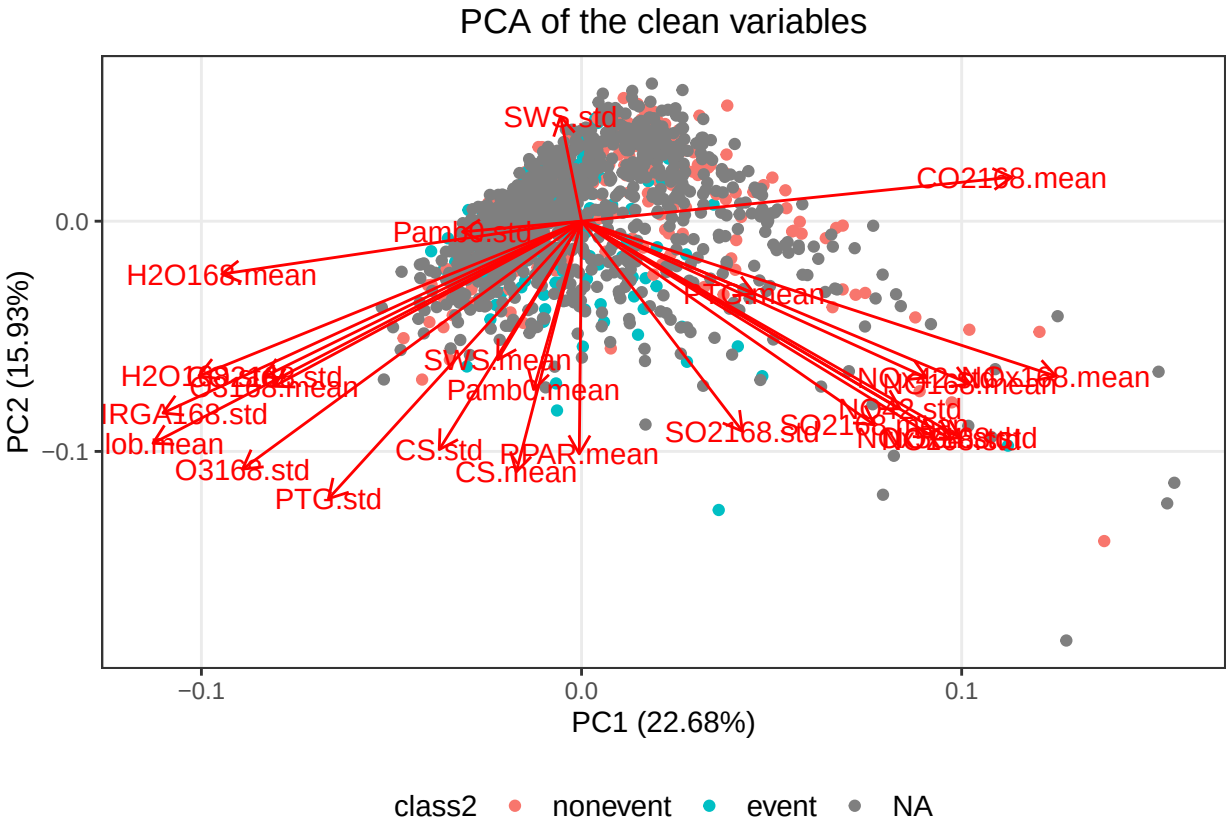
variables and leaving only one of each in the data. We can also see that the first PC (PC1) explains only 39.61 % of the variance in the data and the second PC (PC2) 14,95 %. Together they explain only 54,56 % of the variance which is quite poor result the target being more than 80 %. —Can we conclude from the picture above if there are any significant outliers?

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:



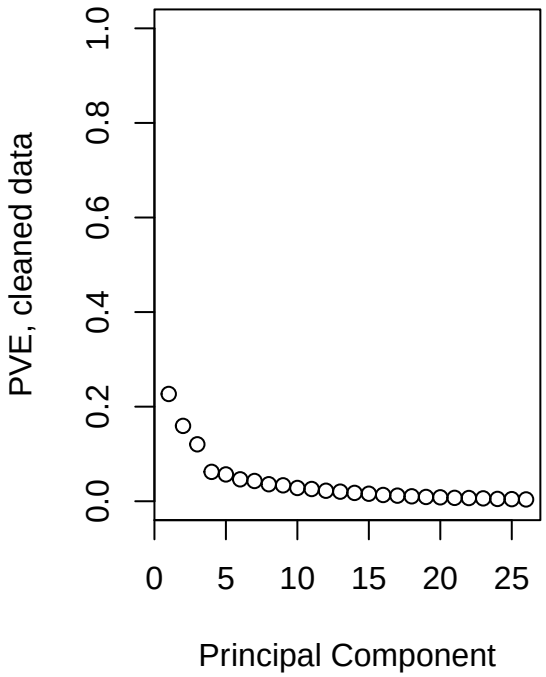
For the original data we would need 10-20 PCs to explain most of the variance.

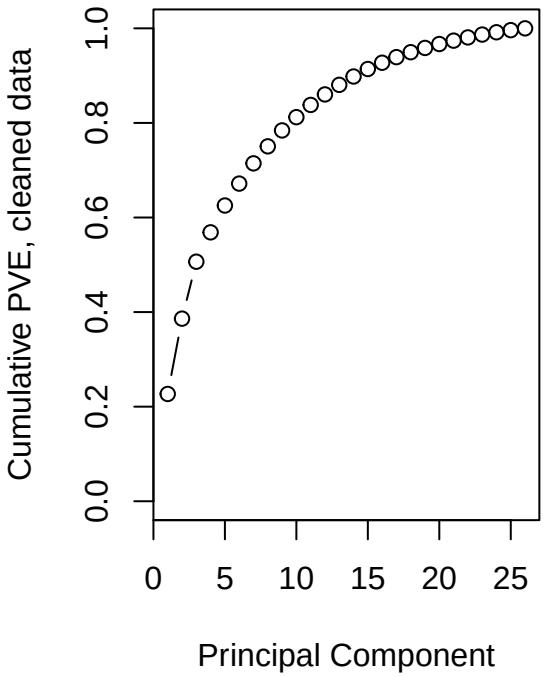
To see how the results are changed after cleaning the data from correlated variables PCA is also done for the cleaned data. The biplot is shown below:



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. 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. The lambda is

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

The estimated coefficients with the selected lambda are

term	step	estimate	lambda	dev.ratio
(Intercept)	1	0.518	0.009	0.69
NO42.std	1	0.074	0.009	0.69
Pamb0.std	1	0.228	0.009	0.69
PTG.std	1	40.885	0.009	0.69
SO2168.std	1	0.495	0.009	0.69
SWS.mean	1	0.015	0.009	0.69
CS.mean	1	-794.265	0.009	0.69
CS.std	1	321.018	0.009	0.69
CO2168.mean	1	-0.047	0.009	0.69
CO2168.std	1	0.078	0.009	0.69
H2O168.mean	1	-0.242	0.009	0.69
NO168.std	1	2.096	0.009	0.69
O3168.mean	1	0.145	0.009	0.69
O3168.std	1	0.095	0.009	0.69
RHIRGA168.std	1	0.066	0.009	0.69
Glob.mean	1	0.008	0.009	0.69

```
## 27 x 1 sparse Matrix of class "dgCMatrix"
##                            s0
## (Intercept)    5.178482e-01
## NO42.std       7.355450e-02
## NOx42.std      .
## NOx504.std     .
## Pamb0.mean     .
## Pamb0.std      2.284548e-01
## PTG.mean       .
## PTG.std        4.088522e+01
## SO2168.mean    .
## SO2168.std     4.954213e-01
## SWS.mean       1.451465e-02
## SWS.std        .
## CS.mean       -7.942649e+02
## CS.std         3.210184e+02
## CO2168.mean   -4.745895e-02
## CO2168.std     7.771859e-02
## H2O168.mean   -2.422650e-01
## H2O168.std     .
## NO168.mean     .
## NO168.std      2.096218e+00
## NOx168.mean    .
## NOx168.std     .
## O3168.mean     1.449695e-01
## O3168.std      9.487847e-02
## RHIRGA168.std  6.634250e-02
## RPAR.mean      .
## Glob.mean      7.590128e-03
```

As we can see, a variable selection has been done since only 15 variables have a nonzero coefficient and many of them are close to zero. Only 4 variables (CS.mean, CS.std, PTG.std and NO168.std) have a value significantly higher than 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 square root of the number of variables in the data.

— in case of classification tree the number of variables should be 26ˆ0.5 and in regression 26/3 => I suppose we have regression now when using dummy response? That's why I changed the number from 5 to 9. Fix the text!!!

Confusion matrix for training data is

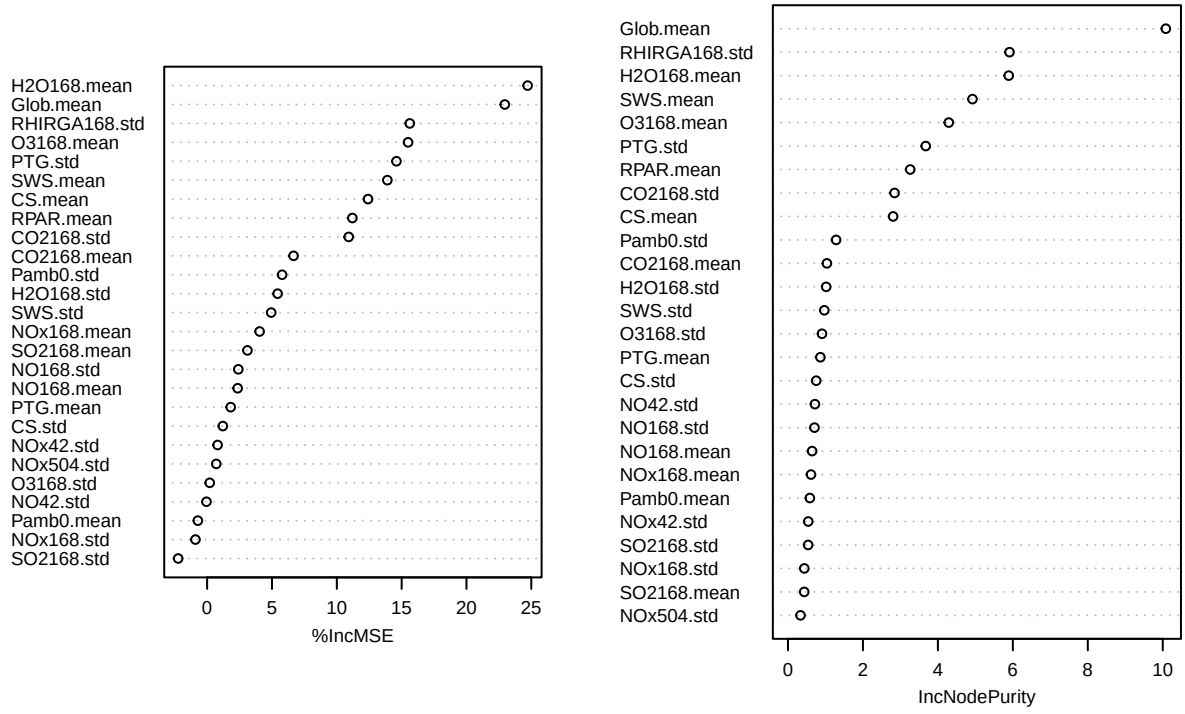
	nonevent	event
0	115	0
1	0	117

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

	nonevent	event
0	101	19
1	16	96

There are 19 observations predicted to be nonevent even though the event has occurred and respectively 16 observations given the false prediction of event. All together, the accuracy in validation set is 84,9 %.

The importance of variables are:



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. — we should add explanations to %incMSE and IncNodePurity and do some conclusions of the pictures

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	0.836	0.881	1.24	1.348	1.319
Random Forest	1	0.849	0.88	1.108	1.383	1.319

Even though the accuracy of random forest in training and validation set is better than logistic regression with Lasso the performance estimated by cross-validation is not as good. Especially, perplexity of logistic regression with Lasso in validation set and cross-validation is better than of random forest.

Other models

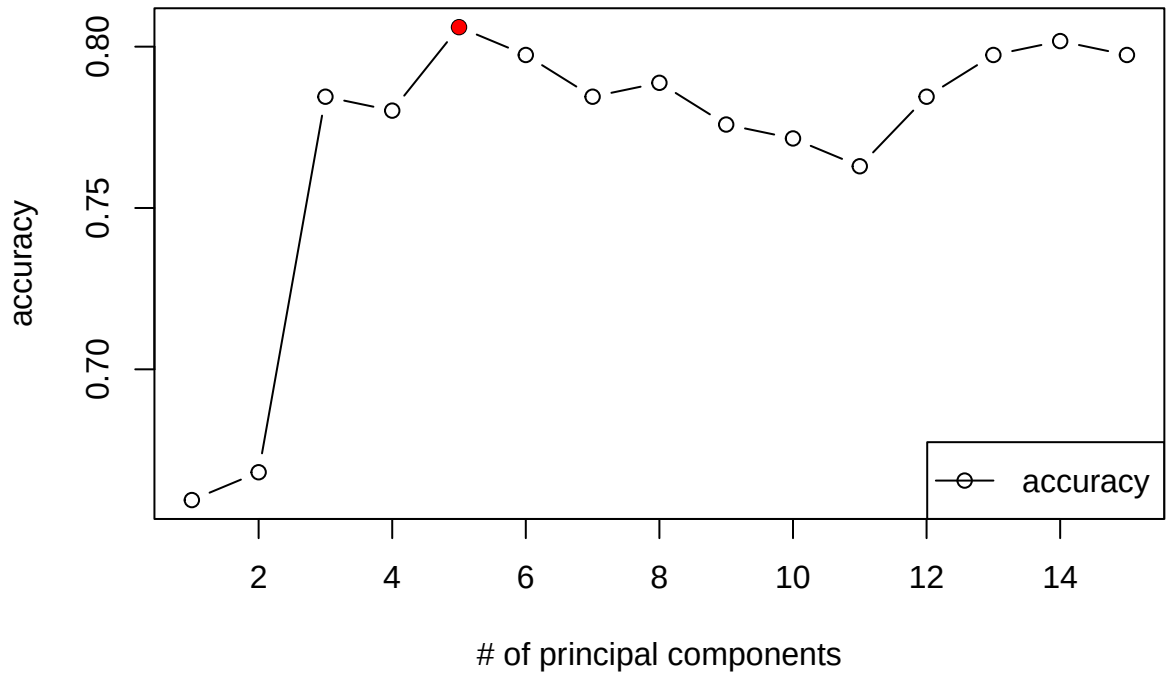
The models tested in addition to the logistic regression with Lasso and random forest are a dummy model, Naive Bayes using reduced dimensionality, logistic regression using all variables in cleaned data (to compare the results) and k nearest neighbor.

1) Dummy: —- write a description of the method and the code itself...

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.

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 — with or without interactions? now it's without — Do we need this at all?

4) k-NN K nearest neighbor is tried with different values of k: 1, 5, 10, 15, 20 and 50. Accuracy on the validation set for each of them are respectively

```
## [1] 0.780 0.836 0.836 0.828 0.823 0.780
```

When k = 5 (or 10), the accuracy is highest so 5-NN is added

Performance measures for the methods are:

Model	Train Accuracy	Validation Accuracy	CV Accuracy	Train Perplexity	Validation Perplexity	CV Perplexity
Log reg Lasso	0.918	0.836	0.881	1.24	1.348	1.319
Random Forest	1	0.849	0.88	1.108	1.383	1.319
Naive Bayes	0.853	0.746	0.753	2.106	Inf	7.597
Naive Bayes with PCA	0.875	0.806	0.816	1.46	1.471	1.627
Logistic regression	0.94	0.866	0.886	1.184	1.411	1.37
5-NN	0.892	0.836	1			

As we already concluded logistic regression with Lasso seems to perform slightly better than random forest. Naive Bayes with all 26 variables performs poorly producing infinitive perplexity in validation data. When the dimensionality is reduced using PCA the performance improves significantly but does not outperform logistic regression with Lasso. 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!!!

Summary

We started by looking at the data and especially the correlations between variables. — summary of data investigations should still be completed...

— The following text is just an example what to write if we happen to choose log reg with Lasso :) According to the performance comparison we select logistic regression with Lasso as our classifier. It gives the highest accuracy and smallest perplexity in cross-validation. Pros of this method is the variable selection 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. 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 table that the performance measures of the model are pretty good.

— should we repeat the performance of the selected model here?

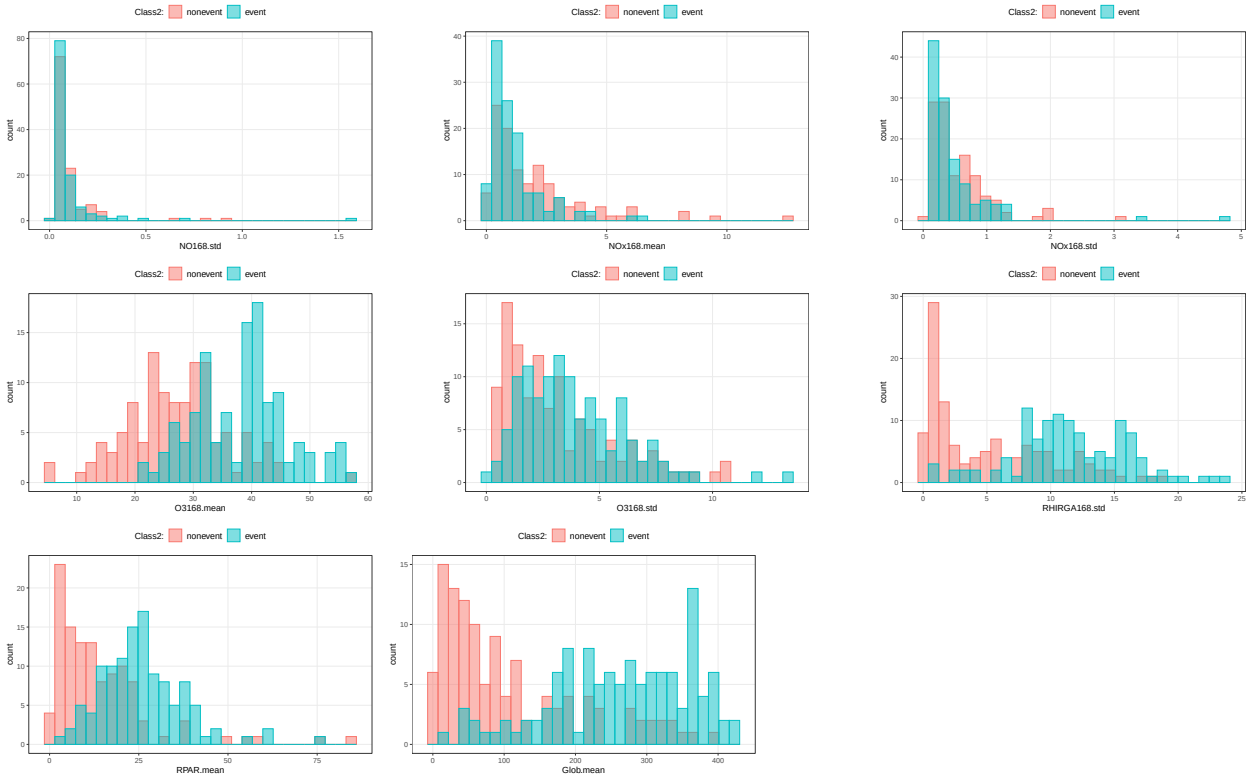
Challenge 11.12.2022

— After selecting the model we should train it using the npf_train.csv after cleaning (npf.all.clean) — Then we should calcluate CV Accuracy and predict probabilities in npf_hidden and give labels to the data — we need to convert class "event" to one of the types Ia, Ib or II - we should use the one that has the highest probability in the data (npf_train.csv). — I have started the code and have used log reg with lasso just to get an idea what should be done. We can change the model still :)

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
NO42.std
NOx42.std
NOx504.std
Pamb0.mean
Pamb0.std
PTG.mean
PTG.std
SO2168.mean
SO2168.std
SWS.mean
SWS.std
CS.mean
CS.std
CO2168.mean
CO2168.std
H2O168.mean
H2O168.std
NO168.mean



END

Some testing and old codes (keep or remove?)

Testing the CV code

if perplexity should be calculated in each iteration in CV and take an average of the 10 results. Test is done to logistic regression on selected data. CV Perplexity in the result table is 1.274 and now the result is

Should we remove this old chapter: Conclusions and feature selection

— !!! This whole part can be removed if it's not needed. The text is not updated after cleaning the data so at least the conclusions should be updated. The models in the next session is set to use cleaned data instead of the selection previously done here. !!!

The accuracy is best for Random Forest, but also Logistic Regression with Lasso where lambda is selected using CV ("log reg CV") performs very well. —- Should we investigate diagnostic plots for log reg? Should be give weight to well performed classifiers when selecting the variables?

All models use the following variables - RHIRGA: mean is more important than std according to tree and RF - H2O mean - O3 mean (Logistic Regression with Lasso, lambda = CV ("log reg CV") gives small value to std) - CO2: tree and RF uses only std, log regs give value to mean too - T std - SWS.mean

The following variables are used in all other models but RF - NO.std - Pamb0.mean or .std

The following variables are used in all other models but "normal" tree - CS.mean (.std only in log reg CV)

These variables are used in some models - UV_B.std is used only on log reg CV - UV_A.mean is used only in RF - RGlob.mean and RGlob.std (RF, mean also in log reg 0.1) - PTG.std (tree, RF) - NET.mean - RPAR.mean - SO2

Because the measures in different heights/levels are highly correlated, we select just one of them. —- which ones? Should we put more weight to the selections in random forest and log reg CV? —- Should be produce boxplots, histograms and/or scatterplots to selected variables

We select the following variables to be used in models: —- selected for testing the same variables than in Log Reg Lasso CV but only once if there are multiple values with different heights ————

Let's check if there are any correlation left:

Do we remove this: A basic decision tree

2) A classification tree selects the following variables with the misclassification as follows

Accuracy of the tree can be calculated from the confusion matrix. Predicted classes vs. actual classes for validation data:

Predicted classes vs. actual classes for training data: — this can actually be seen from the results above (summary(tree.npf)) - do we want to leave this confusion matrix away?

The accuracies are respectively

Do we need the CV-code from problem 2?