UT2_PD1

Property	COMPLETED
Date	@September 5, 2023
# UT	2

There are two general groups of data handling: blending and cleansing. We are going to see operations for data cleansing.

Blending	is about transforming a data set from one state to another or combining multiple data sets.
Cleansing	is about improving the data so that modeling will deliver better results.

Handling Missing values:

Se puede ver en el análisis de los datos que (completar)

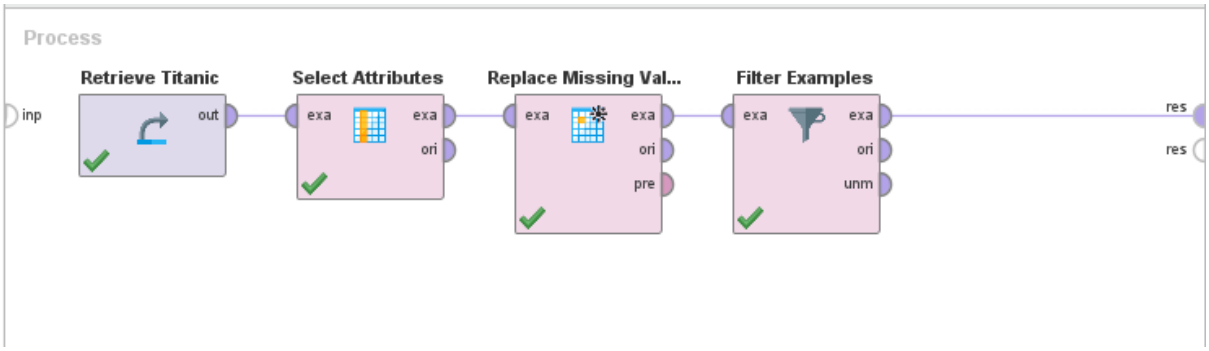
Lo primero que se hace es una selección de atributos, en los cuales se excluye los atributos Cabina y Lifeboat pues no son relevantes para el modelo. Cabina porque tiene demasiados valores faltantes y Lifeboat porque no aporta.

Una vez que se ejecuta se puede ver que falta todavía tenemos missing values en algunos atributos para ello lo que se realiza es utilizar el Replace Missing Values operator. Se aplica sobre el atributo Age.

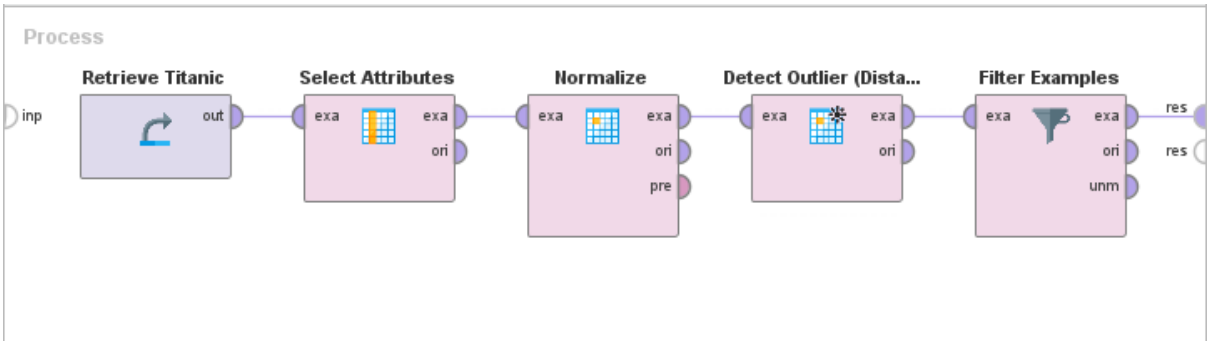
Missing *Age* was replaced with the average value of *Age* - this is a common technique for handling a lot of missing values for an attribute. Because there are only a few missing values left, we can safely filter these examples out of the data set.

Por ultimo se realiza una Filtro donde se eliminan las tuplas con valores faltantes

Modelo:



Normalization and Outlier detection



> Another important step of data cleansing is to identify unusual cases and remove them from the data set. In some situations, the outliers themselves might be the most interesting cases (detecting fraudulent credit card transactions, for example), but in most cases outliers are simply the result of an incorrect measurement and should be removed from the data set.
> -RapidMiner

La primera parte es igual a la anterior, es decir excluimos los atributos que no nos van a servir en el modelo en este caso: *Cabin*, *Life Boat*, *Name*, and *Ticket Number.*

Luego se Normalizan los valores del Dataset. En general una normalizacion es siempre necesaria cuando se trata de aplicar algoritmos en base a distancia como es outlier detection or k-Means clustering. (El cual vamos a aplicar mas adelante).

> Using the default parameters, the **Normalize** operator will perform a *z-Transformation* (also known as Standardization) which results in a mean value of 0 and a standard deviation of 1

> for each attribute. In other words, all of the attributes are on the same scale after ormalization and can be compared with one another.

Luego se agrega el operador para detectar outliers el cual detecta los 10 datos mas alejados.

Por ultimo se filtra el dataset donde los atributos outlier = false.

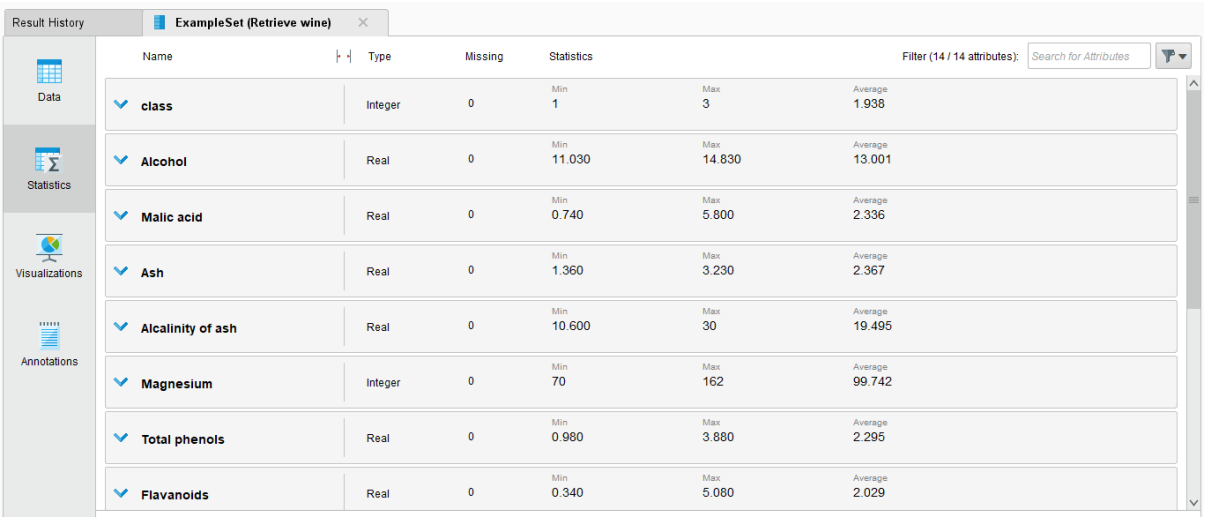
Ejercicio 2: Analisis en RapidMiner del dataset Wine.

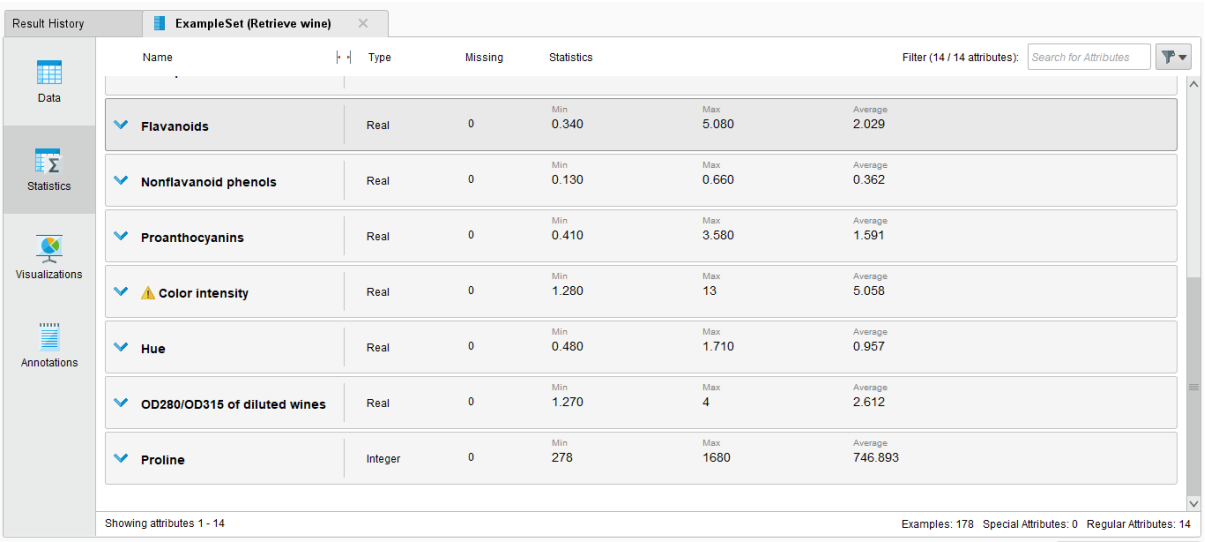
El dataset son los resultados de un análisis químico de vinos cultivados en la misma región de Italia pero procedentes de tres cultivares diferentes. El análisis determinó las cantidades de 13 componentes presentes en cada uno de los tres tipos de vino.

Nos encontramos ante un dataset con los siguientes atributos: (Extraído de UCI)

Attribute Name	Role	Type	Missing Values
class	Target	Categorical	false
Alcohol	Feature	Continuous	false
Malicacid	Feature	Continuous	false
Ash	Feature	Continuous	false
Alcalinity_of_ash	Feature	Continuous	false
Magnesium	Feature	Integer	false
Total_phenols	Feature	Continuous	false
Flavanoids	Feature	Continuous	false
Nonflavanoid_phenols	Feature	Continuous	false
Proanthocyanins	Feature	Continuous	false
Color_intensity	Feature	Continuous	false
Hue	Feature	Continuous	false
0D280_0D315_of_diluted_wines	Feature	Continuous	false
Proline	Feature	Integer	

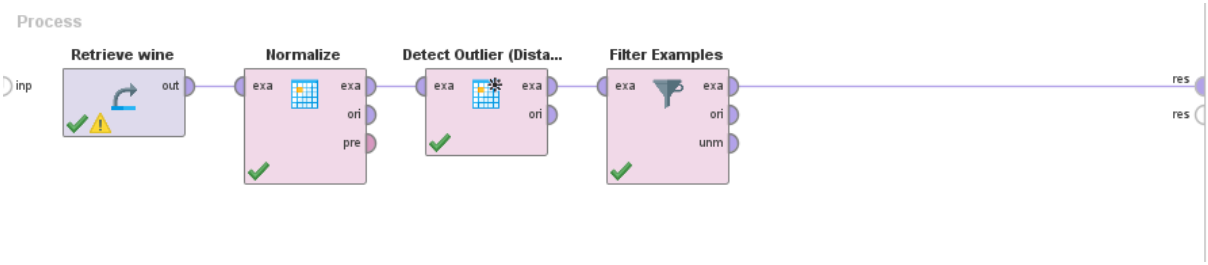
Tras un analisis con el Statics de RapidMiner encontramos que:





Destacando que no hay valores perdidos en el mismo.

Para los outliers se realizara la siguiente conexion de operadores:



Se encuentrarn aproximadamente 10 datos outliers.

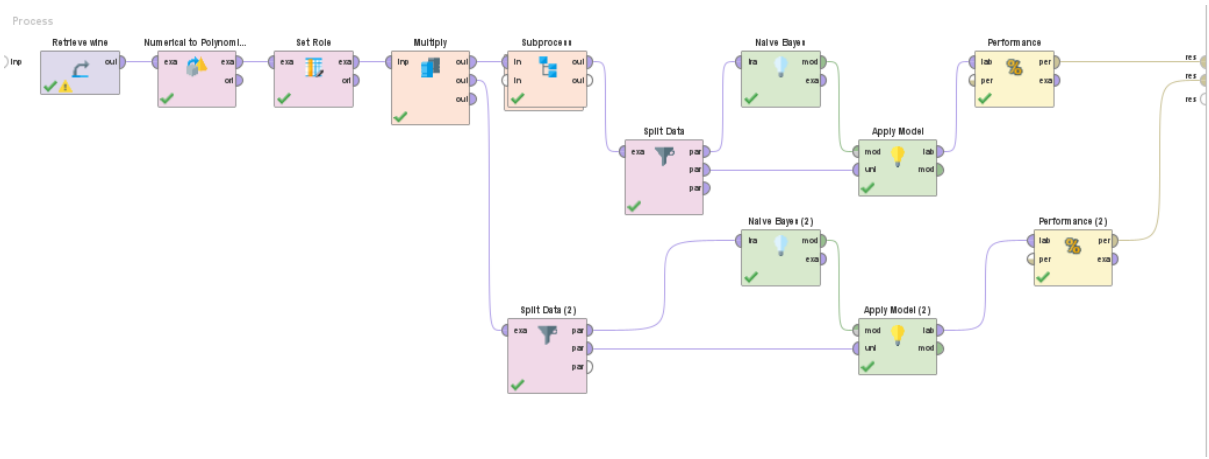
Luego se utiliza un modelo sencillo de Naive Bayes para realizar la clasificación , y bloques de

evaluación de la performance pertinentes (utilizando el dataset de test) para obtener los resultados.

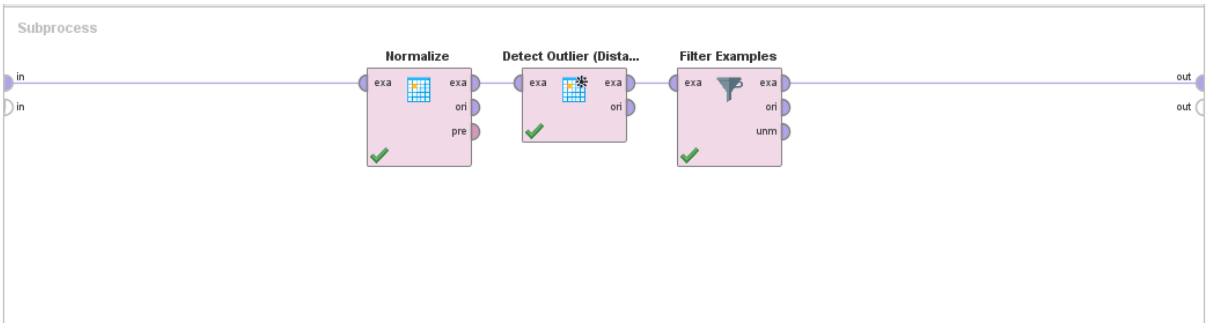
A grosso modo podemos ver que el modelo consiste en el dataset del titanic, el cual se le realiza un parse en el atributo classe, de integer a polinomial (esto debido a que el wizard de rapidminer comprendio mal el dato), luego se establece el rol de la variable objetivo al mismo.

Luego se puede ver que tenemos dos canales uno con un subproceso el cual realiza una normalizacion para luego realizar un filtrado de datos donde se quiten estos outliers para luego realizar un split de esa data en un 70 training 30 test. En el otro canal no hay subproceso y por ende se hace el split sobre el dataset mismo.

A los datos de training se aplica el algoritmo Naive Bayes, para luego aplicar el modelo sobre los datos de training. Por ultimo se utiliza un operador de performance para comparar los resultados de performance de cada canal.



Detección y filtrado de outliers:



Resultados sin normalización:

accuracy: 96.23%

	true 1	true 2	true 3	class precision
pred. 1	16	0	0	100.00%
pred. 2	2	21	0	91.30%
pred. 3	0	0	14	100.00%
class recall	88.89%	100.00%	100.00%	

Resultado con normalización:

accuracy: 100.00%

	true 1	true 2	true 3	class precision
pred. 1	0	0	0	0.00%
pred. 2	0	2	0	100.00%
pred. 3	0	0	0	0.00%
class recall	0.00%	100.00%	0.00%	