Intelligent Systems

Team 04 Presentation

Christian-Albrechts-Universität zu Kiel

Feature Selection

Principal Component Analysis

- **Splitting data accodring to day:** We use the preprocessed main_station data from assignment 1, we work on the data according to day so we get an array of 23 columns after split.

    - On the right side, you can see the data from preprocessed main_station data for a day

	time	level_cm	flow_m2_s
0	2014-01-01 01:00:00	0.105304	0.192496
1	2014-01-01 02:00:00	0.105304	0.192496
2	2014-01-01 03:00:00	0.105304	0.192496
3	2014-01-01 04:00:00	0.105304	0.192496
4	2014-01-01 05:00:00	0.105304	0.192496
5	2014-01-01 06:00:00	0.105304	0.192496
6	2014-01-01 07:00:00	0.105304	0.192496
7	2014-01-01 08:00:00	0.105304	0.192496
8	2014-01-01 09:00:00	0.105304	0.192496
9	2014-01-01 10:00:00	0.105304	0.192496
10	2014-01-01 11:00:00	0.105304	0.192496
11	2014-01-01 12:00:00	0.078380	0.168153
12	2014-01-01 13:00:00	0.078380	0.168153
13	2014-01-01 14:00:00	0.078380	0.168153
14	2014-01-01 15:00:00	0.078380	0.168153
15	2014-01-01 16:00:00	0.078380	0.168153
16	2014-01-01 17:00:00	0.024533	0.123524
17	2014-01-01 18:00:00	0.051456	0.145838
18	2014-01-01 19:00:00	0.051456	0.145838
19	2014-01-01 20:00:00	0.051456	0.145838
20	2014-01-01 21:00:00	0.051456	0.145838
21	2014-01-01 22:00:00	0.051456	0.145838
22	2014-01-01 23:00:00	0.024533	0.123524
23	2014-01-02 00:00:00	0.024533	0.123524

Feature Selection

Principal Component Analysis

- Code for splitting the array:

```
water_level_data_day_basis = (pd.DataFrame(station_main.groupby(station_main.index // 23)[col_level_cm]
                              .apply(list)
                              .values
                              .tolist(), columns=features).fillna(0))

water_level_data_day_basis.head()
```

- We split the data for the column "level_cm" according to day basis and we get the following array:

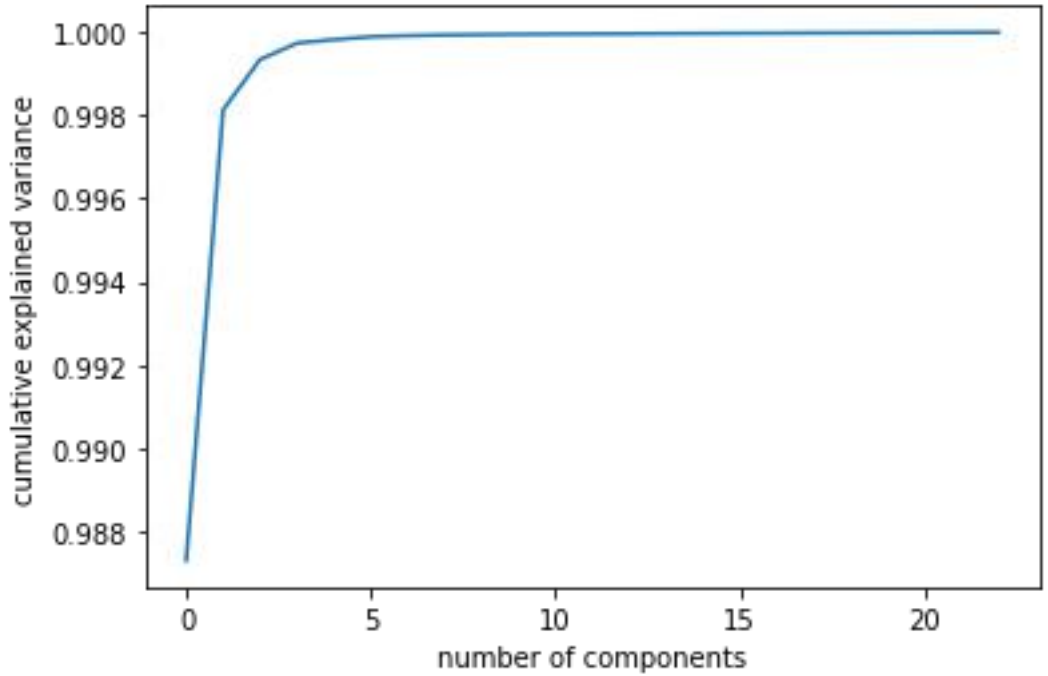
	h1	h2	h3	h4	h5	h6	h7	h8	h9	h10	...	h14	h15	h16	h17	
0	0.105304	0.105304	0.105304	0.105304	0.105304	0.105304	0.105304	0.105304	0.105304	0.105304	...	0.078380	0.078380	0.078380	0.024533	0.05
1	0.024533	-0.002391	-0.029315	-0.029315	-0.029315	-0.015853	-0.002391	-0.002391	-0.002391	-0.002391	...	0.024533	0.024533	0.024533	0.051456	0.05
2	0.078380	0.078380	0.078380	0.078380	0.078380	0.078380	0.078380	0.051456	0.051456	0.051456	...	0.024533	0.024533	0.024533	0.024533	0.02
3	0.078380	0.078380	0.078380	0.078380	0.105304	0.105304	0.132227	0.159151	0.186075	0.186075	...	0.239922	0.239922	0.239922	0.239922	0.21
4	0.159151	0.159151	0.159151	0.159151	0.159151	0.159151	0.159151	0.132227	0.132227	0.132227	...	0.105304	0.105304	0.105304	0.105304	0.10

5 rows × 23 columns

Feature Selection

Principal Component Analysis

- We then standardized the data using StandardScaler from sklearn
- PCA is then applied to the standardized data to find out the number of components from the graph, as seen below



Feature Selection

Principal Component Analysis

- From the graph above, we decide to reduce the dataframe to 4 components
- We then get important_features_set from this reduced dataframe

	h1	h2	h3	h4	h5	h6	h7	h8	h9	h10	...	h14	h15	h16	h17	
0	0.206584	0.207154	0.207669	0.208107	0.208471	0.208755	0.208981	0.209184	0.209327	0.209418	...	0.209463	0.209353	0.209215	0.209015	
1	-0.324091	-0.302164	-0.277077	-0.251802	-0.225349	-0.199219	-0.172303	-0.142291	-0.111979	-0.081469	...	0.074959	0.107273	0.136971	0.168509	
2	0.356098	0.284964	0.217200	0.137486	0.058309	-0.013062	-0.081212	-0.133535	-0.183603	-0.221431	...	-0.236379	-0.206675	-0.160684	-0.098561	-
3	-0.292784	-0.208179	-0.119194	-0.018268	0.078304	0.163558	0.229165	0.270906	0.297942	0.308725	...	-0.252032	-0.246273	-0.228575	-0.187381	-

4 rows × 23 columns

- We get the index of most important feature from each component and print the names of important features, which gives us the following array:

```
['h12', 'h1', 'h23', 'h10']
```

Feature selection

Inconsistency Rate

- We then calculate the inconsistency number “IZ” by subtracting sum of important_features_set (obtained from PCA from last section) with max of important_features_set

h6	h7	h8	h9	h10	...	h15	h16	h17	h18	h19	h20	h21	h22	h23	IZ
.208755	0.208981	0.209184	0.209327	0.209418	...	0.209353	0.209215	0.209015	0.208750	0.208411	0.208022	0.207601	0.207140	0.206620	4.586222
.199219	-0.172303	-0.142291	-0.111979	-0.081469	...	0.107273	0.136971	0.168509	0.200449	0.230238	0.257671	0.281514	0.302937	0.323316	-0.323100
.013062	-0.081212	-0.133535	-0.183603	-0.221431	...	-0.206675	-0.160684	-0.098561	-0.020446	0.066137	0.143966	0.221761	0.293599	0.357353	-0.335858
.163558	0.229165	0.270906	0.297942	0.308725	...	-0.246273	-0.228575	-0.187381	-0.124780	-0.052200	0.025225	0.104218	0.170159	0.236571	-0.309225

- We then calculate the inconsistency rate from the column “IZ” above, we get

```
IR = sum(important_fetures_set["IZ"])/4
IR
```

```
0.9045098651458824
```