Report on Dimensionality reduction of a weather dataset

by Giuseppe Insana, February 2022

The dataset

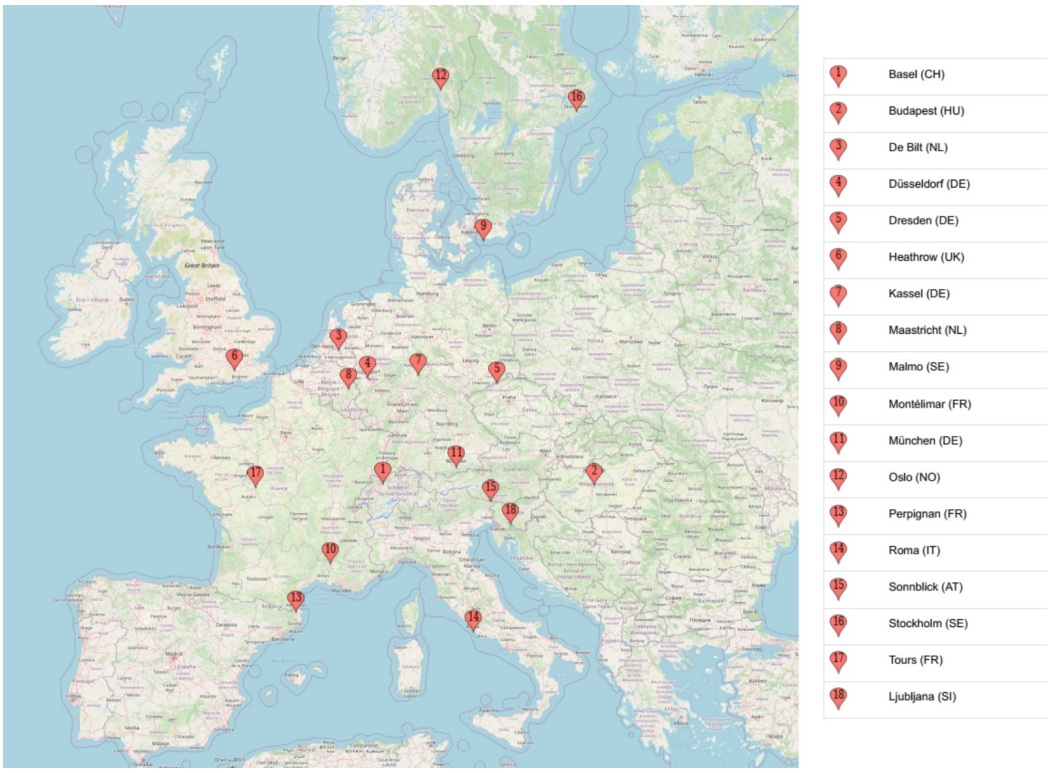
I chose to work on weather data, chiefly because I never worked on this kind of data before (having mostly dealt with biological and linguistic data in the past).

The weather data analysed in this report has been created by Huber Florian from ECA&D data (see section on Data origin at the end for references).

It contains daily weather observations from 18 different European weather stations through the years 2000 to 2010.

The description of the data set says that the minimal set of variables 'mean temperature', 'max temperature' and 'min temperature' are available for all locations. An additional number of measured variables ('cloud_cover', 'wind_speed', 'wind_gust', 'humidity', 'pressure', 'global_radiation', 'precipitation', 'sunshine') are provided, but not for all the locations.

The following map shows the locations which are included in the dataset:



Objective of the analysis

We have recently used this dataset for prediction of tomorrow's weather for one location based on today's weather measures across all locations. During EDA of the dataset we previously observed high correlation among the features.

This is to be expected, both for different measures from the same location (e.g. sunshine hours and global_radiation, or temp_mean and temp_max) and also for measures at different locations; with

our hypothesis being that the closer two locations are, the more correlated their weather measures should be.

The plan for of our analysis is the following:

1) we will explore further the correlation among features

2) we will apply Principal Component Analysis on the dataset and use a series of metrics to understand what proportion of the total variance is explained by PCA in relation to the number of components

3) we will see how adding PCA as part of a ML regression pipeline can affect the prediction accuracy, thus getting another measure of the information loss incurred by dimensionality reduction

4) we will then try to apply multidimensional scaling to see whether we can use the correlation among the features as a way to recover the relative position and distance of the weather locations.

Data exploration

A comprehensive exploratory data analysis has been previously conducted on the model and presented in a previous report (attached as a separate document, see file **EDAcourseproject_report_Giuseppe_Insana.pdf**). We will here briefly summarise the findings of that report to give a description of the data we will process and to show the rationale behind the actions we will take.

We have analysed the types and amount of the available data, checking ranges and distributions of all observations.

Data attributes

The original data is loaded into a pandas dataframe which has 3654 rows (one per each day) and 165 columns:

- DATE, with integer values from 20000101 to 20100101, corresponding to interval from Jan 1st 2000 to Jan 1st 2010
- MONTH, integer 1 to 12
- and another 163 columns for the weather measurements at the different locations.

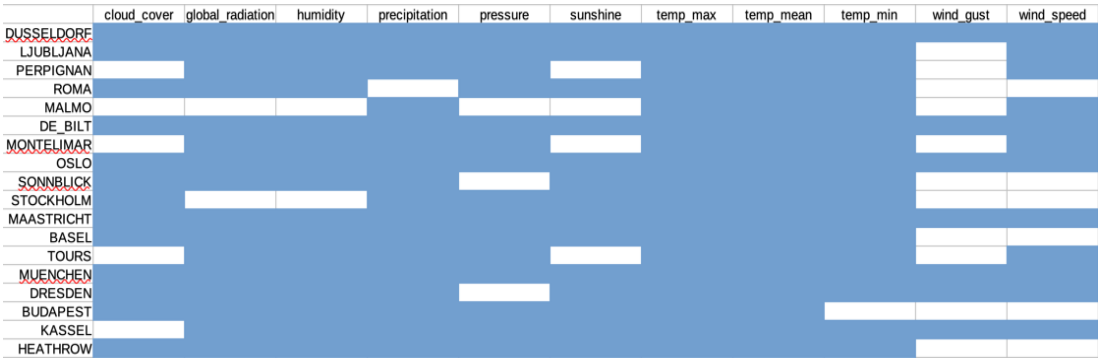
The measurements are labelled as LOCATION_*measure* (e.g. BASEL_cloud_cover, BASEL_pressure, OSLO_precipitation...) with the measured variables being: **cloud_cover, global_radiation, humidity, precipitation, pressure, sunshine, temp_max, temp_mean, temp_min, wind_gust, wind_speed**

All the measurements are loaded as floating point numbers, with the exception of cloud_cover, loaded as integer.

The **physical units** for the variables are described as follows:

> **cloud_cover** in oktas; **wind_speed** and **wind_gust** in m/s; **humidity** in fraction of 100%; **pressure** in 1000 hPa, **global_radiation** in 100 W/m^2; **precipitation** in 10 mm; **sunshine** in 1 Hours; **mean max** and **min temperature** in Celsius degrees.

The following table shows which measures are available for which location:



Null and Out-Of-Range values

The dataset does not contain missing values per se but there are several out-of-range values that can be considered as Null/Missing/Invalid observations:

cloud_cover measures should vary from 0 (sky completely clear) to 8 (sky completely clouded) oktas. But the data contains two data points (both for Stockholm, 20080724 and 20090625) with a value of -99 and one (again Stockholm, 20031108) with a value of 9

pressure: the data contains three entries (again for Stockholm, 20071008, 20000124, 20070603) with value of -0.099 and one entry (Tours, 20081230) with value of 0.0003. These out of range values can again be considered invalid and a decision should be taken for them akin to those mentioned before for cloud_cover.

sunshine: the data contains 29 negative values for hours of sunshine (again for the Stockholm location) which should be treated as invalid/null and dealt appropriately (as mentioned for cloud_cover and pressure). For the location of Oslo there are 24 measurements with more than 18 hours of sunshine, with 20 of them being 24h. While the northern latitude make very long daylight possible, this is for almost 18 hours in midsummer, while these huge values are from Nov-Dec 2006. We must hence treat these as wrong invalid data as well.

Distribution of values

After imputing the out of range values (shown below in the section on Data cleaning and feature engineering), the range, mean and standard deviation for all measures across all locations are the following:

measure	mean	std	range	
cloud_cover:	5.14	2.33	0.00 .. 8.00	*(okta)*
global_radiation:	1.37	0.95	0.01 .. 4.42	*(i.e. 1 to 442 W/m2)*
humidity:	0.75	0.14	0.10 .. 1.00	*(i.e. 1% to 100%)*
precipitation:	0.23	0.58	0.00 .. 16.04	*(i.e. 0 to 160.4 mm)*
pressure:	1.02	0.01	0.96 .. 1.05	*(i.e. 959 to 1016 hPa)*
sunshine:	5	4.41	0.00 .. 17.80	*(hours)*
temp_max:	14.5	9.58	-24.70 .. 41.10	*(°C)*
temp_mean:	10.39	8.41	-26.60 .. 33.10	*(°C)*
temp_min:	6.33	7.58	-30.30 .. 26.30	*(°C)*
wind_gust:	10.06	3.88	1.50 .. 41.00	*(m/s)*
wind_speed:	3.33	1.89	0.00 .. 16.30	*(m/s)*

The dataset was thoroughly explored visually by way of plots, to see the actual distribution of values and to gather insights, plotting the measured features as a whole or grouped spatially or temporally. Again the reader is invited to check the EDA report for the in-depth analysis.

Major findings:

- strong dependence of weather measures by month and by location (obviously) for both ranges, mean and variance; for example pressure has smaller variance in summer months compared to winter months;
- the seasonal component appears to be responsible for multimodality in certain features;
- there are also year to year variations, with for example some years being on average warmer or colder; no overall trend was observed but this is probably due to the scale of the dataset (only ten years period);
- precipitation and wind_speed are the most skewed measures (more than 0.75 skew value) across all locations and for some locations humidity and cloud_cover as well. The most strongly skewed features are:

DRESDEN_precipitation	13.077
PERPIGNAN_precipitation	10.781
MONTELIMAR_precipitation	7.479
BUDAPEST_precipitation	5.662
MALMO_precipitation	5.337
MUENCHEN_precipitation	5.206
BASEL_precipitation	4.529
STOCKHOLM_precipitation	4.481
TOURS_precipitation	4.233
LJUBLJANA_precipitation	3.828

Correlations

The correlation between features was analysed, both for measures in a single location or across different locations.

The major insights were:

- related variables have (as expected) very high correlation:
  - temp_mean with temp_min and temp_max
  - wind speed and wind gust
- global radiation has high positive correlation with sunshine
- global radiation and sunshine correlate negatively with humidity
- precipitation appears to have extreme values concentrated where pressure has average values
- locations nearby have measures more highly correlated compared to locations geographically distant

More will be said about this aspect in Correlated features.

Data cleaning and feature engineering

Out of range values

To clean the dataset we first dealt with the **out-of-range values** identified for the cloud_cover, pressure and sunshine measures.

The number of these invalid values appear in 1.64% of the total rows but as they only affect one location at a time they constitute only 0.01% of the total values. In the previous report we **recommended to not drop** the whole days' measurements but **instead to impute** the invalid values.

To do so we wrote code which gathers the values for the involved measure and location on the 5 days before and 5 days after the date of the out-of-range value. These values are then averaged and the average is used to impute the invalid value. When the invalid values appeared in succession the strategy is modified to use the window of 10 days (5 before and 5 after) but for the year before.

For example, the code identifies the above mentioned out-of-range values for pressure:

idx	DATE	STOCKHOLM_pressure	TOURS_pressure
23	20000124	-0.0990	1.0234
2710	20070603	-0.0990	1.0205
2837	20071008	-0.0990	1.0242
3286	20081230	1.0328	0.0003

and proceeded to impute them as following:

```
 ** Imputing 3 value(s) for column STOCKHOLM_pressure
 idx23: from -0.099 to 1.00449 using mean data from range 18-29
 idx2710: from -0.099 to 1.02079 using mean data from range 2705-2716
 idx2837: from -0.099 to 1.0203099999999998 using mean data from range 2832-2843

** Imputing 1 value(s) for column TOURS_pressure
idx3286: from 0.0003 to 1.02485 using mean data from range 3281-3292
```

Outliers

The **main outliers** identified in the EDA report were:

- the very low values for humidity, with majority of outliers from Sonnblick and (in much lower proportion) Perpignan
- the precipitation extremes
- the highest recorded wind_speed measures (which are almost all from Perpignan location)

These could represent extreme weather conditions, which may or may not hinder the prediction abilities of the ML models.

Similarly to what said for out-of-range values, the outliers account for a negligible portion of the dataset but non-negligible number of rows. Thus it would be best to impute the single values (using the same strategy as above, averaging over a window of days) rather than excluding the whole rows.

As previously observed during regression and classification work on this dataset, imputing these outliers made negligible difference to the trained models.

Feature engineering

The original **dataset does not contain categorical data** which would need to be converted (e.g. by one-hot encoding).

In the pipelines we will use, we will be adding **transformations** (transforming the most skewed variables outlined above) and **scaling** of the features.

The DATE information is dropped from the dataset, leaving 164 columns (MONTH and weather measures across locations).

Correlated features

In previous reports we observed how the MONTH information, although obviously strongly connected to the weather measures which follow seasonal patterns (in particular at the latitudes of the locations of this dataset), was not very helpful for regression or classification.

Checking the correlation of this column with the weather measures show that mostly it is correlated to temperature and humidity measures, but still with a maximum absolute correlation of 0.29. This is to be expected, as the weather during a month could vary quite a lot. The top correlations of weather measures with MONTH are:

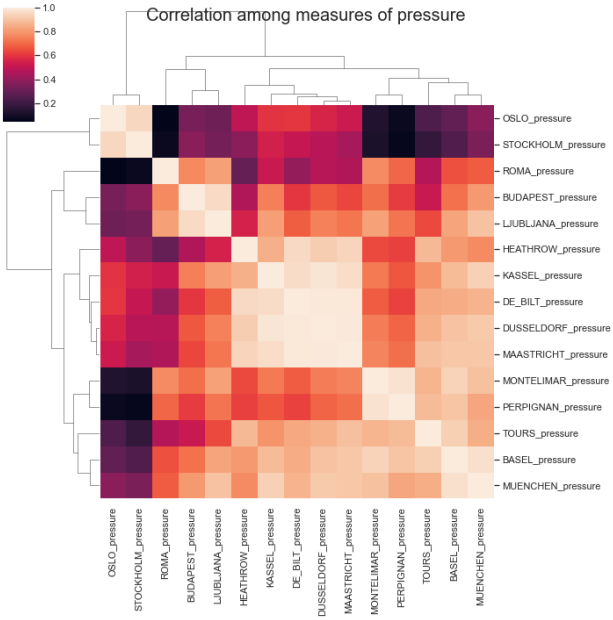
	MONTH
MALMO_temp_min	0.2949
STOCKHOLM_temp_min	0.2897
ROMA_temp_min	0.2828
ROMA_temp_mean	0.2739
SONNBLICK_temp_min	0.2716
SONNBLICK_temp_mean	0.2653
MALMO_temp_mean	0.2604
BASEL_humidity	0.2592
SONNBLICK_temp_max	0.2519
LJUBLJANA_temp_min	0.2421
STOCKHOLM_temp_mean	0.2409
OSLO_temp_min	0.2404
MONTELIMAR_temp_min	0.2349
HEATHROW_temp_min	0.2322
ROMA_temp_max	0.2305
PERPIGNAN_temp_min	0.2277
DRESDEN_temp_min	0.2218
DE_BILT_humidity	0.2217
KASSEL_temp_min	0.2208
PERPIGNAN_temp_mean	0.2197

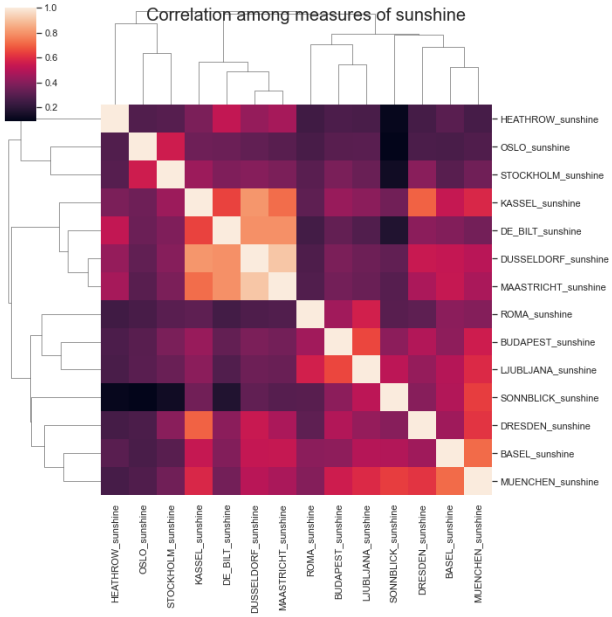
Very high correlations (even up to 0.99) are usually between measures within the same location or for the same weather measure across nearby cities, as seen in the following table showing the 20 highest correlated columns:

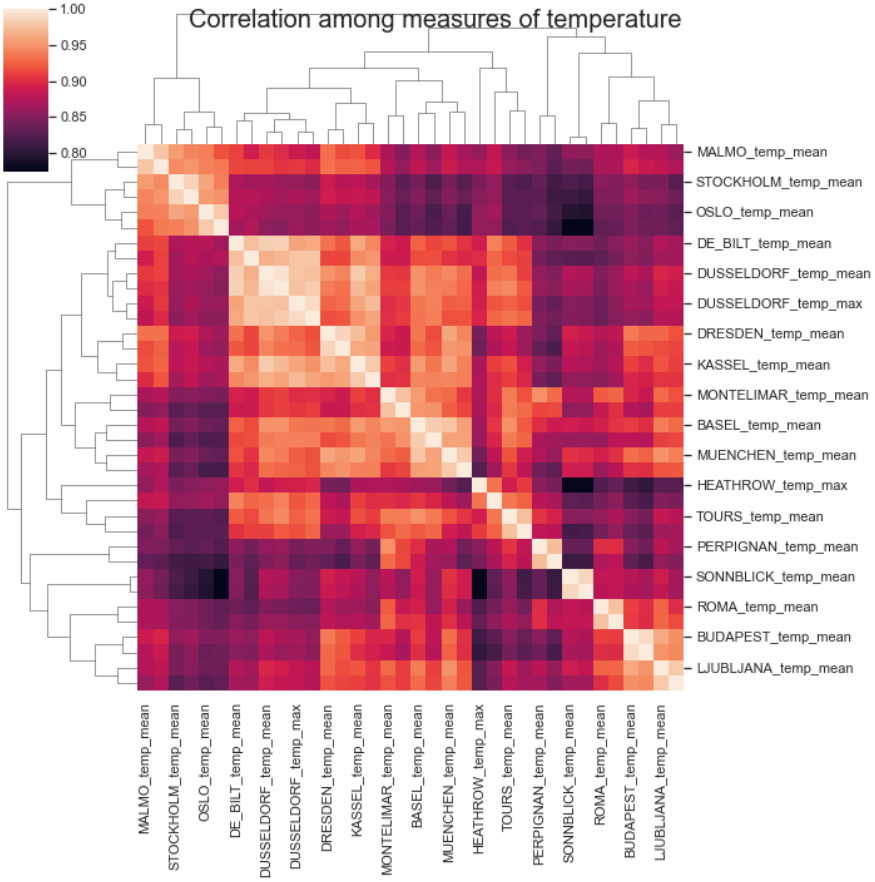
	max_abs_corr	max_corr_with
DUSSELDORF_pressure	0.9955	MAASTRICHT_pressure
DUSSELDORF_temp_mean	0.9941	MAASTRICHT_temp_mean
DUSSELDORF_temp_max	0.9902	MAASTRICHT_temp_max
BUDAPEST_temp_mean	0.9899	BUDAPEST_temp_max
DE_BILT_pressure	0.9891	DUSSELDORF_pressure
SONNBLICK_temp_mean	0.9888	SONNBLICK_temp_min
DRESDEN_temp_mean	0.9867	DRESDEN_temp_max
STOCKHOLM_temp_mean	0.9858	STOCKHOLM_temp_max
DE_BILT_temp_mean	0.9839	MAASTRICHT_temp_mean
MAASTRICHT_temp_mean	0.9815	MAASTRICHT_temp_max
BASEL_temp_mean	0.9809	BASEL_temp_max

	max_abs_corr	max_corr_with
MUENCHEN_temp_mean	0.9806	MUENCHEN_temp_max
OSLO_temp_mean	0.9805	OSLO_temp_max
KASSEL_temp_mean	0.9803	KASSEL_temp_max
LJUBLJANA_temp_mean	0.9798	LJUBLJANA_temp_max
MALMO_temp_mean	0.9778	MALMO_temp_max
DE_BILT_temp_max	0.9773	MAASTRICHT_temp_max
DUSSELDORF_temp_min	0.9761	MAASTRICHT_temp_min
ROMA_temp_mean	0.9758	ROMA_temp_max
MONTELIMAR_temp_mean	0.9754	MONTELIMAR_temp_max

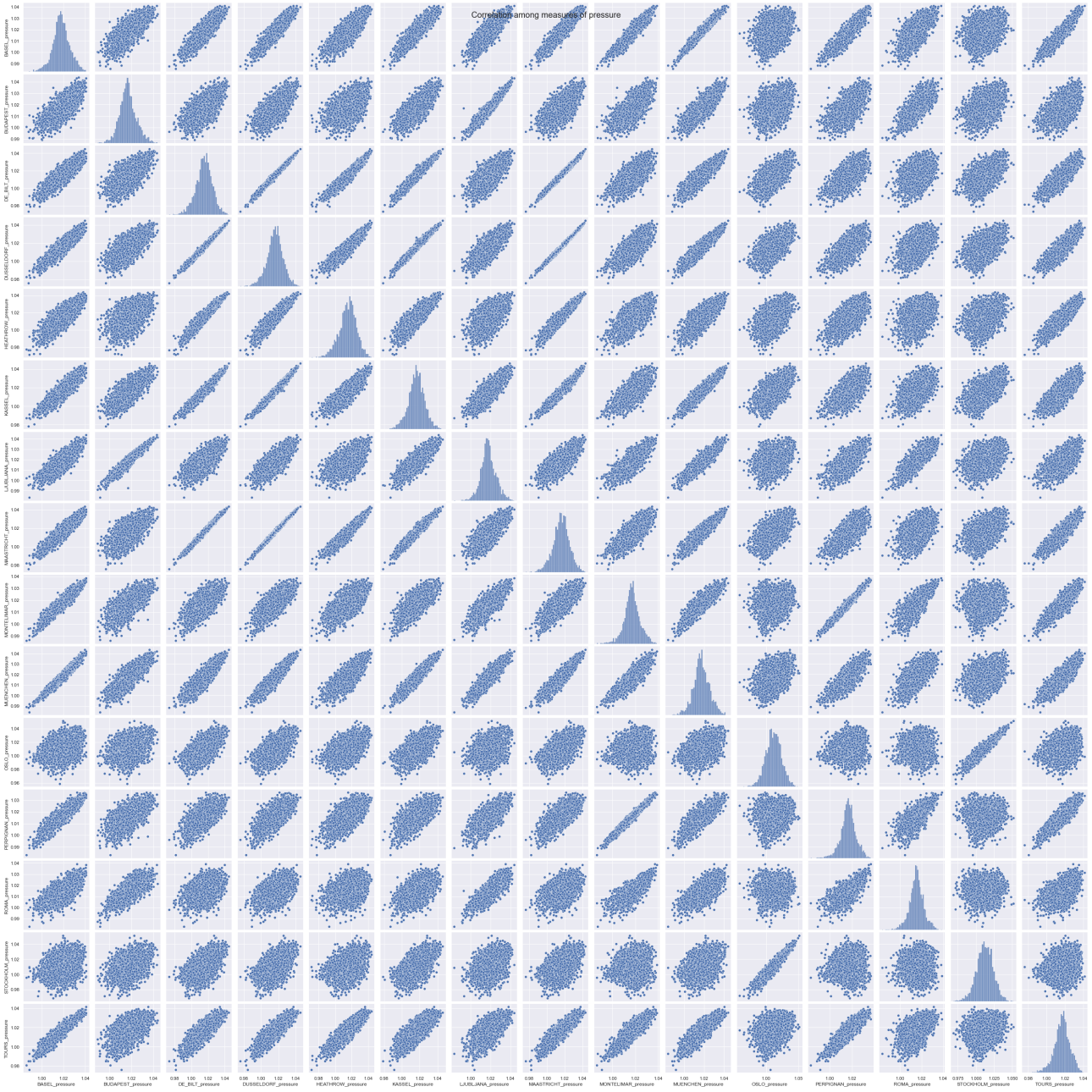
Checking the correlation of the same measure across all the locations one can start to discern a pattern where locations close in space appear close in the dendrogram based on the correlation matrix:







This is also made visible by pairplots, which show high correlation across nearby locations, in this plot computed for the pressure measure:



With that in mind, we think PCA should be able to reduce the number of dimensions of the dataset without losing too much of the variance in the data.

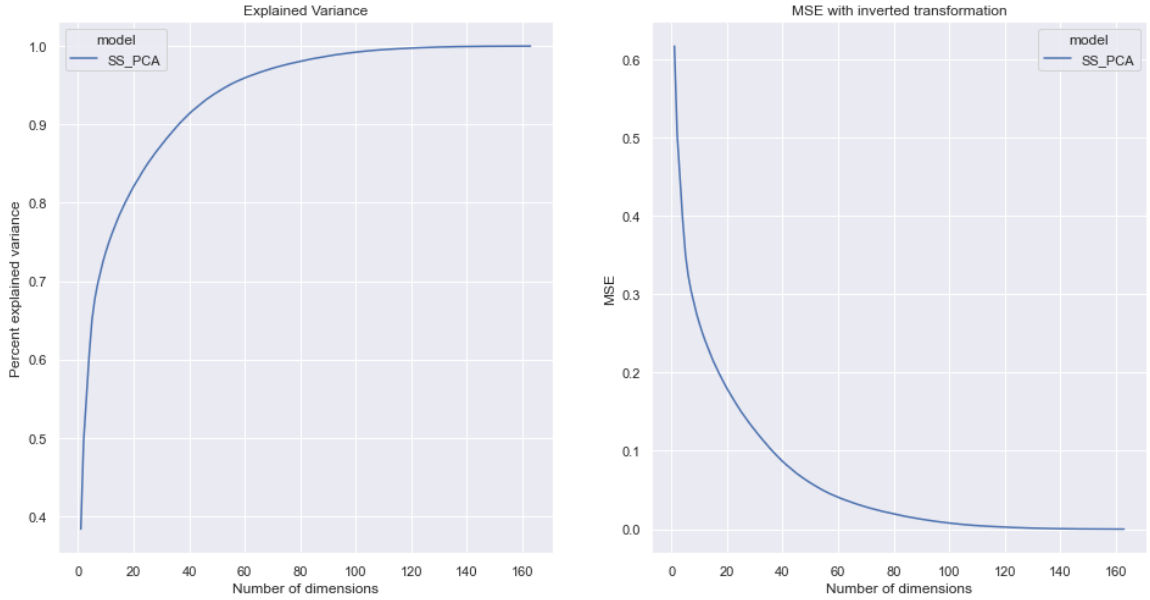
Dimensionality reduction

We will start with linear PCA and use two different ways to measure the amount of information preserved/loss for each choice of the number of components:

- computing the percentage of explained variance (summing the explained_variance_ratio_ attribute for each principal component)

- transforming with PCA and then doing an inverse transformation and computing the MSE (cfr. the definition for MSE in the Procedure section of the regression analysis below) between the data before the PCA and the data recovered after undergoing the forward and inverse transformation.

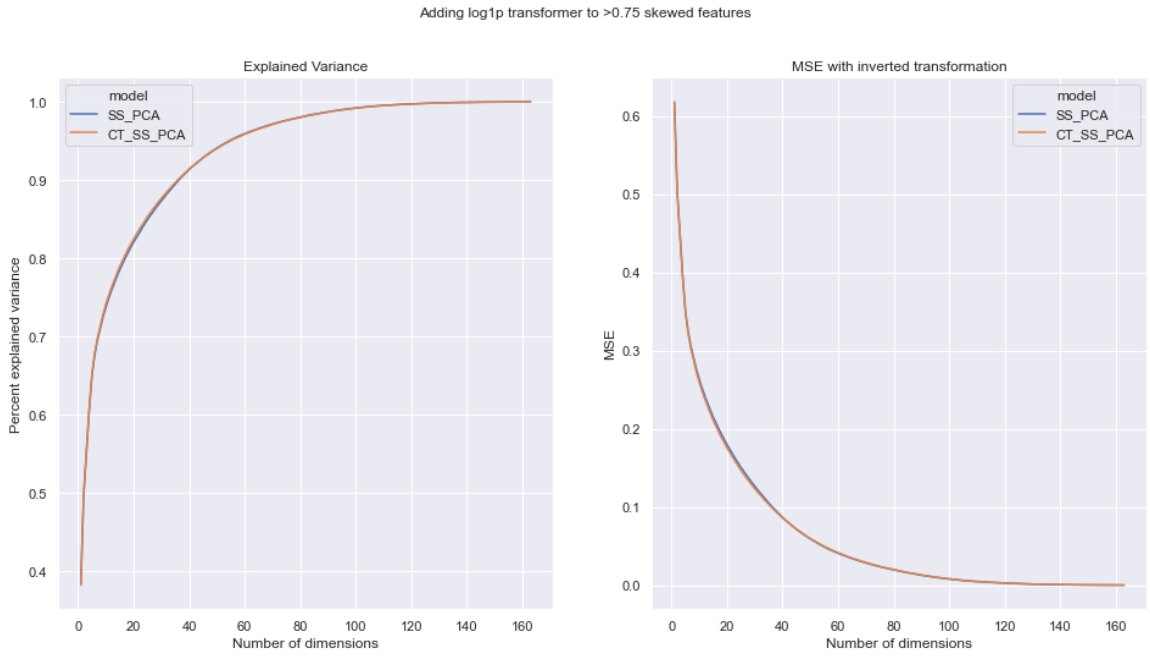
Both metrics are plotted for each amount of "dimensions", i.e. for each choice of number of principal components (n_component parameter of *sklearn*'s PCA). Here are the results for PCA after *StandardScaling* of the data.



Already with 36 principal components the explained variance is 90%, reaching 95% with 54 components and 99% with 95 components. The MSE between data before PCA and reconstructed data is 0.101 with 36 principal components, 0.051 with 54 components and 0.010 with 95 components. 132 components are needed to lower it a further degree of magnitude to 0.001.

So a reduction to *one third* of the total data (54 components from 164 features) still explains 95% of the variance and it is possible to reconstruct the data with a MSE (across almost 600.000 data values: 164 measures for 3654 days) of only 0.051.

We also tested adding a log1p transformation before the scaling step, to the columns with more than 0.75 absolute skew:



This very slightly improves both metrics, but mostly restricted to the range of 15-30 components.

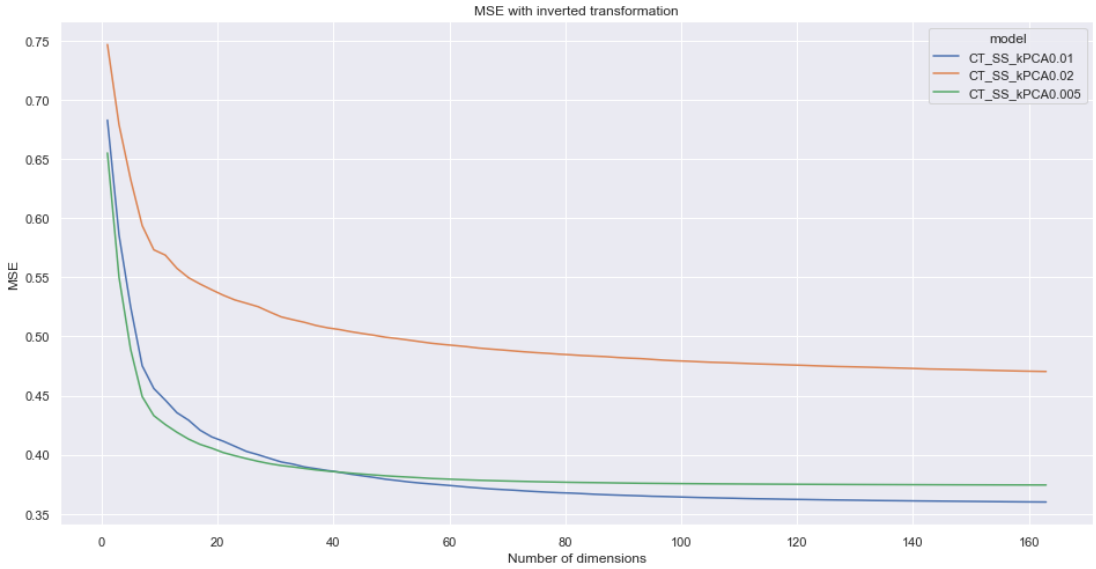
For example with 20 components, the addition of the log1p transformer increases explained variance from 82.1% to 82.5% and decreases MSE from 0.1793 to 0.1752 .

Checking feature importance by comparing components coefficients shows that they vary with each choice of number of components, but without any particular measure taking a large share. Maximum relative component weights of the features are always slightly over or under 1%.

Non linear PCA

We have repeated the analysis using *sklearn*'s *KernelPCA* using *rbf* kernels and optimising for the gamma hyper-parameter. As *KernelPCA* does not have explained_variance, we limited the comparison to the MSE with the reconstructed data.

We tried both the default gamma (1/n_features, i.e. 0.006) and several other gamma values above and below. The results were disappointing, with MSE always much higher than those obtained by linear PCA. For example:



Final model recommendation

We recommend linear PCA with StandardScaling and ColumnTransformer applying log1p to the features exhibiting more than 0.75 absolute skew, with 54 principal components, as it explains 95% of the total variance and is shown to be able to reconstruct the data to an MSE of 0.051. We will now apply this model in a regression pipeline.

Adding PCA to a regression pipeline

Procedure

We will try to predict tomorrow's sunshine hours for one location (BASEL) based on today's weather measures across all locations. We will compare the results with and without the addition of PCA.

The models are setup using a *scikit-learn* **Pipeline**, grouping together the list of steps to be applied to the data, with the final one being the estimator and the preceding ones being transforms, with or without a PCA step.

At the beginning of the analysis we set aside a sample of 20% of the entire dataset to be used as **test** for evaluation. The remaining 80% is the data on which we **train** the models on.

For **cross validation**, we have used *KFold* with **5 splits**. This means that the training data is split in 5 subsets and each fold is then used once as a validation while the remaining folds form the training set for that iteration.

For **scoring**, two metrics have been used: the mean squared error (MSE) and the R^2 score (coefficient of determination).

With $\hat{y}_i$ being the predicted value of the i^{th} sample, and y_i as the corresponding true value, then the mean squared error (MSE) estimated over n samples is defined as:

$$\mathrm{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2$$

while the R^2 score is defined as:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance.

The train data metrics MSE and R^2 (indicated respectively as columns **trainMSE** and **trainR2** in the tables below) are computed averaging over the 5 folds according to the cross validation explained above (using *cross_val_score* function). The test data scores (indicated as **testMSE** and **testR2**) are computed after training the model on the entire train dataset and scoring against the test data which has been set aside.

The train scores are hence the averages of 5 fold cross validation and thus it can often be the case that the train scores are lower than the test scores.

The cross validation ensures a good comparison of the models for the training, and the final scoring on completely unseen test data is best practice to evaluate the prediction ability of the trained models.

To tune hyper-parameters for regularisation, *sklearn*'s *GridSearchCV* was used.

Results

With log1p transformation of the most skewed features, standard scaling and the Lasso estimator, prediction of tomorrow's sunshine hours in BASEL obtains a cross-validated R^2 score of 0.453 on the test set:

	trainMSE	trainR2	testMSE	testR2
Lasso α=0.018	9.919	0.461	10.647	0.454

The addition of a PCA step with 54 components (shown above to explain 95% of the variance) gives results which are about 5-6% worse:

	trainMSE	trainR2	testMSE	testR2
PCA_Lasso α=0.023	10.060	0.454	11.157	0.428

But the advantage is that the time to train the model is greatly reduced: fitting five folds for each of 40 candidates (200 fits) in a *GridSearchCV* setup was **8.5 times faster** when adding the PCA step.

Multidimensional scaling

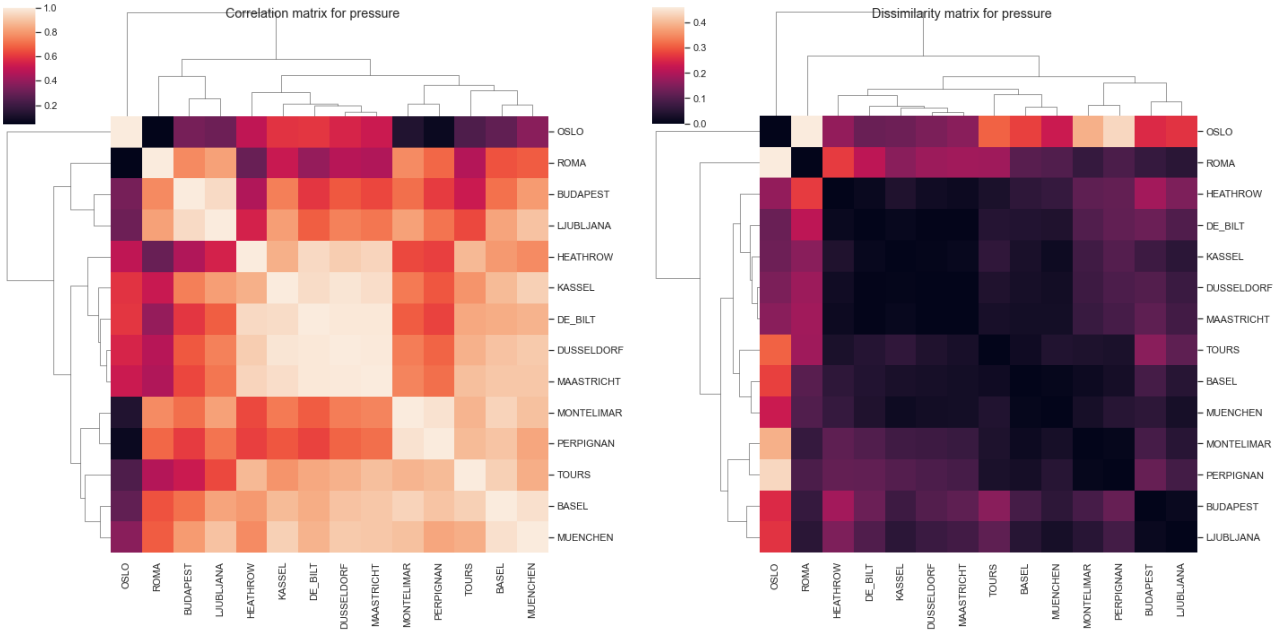
The high correlations observed gave us the idea to see if we can recover the relative position and proximity of the weather locations from how correlated they are, with the idea that locations closer in space should exhibit higher correlation between their measures.

We can start with a set of locations and features for which we have data for all measures. For example the subset of 14 locations BASEL, BUDAPEST, DE_BILT, DUSSELDORF, HEATHROW, KASSEL, LJUBLJANA, MAASTRICHT, MONTELIMAR, MUENCHEN, OSLO, PERPIGNAN, ROMA, TOURS and the measures global_radiation, humidity, pressure, temp_max, temp_mean

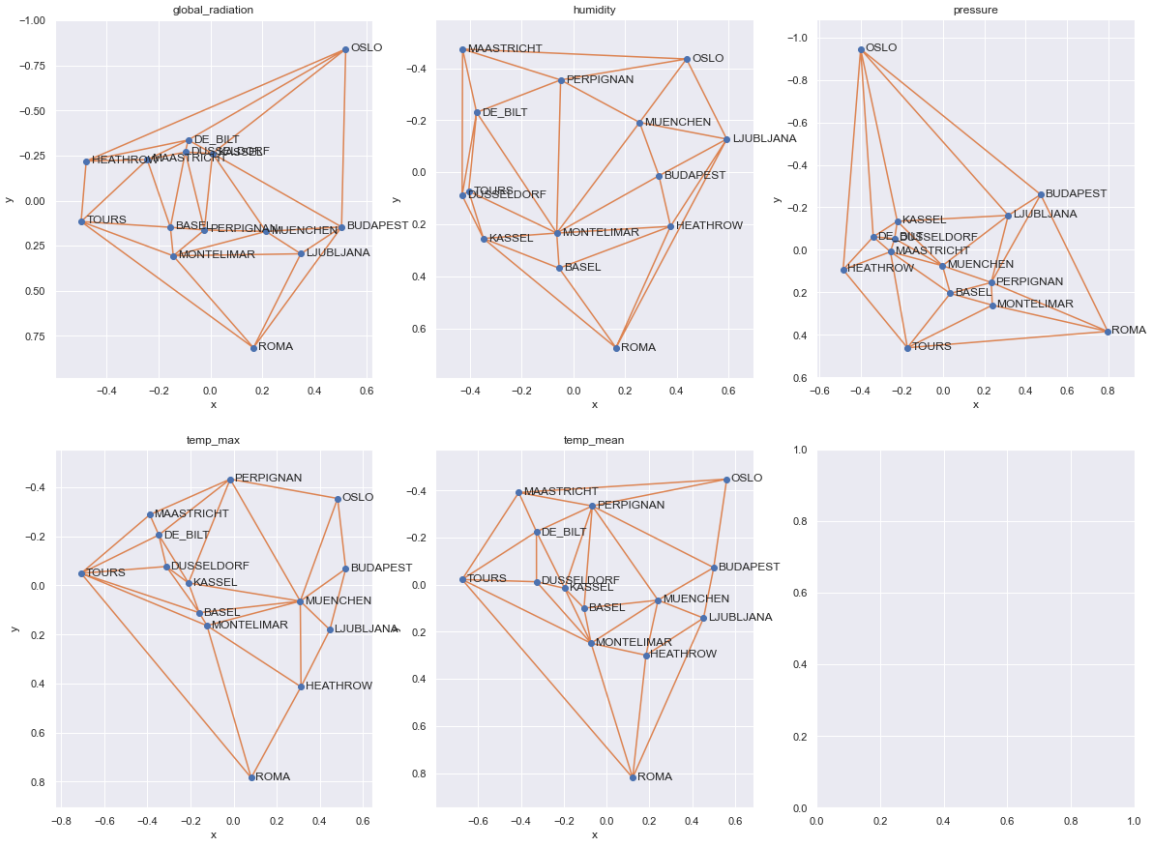
We can then build a correlation matrix between the measurements across all cities for each measure, similarly to those used in the Correlated features section above.

As there are some (slightly) negative correlations (for example humidity of PERPIGNAN and DUSSELDORF have pairwise Pearson correlation of -0.092) we can either treat them as even more distant or use absolute correlation in order to treat a negative correlation (if high) as a sign that they are close. Both strategies were tried and it was found that the best results are obtained by treating negative correlations as more distant.

After adjusting for the negative correlations and inverting the correlation matrix we hence get a dissimilarity matrix for each measure.

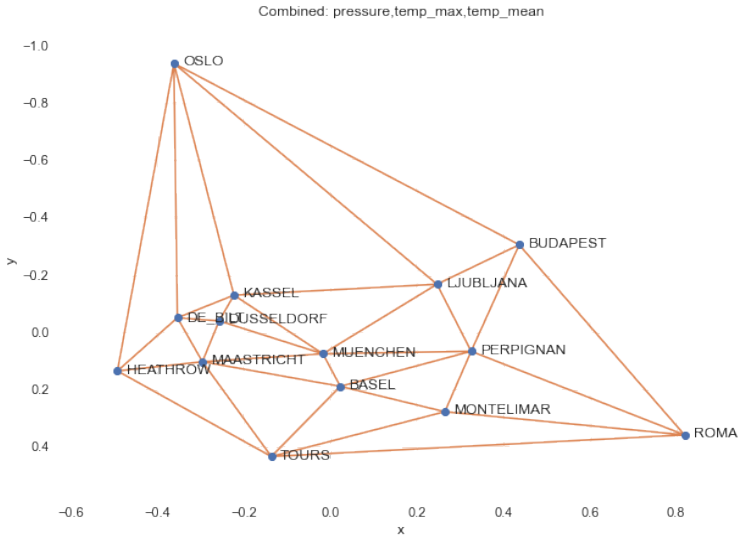


We can then pass the dissimilarity matrix to *sklearn*'s *MDS* (MultiDimensionalScaling) to plot in two dimensions (getting an x and y coordinates for each location) a reconstructed approximate topology based on the dissimilarity matrix for each measure. *Scipy*'s *Delaunay* triangulation is also shown overlaid in the following plots:

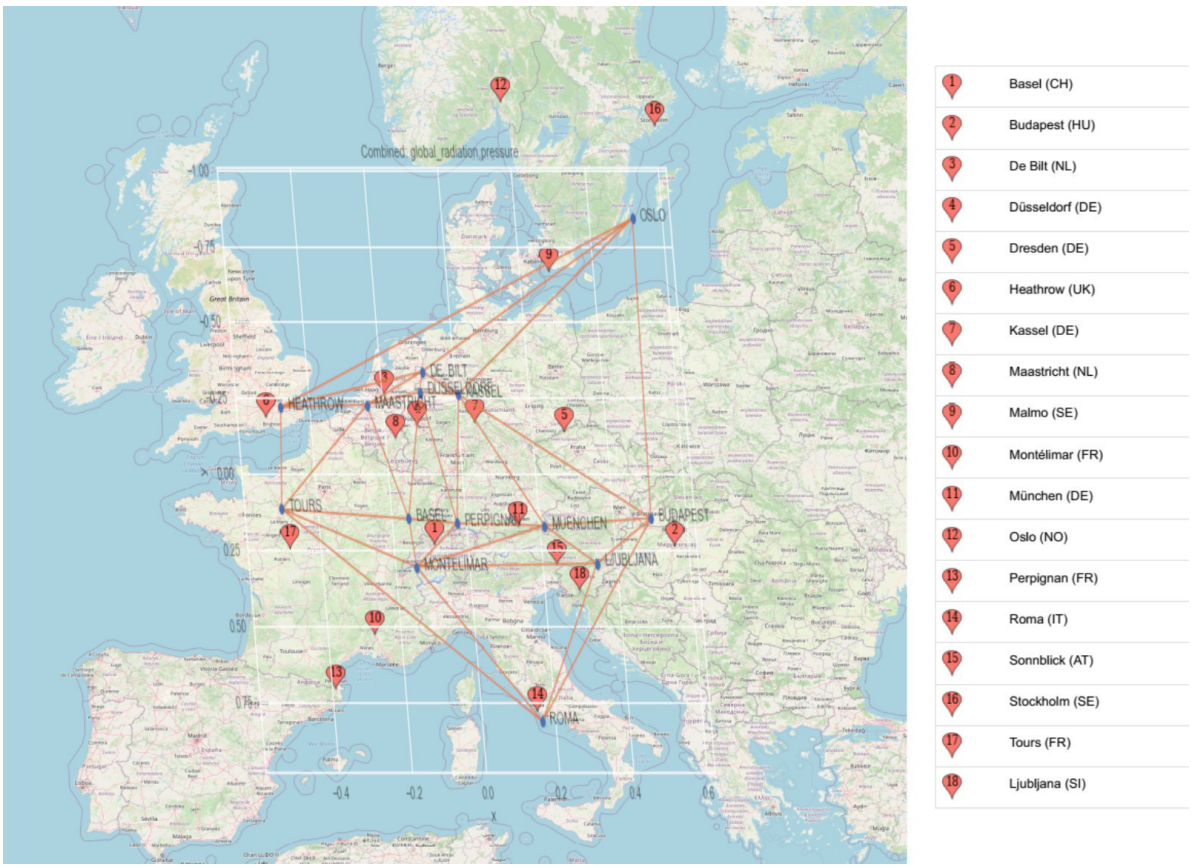


It is already possible to see a very good correspondence between the reconstructed topology and the real geographical location of the weather stations. Note that obviously MDS maps have no real cardinal directions and they could also come out "mirrored".

We can then aggregate dissimilarity matrices of different measures into a combined dissimilarity matrix to be used for MDS. For example combining the measures which we have seen show the highest correlation (temperature and pressure) we come up with the following two dimensional plot:



After appropriate rotation, scaling and skewing, we can overlay this plot to the map of the locations to show how close we've gotten to reconstruct the approximate topology:



Key findings and insights

We have investigated the high correlations in this weather dataset, and how this can lead to efficient PCA dimensionality reduction which was shown to explain 95% of the total variance with only 54 components from the 164 original features.

This is consistent with what was observed during trained regression and classification models, where for example L1 regularisation would zero over one hundred coefficients.

We've shown how PCA can be used in a regression pipeline to reduce the number of data features and hence speed up dramatically the training of the models. This is very important when a lot of models need to be fit in a cross validated grid search tuning the hyper-parameters and especially when time or computational resources are scarce. This comes at a price: a reduced final accuracy, which needs to be considered.

It was also possible to show how the correlation among features across different location could be used to reconstruct an approximate topology which was surprisingly close to the real geographical position.

Next steps

It would be interesting to try other models for dimensionality reduction and see how they compare to PCA.

It was also puzzling that the rbf kernel PCA appeared to perform much worse than linear PCA, and this merits appropriate investigation. Also different kernels could be tried.

Another interesting extension would be to work with a time frame of observations and see how PCA would work to "compress" the sequence of measurements for subsequent days (e.g. the weather of the past week) as these should also be very highly correlated.

Finally, for the aspect of reconstructing an approximate topology using MultiDimensionalScaling, we would like to examine how to best deal with missing information, in order to use all the dataset (rather than a subset of locations where we have measure information for each).

Conclusions

The dataset looked like a very promising one to use for machine learning with limited amount of data cleaning necessary.

The main objective was to examine the correlation among the measures and to understand the amount of information preserved/lost while reducing dimensionality of the dataset. We also showed how dimensionality reduction can be used to greatly speed up model training, but at the cost of reduced accuracy.

We also managed to recover to a surprisingly good degree the approximate topology of the locations in the dataset, simply using as starting point the correlation among the weather measures.

Methods and Materials

Data manipulation and analysis was performed on a MacOS laptop using own written code in python language, working in a jupyter notebook and taking advantage of the following python libraries: pandas, seaborn, scikit-learn, scipy

Data origin acknowledgement

Dataset compiled by Huber, Florian (2021); Zenodo. https://doi.org/10.5281/zenodo.4770937

ORIGINAL DATA TAKEN FROM:

EUROPEAN CLIMATE ASSESSMENT & DATASET (ECA&D), file created on 22-04-2021 THESE DATA CAN BE USED FREELY PROVIDED THAT THE FOLLOWING SOURCE IS ACKNOWLEDGED:

Klein Tank, A.M.G. and Coauthors, 2002. Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment. Int. J. of Climatol., 22, 1441-1453. Data and metadata available at http://www.ecad.eu