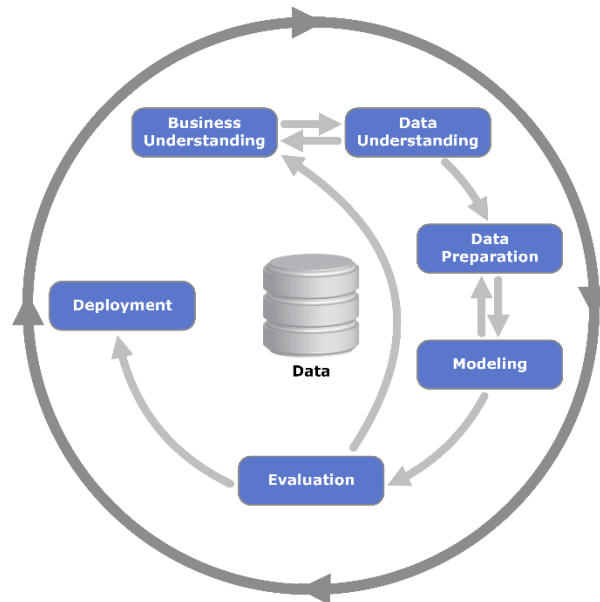


## Predicción de padecer una enfermedad cardíaca

Rafael Filardi – IA1

Objetivo: El objetivo de este caso de estudio es lograr predecir la probabilidad de que una persona posea una enfermedad cardíaca.

El estudio es realizado en el marco del proceso CRISP-DM:



### Comprensión del negocio:

Se entiende como necesidad médica la posibilidad de generar algoritmos predictivos que dado ciertos valores de una persona detecten una enfermedad. En este caso una enfermedad cardíaca.

### Comprensión de la data:

Se obtienen 4 dataset recuperados de:

Estos son:

1. Cleveland Clinic Foundation (cleveland.data)
2. Hungarian Institute of Cardiology, Budapest (hungarian.data)
3. V.A. Medical Center, Long Beach, CA (long-beach.va.data)
4. University Hospital, Zurich, Switzerland (switzerland.data)

De acuerdo con el archivo de presentación, se extrae que todos los dataset poseen 76 atributos por fila (todos con los mismos campos), pero que en los estudios realizados solo son 14. Es más, incluso destaca que solo el de Cleveland fue utilizado como entrenamiento para diferentes aplicaciones de ML.

Ningún data set posee los 76 atributos.

Los 14 más usados:

Variable Name	Role	Type	Demographic	Description	Units	Missing Values
age	Feature	Integer	Age	Age	years	no
sex	Feature	Categorical	Sex	Sex		no
cp	Feature	Categorical				no
trestbps	Feature	Integer		Resting Blood Pressure (on admission to the hospital)	mm Hg	no
chol	Feature	Integer		Serum Cholesterol	mg/dl	no
fbs	Feature	Categorical		Fasting Blood Sugar > 120 mg/dl		no
restecg	Feature	Categorical				no
thalach	Feature	Integer		Maximum Heart Rate Achieved		no
exang	Feature	Categorical		Exercise Induced Angina		no
oldpeak	Feature	Integer		ST Depression Induced by Exercise Relative to Rest		no
slope	Feature	Categorical				no
ca	Feature	Integer		Number of Major Vessels (0-3) Colored by Flourosopy		yes
thal	Feature	Categorical				yes
num	Target	Integer		Diagnosis of Heart Disease		no

Features	Description
Age	Age in year
Sex	Gender
CP	Chest pain type

Features	Description
Trestbps	Resting blood pressure
Chol	Serum cholesterol
Fbs	Fasting blood sugar
Resteg	Resting electrographic results
Talach	Maximum heart rate achieved
Exang	Exercise induce angina
Oldpeak	ST depression induced by exercise relative to rest
Slope	The slope of the peak exercise ST segment
CA	Number of major vessels coloured by fluoroscopy
Thal	Thallium heart scan
Goal	Diagnosis of heart disease

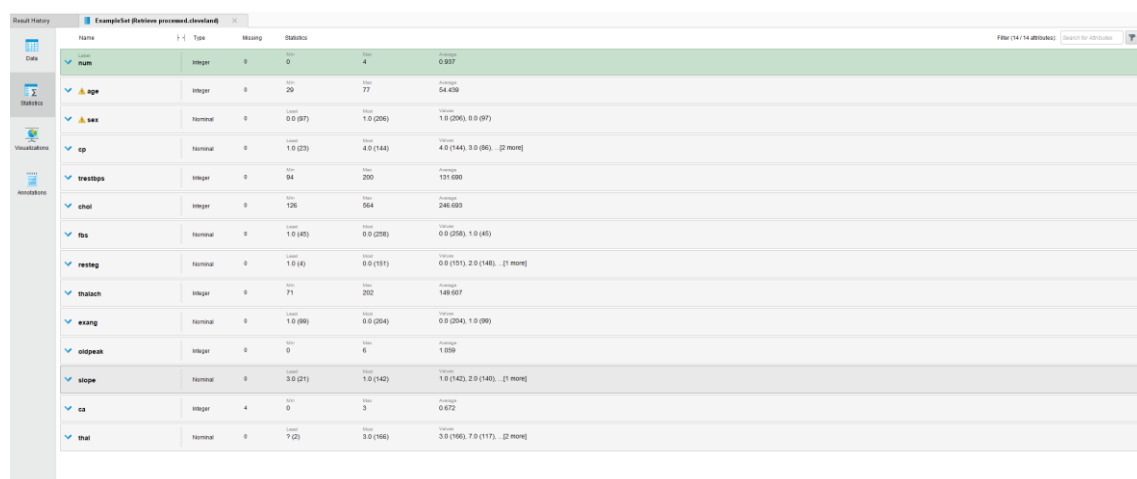
Podemos observar que de esos 76 hay muchos los cuales o no tienen descripción, o son datos identificativos (como son nombre, id, etc) los cuales no generan un aporte significativo al modelo, y deben ser retirados para ahorrar tiempo de ejecución.

Para este caso se utilizarán los 14 atributos. Los cuales están disponibles en el repositorio de UCL como la versión procesada de los datos.

Se realizará de la siguiente forma: se verá cada dataset por separado para luego realizarlos sobre los 4 dataset unidos.

## Estadística

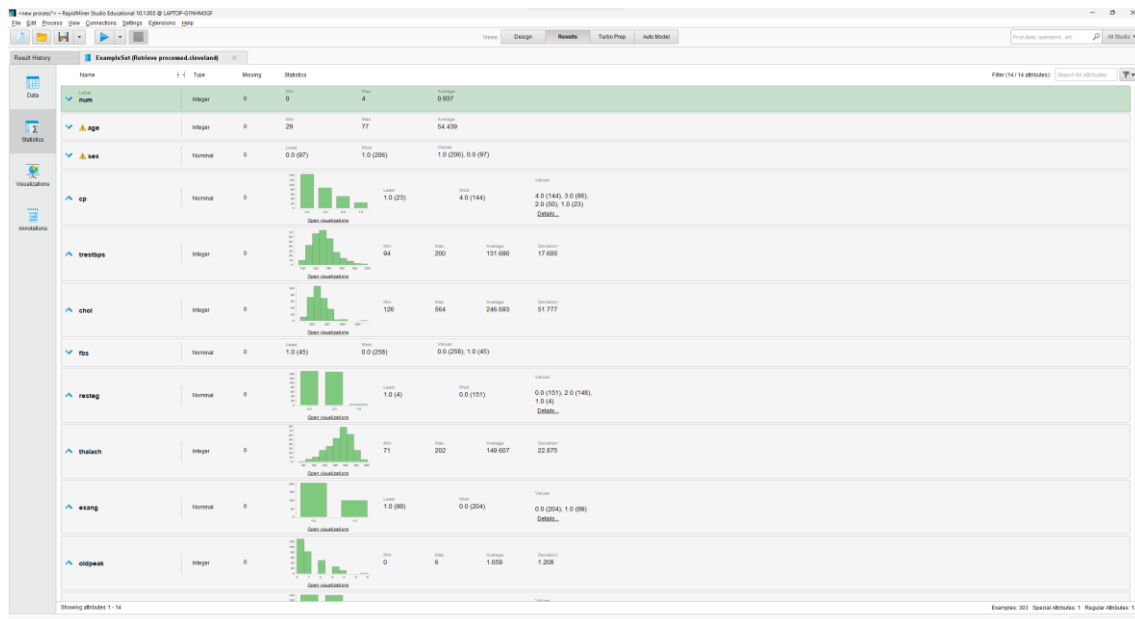
### - Cleveland



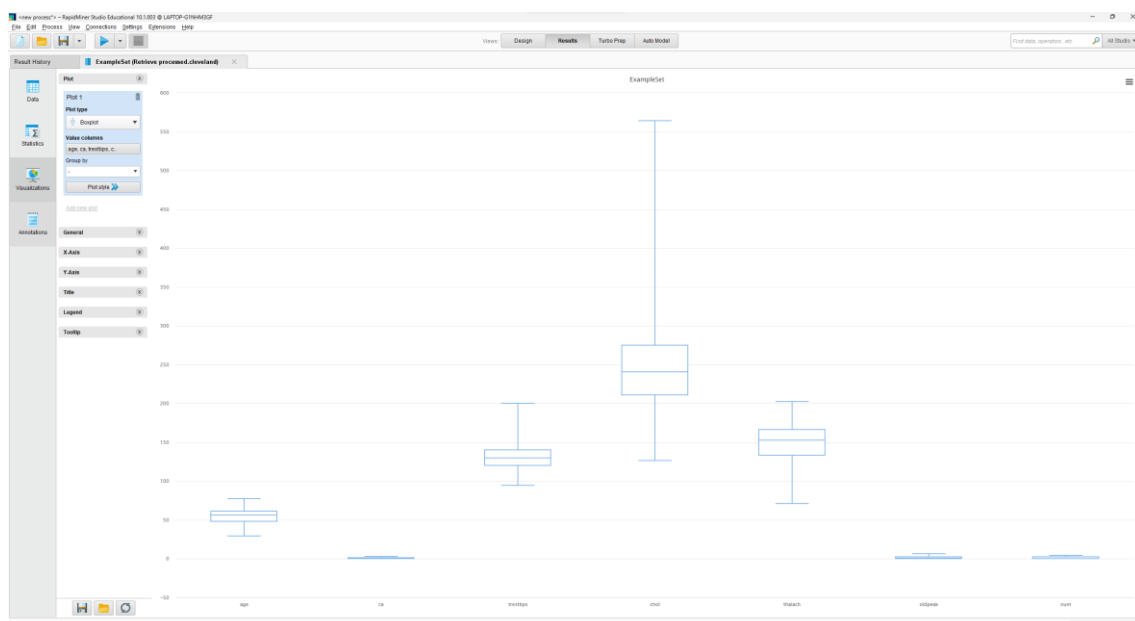
Name	Type	Missing	Statistics
num	Integer	0	Min: 0, Max: 4, Average: 0.937
age	Integer	0	Min: 29, Max: 77, Average: 54.439
sex	Nominal	0	Min: 0.0 (97), Max: 1.0 (206), Average: 1.0 (206), 0.0 (97)
cp	Nominal	0	Min: 1.0 (27), Max: 4.0 (144), Average: 4.0 (144), 3.0 (96), ... (2 more)
trestbps	Integer	0	Min: 94, Max: 200, Average: 131.690
chol	Integer	0	Min: 126, Max: 564, Average: 246.693
fbs	Integer	0	Min: 1.0 (45), Max: 0.0 (255), Average: 0.0 (255), 1.0 (45)
resteg	Integer	0	Min: 1.0 (46), Max: 0.0 (151), Average: 0.0 (151), 2.0 (140), ... (1 more)
thalach	Integer	0	Min: 71, Max: 202, Average: 149.657
exang	Integer	0	Min: 1.0 (98), Max: 0.0 (264), Average: 0.0 (264), 1.0 (98)
oldpeak	Integer	0	Min: 0, Max: 6, Average: 1.659
slope	Integer	0	Min: 3.0 (27), Max: 1.0 (142), Average: 1.0 (142), 2.0 (140), ... (1 more)
ca	Integer	4	Min: 0, Max: 3, Average: 0.672
thal	Integer	0	Min: 7 (2), Max: 3.0 (166), Average: 3.0 (166), 7.0 (117), ... (2 more)

Como se puede observar se destaca la casi nula presencia de valores faltantes, solo 4 en la columna ca. Los cuales representan un 0.1% del dataset, por lo tanto, se eliminarán.

A su vez podemos ver que los atributos tienen una relativa distribución gaussiana



- Obs1: La clase num (variable objetivo), está bastante equilibrada con 164 con NO padecieron una enfermedad contraria a los 139 que SI padecieron.
- Obs2: Usando un boxplot se ve que hay relativamente pocos outliers, salvo chol el cual tiene valores bastantes grandes



- Obs3: Se puede ver que en thal tenemos dos valores con “?” lo que significa que no son valores verdaderos, también se eliminara pues representa menos del 0.1%

## Visualización



Se puede observar como en general parece haber mas mujeres sin enfermedad que hombres, y que a su vez no depende tanto de la edad.

Lo cual en parte puede explicarse debido a que predominan mas hombres en el dataset que mujeres.



Otra forma de visualización, verde representa a las mujeres.



Realizando un scatterplot entre oldpeak y chol podemos notar que a medida que oldpeak aumenta más propensos son a tener una enfermedad cardiaca.



Se ve que a mayor Ca aumenta considerablemente

- Switzerland

Name	Type	Missing	Statistics
age	Integer	0	Min: 29, Max: 74, Average: 55.317
sex	Enumeral	0	Min: 1, Max: 2, Average: 1.1713, 0 (10)
cp	Nominal	0	Min: 1, Max: 4, Average: 4.086, 3 (17), 2 (more)
trestbps	Integer	2	Min: 80, Max: 200, Average: 130.207
chol	Integer	0	Min: 0, Max: 0, Average: 0
lbs	Nominal	0	Min: 1, Max: 7, Average: 7.775, 0 (43), 1 (5), 2 (1), 3 (1)
restecg	Nominal	0	Min: 1, Max: 7, Average: 0.085, 1 (30), 2 (1), 3 (1)
thalach	Integer	1	Min: 69, Max: 162, Average: 121.557
exang	Nominal	0	Min: 1, Max: 7, Average: 0.088, 1 (54), 2 (1), 3 (1)
oldpeak	Integer	0	Min: -3, Max: 4, Average: 0.759
slope	Nominal	0	Min: 1, Max: 2, Average: 2.081, 1 (33), 2 (more)
ca	Integer	118	Min: 1, Max: 2, Average: 1.658
thal	Nominal	0	Min: 1, Max: 7, Average: 7.522, 7 (42), 3 (1), 4 (1)
num	Integer	0	Min: 0, Max: 4, Average: 1.805

Obs1: faltan muchos datos, una columna no tiene registros (ca) lo cual afecta los datos, mas tras ver que en si es una variable que puede afectar y es condicionante.

Obs2: muchos valores faltantes en los demás con “?”

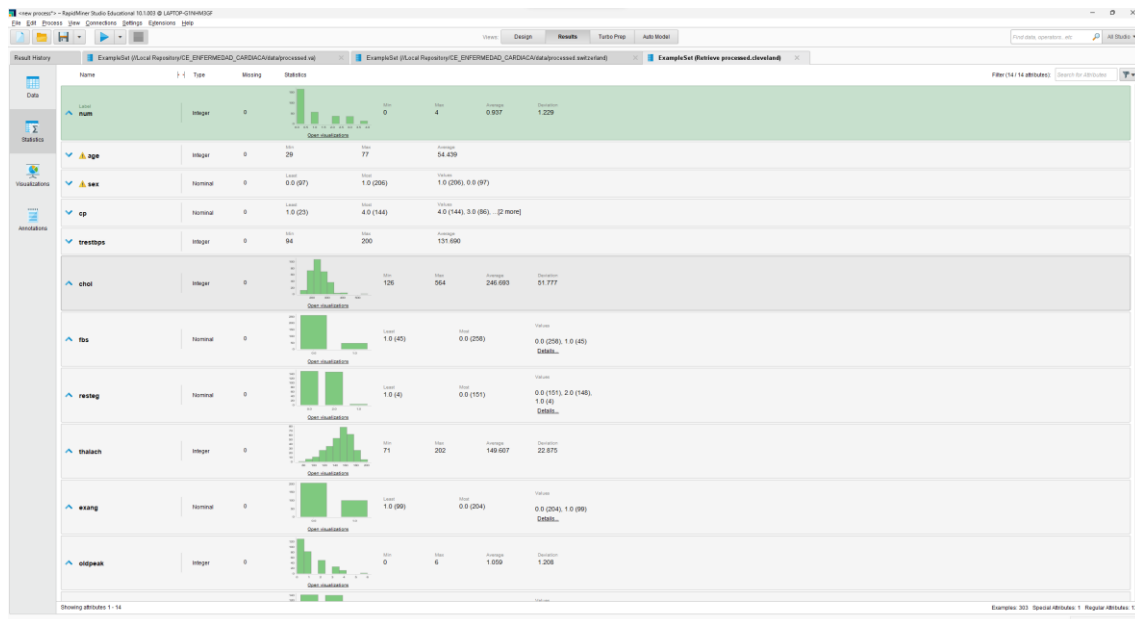
Conclusión: Se descarta el dataset para entrenar modelos, puesto que el tratamiento no puede ser el mejor, aun así, no se descarta el hecho de usarlo para evaluar performance del modelo final.

- Hungary

Name	Type	Missing	Statistics
age	Integer	0	Min: 29, Max: 66, Average: 47.627
sex	Enumeral	0	Min: 1, Max: 2, Average: 1.213, 0 (81)
cp	Nominal	0	Min: 1, Max: 4, Average: 4.123, 2 (106), 3 (more)
trestbps	Integer	0	Min: -9, Max: 250, Average: 132.102
chol	Integer	0	Min: -9, Max: 605, Average: 230.529
lbs	Nominal	0	Min: 1, Max: 9, Average: 0.206, 1 (20), 2 (1), 3 (1)
restecg	Nominal	0	Min: -9, Max: 7, Average: 0.235, 1 (52), 2 (more)
thalach	Integer	0	Min: -9, Max: 190, Average: 138.626
exang	Nominal	0	Min: 1, Max: 7, Average: 0.204, 1 (86), 2 (1), 3 (1)
oldpeak	Integer	0	Min: 0, Max: 5, Average: 0.622
slope	Nominal	0	Min: 1, Max: 2, Average: 2.081, 1 (33), 2 (more)
ca	Integer	9	Min: -9, Max: 9, Average: -8.847
thal	Nominal	0	Min: 1, Max: 7, Average: 7.522, 7 (42), 3 (1), 4 (1)
num	Integer	0	Min: 0, Max: 4, Average: 1.805

Obs1: Podemos ver que varios de hungary tienen valores faltantes en -9 lo cual hace que el dataset no sea útil para entrenar modelos.

- Va

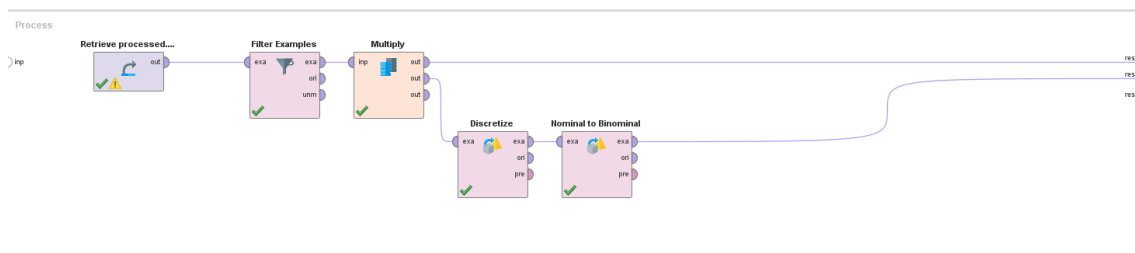


- Obs1: Igual que en los casos anteriores, este dataset tambien tiene muchos valores faltantes, por lo cual su uso podría perjudicar el modelo.

En resumen: Se puede observar que tanto Hungry, Switzerland y VA no son buenos para realizar un modelo debido a la gran cantidad de valores faltantes.

## Preparación de los datos:

Se van a aplicar los cambios definidos en la parte anterior.



Primero se quitan tanto outliers, como valores faltantes y luego se realiza un multiply para tener dos canales posibles: uno donde tenemos las cinco clases y otro donde tenemos solo dos.

## Modelado:

Se va a trabajar tanto sobre Cleveland, debido a que los otros 3 tienen muchos valores faltantes como para poder trabajar sobre ellos.

En esta sección se utilizarán 4 algoritmos de clasificación:

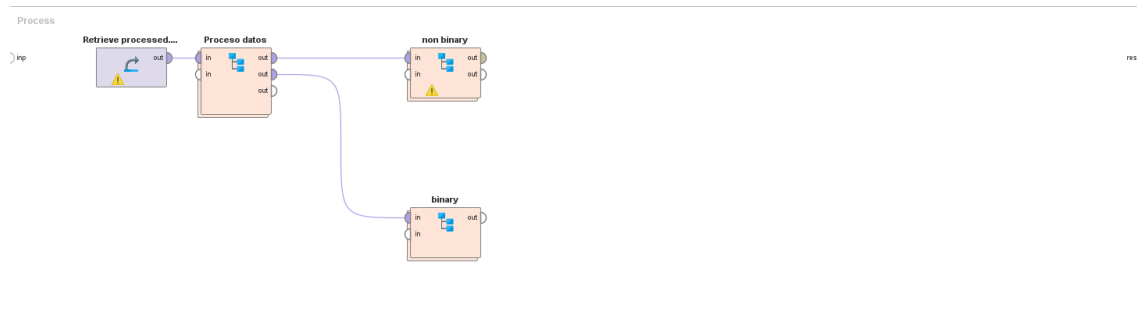
- Regresión logística
- KNN
- Naive Bayes



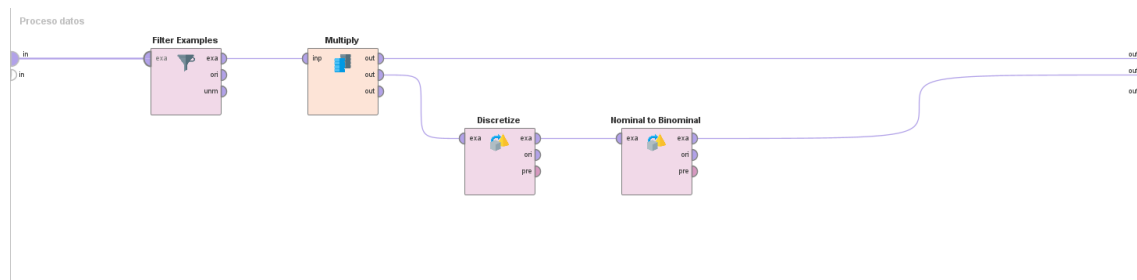
En cada modelo se utilizará la misma estructura, primero un tratamiento previo de datos en donde se trabajará los datos para así poder aplicarlos al algoritmo correspondiente. Luego se multiplicará el dataset para así tener cuatro canales diferentes uno para un cross validation común con los 14 atributos y los otros tres con feature selection. Se realizará tanto en Python como en RM.

Se probará como una clasificación binomial, así como una clasificación de 4 clases. Para comparar, aunque se entiende que el problema puede agruparse la variable de objetivo en dos (como se ve en preparación de los datos).

La estructura de RM para cada algoritmo es la siguiente:



Donde proceso de datos es el subproceso que realiza el siguiente tratamiento previo:

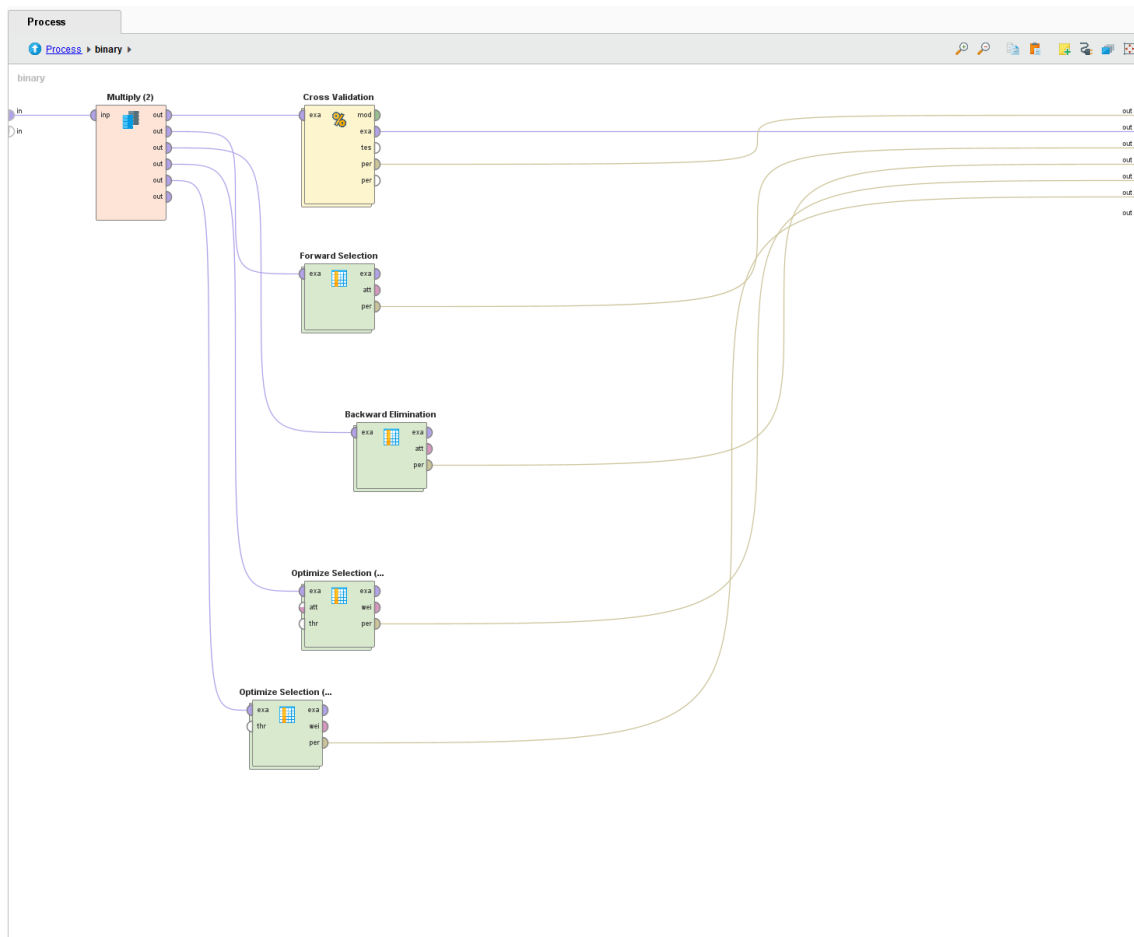


Luego el subproceso tanto de non binary como de binary se componen de:

- Cross validation
- Forward selection
- Backward Elimination
- Evolutionary

Los resultados se medirán en performance.

Dentro de cada proceso de modelo se vera de esta manera:



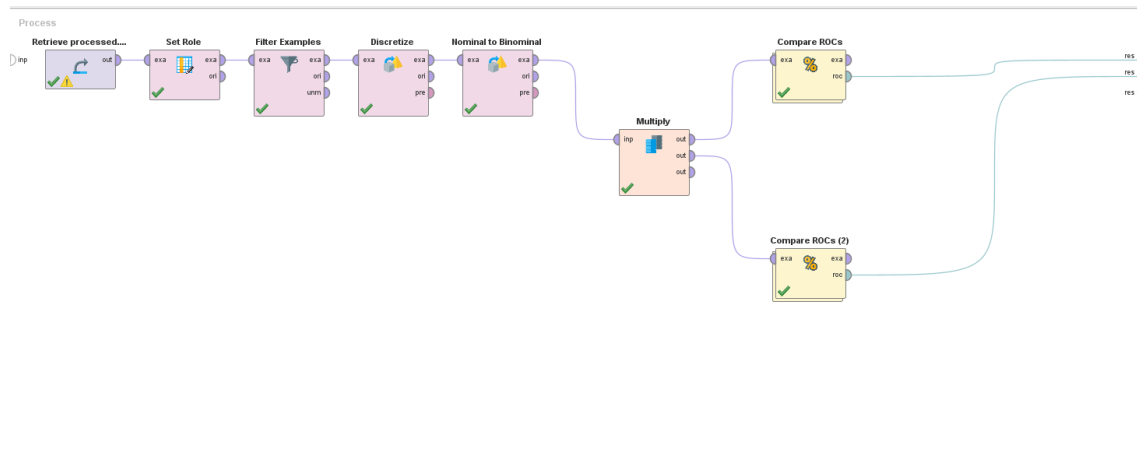
Bloque de cross validation:

1. Bloque del algoritmo del modelo: configuración por defecto
2. Bloque de Aplicar modelo
3. Bloque de performance por defecto

Foward y backward selection ambos tienen la configuración por defecto.

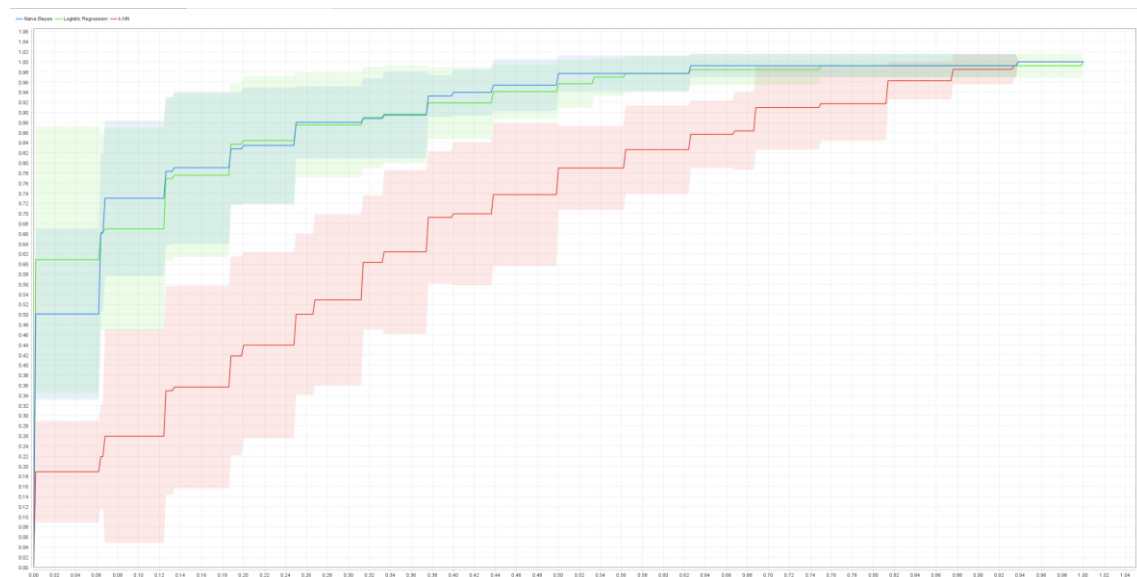
Optimize selection (Evolutionary) tiene 30 como número máximo de generaciones y 20 de tamaño de población.

Paso previo: Puesto que en un caso vamos a trabajar con una clasificación binaria se puede realizar una evaluación ROC. Para eso realizamos lo siguiente:

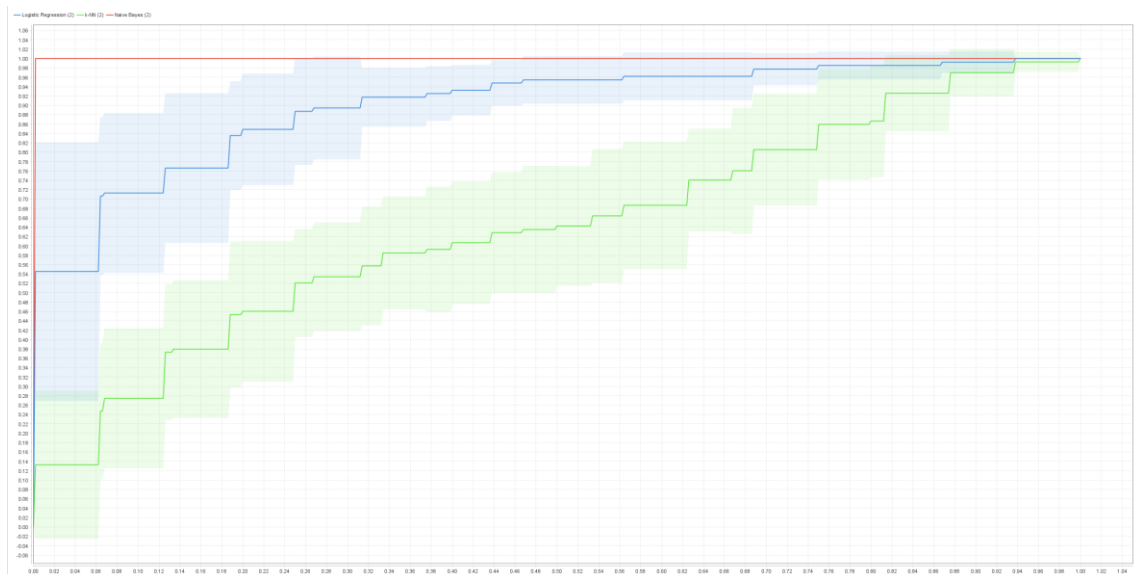


Tenemos dos canales uno con datos sin normalizar y el otro con datos normalizados. En cada ROC se comparará KNN, RL y NB.

Sin normalizar:



Con los datos normalizados:



- Obs1: Knn no esta correcto tras la normalización
- Obs2: En ambos casos la regresión logística parece ser el modelo mas efectivo.

### Regresión logística:

*Consideración importante: La regresión logística es un algoritmo de clasificación binaria, lo que significa que solo podemos tener dos clases para clasificar. Esto con nuestro dataset, no es posible debido a que hay cinco clases posibles para una fila. La decisión que se toma es realizar un Discretize y agrupar en dos clases diferentes. Por un lado, las NO = 0 y las SI = 1,2,3,4. Esto no afecta el modelo, puesto que es una agrupación de padecer una enfermedad o no padecer.*

Los resultados obtenidos son los siguientes

- Cross-Val

accuracy: 82.62% +/- 6.66% (micro average: 82.59%)

	true NO	true YES	class precision
pred NO	136	29	82.42%
pred YES	22	106	82.81%
class recall	85.08%	79.52%	

- Foward

accuracy: 85.33% +/- 8.99% (micro average: 85.32%)

	true NO	true YES	class precision
pred NO	138	23	85.71%
pred YES	20	112	84.95%
class recall	87.34%	82.98%	

- Backward

accuracy: 84.89% +/- 6.79% (micro average: 84.98%)

	true NO	true YES	class precision
pred NO	140	26	84.34%
pred YES	18	109	85.83%
class recall	88.61%	80.74%	

- Evolutionary

accuracy: 85.71% +/- 6.47% (micro average: 85.67%)

	true NO	true YES	class precision
pred NO	142	26	84.52%
pred YES	16	109	87.20%
class recall	89.87%	80.74%	

KNN:

## Binario

### - Cross-Val

accuracy: 56.89% +/- 6.38% (micro average: 56.89%)			
	true NO	true YES	class precision
pred NO	158	127	55.44%
pred YES	0	8	100.00%
class recall	100.00%	5.93%	

### - Foward

accuracy: 81.64% +/- 7.83% (micro average: 81.57%)			
	true NO	true YES	class precision
pred NO	132	28	82.50%
pred YES	29	107	80.45%
class recall	83.54%	79.29%	

### - Backward

accuracy: 59.40% +/- 10.04% (micro average: 59.39%)			
	true NO	true YES	class precision
pred NO	155	117	57.14%
pred YES	2	18	90.00%
class recall	98.73%	13.33%	

### - Evolutionary

accuracy: 59.67% +/- 9.39% (micro average: 59.73%)			
	true NO	true YES	class precision
pred NO	155	115	57.41%
pred YES	3	29	86.96%
class recall	98.10%	14.81%	

## No binario

### - Cross-Val

accuracy: 61.11% +/- 6.11% (micro average: 61.09%)						
	true 0	true 2	true 1	true 3	true 4	class precision
pred 0	142	5	23	5	1	80.00%
pred 2	1	13	7	7	3	41.54%
pred 1	9	11	12	9	2	27.91%
pred 3	5	4	11	12	6	30.77%
pred 4	0	1	1	2	0	0.00%
class recall	89.87%	38.24%	22.22%	34.29%	0.00%	

### - Foward

accuracy: 53.84% +/- 1.80% (micro average: 53.92%)						
	true 0	true 2	true 1	true 3	true 4	class precision
pred 0	158	34	54	35	12	53.92%
pred 2	0	0	0	0	0	0.00%
pred 1	0	0	0	0	0	0.00%
pred 3	0	0	0	0	0	0.00%
pred 4	0	0	0	0	0	0.00%
class recall	100.00%	0.00%	0.00%	0.00%	0.00%	

### - Backward

accuracy: 59.38% +/- 5.73% (micro average: 59.39%)

	true 0	true 2	true 1	true 3	true 4	class precision
pred. 0	151	17	36	16	8	66.23%
pred. 2	1	5	2	0	0	62.50%
pred. 1	0	10	12	13	3	27.27%
pred. 3	0	2	4	6	1	46.15%
pred. 4	0	0	0	0	9	0.00%
class recall	95.57%	14.71%	22.22%	17.14%	0.00%	

## - Evolutionary

accuracy: 53.94% +/- 1.83% (micro average: 53.92%)

	true 0	true 2	true 1	true 3	true 4	class precision
pred. 0	158	34	54	35	12	53.82%
pred. 2	0	0	0	0	0	0.00%
pred. 1	0	0	0	0	0	0.00%
pred. 3	0	0	0	0	0	0.00%
pred. 4	0	0	0	0	0	0.00%
class recall	100.00%	0.00%	0.00%	0.00%	0.00%	

## Naive Bayes:

### Binario

## - Cross-Val

accuracy: 46.06% +/- 1.83% (micro average: 46.08%)

	true NO	true YES	class precision
pred. NO	0	0	0.00%
pred. YES	158	135	46.08%
class recall	0.00%	100.00%	

## - Forward

accuracy: 82.94% +/- 6.71% (micro average: 82.94%)

	true NO	true YES	class precision
pred. NO	130	22	85.53%
pred. YES	28	113	80.14%
class recall	82.28%	83.70%	

## - Backward

accuracy: 82.83% +/- 7.72% (micro average: 82.59%)

	true NO	true YES	class precision
pred. NO	120	13	90.23%
pred. YES	38	122	76.25%
class recall	75.95%	90.37%	

## - Evolutionary

accuracy: 81.63% +/- 8.83% (micro average: 81.57%)

	true NO	true YES	class precision
pred. NO	128	24	84.21%
pred. YES	30	111	78.72%
class recall	81.01%	82.22%	

### No binario

## - Cross-Val

accuracy: 57.68% +/- 4.82% (micro average: 57.68%)

	true 0	true 2	true 1	true 3	true 4	class precision
pred. 0	134	6	25	4	0	79.29%
pred. 2	5	9	6	12	2	26.47%
pred. 1	17	11	15	8	5	26.79%
pred. 3	1	8	8	11	5	33.33%
pred. 4	1	0	0	0	0	0.00%
class recall	84.81%	26.47%	27.78%	31.43%	0.00%	

## - Forward

accuracy: 56.68% +/- 5.71% (micro average: 56.68%)

	true 0	true 2	true 1	true 3	true 4	class precision
pred 0	131	6	25	3	1	79.92%
pred 2	5	7	4	8	0	29.17%
pred 1	20	7	13	8	2	26.00%
pred 3	2	13	11	14	8	29.17%
pred 4	0	1	1	2	1	20.00%
class recall	82.91%	20.59%	24.07%	40.00%	8.33%	

## - Backward

accuracy: 60.11% +/- 5.84% (micro average: 60.07%)

	true 0	true 2	true 1	true 3	true 4	class precision
pred 0	139	9	27	9	3	74.33%
pred 2	0	0	0	0	0	0.00%
pred 1	9	13	18	7	3	36.00%
pred 3	10	12	9	19	6	33.83%
pred 4	0	0	0	0	0	0.00%
class recall	87.97%	0.00%	33.33%	54.29%	0.00%	

## - Evolutionary

accuracy: 11.94% +/- 2.40% (micro average: 11.95%)

	true 0	true 2	true 1	true 3	true 4	class precision
pred 0	0	0	0	0	0	0.00%
pred 2	0	0	0	0	0	0.00%
pred 1	0	0	0	0	0	0.00%
pred 3	158	34	54	35	12	11.95%
pred 4	0	0	0	0	0	0.00%
class recall	0.00%	0.00%	0.00%	100.00%	0.00%	

En Python se realizó el siguiente código disponible en: [https://github.com/RafaFil/ia-portafolio-docs/blob/main/CASOS%20DE%20ESTUDIO/Enfermedad%20Cardiaca/CE Enfermedad Cardiac a.ipynb](https://github.com/RafaFil/ia-portafolio-docs/blob/main/CASOS%20DE%20ESTUDIO/Enfermedad%20Cardiaca/CE%20Enfermedad%20Cardiaca.ipynb)

Resultados de Python:

Clasificación binaria:

	Model	Accuracy
3	LDA	0.875000
0	Logistic Regression	0.863636
2	Gaussian Naive Bayes	0.852273
4	Bernouli NB	0.818182
1	K Nearest Neighbors	0.613636

Clasificación no binaria:

	<b>Model</b>	<b>Accuracy</b>
<b>2</b>	LDA	0.590909
<b>1</b>	Gaussian Naive Bayes	0.579545
<b>0</b>	K Nearest Neighbors	0.534091
<b>3</b>	Multinomial NB	0.454545

#### Evaluación:

Knn para 4 clases se descarta puesto que el algoritmo no fue preciso para la clase num = 4, se estima que este error es debido a la falta de valores 4

Se llevo a los siguientes resultados (comparando entre py y rm), considerando aquellos mayores a 60.

LDA	87.5
LR	86.3
LR	85.7
LR	85.33
NB	85.2
LR	84.99
LR	82.94
LR	82.63
NB	82.62
NB	81.8
NB	81.63
KNN	81
KNN	61

#### Conclusión:

RapidMiner: La regresión logística fue la que obtuvo mejores resultados. KNN para 4 clases no es optimo puesto que al disminuir las clases y no tener una distribución uniforme de la misma.

Métodos de feature selection: aumentan el rendimiento y el mas efectivo en la gran mayoría de casos fue Foward selection. Aunque en los que Evolutionary no tenia baja performance el modelo mejoraba contra el mencionado anteriormente.

Python: LDA fue el mejor, pero la regresión logística dio un resultado muy similar, el cual considerando el margen de error podríamos decir que ambos tienen un rendimiento similar



En general:

Se puede decir que el mejor algoritmo para enfrentar este caso es la regresión logística si lo vemos como un problema de clasificación binaria, sino en todo caso LDA tiene una eficiencia de casi 60% lo cual considerando que la probabilidad es 20% si se escoge aleatoriamente una clase es para tener en cuenta. Mismo con naive bayes.

Los resultados coinciden con UCL, en donde se trataba a RL como el mejor algoritmo para este problema.

Se puede observar que se dio que el rendimiento del modelo fue mejor en la clasificación binaria que en la clasificación con varias clases, esto principalmente se puede deber a que la distribución de los datos con varias clases no era la más uniforme, teníamos muchísimos 0 menos 1 y así descendiendo hasta tener unos pocos 4.

Como mejora continua se podría partir de la base de la clasificación binaria y a partir de la respuesta de este modelo evaluarlo con uno de 5 clases.

Deploy:

Se utilizaría como modelo: LDA, RL, NB y se tomaría como una clasificación binaria.