Ejercicio 1 y 2:

Dataset wine

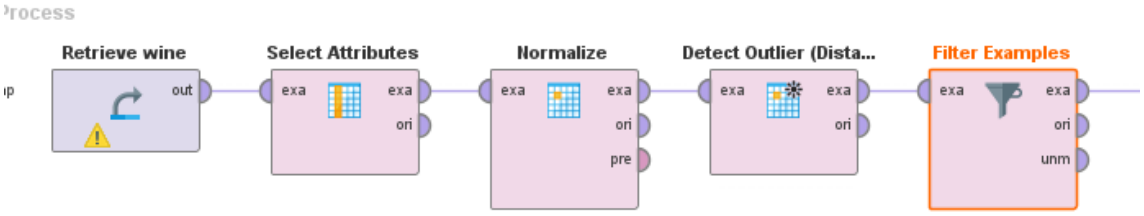
Este dataset tiene como objetivo determinar el tipo de vino que se está analizando. Para ello, se cuenta con un conjunto de datos de vinos crecidos en la misma región de Italia pero provenientes de distintos cultivos. Los atributos del dataset son:

- Alcohol
- Malic acid
- Ash
- Alcalinity of ash
- Magnesium
- Total phenols
- Flavanoids
- Nonflavanoid phenols
- Proanthocyanins
- Color intensity
- Hue
- OD280/OD315 of diluted wines
- Proline

Analisis

Name	Type	Missing	Statistics		Filter (12 / 12 attributes): Search for Attributes	
Alcohol	Integer	0	Min 1	Max 3	Average 1.938	
Malic acid	Real	0	Min 11.030	Max 14.830	Average 13.001	
Ash	Real	0	Min 0.740	Max 5.800	Average 2.336	
Alcalinity of ash	Real	0	Min 1.360	Max 3.230	Average 2.367	
Magnesium	Real	0	Min 10.600	Max 30	Average 19.495	
Flavanoids	Integer	0	Min 70	Max 162	Average 99.742	
Nonflavanoid phenols	Real	0	Min 0.980	Max 3.880	Average 2.295	
Proanthocyans	Real	0	Min 0.340	Max 5.080	Average 2.029	
⚠ Color Intensity	Real	0	Min 0.130	Max 0.660	Average 0.362	
Hue	Real	0	Min 0.410	Max 3.580	Average 1.591	
OD280/OD315 of diluted wines	Real	0	Min 1.280	Max 13	Average 5.058	
Proline	Real	0	Min 0.480	Max 1.710	Average 0.957	

Dados los datos de las estadísticas de Rapidminer vemos como ningún atributo tiene valores faltantes, por ello pasamos directo al análisis de outliers.



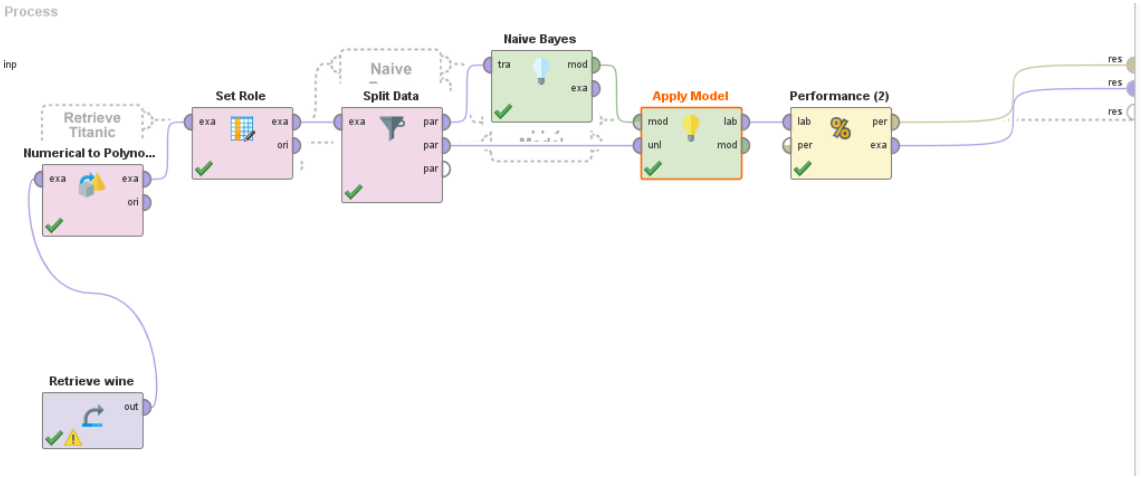
Se crea una sencilla red en rapidminer donde se seleccionan todos los atributos excepto “alcohol” (nuestra salida), normalizamos y luego detectamos los outliers. Por último, filtramos todos aquellos datos detectados como outliers.

Row No.	outlier	Malic acid	Ash	Alcalinity of ...	Magnesium	Flavanoids	Nonflavanoi...	Proanthocy...	Color
1	true	-0.777	-1.250	-3.669	-2.664	-0.822	-0.503	-1.461	-0.65
2	true	-0.974	-1.026	-2.247	-0.807	3.589	-0.711	-0.750	-1.78
3	true	-0.013	-0.596	0.851	3.146	2.749	1.606	0.862	-1.22
4	true	-0.826	-1.205	-1.518	-1.406	2.539	-0.631	-0.179	-0.09
5	true	-0.654	-0.731	-0.607	-0.148	4.359	0.327	0.241	-0.33
6	true	-1.467	-0.194	1.361	0.600	2.399	-1.111	-1.040	-1.78
7	true	-1.898	1.256	-1.992	0.002	0.508	1.414	0.551	-0.97
8	true	-2.427	-0.740	-0.607	0.600	-1.032	0.263	0.141	1.271
9	true	-1.775	-0.256	3.147	2.696	1.348	1.414	3.054	0.869
10	true	1.650	-0.588	1.216	1.648	-0.122	0.807	-0.720	1.351

Analizando la data, vemos como rapidminer encontró solamente 10 outliers.

Se concluye que el dataset está muy bien distribuido y tiene data de calidad ya que se encuentran pocos outliers y no hay datos faltantes.

Análisis de performance naive bayes



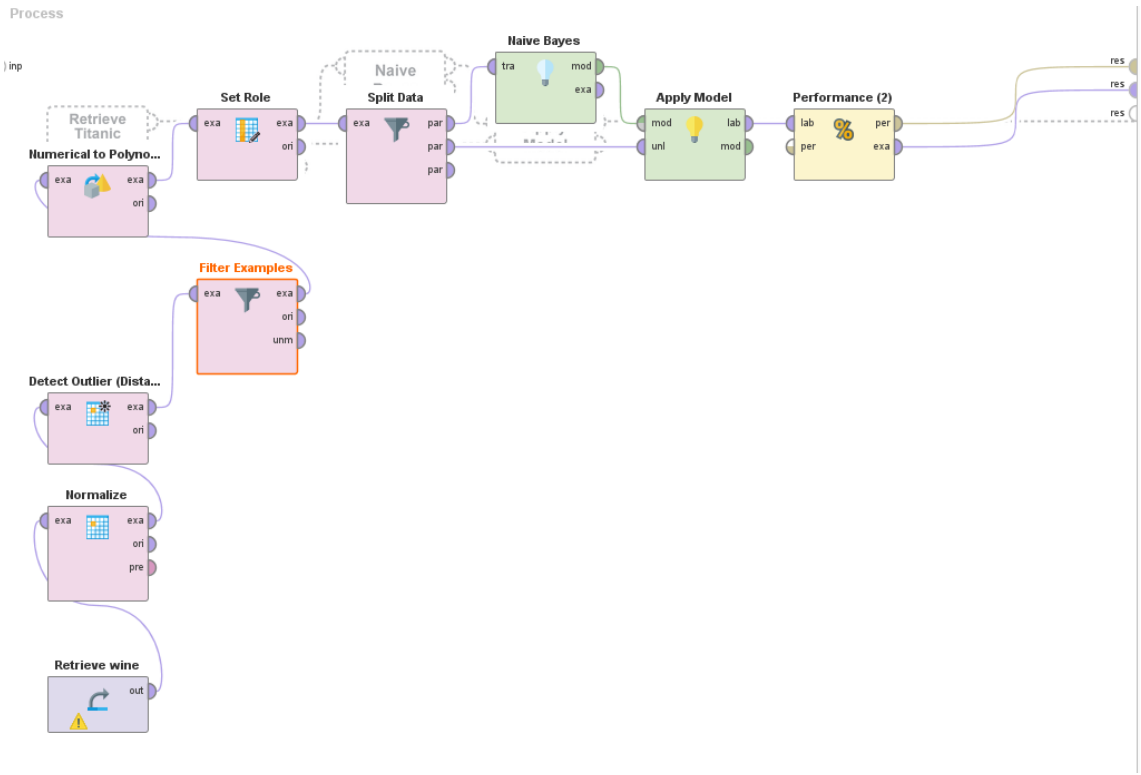
Dada la red mostrada, se realiza la comparación de los datos.

Sin normalización:

accuracy: 96.23%

	true 1	true 2	true 3	class precision
pred. 1	17	1	0	94.44%
pred. 2	1	20	0	95.24%
pred. 3	0	0	14	100.00%
class recall	94.44%	95.24%	100.00%	

Con normalización y filtrado de outliers:



accuracy: 92.00%

	true -1.211	true 0.080	true 1.370	class precision
pred. -1.211	13	0	0	100.00%
pred. 0.080	4	19	0	82.61%
pred. 1.370	0	0	14	100.00%
class recall	76.47%	100.00%	100.00%	

Solo con normalización:

accuracy: 96.23%

	true -1.211	true 0.080	true 1.370	class precision
pred. -1.211	17	1	0	94.44%
pred. 0.080	1	20	0	95.24%
pred. 1.370	0	0	14	100.00%
class recall	94.44%	95.24%	100.00%	

Conclusión

El aprendizaje con naive bayes se comporta peor con los datos con limpieza del dataset que sin la limpieza. En este caso, se puede deber a la poca cantidad de datos que se tienen y asimismo, la cantidad de outliers que constituyen un 8% de los datos. Esto lleva a que el modelo se entrene teniendo muy en cuenta los outliers y generando distintos resultados. Para ver una mejora en la performance, se debería probar con un dataset más grande.