Exercise : PCA

The following table represents PCA output on wine data (non-normalized) in which the variables represent chemical characteristics of wine, and each case is a different wine.

a) Consider the row near the bottom labeled “variance”. Explain why column 1’s variance is so much greater that of any other column?

เพราะว่าในคอลัมน์ที่ 1 คือ Component1 ที่มีความสัมพันธ์ในทิศทางบวกกับ Attribute Proline (correlation coefficient = 1.000) ซึ่งมีค่า Standard Deviation สูงถึง 351.5 แสดงให้เห็นถึงการกระจายตัวของข้อมูลที่สูงกว่า Attribute อื่นๆเป็นอย่างมาก ส่งผลให้ค่า Variance หรือค่าความแปรปรวนสูงตามไปด้วย แต่ใน Component อื่นๆมีค่าความสัมพันธ์กับ Attribute Proline ค่อนข้างน้อยและไปมีความสัมพันธ์กับ Attribute ตัวอื่นๆ ที่มีค่า Standard Deviation ต่ำ ทำให้ Variance ต่ำตามไปด้วย

b) Comment on the use of normalization?

การ normalization เป็นการทำให้ข้อมูลทั้งหมดอยู่ใน range ที่ใกล้เคียงกันและมีค่าการกระจายตัวที่เท่ากันในทุกๆ attribute เพื่อเป็นการทำให้ เวลาสร้าง Principal Component ไม่เลือก attribute ที่มีการกระจายตัวสูงให้มีความสำคัญกว่า attribute อื่นๆ

c) Use RapidMiner. Compare and comment the results between using normalization+PCA and without using normalization+PCA. How the result different of a)?

การทำ normalization+PCA จะทำให้ได้ทั้งหมด 10 components เมื่อกำหนด Variance threshold เท่ากับ 0.95 แต่ถ้าทำ PCA โดยที่ไม่ได้ทำการ normalization จะได้ 1 component ที่ Variance threshold เท่ากับ 0.95 ซึ่งทั้งสองแบบต่างจากข้อ a) เพราะ ข้อ a) มี 5 components แต่ค่า Variance threshold ในข้อ a) ไม่ใช่ 0.95

d) Use Python or R. Do you obtain same results?

- Without Normalization

Row No.	pc_1
1	318.563
2	303.097
3	438.061
4	733.240
5	-11.571
6	703.231
7	542.972
8	548.402
9	298.037
10	298.050
11	763.080
12	532.943
13	572.834
14	402.925
15	800.053

From RapidMiner

	principal component 1
0	318.562979
1	303.097420
2	438.061133
3	733.240139
4	-11.571428
5	703.231192
6	542.971581
7	548.401860
8	298.036863
9	298.049553
10	763.079712
11	532.943228
12	572.834410
13	402.925358
14	800.053394

From Python

Ans ตามตารางด้านบนจะเห็นได้ว่าผลลัพธ์ที่ไม่ได้ทำการ normalization จะออกมาเหมือนกัน

- With Normalization

Row No.	pc_1	pc_2	pc_3	pc_4	pc_5	pc_6	pc_7	pc_8	pc_9	pc_10
1	3.307	1.439	-0.165	0.215	-0.691	-0.223	-0.595	0.065	-0.640	1.018
2	2.203	-0.332	-2.021	0.291	0.257	-0.925	-0.054	1.022	0.308	0.159
3	2.510	1.028	0.980	-0.723	0.250	0.548	-0.423	-0.343	1.175	0.113
4	3.746	2.749	-0.176	-0.566	0.311	0.114	0.382	0.642	-0.052	0.239
5	1.006	0.867	2.021	0.409	-0.298	-0.405	-0.443	0.416	-0.326	-0.078
6	3.042	2.116	-0.628	0.514	0.630	0.123	-0.401	0.394	0.152	-0.102
7	2.442	1.172	-0.974	0.066	1.025	-0.618	-0.053	-0.371	0.456	1.014
8	2.054	1.604	0.146	1.189	-0.077	-1.436	-0.032	0.232	-0.123	0.734
9	2.504	0.915	-1.766	-0.056	0.890	-0.129	-0.125	-0.498	-0.605	0.174
10	2.746	0.787	-0.981	-0.348	0.467	0.163	0.872	0.150	-0.230	0.179
11	3.470	1.299	-0.422	-0.027	0.337	-0.182	-0.247	-1.203	0.523	-0.214
12	1.750	0.610	-1.188	0.888	0.736	-0.551	0.433	-0.982	0.473	0.220
13	2.108	0.674	-0.863	0.355	1.207	-0.214	0.242	-0.460	0.876	-0.096
14	3.448	1.127	-1.201	-0.162	2.017	0.744	-1.472	-0.379	0.026	-0.244
15	4.301	2.090	-1.260	-0.305	1.027	0.793	-0.997	-0.404	0.838	-0.363

From RapidMiner

	pc_1	pc_2	pc_3	pc_4	pc_5	pc_6	pc_7	pc_8	pc_9	pc_10
0	3.316751	-1.443463	-0.165739	-0.215631	0.693043	-0.223880	0.596427	0.065139	0.641443	1.020956
1	2.209465	0.333393	-2.026457	-0.291358	-0.257655	-0.927120	0.053776	1.024416	-0.308847	0.159701
2	2.516740	-1.031151	0.982819	0.724902	-0.251033	0.549276	0.424205	-0.344216	-1.177834	0.113361
3	3.757066	-2.756372	-0.176192	0.567983	-0.311842	0.114431	-0.383337	0.643593	0.052544	0.239413
4	1.008908	-0.869831	2.026688	-0.409766	0.298458	-0.406520	0.444074	0.416700	0.326819	-0.078366
5	3.050254	-2.122401	-0.629396	-0.515637	-0.632019	0.123431	0.401654	0.394893	-0.152146	-0.101996
6	2.449090	-1.174850	-0.977095	-0.065831	-1.027762	-0.620121	0.052891	-0.371934	-0.457016	1.016563
7	2.059437	-1.608963	0.146282	-1.192608	0.076903	-1.439806	0.032376	0.232979	0.123370	0.735600
8	2.510874	-0.918071	-1.770969	0.056270	-0.892257	-0.129181	0.125285	-0.499578	0.606589	0.174107
9	2.753628	-0.789438	-0.984247	0.349382	-0.468553	0.163392	-0.874352	0.150580	0.230489	0.179420
10	3.479737	-1.302333	-0.422735	0.026842	-0.338375	-0.182902	0.248162	-1.206611	-0.524574	-0.214538
11	1.754753	-0.611977	-1.190878	-0.890164	-0.738573	-0.553055	-0.434266	-0.985127	-0.474030	0.220283
12	2.113462	-0.675706	-0.865086	-0.356438	-1.209929	-0.215076	-0.242597	-0.461506	-0.878813	-0.096505
13	3.458157	-1.130630	-1.204276	0.162458	-2.023127	0.745781	1.475773	-0.380386	-0.025702	-0.244653
14	4.312784	-2.095976	-1.263913	0.305773	-1.029693	0.795643	0.999971	-0.404891	-0.840343	-0.364433

From Python

Ans ตามตารางด้านบนจะเห็นได้ว่าผลลัพธ์ที่ได้จากการทำ normalization จะออกมาไม่เหมือนกัน