Práctica 1
Análisis Predictivo Mediante Clasificación

Paula Villanueva Núñez

49314567Z

pvillanunez@correo.ugr.es

Cuarto Curso del Grado en Ingeniería Informática Curso 2021-2022 Grupo 1

Universidad de Granada

Índice

1 Introducción

En esta práctica abordaremos problemas reales de clasificación mediante el uso de algoritmos de aprendizaje supervisado de clasificación de forma que aportarán valor en forma de conocimiento para ayudar en la toma de decisiones. Se trabajarán con cuatro conjuntos de datos reales sobre los que se emplearán diferentes algoritmos de clasificación, para su comparación, y se examinarán las predicciones obtenidas y concluir estrategias para resolver cada problema.

Los problemas con los que se han trabajado combinan distintas propiedades, como la clasificación binaria y multiclase, clases balanceadas o no, atributos nominales y numéricos, existencia o no de valores perdidos, etc. Dichos problemas son los siguientes: Heart Failure Prediction, Mobile Price Classification, Bank Marketing y Tanzania water Pump que se describirán a continuación.

1.1 Conjuntos de datos

1.1.1 Heart Failure

Este dataset proporciona una predicción sobre si una persona puede sufrir una insuficiencia cardíaca.

De esta forma, este problema tendrá dos clases diferentes a clasificar (0: normal, 1: insuficiencia cardíaca) a partir de unas variables, como la edad, el sexo, el tipo de dolor de pecho, la presión arterial en reposo, el colesterol... de cada persona. Con esta predicción, se podrá ayudar a detectar y tratar de forma precoz las enfermedades relacionadas con el corazón.

Este conjunto de datos tiene 918 instancias clasificadas según si el corazón es normal, 0, o si tiene una insuficiencia cardíaca, 1, de la siguiente forma.

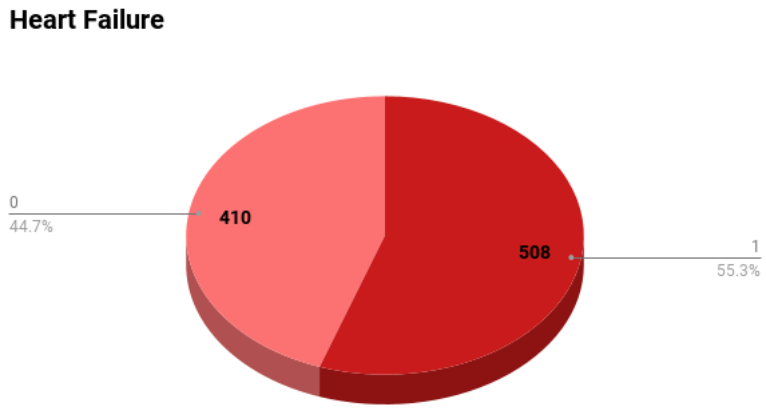

Figura 1: Porcentaje de instancias clasificadas de Heart Failure

Observamos que las clases están desbalanceadas, más de la mitad (un 55.3%) tienen insuficiencia cardíaca y el resto, un 44.7%, tienen el corazón normal. Además, posee valores desconocidos en la variable *Cholesterol*, donde aparecen ceros, por lo que he decidido sustituirlos por interrogaciones para que el programa los detecte debidamente.

Para este conjunto de datos consideraremos la clase positiva "1", pues nos interesa saber si una persona podría desarrollar alguna insuficiencia cardíaca.

1.1.2 Mobile Price

Este dataset proporciona una predicción del rango del precio indicando cómo de alto es el precio.

Este problema tendrá cuatro clases diferentes a clasificar: 0, 1, 2 o 3; ordenando de menor a mayor dicho precio. Cada instancia tiene una serie de variables, como la energía de la batería, bluetooth, megapíxeles de la cámara, 4G, memoria interna... Con esta predicción, podremos estimar el precio de los móviles que una compañía produce.

Este conjunto de datos tiene 2000 instancias clasificadas de 0 a 3, en orden ascendente del precio, de la siguiente forma.

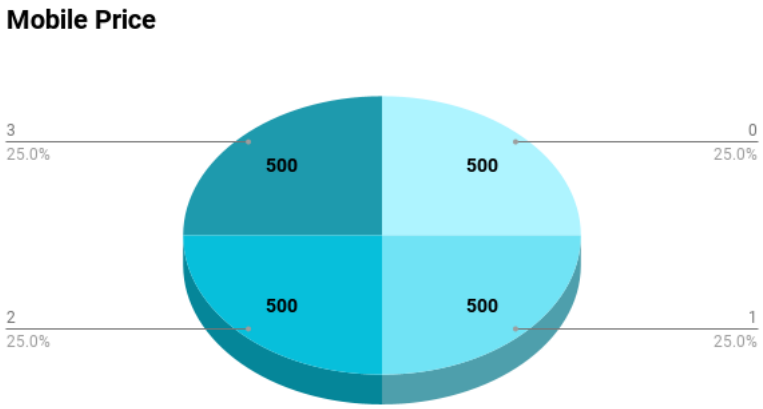

Figura 2: Porcentaje de instancias clasificadas de Mobile Price

Observamos que las clases están perfectamente balanceadas, cada clase posee el mismo número de instancias. Además, tampoco observamos valores perdidos.

Para este conjunto de datos, como las clases están perfectamente balanceadas, hemos agrupado las clases en dos grupos. Una primera clase, 0, que tendrá las clases 0 y 1; y la segunda clase, 2, que tendrá las clases 2 y 3.

1.1.3 Bank Marketing

Los datos están relacionados con campañas de marketing de una institución bancaria portuguesa. Este dataset proporciona una predicción sobre si una persona se subscribiría a un depósito a plazo.

De esta forma, este problema tendrá dos clases diferentes a clasificar (sí o no) a partir de unas variables, como la edad, el trabajo, el estado civil, el nivel de educación, la hipoteca, los préstamos... que tiene cada persona.

Este conjunto de datos tiene 41188 instancias clasificadas según si la persona se subscribiría o no de la siguiente forma.

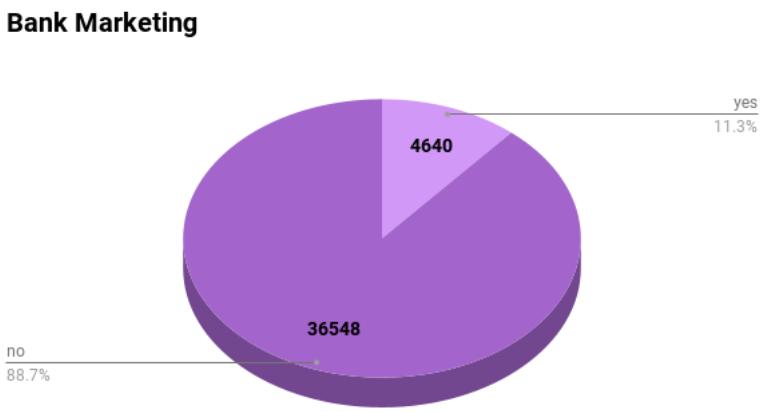

Figura 3: Porcentaje de instancias clasificadas de Bank Marketing

Observamos que hay un desbalanceo bastante notable entre las distintas clases, la mayoría (un 88.7%) responden que "no", mientras que apenas el 11.3% responden que "sí"se subscribirían a un depósito a plazo. Además, se contemplan valores desconocidos, *unknown*.

Para este conjunto de datos consideraremos la clase positiva zes", pues nos interesa saber quién sí se subscribiría a dicho depósito a plazo.

1.1.4 Tanzania Water Pump

Este dataset proporciona una predicción sobre si una bomba de agua funciona o no.

Este problema tendrá dos clases diferentes a clasificar (functional o non functional) a partir de unas variables, como la altura, la fecha, el financiador, la altitud, la organización, el nombre, la región, la población de alrededor... de cada nomba de agua. Con esta predicción, podremos saber qué puntos de agua fallarán y así podremos mejorar las operaciones de mantenimiento y garantizar que se disponga de agua potable y limpia.

Este conjunto de datos tiene 55083 instancias clasificadas según si es funcional o no de la siguiente forma.

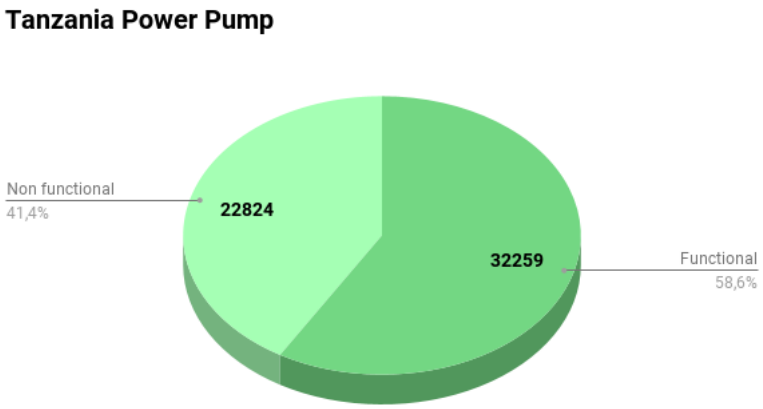

Figura 4: Porcentaje de instancias clasificadas de Tanzania Power Pump

Observamos que hay un desbalanceo entre las distintas clases, más de la mitad (un 58.6%) son

bombas de agua funcionales, mientras que el 41.4% no son funcionales. Además, se perciben valores desconocidos, *unknown*, y valores perdidos, *?*.

Para este conjunto de datos consideraremos la clase positiva "non functional", pues nos interesa saber qué bombas de agua no funcionan y así poder mejorar las operaciones de mantenimiento.

1.2 Diseño experimental

Para realizar esta práctica, se han considerado cinco algoritmos de clasificación: árbol de decisión, k-NN, Naive Bayes, Random Forest y XGBoost.

Toda la experimentación se ha realizado con validación cruzada de 5 particiones y la semilla aleatoria utilizada es 4567. Para ello, se han empleado los nodos de KNIME `X-Partitioner` y `X-Aggregator`.

De esta forma, Heart Failure tendrá un conjunto de entrenamiento de tamaño 735 y un conjunto de prueba de tamaño 183. Por otra parte, Mobile Price tendrá un conjunto de entrenamiento de tamaño 1600 y un conjunto de prueba de tamaño 400. De la misma forma, Bank Marketing tendrá un entrenamiento de tamaño 32951 y un conjunto de prueba de tamaño 8237. Por último, Tanzania Water Pump tendrá un conjunto de entrenamiento de tamaño 44067 y un conjunto de prueba de tamaño 11016.

Además, para sustentar el análisis comparativo se han empleado tablas de errores, matrices de confusión y curvas ROC. Además de la precisión, se han añadido las medidas de rendimiento TPR, TNR, Valor-F_1, G-mean y AUC así como medidas de complejidad del modelo.

2 Resultados obtenidos

En esta sección se muestran los algoritmos estudiados junto con el flujo de trabajo empleado y una tabla con los resultados obtenidos por el algoritmo en todos los problemas.

2.1 Árbol de decisión

Un **árbol de decisión** es un clasificador que en función de un conjunto de atributos permite determinar a qué clase pertenece el caso objeto de estudio. Esto es, divide el conjunto de ejemplos según el valor de unos atributos seleccionados previamente. El criterio de selección de variables elegido es GINI (CART), es decir, si un conjunto de datos T tiene ejemplos pertenecientes a n clases, el índice Gini se define como

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2$$

donde p_j es la frecuencia relativa de la clase j en T.

Se elige para dividir el nodo, el atributo que proporciona el índice $gini(T)$ más pequeño.

Los árboles de decisión tratan bien los datos con ruido. Sin embargo, no manejan de forma fácil los atributos continuos y tienen dificultad para trabajar con valores perdidos.

El flujo de trabajo de este algoritmo en KNIME es el siguiente, común para todos los datasets.

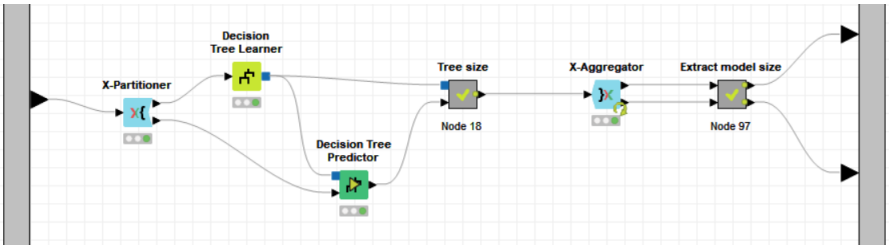

Figura 5: Metanodo Árbol de decisión

Se han utilizado los nodos `Decision Tree Learner`, usando Gini, y `Decision Tree Predictor`.

No ha sido necesario tratar los valores perdidos o desconocidos que presentaban algunos conjuntos de datos. Sin embargo, los datasets Heart Failure y Mobile Price han necesitado transformar su clase a un valor nominal. El resto de atributos pueden ser nominales o numéricos.

Con respecto a los valores numéricos, como vimos en teoría, los árboles de decisión no manejan de forma fácil los atributos continuos. El nodo `Decision Tree Learner`, para resolver este problema, divide el dominio en dos subconjuntos para tratarlos de forma categórica.

Además, el nodo `Decision Tree Predictor` nos proporciona el árbol obtenido, que es fácil de interpretar.

Por otra parte, se han creado dos metanodos, `Tree size` para calcular el tamaño del árbol obtenido y `Extract model size` para representarlo en una tabla.

En el siguiente apartado veremos los resultados obtenidos de este algoritmo en cada conjunto de datos, detallando más la tabla que se muestra a continuación.

Tabla 1: Criterios de precisión del árbol de decisión

Árbol de decisión en	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	Model size	AUC
Heart Failure	422	87	323	86	0.8307	0.7878	0.8290	0.8115	0.8299	0.8090	69.2	0.8157
Mobile Price	947	59	941	53	0.947	0.941	0.9414	0.944	0.9442	0.9439	36	0.9502
Bank Marketing	2284	1864	34387	2149	0.5152	0.9486	0.5506	0.9014	0.5323	0.6991	2686.2	0.7207
Tanzania Power Pump	17748	4520	27493	4871	0.7847	0.8588	0.7970	0.8281	0.7908	0.8209	4882	0.8385

2.1.1 Heart Failure

En este dataset ha sido necesario usar el nodo `Number to String` para transformar la clase `HeartDisease` a un `String`, pues así lo requiere este algoritmo.

La matriz de confusión obtenida es la siguiente.

Tabla 2: Matriz de confusión del árbol de decisión de Heart Failure

Heart Disease	0	1
0	329	81
1	88	420

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 3: Criterios de precisión del árbol de decisión de Heart Failure

Hear Failure	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	Model size	AUC
Árbol de decisión	422	87	323	86	0.8307	0.7878	0.8290	0.8115	0.8299	0.8090	69.2	0.8157

2.1.2 Mobile Price

En este dataset ha sido necesario usar el nodo `Number to String` para transformar la clase `price_range` a un `String`, pues así lo requiere este algoritmo.

La matriz de confusión obtenida es la siguiente.

Tabla 4: Matriz de confusión del árbol de decisión de Mobile Price

Mobile Price	0	2
0	941	59
2	53	947

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 5: Criterios de precisión del árbol de decisión de Mobile Price

Mobile Price	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	Model size	AUC
Árbol de decisión	947	59	941	53	0.947	0.941	0.9414	0.944	0.9442	0.9439	36	0.9502

2.1.3 Bank Marketing

La matriz de confusión obtenida es la siguiente.

Tabla 6: Matriz de confusión del árbol de decisión de Bank Marketing

Bank Marketing	no	yes
no	34387	1864
yes	2149	2284

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 7: Criterios de precisión del árbol de decisión de Bank Marketing

Bank Marketing	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	Model size	AUC
Árbol de decisión	2284	1864	34387	2149	0.5152	0.9486	0.5506	0.9014	0.5323	0.6991	2686.2	0.7207

2.1.4 Tanzania Water Pump

La matriz de confusión obtenida es la siguiente.

Tabla 8: Matriz de confusión del árbol de decisión de Tanzania Water Pump

Tanzania Water Pump	functional	non functional
functional	27493	4520
non functional	4871	17748

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 9: Criterios de precisión del árbol de decisión de Tanzania Water Pump

Tanzania Water Pump	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	Model size	AUC
Árbol de decisión	17748	4520	27493	4871	0.7847	0.8588	0.7970	0.8281	0.7908	0.8209	4882	0.8385

2.2 k-NN

Como segundo algoritmo se ha elegido el clasificador del vecino más cercano (**k-NN**). Su proceso de aprendizaje consiste en almacenar una tabla con los ejemplos disponibles, junto a la clase asociada a cada uno de ellos. Ante un nuevo ejemplo a clasificar, se calcula su distancia (euclídea) con respecto a los n ejemplos existentes en la tabla, y se consideran los k más cercanos. El nuevo ejemplo se clasifica según la clase mayoritaria de los k ejemplos más cercanos. Para aplicar este algoritmo, es necesario que los atributos estén discretizados y normalizados en $[0, 1]$ para no priorizarlos sobre otros.

En todos los conjuntos de datos se ha utilizado el nodo `K Nearest Neighbor` tomando el número de vecinos $k = 3$. Además, en todos los conjuntos de datos se ha utilizado el nodo `Column Rename` puesto que el anterior nodo definía la columna de predicción como `Class [kNN]`. De esta forma dicha columna pasará a llamarse al igual que las otras en los distintos algoritmos, `Prediction ( class)`.

En algunos casos ha sido necesario tratar los valores perdidos que presentaban algunos conjuntos de datos, incluso ha sido imprescindible discretizar y normalizar los atributos en algunos datasets.

En el siguiente apartado veremos los resultados obtenidos de este algoritmo en cada conjunto de datos, detallando más la tabla que se muestra a continuación.

Tabla 10: Criterios de precisión de 3-NN

3-NN en	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Heart Failure	452	83	327	56	0.8898	0.7976	0.8449	0.8486	0.8667	0.8424	0.8741
Mobile Price	790	204	796	210	0.79	0.796	0.7947	0.793	0.7923	0.7929	0.8475
Bank Marketing	1438	1347	35201	3202	0.3099	0.9631	0.5163	0.8896	0.3873	0.5463	0.7400
Tanzania Power Pump	16480	4781	27478	6344	0.7220	0.8518	0.7751	0.7980	0.7476	0.7842	0.8480

2.2.1 Heart Failure

En este conjunto ha sido necesario, antes de aplicar este clasificador, el nodo `Number to String` para transformar `HeartDisease` a un `String`. También se han usado los nodos `Category To Number` y `Normalizer` para transformar el resto de atributos a valores numéricos normalizados. Además, como este dataset posee valores perdidos, también ha sido necesario utilizar los nodos `Missing Value` y `Missing Value (apply)` para tratarlos, se ha utilizado la mediana para los valores numéricos y el valor más frecuente para los nominales. El flujo queda como muestra la siguiente figura.

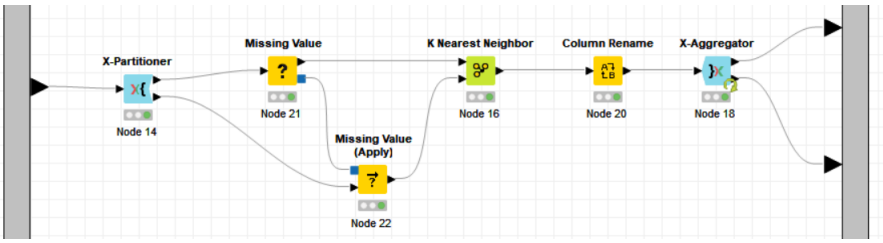

Figura 6: Metanodo k-NN de Heart Failure

La matriz de confusión obtenida es la siguiente.

Tabla 11: Matriz de confusión de 3-NN de Heart Failure

Heart Disease	0	1
0	329	81
1	58	450

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 12: Criterios de precisión de 3-NN de Heart Failure

Hear Failure	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
3-NN	452	83	327	56	0.8898	0.7976	0.8449	0.8486	0.8667	0.8424	0.8741

2.2.2 Mobile Price

En este conjunto ha sido necesario, antes de aplicar este clasificador, el nodo `Number to String` para transformar `price_range` a un `String`. Como el resto de atributos eran numéricos, solo ha

sido necesario utilizar el nodo `Normalizer` para normalizarlos. Como no posee valores perdidos, no ha sido necesario ningún procesamiento adicional. El flujo queda como muestra la siguiente figura.

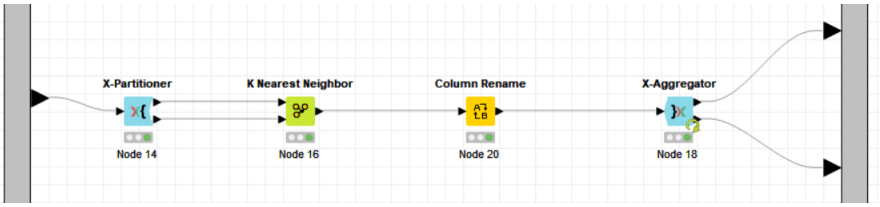

Figura 7: Metanodo k-NN de Mobile Price

La matriz de confusión obtenida es la siguiente.

Tabla 13: Matriz de confusión de 3-NN de Mobile Price

Mobile Price	0	2
0	796	204
2	210	790

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 14: Criterios de precisión de 3-NN de Mobile Price

Mobile Price	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
3-NN	790	204	796	210	0.79	0.796	0.7947	0.793	0.7923	0.7929	0.8475

2.2.3 Bank Marketing

Para este conjunto de datos se han usado los nodos `Category To Number` y `Normalizer` para transformar el resto de atributos a valores numéricos normalizados. Al igual que el conjunto de datos anterior, no ha sido necesario tratar los valores perdidos. Por lo que el flujo de trabajo es igual a la figura 7.

La matriz de confusión obtenida es la siguiente.

Tabla 15: Matriz de confusión de 3-NN de Bank Marketing

Bank Marketing	no	yes
no	35201	1347
yes	3202	1438

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 16: Criterios de precisión de 3-NN de Bank Marketing

Bank Marketing	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
3-NN	1438	1347	35201	3202	0.3099	0.9631	0.5163	0.8896	0.3873	0.5463	0.7400

2.2.4 Tanzania Water Pump

Para este conjunto de datos se han usado los nodos `Category To Number` y `Normalizer` para transformar el resto de atributos a valores numéricos normalizados. Al igual que el conjunto de datos Heart Failure, también posee valores perdidos y ha sido necesario utilizar los nodos `Missing Value` y `Missing Value (apply)` para tratarlos, se ha utilizado la mediana para los valores numéricos y el valor más frecuente para los nominales. El flujo es igual al de la figura 6

La matriz de confusión obtenida es la siguiente.

Tabla 17: Matriz de confusión de 3-NN de Tanzania Water Pump

Tanzania Water Pump	functional	non functional
functional	27478	4781
non functional	6344	16480

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 18: Criterios de precisión de 3-NN de Tanzania Water Pump

Tanzania Water Pump	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
3-NN	16480	4781	27478	6344	0.7220	0.8518	0.7751	0.7980	0.7476	0.7842	0.8480

2.3 Naïve Bayes

El **teorema de Bayes** orientado a un problema de clasificación con n variables tiene la siguiente expresión

$$P(C|A_1, ..., A_n) = \frac{P(A_1, ..., A_n|C)P(C)}{P(A_1, ..., A_n)}$$

El clasificador **Naïve Bayes** es el modelo de red bayesiana orientada a clasificación más simple. Supone que todos los atributos son independientes conocida la variable clase y calcula la clase más probable condicionando el resto de atributos.

El flujo de trabajo de este algoritmo en KNIME es el siguiente, común para todos los datasets.

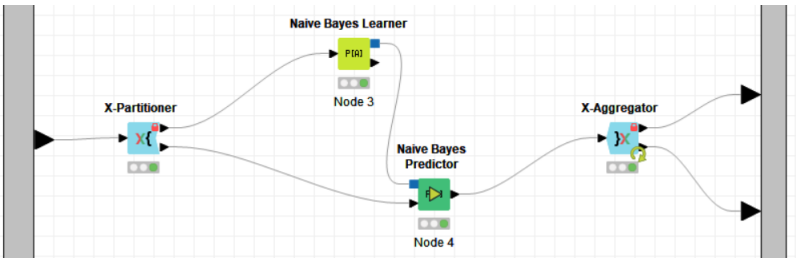

Figura 8: Metanodo Naive Bayes

En todos los conjuntos de datos se han utilizado los nodos `Naive Bayes Learner` y `Naive Bayes Predictor`.

En algunos casos ha sido necesario transformar la clase principal a un valor nominal, puesto que este algoritmo así lo requiere. Sin embargo, algunos conjuntos de datos poseen valores perdidos o desconocidos y no ha sido necesario tratarlos, pues el algoritmo es capaz de trabajar con ellos.

En el siguiente apartado veremos los resultados obtenidos de este algoritmo en cada conjunto de datos, detallando más la tabla que se muestra a continuación.

Tabla 19: Criterios de precisión de Naïve Bayes

Naïve Bayes en	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Heart Failure	439	68	342	69	0.8642	0.8341	0.8659	0.8508	0.8650	0.8490	0.9202
Mobile Price	926	75	925	74	0.926	0.925	0.9250	0.9255	0.9255	0.9254	0.9763
Bank Marketing	2510	3565	32983	2130	0.5409	0.9025	0.4132	0.8617	0.4685	0.6987	0.8337
Tanzania Power Pump	15125	6716	25543	7699	0.6627	0.7918	0.6925	0.7383	0.6773	0.7244	0.8040

2.3.1 Heart Failure

Ha sido necesario transformar la clase principal `HeartDisease` a un valor nominal con el nodo `Number to String`.

La matriz de confusión obtenida es la siguiente.

Tabla 20: Matriz de confusión de Naive Bayes de Heart Failure

Heart Disease	0	1
0	342	68
1	69	439

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 21: Criterios de precisión de Naive Bayes de Heart Failure

Hear Failure	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Naive Bayes	439	68	342	69	0.8642	0.8341	0.8659	0.8508	0.8650	0.8490	0.9202

2.3.2 Mobile Price

Ha sido necesario transformar la clase principal `price_range` a un valor nominal con el nodo `Number to String`.

La matriz de confusión obtenida es la siguiente.

Tabla 22: Matriz de confusión de Naive Bayes de Mobile Price

Mobile Price	0	2
0	925	75
2	74	926

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 23: Criterios de precisión de Naive Bayes de Mobile Price

Mobile Price	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Naive Bayes	926	75	925	74	0.926	0.925	0.9250	0.9255	0.9255	0.9254	0.9763

2.3.3 Bank Marketing

La matriz de confusión obtenida es la siguiente.

Tabla 24: Matriz de confusión de Naive Bayes de Bank Marketing

Bank Marketing	no	yes
no	32983	3565
yes	2130	2510

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 25: Criterios de precisión de Naive Bayes de Bank Marketing

Bank Marketing	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Naive Bayes	2510	3565	32983	2130	0.5409	0.9025	0.4132	0.8617	0.4685	0.6987	0.8337

2.3.4 Tanzania Water Pump

La matriz de confusión obtenida es la siguiente.

Tabla 26: Matriz de confusión de Naive Bayes de Tanzania Water Pump

Tanzania Water Pump	functional	non functional
functional	25543	6716
non functional	7699	15125

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 27: Criterios de precisión de Naive Bayes de Tanzania Water Pump

Tanzania Water Pump	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Naive Bayes	15125	6716	25543	7699	0.6627	0.7918	0.6925	0.7383	0.6773	0.7244	0.8040

2.4 Random Forest

Random Forest es un multiclasificador que consta de varios árboles de decisión. Cada uno se construye con un conjunto diferente de filas y para cada división dentro de un árbol se elige al azar un conjunto de columnas. Los conjuntos de filas para cada árbol de decisión se crean mediante bootstrapping y tienen el mismo tamaño que la tabla de entrada original. Es una modificación de bagging.

El flujo de trabajo de este algoritmo en KNIME es el siguiente, común para todos los datasets (salvo Tanzania Water Pump).

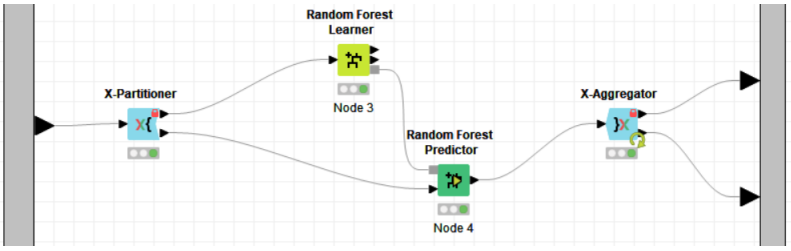

Figura 9: Metanodo Random Forest

Se han utilizado los nodos `Random Forest Learner`, con 100 modelos y usando el índice Gini, y `Random Forest Predictor`.

No ha sido necesario tratar los valores perdidos o desconocidos que presentaban algunos conjuntos de datos. Sin embargo, los datasets Heart Failure y Mobile Price han necesitado transformar su clase a un valor nominal.

En el siguiente apartado veremos los resultados obtenidos de este algoritmo en cada conjunto de datos, detallando más la tabla que se muestra a continuación.

Tabla 28: Criterios de precisión de Random Forest

Random Forest en	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Heart Failure	449	70	340	59	0.8839	0.8293	0.8651	0.8595	0.8744	0.8561	0.9249
Mobile Price	947	42	958	53	0.947	0.958	0.9575	0.9525	0.9522	0.9524	0.9915
Bank Marketing	2044	949	35599	2596	0.4405	0.9740	0.6829	0.9139	0.5356	0.6550	0.9433
Tanzania Power Pump	8905	670	31589	13919	0.3902	0.9792	0.9300	0.7351	0.5497	0.6181	0.8540

2.4.1 Heart Failure

La matriz de confusión obtenida es la siguiente.

Tabla 29: Matriz de confusión de Random Forest de Heart Failure

Heart Disease	0	1
0	340	70
1	59	449

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 30: Criterios de precisión de Random Forest de Heart Failure

Hear Failure	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Random Forest	449	70	340	59	0.8839	0.8293	0.8651	0.8595	0.8744	0.8561	0.9249

2.4.2 Mobile Price

La matriz de confusión obtenida es la siguiente.

Tabla 31: Matriz de confusión de Random Forest de Mobile Price

Mobile Price	0	2
0	958	42
2	53	947

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 32: Criterios de precisión de Random Forest de Mobile Price

Mobile Price	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Random Forest	947	42	958	53	0.947	0.958	0.9575	0.9525	0.9522	0.9524	0.9915

2.4.3 Bank Marketing

La matriz de confusión obtenida es la siguiente.

Tabla 33: Matriz de confusión de Random Forest de Bank Marketing

Bank Marketing	no	yes
no	35599	949
yes	2596	2044

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 34: Criterios de precisión de Random Forest de Bank Marketing

Bank Marketing	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Random Forest	2044	949	35599	2596	0.4405	0.9740	0.6829	0.9139	0.5356	0.6550	0.9433

2.4.4 Tanzania Water Pump

En este conjunto de datos ha sido necesario utilizar el nodo `Domain Calculator`, pues el algoritmo lo necesitaba para poder tener en cuenta todas las variables. De esta forma, el flujo de datos se ha modificado como muestra la siguiente imagen.

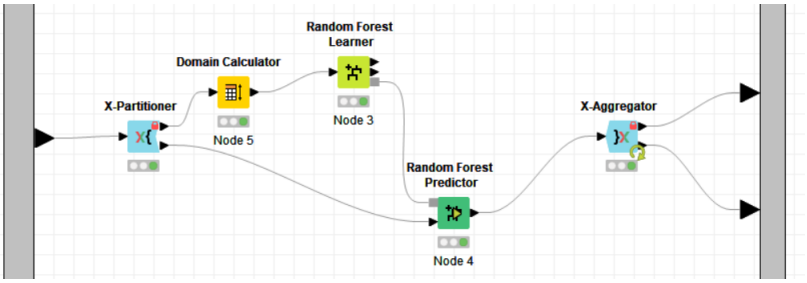

Figura 10: Metanodo Random Forest de Tanzania Water Pump

La matriz de confusión obtenida es la siguiente.

Tabla 35: Matriz de confusión de Random Forest de Tanzania Water Pump

Tanzania Water Pump	functional	non functional
functional	31589	670
non functional	13919	8905

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 36: Criterios de precisión de Random Forest de Tanzania Water Pump

Tanzania Water Pump	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Random Forest	8905	670	31589	13919	0.3902	0.9792	0.9300	0.7351	0.5497	0.6181	0.8540

2.5 XGBoost

XGBoost es un multiclasificador basado en árboles para la clasificación. Se basa en un algoritmo de boosting, es decir, se tiene en cuenta los fallos del anterior clasificador.

El flujo de trabajo de este algoritmo en KNIME es el siguiente, común para todos los datasets.

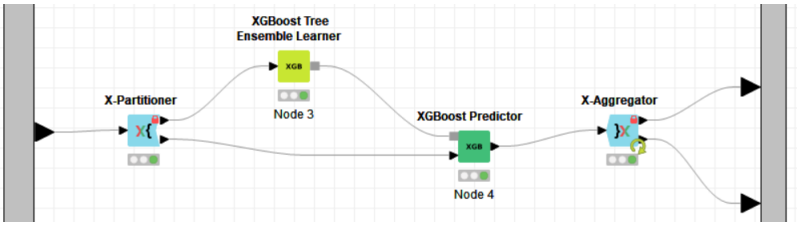

Figura 11: Metanodo XGBoost

Se han utilizado los nodos `XGBoost Tree Ensemble Learner`, usando Gini, y `XGBoost Predictor` .

No ha sido necesario tratar los valores perdidos o desconocidos que presentaban algunos conjuntos de datos. Sin embargo, en algunos casos ha sido necesario transformar los valores de los atributos categóricos a numéricos y/o transformar su clase principal a un valor nominal.

En el siguiente apartado veremos los resultados obtenidos de este algoritmo en cada conjunto de datos, detallando más la tabla que se muestra a continuación.

Tabla 37: Criterios de precisión de XGBoost

XGBoost en	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Heart Failure	453	75	335	55	0.8917	0.8171	0.8580	0.8584	0.8745	0.8536	0.9215
Mobile Price	968	30	970	32	0.968	0.97	0.9699	0.969	0.9689	0.9689	0.9963
Bank Marketing	2514	1408	35140	2126	0.5418	0.9615	0.6410	0.9142	0.5872	0.7218	0.9473
Tanzania Power Pump	17322	2720	29539	5502	0.759	0.9157	0.8643	0.8507	0.8082	0.8336	0.9208

2.5.1 Heart Failure

En este dataset ha sido necesario usar el nodo `Number to String` para transformar la clase `HeartDisease` a un `String`, pues así lo requiere este algoritmo. También se ha usado el nodo `Category To Number` para transformar el resto de atributos a valores numéricos.

La matriz de confusión obtenida es la siguiente.

Tabla 38: Matriz de confusión de XGBoost de Heart Failure

Heart Disease	0	1
0	335	75
1	55	453

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 39: Criterios de precisión de XGBoost de Heart Failure

Hear Failure	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
XGBoost	453	75	335	55	0.8917	0.8171	0.8580	0.8584	0.8745	0.8536	0.9215

2.5.2 Mobile Price

En este dataset ha sido necesario usar el nodo `Number to String` para transformar la clase `HeartDisease` a un `String`, pues así lo requiere este algoritmo.

La matriz de confusión obtenida es la siguiente.

Tabla 40: Matriz de confusión de XGBoost de Mobile Price

Mobile Price	0	2
0	970	30
2	32	968

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 41: Criterios de precisión de XGBoost de Mobile Price

Mobile Price	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
XGBoost	968	30	970	32	0.968	0.97	0.9699	0.969	0.9689	0.9689	0.9963

2.5.3 Bank Marketing

Se ha usado el nodo `Category To Number` para transformar el resto de atributos a valores numéricos.

La matriz de confusión obtenida es la siguiente.

Tabla 42: Matriz de confusión de XGBoost de Bank Marketing

Bank Marketing	no	yes
no	35140	1408
yes	2126	2514

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 43: Criterios de precisión de XGBoost de Bank Marketing

Bank Marketing	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
XGBoost	2514	1408	35140	2126	0.5418	0.9615	0.6410	0.9142	0.5872	0.7218	0.9473

2.5.4 Tanzania Water Pump

Se ha usado el nodo `Category To Number` para transformar el resto de atributos a valores numéricos.

La matriz de confusión obtenida es la siguiente.

Tabla 44: Matriz de confusión de XGBoost de Tanzania Water Pump

Tanzania Water Pump	functional	non functional
functional	29539	2720
non functional	5502	17322

En la siguiente tabla se puede contemplar la interpretación de dicha matriz.

Tabla 45: Criterios de precisión de XGBoost de Tanzania Water Pump

Tanzania Water Pump	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
XGBoost	17322	2720	29539	5502	0.759	0.9157	0.8643	0.8507	0.8082	0.8336	0.9208

3 Análisis de resultados

Para analizar los resultados obtenidos en la sección anterior, realizaremos un análisis comparativo en cada conjunto de datos estudiado.

3.1 Heart Failure

En la siguiente tabla se muestran los criterios de precisión del dataset Heart Failure que hemos obtenido con los algoritmos que hemos aplicado.

Tabla 46: Criterios de precisión de Heart Failure

Heart Failure	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	Model size	AUC
Árbol de decisión	422	87	323	86	0.8307	0.7878	0.8290	0.8115	0.8299	0.8090	69.2	0.8157
3-NN	452	83	327	56	0.8898	0.7976	0.8449	0.8486	0.8667	0.8424	NaN	0.8741
Naive Bayes	439	68	342	69	0.8642	0.8341	0.8659	0.8508	0.8650	0.8490	NaN	0.9202
Random Forest	449	70	340	59	0.8839	0.8293	0.8651	0.8595	0.8744	0.8561	NaN	0.9249
XGBoost	453	75	335	55	0.8917	0.8171	0.8580	0.8584	0.8745	0.8536	NaN	0.9215

En la siguiente figura se muestra una gráfica para representar la tasa de aciertos y fallos de cada algoritmo empleado, según nuestra clase positiva "1".

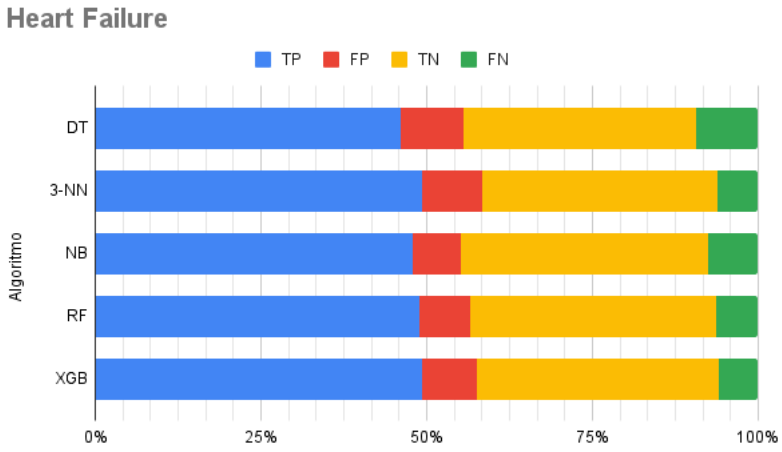

Figura 12: Matriz de confusión de Heart Failure

Nuestro objetivo es que haya el máximo número de TP (Verdaderos positivos)), pues así podemos saber con certeza cuántas predicciones son ciertas. De esta forma, observamos que en la gráfica XGBoost lo consigue, seguido de 3-NN y Random Forest. También es importante que haya el mínimo número de FN (Falsos Negativos), que también lo consiguen los tres algoritmos que maximizaban el número de TP. Observamos que, en general, todos los algoritmos producen resultados similares y el que peor resultados proporciona es el árbol de decisión.

En la siguiente figura se ha representado la precisión de los algoritmos.

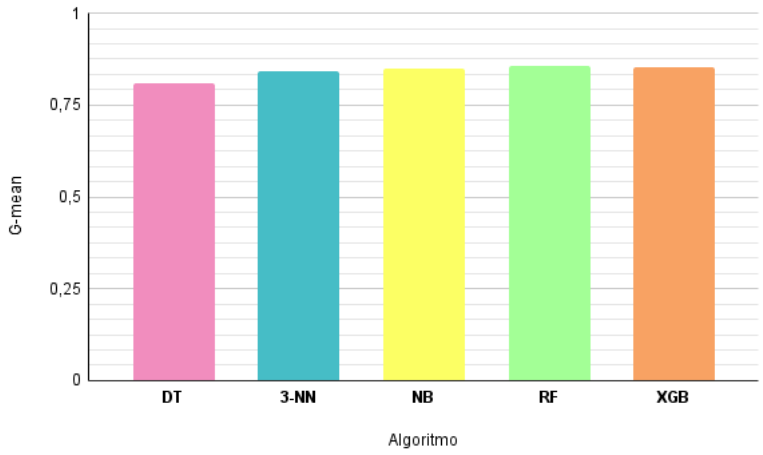

Figura 13: G-mean de Heart Failure

Como bien observábamos, el árbol de decisión es el que menos precisión tiene, y el resto de algoritmos tienen precisión similar. Todos los algoritmos sobrepasan el umbral de 0.8. Es posible que el árbol de decisión no produzca tan buenos resultados porque no manejan de forma fácil los atributos continuos y tienen dificultad para trabajar con valores perdidos, y este conjunto de datos los posee.

En el algoritmo de Naïve Bayes suponemos que todas las variables son independientes, lo cual puede influir negativamente en los resultados.

El algoritmo 3-NN es válido para la clasificación y para la predicción numérica, lo cual se puede aprovechar en este dataset.

Los algoritmos Random Forest y XGBoost producen también buenos resultados, esto se debe a que son multiclasificadores.

En general, todos los algoritmos producen resultados muy similares y no se observan grandes diferencias. Además, tienen una precisión bastante buena por lo que podemos decir que todos producen buenos resultados.

A continuación se ha representado la curva ROC junto al índice AUC (Área bajo la curva). Esta gráfica representa cómo aumenta el número de errores en función de aumentar el número de aciertos en nuestra clase positiva. Esto es, cuanto más cercano a 1 sea el índice AUC, mejor será el algoritmo.

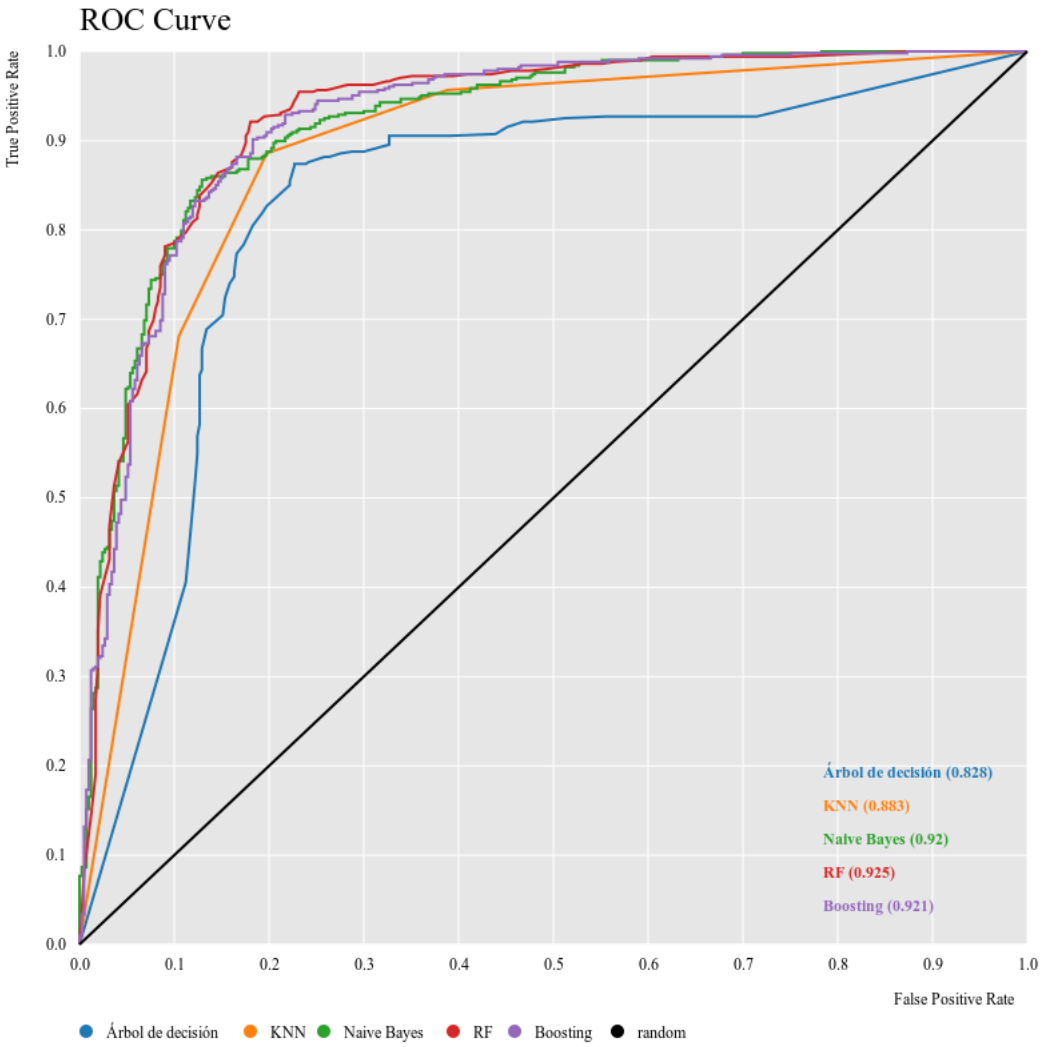

Figura 14: Curva ROC de Heart Failure

Como ya habíamos observado, el árbol de decisión es el que peor resultados obtiene, con un índice AUC de 0.828. Los algoritmos Naïve Bayes, Random Forest y XGBoost producen los mejores resultados en este conjunto de datos, con un índice AUC superior a 0.92.

3.2 Mobile Price

En la siguiente tabla se muestran los criterios de precisión del dataset Mobile Price que hemos obtenido con los algoritmos que hemos aplicado.

Tabla 47: Criterios de precisión de Mobile Price

Mobile Price	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	Model size	AUC
Árbol de decisión	947	59	941	53	0.947	0.941	0.9414	0.944	0.9442	0.9439	36	0.9502
3-NN	790	204	796	210	0.79	0.796	0.7947	0.793	0.7923	0.7929	NaN	0.8475
Naive Bayes	926	75	925	74	0.926	0.925	0.9250	0.9255	0.9255	0.9254	NaN	0.9763
Random Forest	947	42	958	53	0.947	0.958	0.9575	0.9525	0.9522	0.9524	NaN	0.9915
XGBoost	968	30	970	32	0.968	0.97	0.9699	0.969	0.9689	0.9689	NaN	0.9963

En la siguiente figura se muestra una gráfica para representar la tasa de aciertos y fallos de cada algoritmo empleado, según nuestra clase positiva "2".

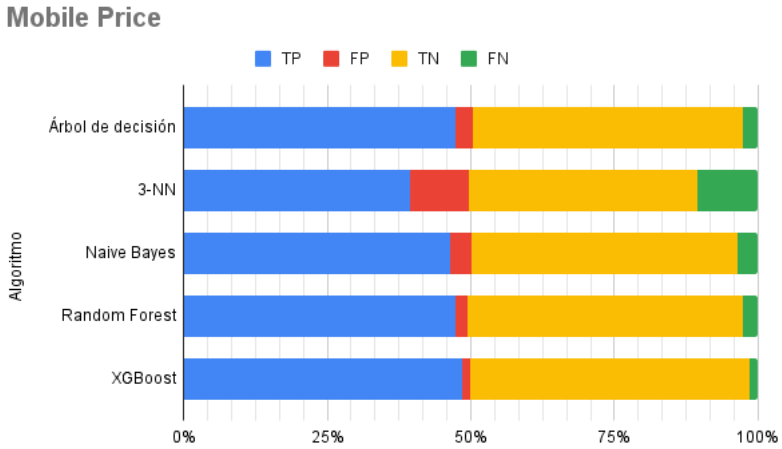

Figura 15: Matriz de confusión de Mobile Price

Al igual que hemos comentado en el apartado anterior, nos fijaremos en los algoritmos que maximizan el número de verdaderos positivos (TP) y minimizan los falsos negativos (FN).

Observamos que, salvo 3-NN, producen valores similares de TP. El algoritmo 3-NN tiene un número de verdaderos positivos algo más bajo. Lo mismo sucede con los falsos negativos, todos producen valores muy pequeños excepto 3-NN.

En la siguiente figura se ha representado la precisión de los algoritmos.

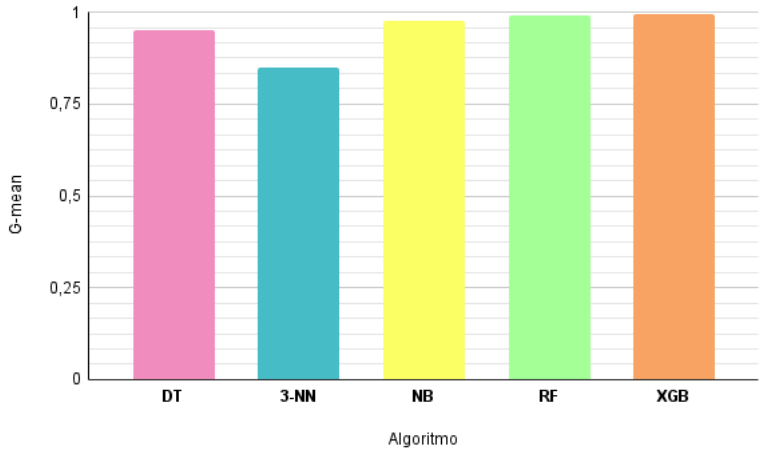

Figura 16: G-mean de Mobile Price

Podemos contemplar que los algoritmos basados en árboles (árbol de decisión, Random Forest y XGBoost) junto a Naïve Bayes son los que proporcionan mejores resultados, superando el umbral de 0.95, e incluso muy cercanos a 1. Sin embargo, el algoritmo 3-NN se queda en 0.8475, esto puede deberse a los problemas en la frontera o que le dé la misma importancia a cada atributo en vez de darle más importancia a los atributos relevantes, esto podría hacerse asignando pesos.

A continuación se ha representado la curva ROC junto al índice AUC (Área bajo la curva). Esta gráfica representa cómo aumenta el número de errores en función de aumentar el número de aciertos en nuestra clase positiva. Esto es, cuanto más cercano a 1 sea el índice AUC, mejor será el algoritmo.

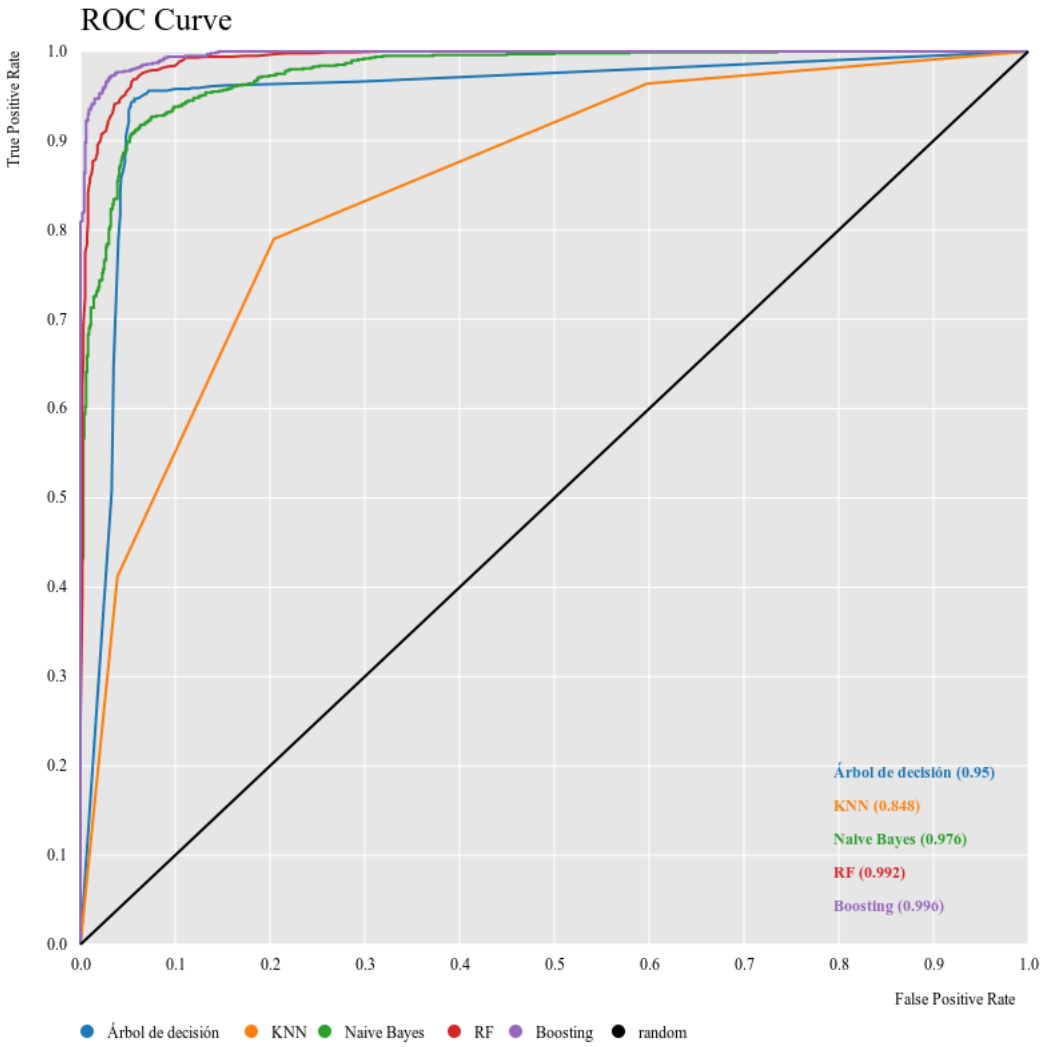

Figura 17: Curva ROC de Mobile Price

Vemos que los mejores algoritmos son los que están basados en multiclasificadores, Random Forest y XGBoost. Destacamos que sus índices AUC son muy proximos a 1. Los algoritmos Naïve Bayes y árbol de decisión también producen índices AUC muy altos. Por último, el algoritmo 3-NN tiene un índice AUC más bajo, 0.848, como ya lo veíamos venir.

3.3 Bank Marketing

En la siguiente tabla se muestran los criterios de precisión del dataset Bank Marketing que hemos obtenido con los algoritmos que hemos aplicado.

Tabla 48: Criterios de precisión de Bank Marketing

Bank Marketing	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	Model size	AUC
Árbol de decisión	2284	1864	34387	2149	0.5152	0.9486	0.5506	0.9014	0.5323	0.6991	2686.2	0.7207
3-NN	1438	1347	35201	3202	0.3099	0.9631	0.5163	0.8896	0.3873	0.5463	NaN	0.7400
Naive Bayes	2510	3565	32983	2130	0.5409	0.9025	0.4132	0.8617	0.4685	0.6987	NaN	0.8340
Random Forest	2044	949	35599	2596	0.4405	0.9740	0.6829	0.9139	0.5356	0.6550	NaN	0.9433
XGBoost	2514	1408	35140	2126	0.5418	0.9615	0.6410	0.9142	0.5872	0.7218	NaN	0.9473

En la siguiente figura se muestra una gráfica para representar la tasa de aciertos y fallos de cada algoritmo empleado, según nuestra clase positiva ɀes".

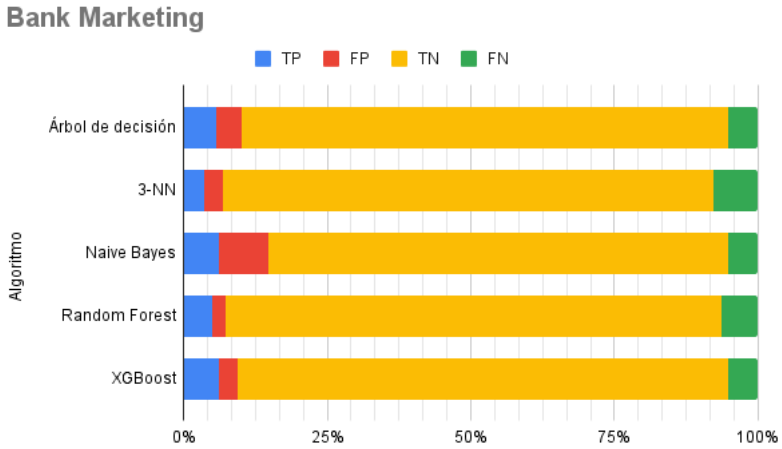

Figura 18: Matriz de confusión de Bank Marketing

En este conjunto de datos, los algoritmos Naïve Bayes y XGBoost son los que mayor número de verdaderos positivos (TP) consiguen. Sin embargo, 3-NN es el que menos consigue. Con respecto a los falsos negativos, el árbol de decisión, Naïve Bayes y XGBoost son los que menor número de falsos negativos consiguen. De la misma forma, 3-NN es el que más consigue.

En la siguiente figura se ha representado la precisión de los algoritmos.

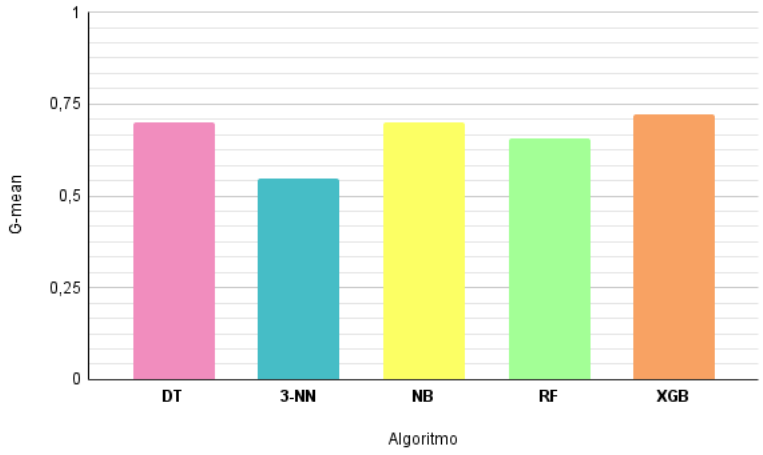

Figura 19: G-mean de Bank Marketing

Observamos que XGBoost es el que mejor G-mean tiene, cabe destacar que es un multiclasificador y tiene en cuenta los fallos de la anterior clasificador lo cual hace que sea tan bueno. A este algoritmo le siguen el árbol de decisión y Naïve Bayes. Puede que el árbol de decisión no funcione tan bien porque no maneja bien los atributos continuos o incluso porque haya sobreaprendizaje.

En esta gráfica también podemos contemplar que 3-NN es el peor, puede deberse a que haya problemas en la frontera o que le dé la misma importancia a cada atributo en vez de darle más importancia a los atributos relevantes, esto podría hacerse asignando pesos.

A continuación se ha representado la curva ROC junto al índice AUC (Área bajo la curva). Esta gráfica representa cómo aumenta el número de errores en función de aumentar el número de aciertos en nuestra clase positiva. Esto es, cuanto más cercano a 1 sea el índice AUC, mejor será el algoritmo.

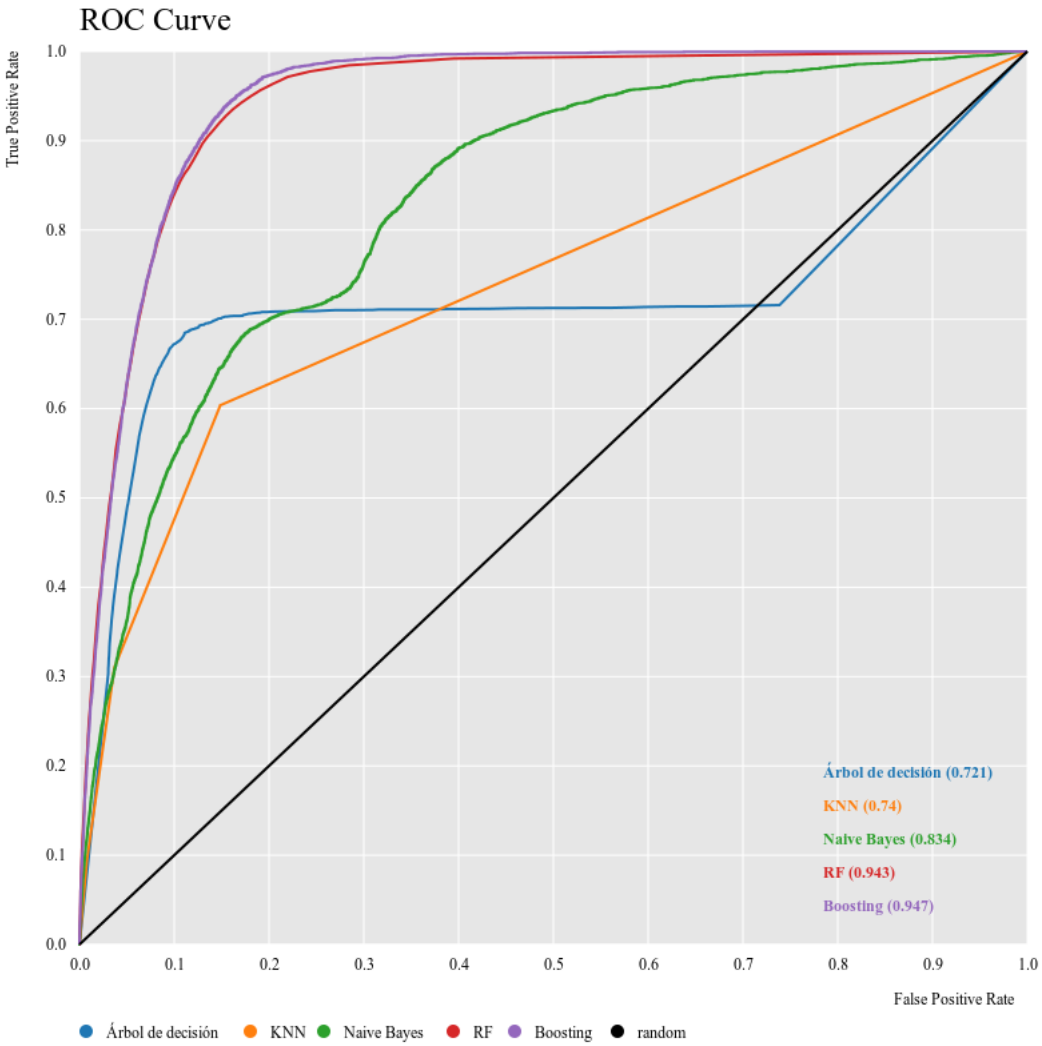

Figura 20: Curva ROC de Bank Marketing

Cabe destacar la gráfica del algoritmo del árbol de decisión, pues conforme aumenta el número de falsos positivos, el número de verdaderos positivos permanece constante la mayor parte e incluso llega a quedarse por debajo del algoritmo random. Como ya habíamos observado, los algoritmos basados en multiclasificadores, Random Forest y XGBoost, son los que mejor índice AUC tienen, un 0.94 aproximadamente. También sobresale lo que ocurre con la gráfica de 3-NN, pues habíamos visto que era el peor algoritmo pero a partir de un número de falsos negativos consigue superar al árbol de decisión.

El algoritmo Naïve Bayes tiene un buen índice AUC, aunque no es tan bueno como los multiclasificadores pues este algoritmo supone independencia entre las distintas clases.

3.4 Tanzania Power Pump

En la siguiente tabla se muestran los criterios de precisión del dataset Tanzania Power Pump que hemos obtenido con los algoritmos que hemos aplicado.

Tabla 49: Criterios de precisión de Tanzania Power Pump

Tanzania Power Pump	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	Model size	AUC
Árbol de decisión	17748	4520	27493	4871	0.7847	0.8588	0.7970	0.8281	0.7908	0.8209	4882	0.8385
3-NN	16480	4781	27478	6344	0.7220	0.8518	0.7751	0.7980	0.7476	0.7842	NaN	0.8480
Naive Bayes	15125	6716	25543	7699	0.6627	0.7918	0.6925	0.7383	0.6773	0.7244	NaN	0.8010
Random Forest	8905	670	31589	13919	0.3902	0.9792	0.9300	0.7351	0.5497	0.6181	NaN	0.8540
XGBoost	17322	2720	29539	5502	0.759	0.9157	0.8643	0.8507	0.8082	0.8336	NaN	0.9208

En la siguiente figura se muestra una gráfica para representar la tasa de aciertos y fallos de cada algoritmo empleado, según nuestra clase positiva "non functional".

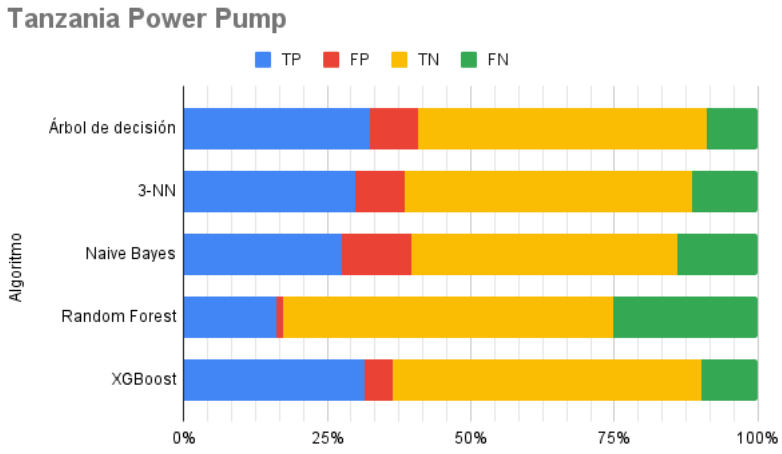

Figura 21: Matriz de confusión de Tanzania Power Pump

Vemos que el árbol de decisón es el que más número de verdaderos positivos (TP) tiene. Los algoritmos 3-NN, Naïve Bayes y XGBoost también producen resultados similares. Sin embargo, el algoritmo Random Forest produce unos resultados peores.

Con respecto a los falsos negativos (FN), el árbol de decisión también obtiene un número muy pequeño de ellos. De la misma forma, los algoritmos 3-NN, Naïve Bayes y XGBoost también tienen resultados parecidos. Destaca Random Forest porque tiene más del triple de falsos negativos que el árbol de decisión.

En la siguiente figura se ha representado la precisión de los algoritmos.

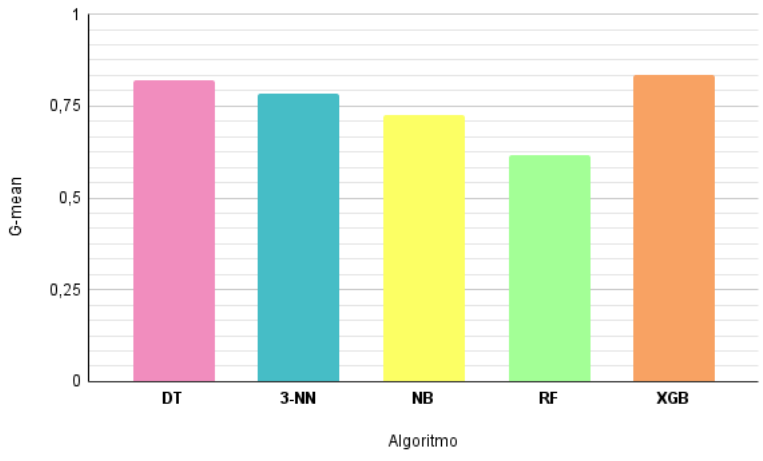

Figura 22: G-mean de Tanzania Power Pump

En esa gráfica seguimos observando que Random Forest es el que peor resultados obtiene. El multiclasificador XGBoost es el que mejor resultados tiene, le sigue el árbol de decision y 3-NN.

A continuación se ha representado la curva ROC junto al índice AUC (Área bajo la curva). Esta gráfica representa cómo aumenta el número de errores en función de aumentar el número de aciertos en nuestra clase positiva. Esto es, cuanto más cercano a 1 sea el índice AUC, mejor será el algoritmo.

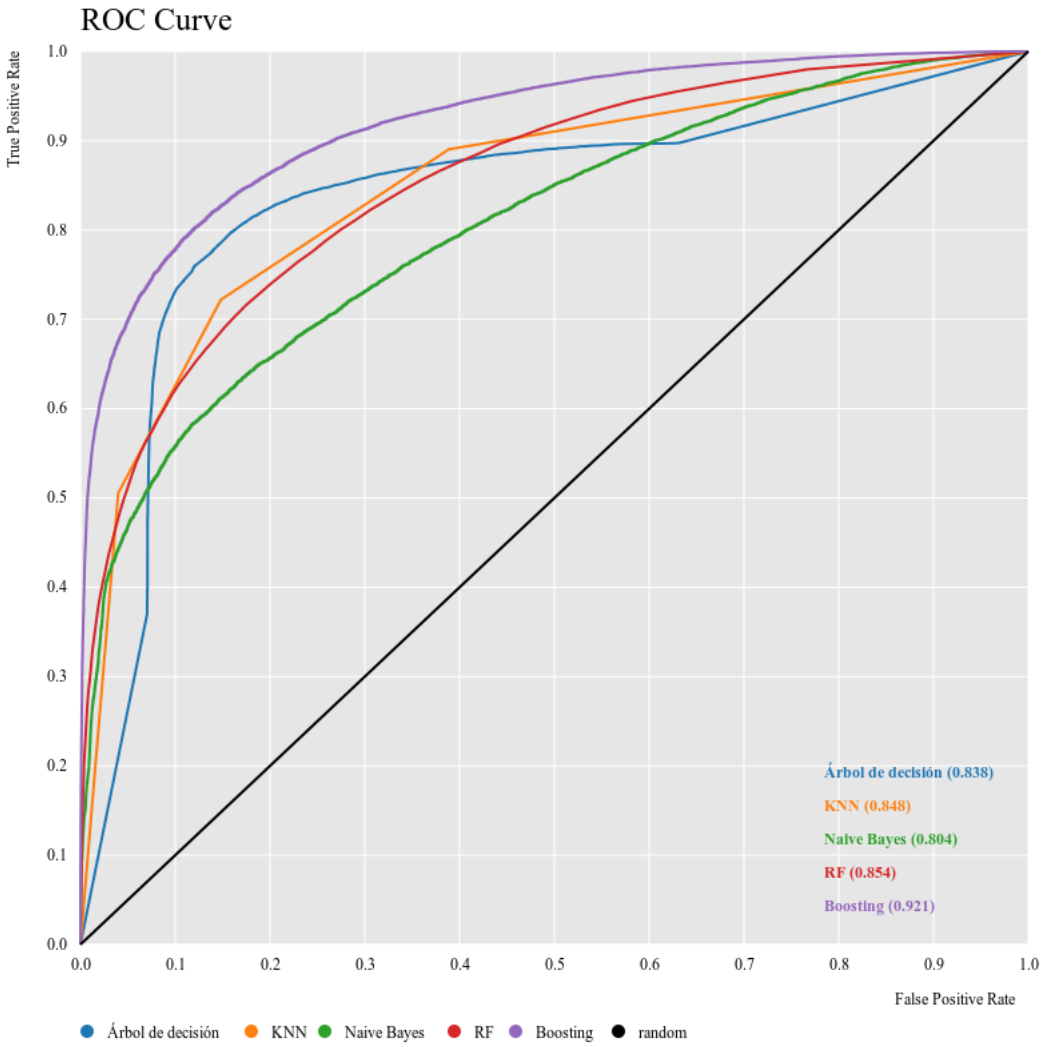

Figura 23: Curva ROC de Tanzania Power Pump

Tenemos que XGBoost proporciona los mejores resultados, pues lo que destaca en este algoritmo es que tiene en cuenta los fallos de la anterior clasificación. A este algoritmo le sigue Random Forest, KNN, árbol de decisión y Naïve Bayes en último lugar.

3.5 Ranking

Por último, se ha realizado un ranking entre los distintos algoritmos en cada dataset. Para tener una vista global, se han representado los resultados comentados anteriormente en una misma gráfica, que se muestra a continuación.

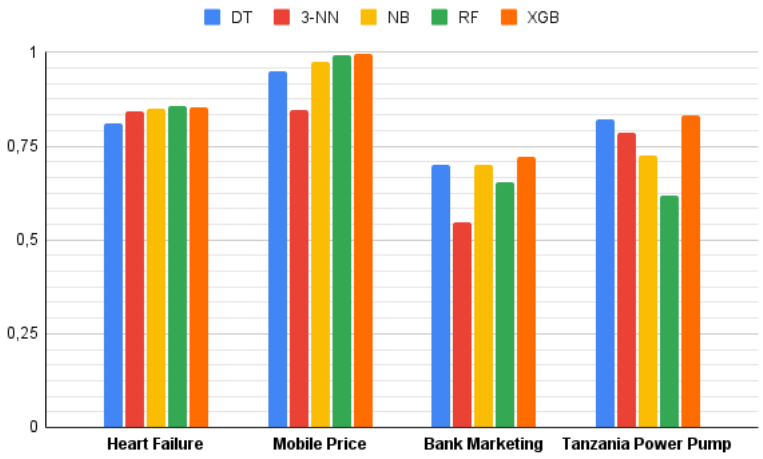

Figura 24: G-mean en cada dataset

Hemos realizado la media de cada algoritmo en cada dataset y se obtienen los siguientes resultados.

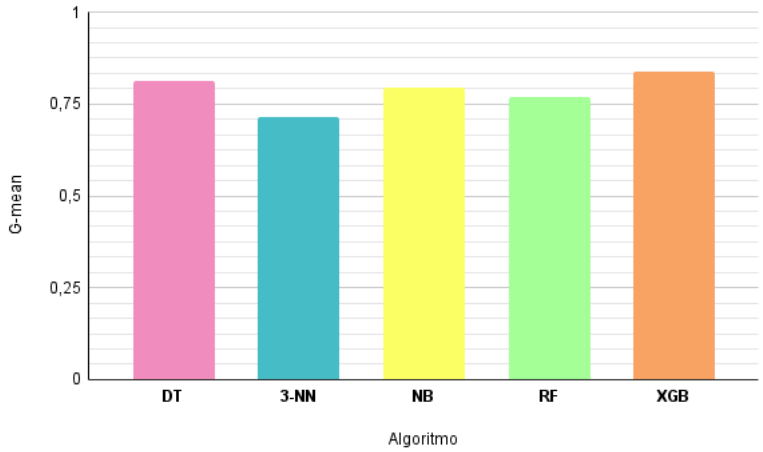

Figura 25: Media de G-mean

Para realizar el ranking, hemos analizado la posición, del 1 al 5, de cada algoritmo en cada dataset según si es el mejor, 1, o si ha obtenido los peores resultados, 5. Hemos realizado la media de la posición de cada algoritmo y el podium queda como indica la siguiente gráfica.

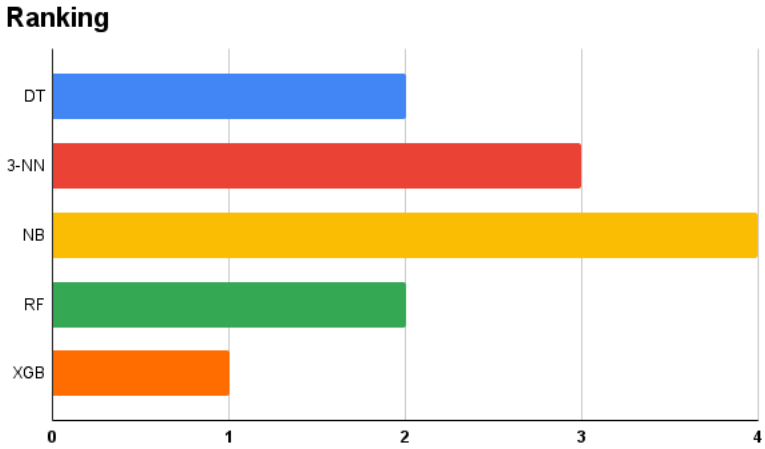

Figura 26: Podium

En primer lugar, tenemos que XGBoost es el mejor algoritmo en todos los datasets, puede deberse a que es un multiclasificador que tiene en cuenta los fallos de la anterior clasificicación. En segundo lugar, tenemos un empate entre el árbol de decisión y Random Forest. El tercer puesto es para 3-NN y el último puesto es para Naïve Bayes.

4 Configuración de algoritmos

Los resultados que hemos obtenido y estudiado hasta ahora se correspondían a algoritmos cuya configuración había sido por defecto. En algunos casos hemos observado que no siempre funcionan tan bien como otros algoritmos, por lo que en esta sección estudiaremos configuraciones alternativas de los parámetros de dos de los algoritmos empleados. Además, estudiaremos los resultados obtenidos proporcionando tablas comparativas.

4.1 k-NN

Habíamos estudiado el caso particular en el que $k = 3$, es decir, el número de vecinos cercanos que consideraba para clasificar las instancias. Este algoritmo nos proporcionaba resultados buenos en algunos conjuntos de datos, pero empeoraba en otros. Para realizar un estudio de este algoritmo, probaremos con diferentes valores de k y analizaremos si se comporta mejor o peor en cada dataset.

El flujo de datos utilizado es el siguiente, donde en el metanodo KNN variaremos el número de vecinos y el resultado se añadirá al final de un fichero .csv

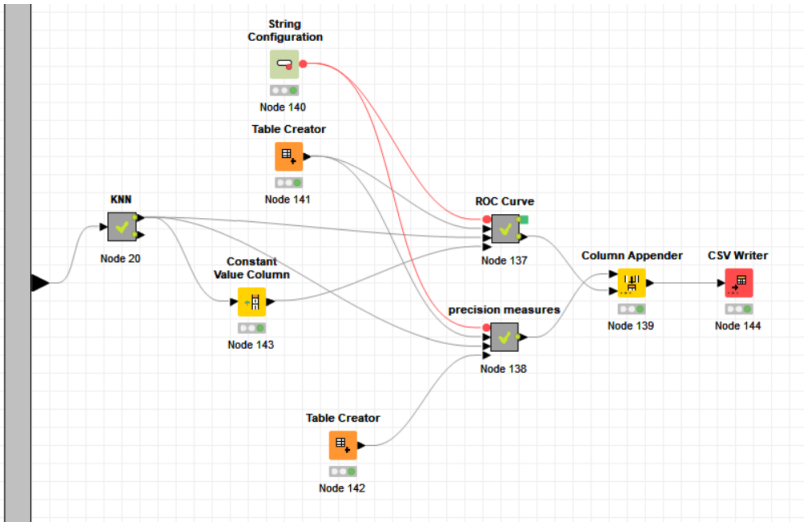

Figura 27: Metanodo Configuración KNN

4.1.1 Heart Failure

Ejecutando en algoritmo k-NN en el dataset Heart Failure variando los valores de k, obtenemos los siguientes resultados.

Tabla 50: Criterios de precisión de K-NN de Heart Failure

Hear Failure	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
1-NN	432	83	327	76	0.8503	0.7975	0.8388	0.8267	0.8445	0.8235	0.82397
2-NN	478	123	287	30	0.9409	0.7	0.7953	0.83333	0.862037	0.8115795	0.87707
3-NN	450	81	329	58	0.8858	0.8024	0.8474	0.84858	0.8663	0.8431025	0.882533
5-NN	455	86	324	53	0.8956	0.7902	0.8410	0.84858	0.86749	0.8413068	0.8918
7-NN	458	83	327	50	0.9015	0.7975	0.84658	0.8551	0.87321	0.8479745	0.89835
10-NN	462	94	316	46	0.9094	0.7707	0.8309	0.84749	0.8684	0.837222	0.90160
15-NN	463	90	320	45	0.9114	0.7804	0.8372	0.8529	0.87276	0.8434157	0.903557
20-NN	465	98	312	43	0.9153	0.7609	0.82593	0.84649	0.8683	0.834603	0.9056654
25-NN	462	99	311	46	0.9094	0.7585	0.8235	0.84204	0.86435	0.830572	0.904433
30-NN	462	102	308	46	0.9094	0.7512	0.819188	0.83877	0.8619	0.8265565	0.90172
35-NN	460	101	309	48	0.9055	0.7536	0.81996	0.83769	0.86061	0.82610	0.899603
40-NN	459	100	310	49	0.9035	0.7560	0.82110	0.8376	0.86035	0.826539	0.898890
50-NN	454	97	313	54	0.8937	0.7634	0.82395	0.8355	0.8574	0.82599	0.8969440176
60-NN	439	90	320	69	0.8641	0.7804	0.82986	0.82679	0.84667	0.821265	0.89485
70-NN	432	89	321	76	0.8503	0.7829	0.82917	0.82026	0.8396	0.81596	0.893419

Para interpretar estos resultados con más facilidad, hemos representado algunas medidas en las siguientes gráficas.

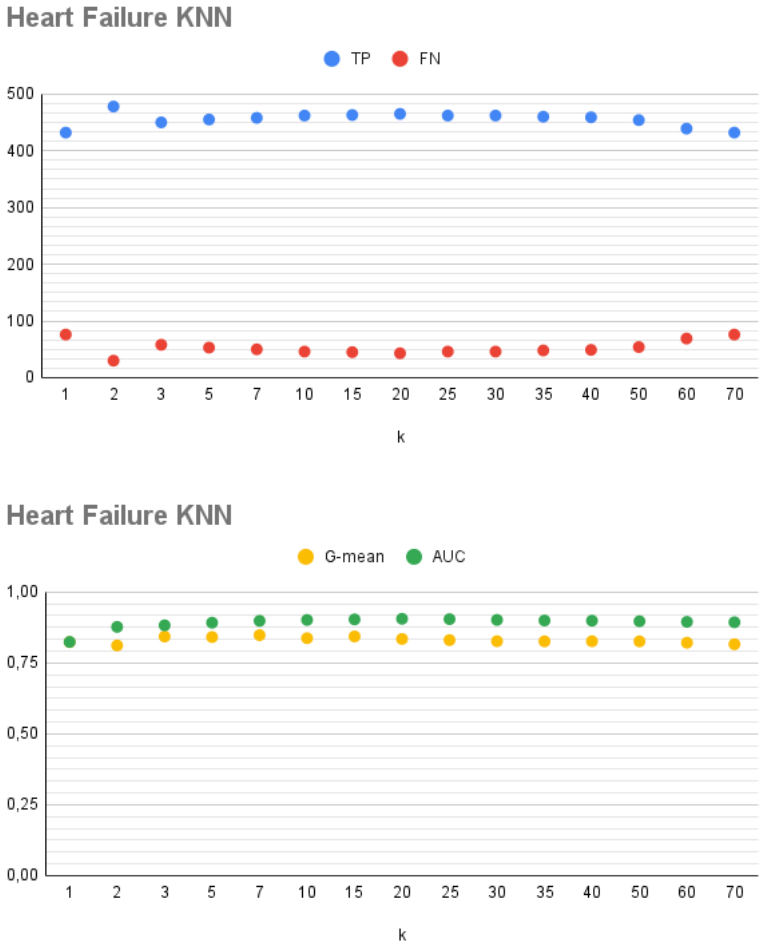

Figura 28: Configuración de KNN en Heart Failure

Observamos que en $k = 1$ obtenemos resultados muy malos en todas las medidas. Para $k = 2$ el número de verdaderos positivos es máximo y el número de falsos negativos es mínimo, sin embargo sus medidas de precisión son más bajas, salvo el TPR, pues el número de falsos positivos es elevado y eso provoca que el valor de AUC sea más bajo. Para el resto de valores de k, tenemos que el número de verdaderos positivos y falsos negativos son muy similares. Si nos fijamos en las medidas

G-mean y AUC también permanecen muy similares. La configuración por defecto que habíamos elegido era cuando $k = 3$ y vemos que si tomamos $k = 7$, podemos incrementar ligeramente los valores de G-mean y AUC. En conclusión, para este dataset el mejor valor de k sería 7, aunque realmente tampoco hay mucha diferencia entre todos los valores.

4.1.2 Mobile Price

Ejecutando en algoritmo k-NN en el dataset Mobile Price variando los valores de k, obtenemos los siguientes resultados.

Tabla 51: Criterios de precisión de K-NN de Mobile Price

Mobile Price	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
1-NN	737	264	736	263	0.737	0.736	0.7362637362637363	0.7365	0.736631684157921	0.736499830278324	0.7365
2-NN	737	264	736	263	0.737	0.736	0.7362637362637363	0.7365	0.736631684157921	0.736499830278324	0.803063
3-NN	790	204	796	210	0.79	0.796	0.7947686116700201	0.793	0.7923771313941825	0.7929943253264805	0.8475010000000001
5-NN	812	187	813	188	0.812	0.813	0.8128128128128128	0.8125	0.8124062031015508	0.8124998461538315	0.885127
7-NN	824	161	839	176	0.824	0.839	0.8365482233502538	0.8315	0.8302267002518892	0.8314661748982937	0.9103055
10-NN	844	143	857	156	0.844	0.857	0.8551165146909828	0.8505	0.8495218922999498	0.8504751613069014	0.931530999999999
15-NN	857	120	880	143	0.857	0.88	0.8771750255885363	0.8685	0.8669701568032373	0.8684238596445862	0.944655999999999
20-NN	871	112	888	129	0.871	0.888	0.8860630722278738	0.8795	0.8784669692385274	0.8794589245666906	0.953279
25-NN	873	107	893	127	0.873	0.893	0.8908163265306123	0.883	0.881818181818182	0.8829433730426883	0.96054
30-NN	883	101	899	117	0.883	0.899	0.8973577235772358	0.891	0.8901209677419355	0.890964084573559	0.965494
35-NN	899	93	907	101	0.899	0.907	0.90625	0.903	0.9026104417670683	0.9029911405988433	0.9704885
40-NN	904	86	914	96	0.904	0.914	0.9131313131313131	0.909	0.9085427135678392	0.9089862485208454	0.973174
50-NN	908	93	907	92	0.908	0.907	0.9070929070929071	0.9075	0.9075462268865568	0.9074998622589427	0.975270999999999
60-NN	909	82	918	91	0.909	0.918	0.917255297679112	0.9135	0.9131089904570568	0.9134889161889158	0.978245500000000
70-NN	918	76	924	82	0.918	0.924	0.9235412474849095	0.921	0.9207622868605818	0.9209951139935543	0.979741

Para interpretar estos resultados con más facilidad, hemos representado algunas medidas en las siguientes gráficas.

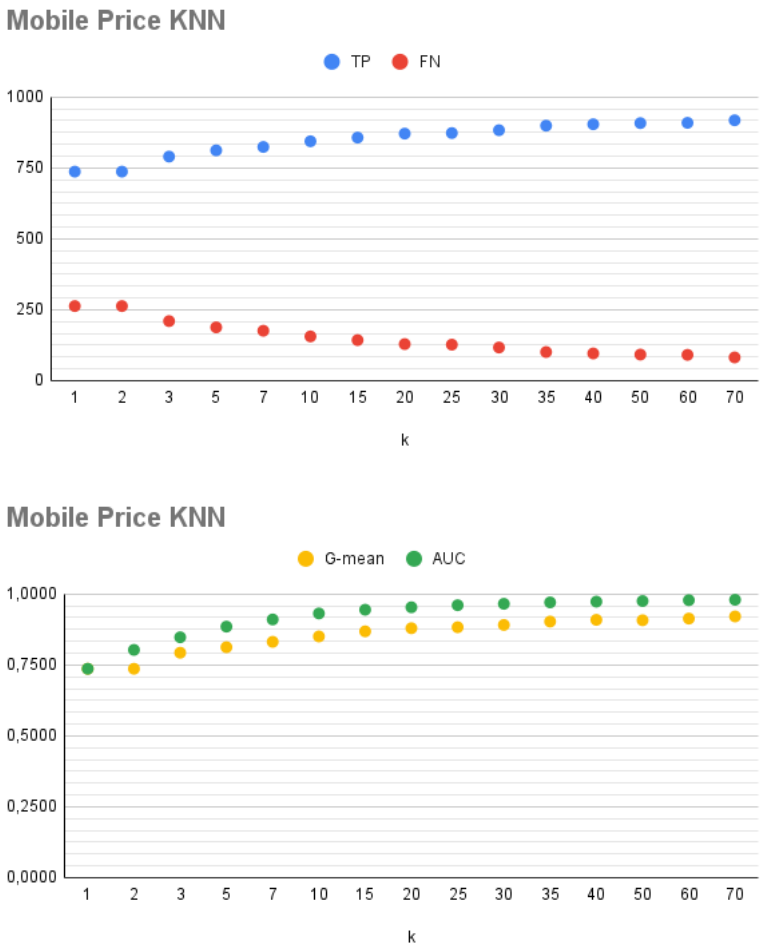

Figura 29: Configuración de KNN en Mobile Price

En este dataset observamos más variedad para los distintos valores de k. Los valores de $k = 1, 2, 3$ producen resultados bastante malos en comparación con el resto. Observamos que si aumentamos el número de vecinos, los resultados mejoran notablemente. Por lo que el mejor valor de k en este conjunto de datos es el más alto que se pueda, en este caso $k = 70$.

4.1.3 Bank Marketing

Ejecutando en algoritmo k-NN en el dataset Bank Marketing variando los valores de k, obtenemos los siguientes resultados.

Tabla 52: Criterios de precisión de K-NN de Bank Marketing

Bank Marketing	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
1-NN	1647	2265	34283	2993	0.3549563	0.9380267	0.421012	0.8723414	0.385173	0.5770260	0.64672
2-NN	794	649	35899	3846	0.171120	0.98224	0.55024	0.89086	0.26105	0.40997807	0.71058
3-NN	1438	1347	35201	3202	0.30991	0.96314	0.5163	0.889555	0.3873	0.5463439	0.73999
5-NN	1331	1104	35444	3309	0.286853	0.969793	0.546611	0.892857	0.37625	0.5274357	0.7710
7-NN	1281	982	35566	3359	0.27607758	0.973131	0.566062	0.8946052	0.3711	0.5183239	0.78820
10-NN	1077	678	35870	3563	0.232112	0.98144	0.61367	0.897033	0.33682	0.477290	0.8018
15-NN	1208	750	35798	3432	0.26034	0.979479	0.616956	0.8984655	0.36617	0.504977	0.811194
20-NN	1077	601	35947	3563	0.23211206	0.9835558	0.64183	0.89890	0.340930	0.477802	0.81427
25-NN	1129	640	35908	3511	0.243318	0.982488	0.638213	0.8992	0.35231	0.48893	0.817229
30-NN	1074	563	35985	3566	0.2314655	0.984595	0.656078	0.89975	0.3422	0.4773886	0.8210433
35-NN	1113	597	35951	3527	0.239870	0.983665	0.65087	0.89987	0.35055	0.4857493	0.8235864
40-NN	1054	537	36011	3586	0.227155	0.9853069	0.66247	0.8998980	0.3383084	0.4730936	0.825879
50-NN	1041	523	36025	3599	0.2243534	0.9856900	0.665601	0.899922	0.335589	0.4702	0.8290
60-NN	1030	501	36047	3610	0.2219827	0.98629	0.67276	0.9001893	0.3338194	0.46791	0.830250
70-NN	1014	492	36056	3626	0.2185344	0.98653	0.673306	0.900019	0.3299707	0.46431	0.83028

Para interpretar estos resultados con más facilidad, hemos representado algunas medidas en las siguientes gráficas.

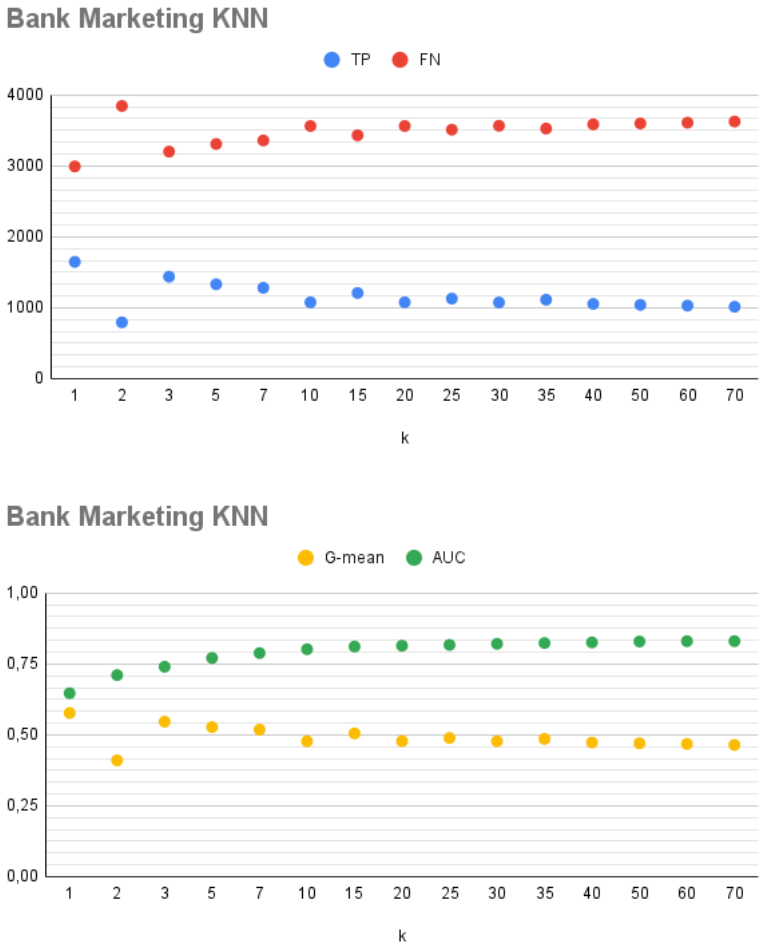

Figura 30: Configuración de KNN en Bank Marketing

En este conjunto de datos no observamos tanta variedad en las distintas medidas de precisión. Lo que destaca en este dataset es que hay mayor número de falsos negativos que verdaderos positivos. Sobre todo en el caso en el que $k = 2$, que obtenemos resultados malísimos. Para el resto de número de vecinos los resultados se mantienen constantes, sin mucha variedad. Pero el valor de k que mejora este dataset es cuando $k = 1$, pues el número de verdaderos positivos es el máximo, el número de falsos negativos es mínimo y el valor de G-mean es máximo, sin embargo el valor de AUC es mínimo. Esto puede deberse a que los datos se encuentren en la frontera y no sepa clasificarlos correctamente.

4.1.4 Tanzania Power Pump

Ejecutando en algoritmo k-NN en el dataset Tanzania Power Pump variando los valores de k, obtenemos los siguientes resultados.

Tabla 53: Criterios de precisión de K-NN de Tanzania Power Pump

Tanzania Power Pump	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
1-NN	16853	5648	26611	5971	0.7383894	0.824917	0.74898893	0.789063	0.7436514	0.78045501	0.7817
2-NN	13685	2332	29927	9139	0.5995881	0.9277100	0.85440469	0.7917	0.704667	0.74581766	0.83069
3-NN	16480	4781	27478	6344	0.7220469	0.851793	0.7751281	0.798032	0.7476465	0.7842415	0.84825
5-NN	16269	4614	27645	6555	0.7128023	0.85697	0.77905473	0.7972332	0.7444574	0.7815691	0.860129
7-NN	16002	4592	27667	6822	0.701104	0.8576521	0.77702243	0.79278	0.7371136	0.775437	0.863190
10-NN	14747	3536	28723	8077	0.64611	0.890387	0.80659	0.7891727	0.7174933	0.7584822	0.86345
15-NN	15325	4563	27696	7499	0.6714423	0.858551	0.7705651	0.781021	0.71759	0.759254	0.859345
20-NN	14366	3935	28324	8458	0.6294251	0.8780185	0.7849844	0.775012	0.69865	0.743402	0.854382
25-NN	14747	4543	27716	8077	0.646118	0.85917	0.7644893	0.77089	0.700337	0.7450677	0.8501971
30-NN	14113	4136	28123	8711	0.618340	0.8717877	0.7733574	0.766770	0.687215	0.734208	0.846317
35-NN	14459	4622	27637	8365	0.6334998	0.856722	0.757769	0.764228	0.690084	0.73670437	0.842382
40-NN	13958	4271	27988	8866	0.611549	0.86760283	0.7657030	0.761505	0.679998	0.728410504	0.839328
50-NN	13758	4319	27940	9066	0.60278654	0.8661148	0.76107	0.757003	0.67274636	0.7225526	0.833787
60-NN	13535	4337	27922	9289	0.593016	0.865556	0.757329	0.7526278	0.665175938	0.716442040	0.82858143
70-NN	13391	4396	27863	9433	0.5867069	0.86372795	0.7528532	0.748942	0.65947649	0.711867414	0.8237431

Para interpretar estos resultados con más facilidad, hemos representado algunas medidas en las siguientes gráficas.

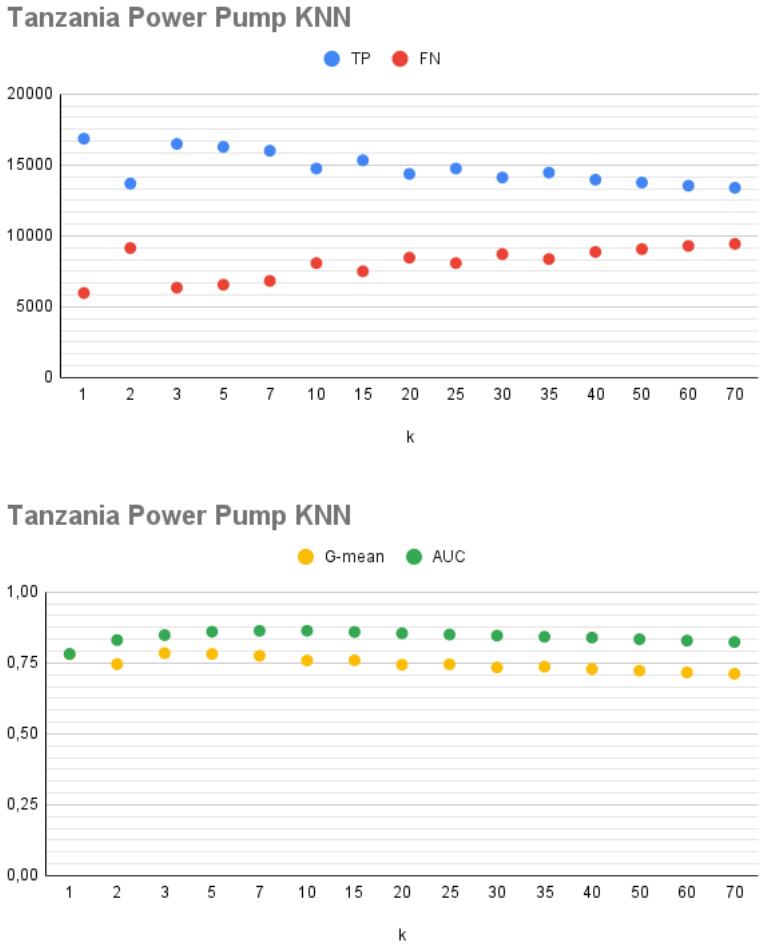

Figura 31: Configuración de KNN en Tanzania Power Pump

Observamos que a medida que aumentamos el número de vecinos, los resultados empeoran. Además, para $k = 2$ obtenemos resultados igualmente malos. Los mejores resultados que obtenemos son cuando el número de vecinos es $3, 5$ o 7, pues a partir de ellos empeoran los resultados. Esto puede deberse a que los datos se encuentren en la frontera y no sepa clasificarlos correctamente.

4.2 Random Forest

La configuración que habíamos elegido previamente para este algoritmo era con el índice Gini y 100 número de modelos. Cambiaremos ambos parámetros y estudiaremos si mejora o empeora en cada dataset.

Figura 32: Configuración de Random Forest por defecto

El flujo de datos utilizado es el siguiente, donde en el metanodo RF variaremos dichos parámetros y el resultado se añadirá al final de un fichero .csv

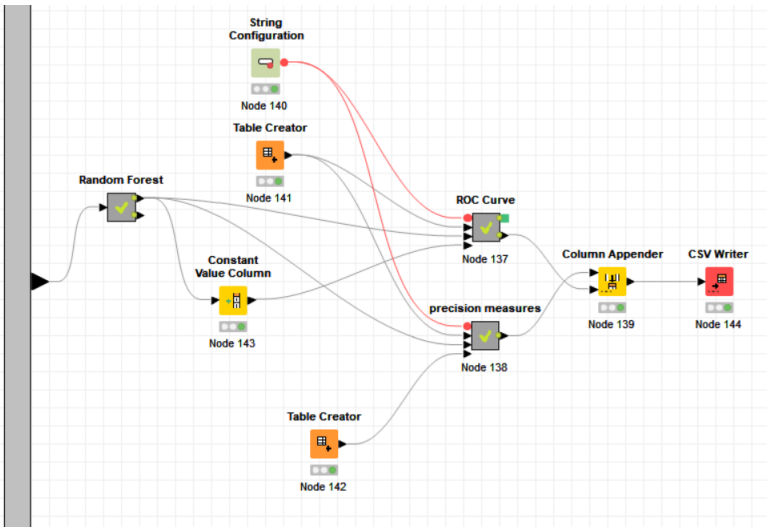

Figura 33: Metanodo Configuración Random Forest

4.2.1 Heart Failure

Ejecutando en algoritmo Random Forest en el dataset Heart Failure variando los parámetros, obtenemos los siguientes resultados.

Primero, habíamos considerado por defecto el índice Gini. Probaremos con los otros criterios para analizar los resultados que obtienen.

Tabla 54: Criterios de precisión de Random Forest de Heart Failure

Heart Failure	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Information Gain	454	71	339	54	0.8937007874015748	0.8268292682926829	0.8647619047619047	0.8638344226579521	0.8789932236205226	0.859615011571947	0.9272541770693298
Information Gain Ratio	455	64	346	53	0.8956692913385826	0.8439024390243902	0.8766859344894027	0.8725490196078431	0.8860759493670887	0.8694006553481984	0.9282048204340312
Gini Index	449	70	340	59	0.8838582677165354	0.8292682926829268	0.8651252408477842	0.8594771241830066	0.8743914313534568	0.8561282828192166	0.9248991741885925

Observamos que los tres producen valores similares, aunque Information Gain Ratio es el que mejores resultados obtiene.

Ahora, probaremos a cambiar el número de modelos elegido, que por defecto era 100.

Tabla 55: Criterios de precisión de Random Forest de Heart Failure

Heart Failure	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
20	443	68	342	65	0.8720472440944882	0.8341463414634146	0.8669275929549902	0.855119825708061	0.8694798822374877	0.8528862868194509	0.9189096408680621
40	450	69	341	58	0.8858267716535433	0.8317073170731707	0.8670520231213873	0.8616557734204793	0.8763388510223953	0.8583406128359281	0.9232187439984636
60	456	67	343	52	0.8976377952755905	0.8365853658536585	0.8718929254302104	0.8703703703703703	0.8845780795344326	0.8665740841755547	0.9249087766468216
80	451	67	343	57	0.8877952755905512	0.8365853658536585	0.8706563706563707	0.8649237472766884	0.8791423001949319	0.8618100344235212	0.9240877664682158
100	449	70	340	59	0.8838582677165354	0.8292682926829268	0.8651252408477842	0.8594771241830066	0.8743914313534568	0.8561282828192166	0.9248991741885925
120	455	68	342	53	0.8956692913385826	0.8341463414634146	0.869980879541109	0.8681917211328976	0.8826382153249273	0.8643606090811913	0.9264211638179377
140	454	68	342	54	0.8937007874015748	0.8341463414634146	0.8697318007662835	0.8671023965141612	0.8815533980582524	0.8634102397898675	0.9263779527559057
160	457	69	341	51	0.8996062992125984	0.8317073170731707	0.8688212927756654	0.869281045751634	0.88394584139265	0.8649908332116787	0.9272877856731326
180	455	70	340	53	0.8956692913385826	0.8292682926829268	0.8666666666666667	0.8660130718954249	0.8809293320425944	0.8618295330498215	0.9267116381793742
200	452	68	342	56	0.889763779527559	0.8341463414634146	0.8692307692307693	0.8649237472766884	0.8793774319066148	0.8615063560180932	0.9269372959477625
250	455	71	339	53	0.8956692913385826	0.8268292682926829	0.8650190114068441	0.8649237472766884	0.8800773694390716	0.8605612033956133	0.9271773574034955
300	454	69	341	54	0.8937007874015748	0.8317073170731707	0.8680688336520076	0.8660130718954249	0.8806983511154218	0.8621470200354137	0.9270501248319573
350	457	71	339	51	0.8996062992125984	0.8268292682926829	0.865530303030303	0.8671023965141612	0.8822393822393821	0.8624504728559438	0.92721816785097
400	456	71	339	52	0.8976377952755905	0.8268292682926829	0.8652751423149905	0.8660130718954249	0.881159420289855	0.8615063560180932	0.92730699058959

Para interpretar estos resultados con más facilidad, hemos representado algunas medidas en las siguientes gráficas.

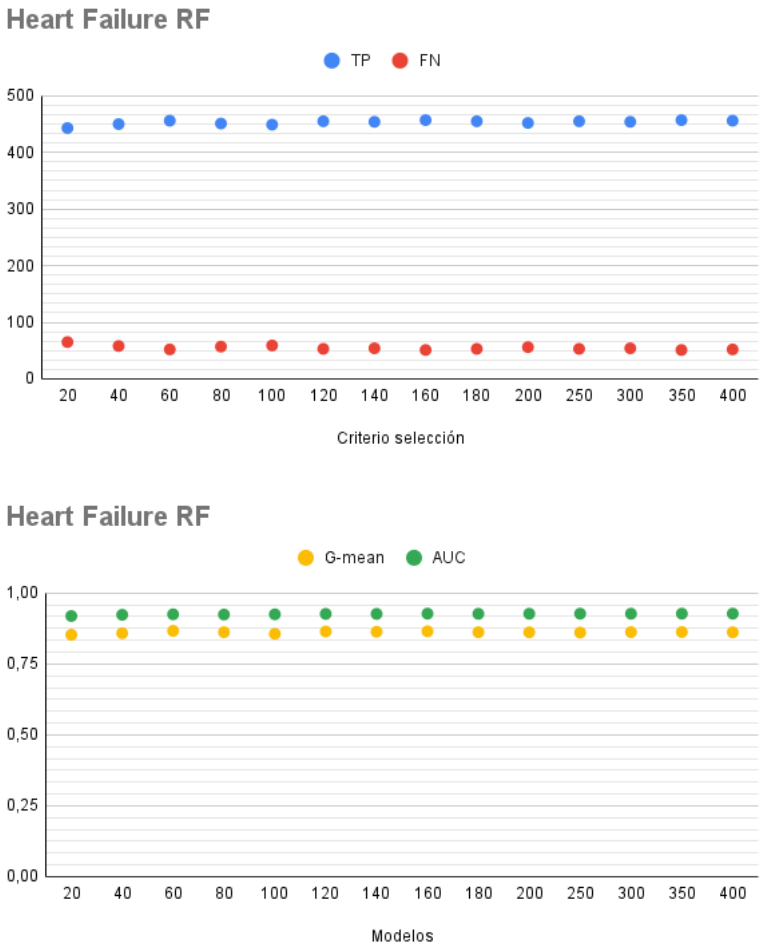

Figura 34: Configuración de RF en Heart Failure

No observamos casi diferencia, todos los distintos valores producen resultados muy similares. Por lo que en este caso es indiferente elegir un número más pequeño o más alto que el que habíamos elegido por defecto.

4.2.2 Mobile Price

Ejecutando en algoritmo Random Forest en el dataset Mobile Price variando los parámetros, obtenemos los siguientes resultados.

Primero, habíamos considerado por defecto el índice Gini. Probaremos con los otros criterios para analizar los resultados que obtienen.

Tabla 56: Criterios de precisión de Random Forest de Mobile Price

Mobile Price	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Information Gain	948	40	960	52	0.948	0.96	0.9595141700404858	0.954	0.9537223340040242	0.9539811318888859	0.9926955
Information Gain Ratio	950	44	956	50	0.95	0.956	0.9557344064386318	0.953	0.9528585757271815	0.9529952780575568	0.9916014999999999
Gini Index	947	42	958	53	0.947	0.958	0.9575328614762386	0.9525	0.9522373051784817	0.9524841206025431	0.9915100000000001

Observamos que los tres producen valores similares, aunque Information Gain es el que mejores resultados obtiene.

Ahora, probaremos a cambiar el número de modelos elegido, que por defecto era 100.

Tabla 57: Criterios de precisión de Random Forest de Mobile Price

Mobile Price	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
20	934	41	959	66	0.934	0.959	0.9579487179487179	0.9465	0.9458227848101266	0.946417455460327	0.989
40	943	43	957	57	0.943	0.957	0.9563894523326572	0.95	0.9496475327291037	0.9499742101762552	0.9896505
60	943	41	959	57	0.943	0.959	0.9583333333333334	0.951	0.9506048387096774	0.9509663506139425	0.9903775
80	947	41	959	53	0.947	0.959	0.958502024291498	0.953	0.9527162977867204	0.9529811120898461	0.9908035000000001
100	947	42	958	53	0.947	0.958	0.9575328614762386	0.9525	0.9522373051784817	0.9524841206025431	0.9915100000000001
120	950	46	954	50	0.95	0.954	0.9538152610441767	0.952	0.9519038076152304	0.9519978991573458	0.991838
140	948	43	957	52	0.948	0.957	0.9566094853683148	0.9525	0.9522852837769965	0.9524893700194244	0.991851
160	946	41	959	54	0.946	0.959	0.9584599797365755	0.9525	0.9521892299949672	0.9524778212640964	0.9921749999999999
180	949	43	957	51	0.949	0.957	0.9566532258064516	0.953	0.9528112449799196	0.952991605419481	0.9922565000000001
200	951	43	957	49	0.951	0.957	0.9567404426559356	0.954	0.9538615847542627	0.9539952830072065	0.9924535000000001
250	949	44	956	51	0.949	0.956	0.9556898288016112	0.9525	0.9523331660812844	0.9524935695320992	0.992432
300	947	44	956	53	0.947	0.956	0.9556004036326943	0.9515	0.9512807634354595	0.951489358847486	0.992405
350	947	44	956	53	0.947	0.956	0.9556004036326943	0.9515	0.9512807634354595	0.951489358847486	0.9924074999999999
400	949	44	956	51	0.949	0.956	0.9556898288016112	0.9525	0.9523331660812844	0.9524935695320992	0.992433

Para interpretar estos resultados con más facilidad, hemos representado algunas medidas en las siguientes gráficas.

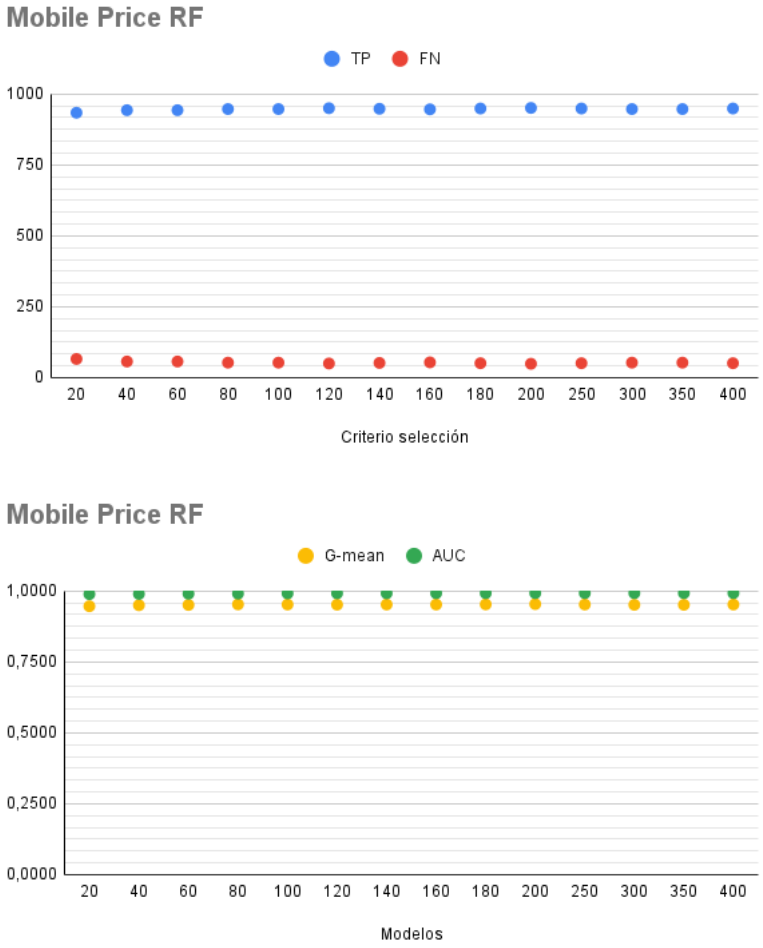

Figura 35: Configuración de RF en Mobile Price

No observamos casi diferencia, todos los distintos valores producen resultados muy similares. A medida que aumentamos el número de modelos, observamos que, ligeramente, mejoran los resultados. Pero apenas hay gran diferencia. Por lo que en este caso es indiferente elegir un número más pequeño o más alto que el que habíamos elegido por defecto.

4.2.3 Bank Marketing

Ejecutando en algoritmo Random Forest en el dataset Bank Marketing variando los parámetros, obtenemos los siguientes resultados.

Primero, habíamos considerado por defecto el índice Gini. Probaremos con los otros criterios para analizar los resultados que obtienen.

Tabla 58: Criterios de precisión de Random Forest de Bank Marketing

Bank Marketing	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Information Gain	2024	938	35610	2616	0.4362068965517241	0.9743351209368502	0.6833220796758946	0.9137127318636497	0.5324914496185215	0.651929213416006	0.9444215012001222
Information Gain Ratio	2083	991	35557	2557	0.4489224137931034	0.9728849731859472	0.6776187378009109	0.9138584053607847	0.5400570391495981	0.660870539898454	0.9436399769976563
Gini Index	2044	949	35599	2596	0.44051724137931036	0.9740341468753421	0.6829268292682927	0.9139312421093523	0.5355692388313901	0.6550410944290256	0.9433032563695167

Observamos que los tres producen valores similares, aunque Information Gain Ratio es el que mejores resultados obtiene.

Ahora, probaremos a cambiar el número de modelos elegido, que por defecto era 100.

Tabla 59: Criterios de precisión de Random Forest de Bank Marketing

Bank Marketing	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
20	1973	990	35558	2667	0.4252155172413793	0.9729123344642662	0.6658791765102936	0.9112120034961639	0.5190056556622384	0.6431931448093494	0.9295309805149959
40	2021	972	35576	2619	0.4355603448275862	0.9734048374740067	0.6752422318743735	0.9128144119646499	0.5295427747936592	0.651134814510036	0.9381727778632163
60	2039	967	35581	2601	0.4394396551724138	0.9735416438656014	0.6783100465735197	0.9133728270370011	0.5333507716453048	0.6540740051984062	0.9410392108346888
80	2044	975	35573	2596	0.4405172413793103	0.9733227536390 5	0.6770453792646571	0.9132999902884336	0.5337511424467946	0.6548018436174324	0.9424813683847033
100	2044	949	35599	2596	0.4405172413793103	0.9740341468753421	0.6829268292682927	0.9139312421093523	0.5355692388313901	0.6550410944290256	0.9433032563695167
120	2052	961	35587	2588	0.4422413793103485	0.9737058115355149	0.6810487885828078	0.9138341264445955	0.5362602900823207	0.656211094950371	0.9438552966953238
140	2054	967	35581	2586	0.4426724137931034	0.9735416438656014	0.6799073154584575	0.9137370107798388	0.5362224252708524	0.6564754598749991	0.944010728215705
160	2065	956	35592	2575	0.4450431034827586	0.9738426179271096	0.6835484938762	0.9142711469360008	0.539094113040073	0.658332697769505	0.9440962852818966
180	2072	958	35590	2568	0.4465517241379310 4	0.9737878953704717	0.6838283828382838	0.9143925415169467	0.5402868318122555	0.6594290436599918	0.9442188537841594
200	2074	958	35590	2566	0.4469827586206896 5	0.9737878953704717	0.6840369393139841	0.914441099349325	0.540667361835245	0.6597472241579565	0.944403244623037
250	2077	944	35604	2563	0.447629310344827 6	0.9741709532669366	0.6875206885137372	0.9148538409245411	0.5422268633337685	0.6603540504675065	0.9448762645156299
300	2079	945	35603	2561	0.4480603448275862	0.9741435919886177	0.6875	0.9148781198407303	0.5425365344467641	0.6606626323154682	0.9450535585229438
350	2080	949	35599	2560	0.4482758620689655	0.9740341468753421	0.6866952789699571	0.9148052830921628	0.5424436041204852	0.6607843800175314	0.9451812425228232
400	2071	949	35599	2569	0.4463362068965517 3	0.9740341468753421	0.6857615894039735	0.9145867728464602	0.5407310704960836	0.6593532486490523	0.9454841654857288

Para interpretar estos resultados con más facilidad, hemos representado algunas medidas en las siguientes gráficas.

Figura 36: Configuración de RF en Bank Marketing

No observamos casi diferencia, todos los distintos valores producen resultados muy similares. Destaca en este dataset que hay mayor número de falsos negativos que verdaderos positivos. A medida que aumentamos el número de modelos, observamos que los resultados mejoran ligeramente, pero apenas hay diferencia.

4.2.4 Tanzania Power Pump

Ejecutando en algoritmo Random Forest en el dataset Tanzania Power Pump variando los parámetros, obtenemos los siguientes resultados.

Primero, habíamos considerado por defecto el índice Gini. Probaremos con los otros criterios para analizar los resultados que obtienen.

Tabla 60: Criterios de precisión de Random Forest de Tanzania Power Pump

Tanzania Power Pump	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
Information Gain	9701	667	31592	13123	0.42503505082369436	0.9793235996156111	0.9356674382716049	0.7496505273859448	0.5845384429983128	0.6451719584230738	0.8664281740561384
Information Gain Ratio	10211	3881	28378	12613	0.4473799509288468	0.8796924889178214	0.7245955151859211	0.7005609716246392	0.5532018636905407	0.6273410416388601	0.7929647954466243
Gini Index	8905	670	31589	13919	0.3901594812478093	0.979230602312532 9	0.9300261096605744	0.7351451445999673	0.5497083243309978	0.6181068708728594	0.8544337582024703

Observamos que los tres producen valores similares, aunque Information Gain es el que mejores resultados obtiene.

Ahora, probaremos a cambiar el número de modelos elegido, que por defecto era 100.

Tabla 61: Criterios de precisión de Random Forest de Tanzania Power Pump

Tanzania Power Pump	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	AUC
20	8201	600	31659	14623	0.35931475639677535	0.9814005393843579	0.9318259288717191	0.7236352413630339	0.5186403162055337	0.5938280018124393	0.8447374108038352
40	8793	674	31585	14031	0.38525236593059936	0.9791066059084287	0.9288053237562057	0.7330392317048817	0.5446099532377443	0.6141686547069144	0.8524861260008387
60	8671	636	31623	14153	0.37990711531720994	0.9802845717474193	0.9316643386698185	0.7315142602980956	0.5397279885468862	0.6102598494432749	0.8542289561983355
80	8731	625	31634	14093	0.38253592709428674	0.9806255618587061	0.9331979478409577	0.73280322422526	0.5426351771286513	0.612474087972688	0.8554967133564413
100	8905	670	31589	13919	0.3901594812478093	0.9792306023125329	0.9300261096605744	0.7351451445999673	0.5497083243309978	0.6181068708728594	0.8544337582024703
120	9078	691	31568	13746	0.397739221871714	0.9785796211909855	0.9292660456546218	0.7379046166693898	0.557052127757494	0.6238745844094141	0.8562463567499761
140	9078	674	31585	13746	0.397739221871714	0.9791066059084287	0.9308859721082855	0.7382132418350489	0.5573428290766208	0.6240425462766729	0.8570212901347769
160	9045	652	31607	13779	0.39629337539432175	0.9797885861310022	0.9327627101165309	0.738013543198446	0.5562559576888779	0.6231241657733112	0.8578450976279908
180	9101	647	31612	13723	0.39874693305292674	0.9799435816361326	0.9336274107509233	0.7391209629105169	0.5588235294117647	0.6250995902592708	0.8587918747412056
200	9162	664	31595	13662	0.40141955835962145	0.9794165969186893	0.9324241807449624	0.7399197574569286	0.5612251148545175	0.6270223104365455	0.85802419064232
250	9202	689	31570	13622	0.4031720995443393	0.9786416193930376	0.9303407137802042	0.740192073779569	0.5625554027204647	0.6281409048869235	0.858234663048084
300	9283	692	31567	13541	0.4067209954433929	0.9785486220899594	0.9306265664160401	0.741608118657299	0.5660538431049728	0.6308694553282705	0.8585255655170998
350	9283	683	31576	13541	0.4067209954433929	0.978827613999194	0.9314669877583784	0.7417715084508832	0.5662092101250381	0.6309593818410448	0.8590740121953919
400	9298	701	31558	13526	0.4073781983876621	0.9782696301807248	0.9298929892989299	0.7417170451863552	0.5665539408341712	0.6312889350213483	0.8587117916929512

Para interpretar estos resultados con más facilidad, hemos representado algunas medidas en las siguientes gráficas.

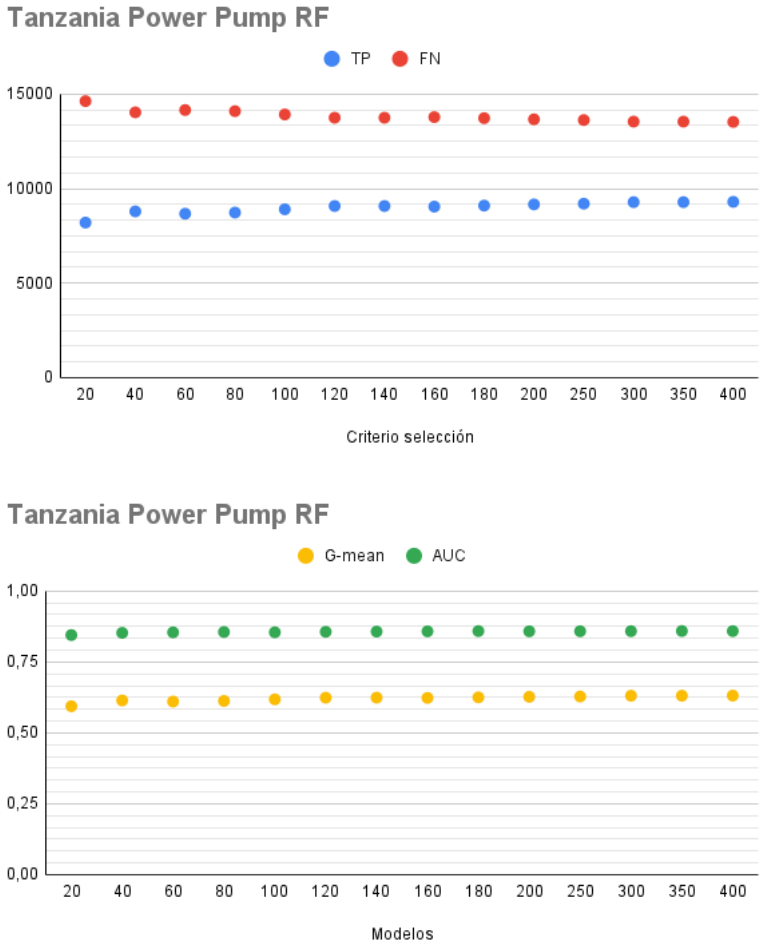

Figura 37: Configuración de RF en Tanzania Power Pump

No observamos casi diferencia, todos los distintos valores producen resultados muy similares. Destaca en este dataset que hay mayor número de falsos negativos que verdaderos positivos. A medida que aumentamos el número de modelos, observamos que los resultados mejoran ligeramente, pero apenas hay diferencia.

5 Procesado de datos

Hasta el momento hemos empleado los algoritmos sobre los datos sin apenas procesarlos, solo hemos realizado un preprocesamiento básico cuando el algoritmo así lo indicaba. En esta sección se estudiará un procesado básico de los datos que mejore la predicción.

Para ello, trabajaremos sobre el conjunto de datos de Tanzania Power Pump por la cantidad de instancias que tiene y porque posee valores desconocidos y perdidos.

Utilizaremos el nodo `Data Explorer` para analizar las propiedades de cada atributo, tales como el número de missing values, el mínimo, el máximo, la media, el número de valores únicos... De esta forma, obtenemos la siguiente información.

- El atributo `num_private` es cero en 54374 variables, por lo que eliminaremos esta variable pues indica que se ha perdido su valor real.
- El atributo `date_recorded` tiene muchos valores posibles. Además, a la hora de determinar si una bomba de agua es funcional o no, no es necesario saber la fecha en la que se almacenó la información de dicha bomba. Por lo tanto, eliminaremos este atributo.
- El atributo `funder` tiene muchos valores posibles e indica el fundador de la bomba de agua, puede que sea útil para determinar si una bomba de agua será funcional o no. También vemos que tiene 3198 missing values, no es gran cantidad. Por lo tanto, no creo que sea necesario eliminar este atributo.
- El atributo `installer` tiene el mismo problema que la anterior variable e indica quién la instaló, información que puede ser relevante. Además, tiene 3215 missing values, no es gran cantidad. Por lo tanto, no creo que sea necesario eliminar este atributo.
- El atributo `wpt_name` indica el nombre del punto de agua y será único para cada bomba de agua. Como tiene muchos valores posibles, no proporciona información relevante, por lo tanto lo eliminaremos.
- El atributo `ward` tiene muchos valores posibles e indica la localización geográfica, información que puede ayudar a clasificarla. Por lo tanto, no creo que sea necesario eliminarlo.
- El atributo `recorder_by` tiene un único valor, GeoData Consultants Ltd. Por lo que lo eliminaremos.
- El atributo `scheme_name` tiene 26162 missing values y muchos valores posibles. Por lo que lo eliminaremos.

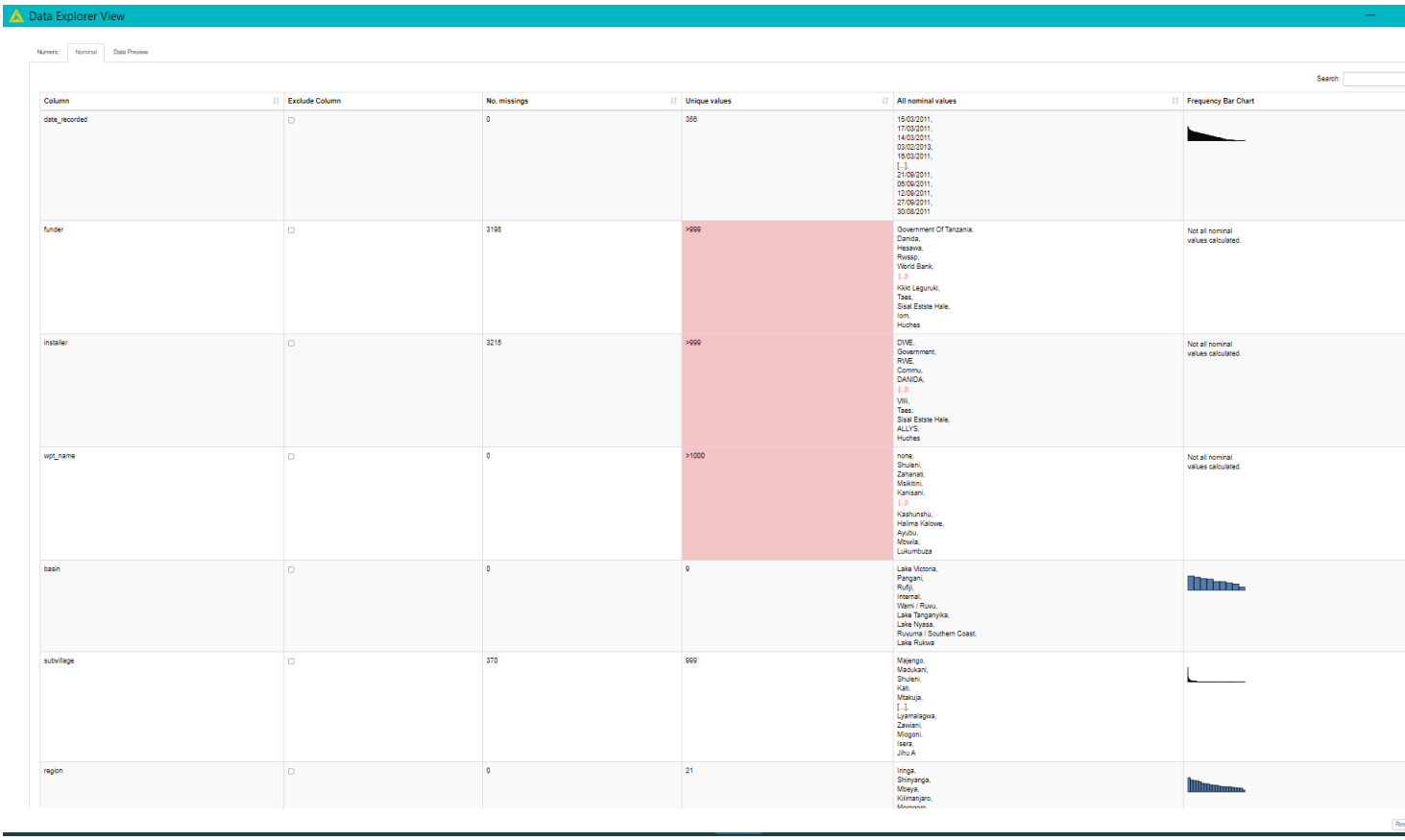

Figura 38: Nodo Data Explorer

También podemos estudiar la correlación entre los atributos. Esto es, si dos atributos están correlados, no será necesario tener a ambos, sino nos quedaremos con uno. Para ello, utilizaremos el nodo `Linear Correlation`, que nos mostrará la correlación entre los distintos atributos. Buscaremos los atributos que estén correlados linealmente, es decir, con coeficiente 1.

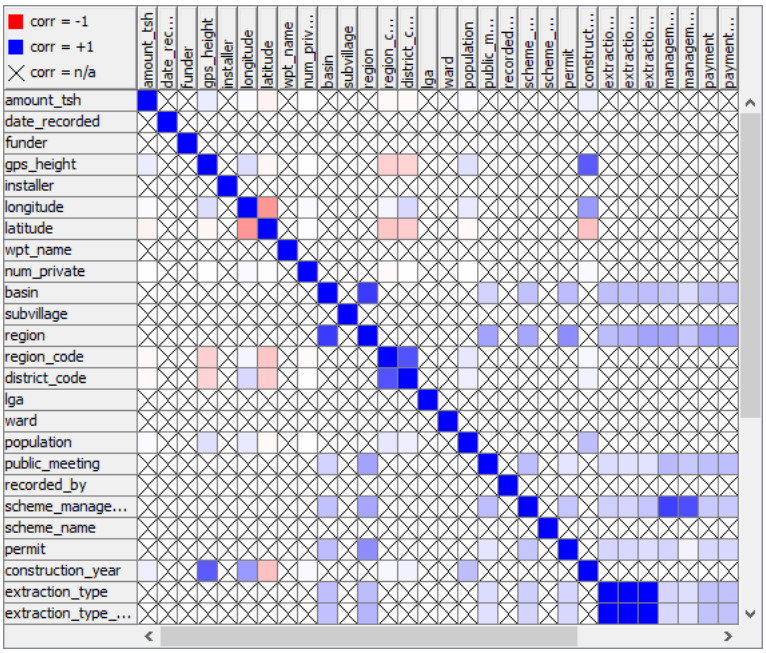

Figura 39: Correlación entre los atributos de Tanzania Power Pump

Observamos que los atributos `extraction_type`, `extraction_type_group` y `extraction_type_class` están correlados. Por lo que solo será necesario uno de ellos, nos quedaremos con `extraction_type_class` .

Los atributos `management` y `management_group` también están correlados. Nos quedaremos con `management_group`.

Los atributos `payment` y `payment_type` están correlados. Nos quedaremos con `payment_type`.

Los atributos `quality_group` y `water_quality` están correlados. Nos quedaremos con `quality_group` .

Los atributos `quantity` y `quantity_group` están correlados. Nos quedaremos con `quantity_group` .

Los atributos `source`, `source_type` y `source_class` están correlados. Por lo que solo será necesario uno de ellos, nos quedaremos con `source_class`.

Los atributos `waterpoint_type` y `waterpoint_type_group` están correlados. Nos quedaremos con `waterpoint_type_group`.

Además, vamos a convertir una característica categórica en varias binarias. Esto lo realizaremos sobre los atributos `source_class`, `scheme_management`, `management_group`, `extraction_type_class` , `public_meeting`, `permit`, `region`, `quantity_group`, `quality_group`, `water_type_group`, `payment_type` y `basin`.

También se ha usado el nodo `Missing Values` en cada algoritmo para tratar los valores perdidos, se ha utilizado la mediana para los valores numéricos y el valor más frecuente para los nominales.

De esta forma, el flujo de trabajo queda como se observa en la siguiente figura.

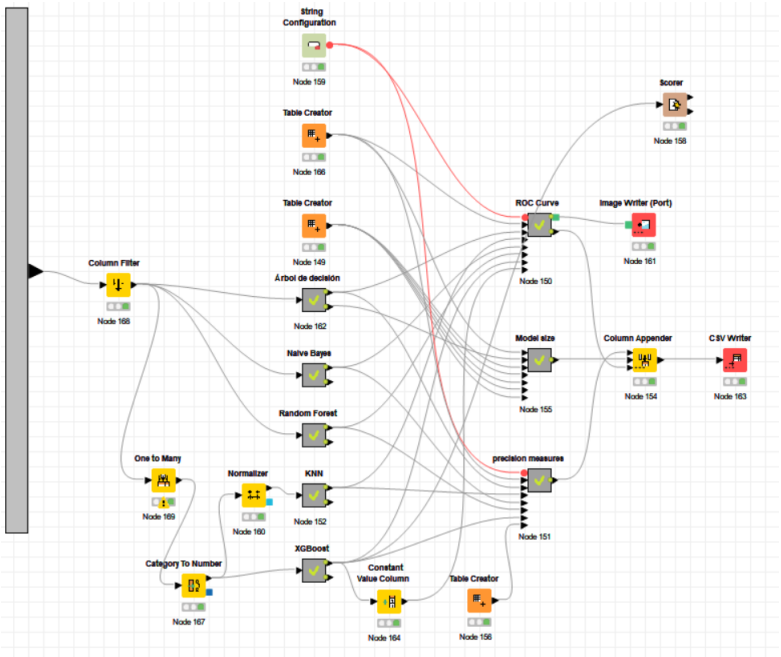

Figura 40: Metanodo Procesado de Tanzania Power Pump

En la siguiente tabla se muestran los resultados que teníamos y los que hemos obtenido al aplicar el procesado (P).

Tabla 62: Criterios de precisión sin y con procesado de Tanzania Power Pump

Tanzania Power Pump	TP	FP	TN	FN	TPR	TNR	PPV	Accuracy	F1-score	G-mean	Model size	AUC
Árbol de decisión	17748	4520	27493	4871	0.7847	0.8588	0.7970	0.8281	0.7908	0.8209	4882	0.8385
Árbol de decisión (P)	17731	4502	27583	4959	0.7814	0.8596	0.7975	0.8272	0.7893	0.8196	4434.6	0.8365
3-NN	16480	4781	27478	6344	0.7220	0.8518	0.7751	0.7980	0.7476	0.7842	NaN	0.8480
3-NN (P)	17305	4312	27947	5519	0.7581	0.8663	0.8005	0.8215	0.7787	0.8104	NaN	0.8713
Naive Bayes	15125	6716	25543	7699	0.6627	0.7918	0.6925	0.7383	0.6773	0.7244	NaN	0.8010
Naive Bayes (P)	16535	10417	21842	6289	0.7244	0.6770	0.6134	0.6967	0.6643	0.7003	NaN	0.7889
Random Forest	8905	670	31589	13919	0.3902	0.9792	0.9300	0.7351	0.5497	0.6181	NaN	0.8540
Random Forest (P)	12235	1734	30525	10589	0.5360	0.9462	0.8758	0.7762	0.6650	0.7122	NaN	0.8720
XGBoost	17322	2720	29539	5502	0.759	0.9157	0.8643	0.8507	0.8082	0.8336	NaN	0.9208
XGBoost (P)	17120	2717	29542	5704	0.7500	0.9157	0.8630	0.8471	0.8026	0.8288	NaN	0.9177

Y las curvas ROC sin y con procesado, respectivamente, son las siguientes.

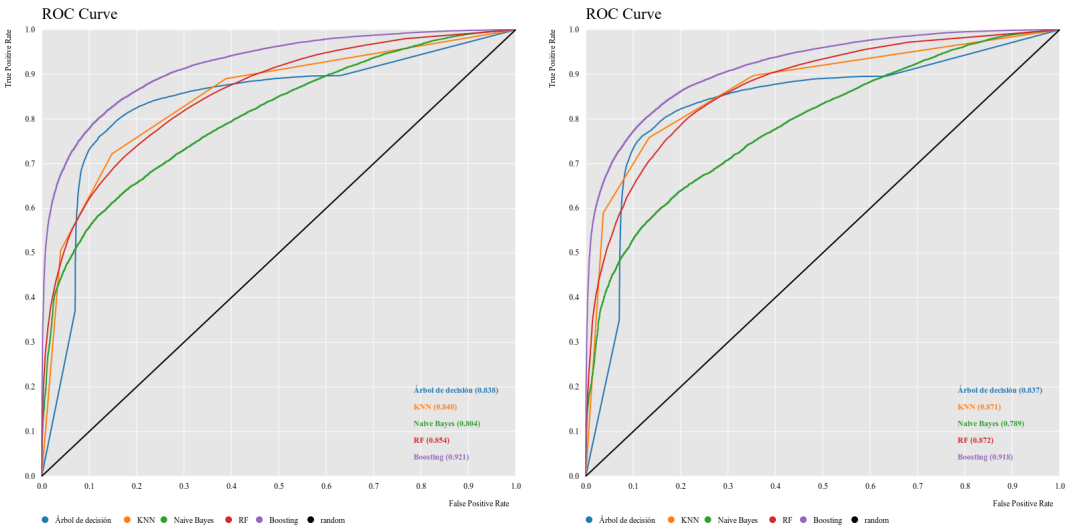

Figura 41: Curvas ROC sin y con procesado, respectivamente, de Tanzania Power Pump

En general, no observamos gran diferencia. El árbol de decisión ha reducido su tamaño, pues hemos eliminado varios atributos. Sin embargo, ha obtenido resultados un poco peores. Lo mismo ocurre con Naive Bayes o con XGBoost. Esto se debe a que puede que hayamos eliminado algunos atributos que eran relevantes para estos algoritmos. Sobre todo en Naive Bayes, pues hemos eliminado atributos que estaban correlados, es decir, que dependían y como este algoritmo se basaba en la independencia, observamos que no ha tenido el efecto que queríamos.

Por otra parte, otros algoritmos han mejorado, como es el caso de 3-NN y Random-Forest, este último ha mejorado notablemente. Esto puede deberse a que hemos convertido características categóricas en varias binarias, en el caso de 3-NN, o la eliminación de atributos con muchos valores perdidos.

6 Interpretación de resultados

En esta sección estudiaremos los factores que determinan cada clase. Para ello, el algoritmo de árboles de decisión proporciona modelos legibles. Esto es, genera una vista del árbol interactivo e interpretable, indicando cuáles son los atributos por los que se divide en cada caso. Esto nos será realmente útil a la hora de averiguar los atributos más relevantes en cada conjunto de datos. Además, también podremos aprovechas el algoritmo Random Forest, pues nos puede proporcionar información sobre los atributos más usados. En Random Forest podemos seleccionar la opción de mostrar las estadísitcas de los atributos para ver cuántas veces ha podido ser elegido cada atributo (columna candidate) y, de esas veces, cuántas veces realmente ha sido elegido (columna splits). Veamos lo que ocurre en cada conjunto de datos.

6.1 Heart Failure

En la siguiente imagen podemos ver una parte del árbol de decisión del conjunto de datos Heart Failure.

Observamos que la primera división que realiza el árbol es según el atributo `St_Slope`, esto significa que este atributo es el más determinante a la hora de clasificar la clase. Dependiendo del valor de esta variable, podemos ver los siguientes atributos relevantes, como `ChestPainType` o `Sex`.

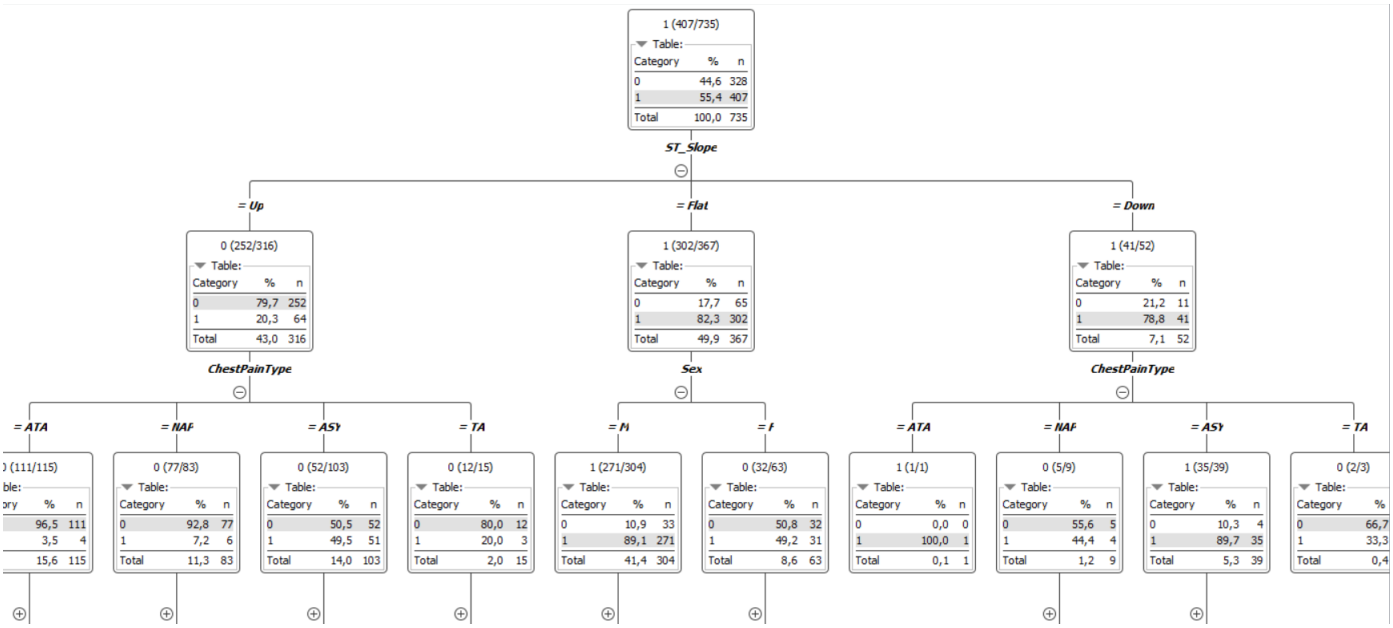

Figura 42: Vista del árbol de decisión de Heart Failure

Veamos las estadísticas de los atributos de Random Forest.

row ID	#splits (level 0)	#splits (level 1)	#splits (level 2)	#candidates (level 0)	#candidates (level 1)	#candidates (level 2)
Age	2	15	38	23	54	122
Sex	4	15	32	31	59	109
ChestPainType	26	31	52	31	54	104
RestingBP	0	3	30	29	66	107
Cholesterol	7	17	45	29	51	117
FastingBS	1	9	23	28	54	103
RestingECG	0	3	10	26	48	107
MaxHR	8	35	42	31	62	100
ExerciseAngina	21	20	27	28	56	108
Oldpeak	9	23	46	22	45	112
ST_Slope	22	29	51	22	51	111

Observamos que la variable `ChestPainType` es la más usada, seguida de `ST_Slope` y `ExerciseAngina` . De las 22 veces que `ChestPainType` ha sido candidata, todas las veces se ha elegido.

6.2 Mobile Price

En la siguiente imagen podemos ver una parte del árbol de decisión del conjunto de datos Mobile Price.

Observamos que la primera división que realiza el árbol es según el atributo `ram`, esto significa que este atributo es el más determinante a la hora de clasificar la clase, es decir, a la hora de determinar el precio de un móvil. Dependiendo del valor de esta variable, podemos ver los siguientes atributos relevantes, como `px_height` o `battery_power`.

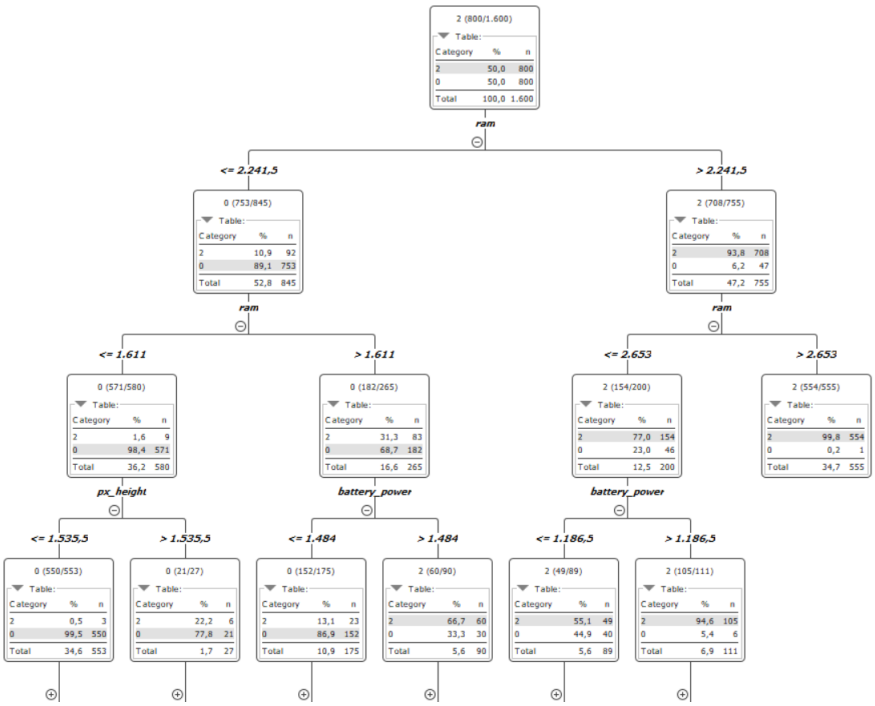

Figura 43: Vista del árbol de decisión de Mobile Price

Veamos las estadísticas de los atributos de Random Forest.

row ID	#splits (level 0)	#splits (level 1)	#splits (level 2)	#candidates (level 0)	#candidates (level 1)	#candidates (level 2)
battery_power	14	26	37	21	42	73
blue	0	1	0	14	37	76
clock_speed	3	11	20	20	39	86
dual_sim	1	1	4	20	44	79
fc	3	5	10	21	28	71
four_g	0	1	0	26	38	79
int_memory	6	7	26	20	40	73
m_dep	2	6	15	15	38	74
mobile_wt	8	15	33	25	42	96
n_cores	4	4	14	21	44	88
pc	4	5	20	19	47	78
px_height	12	21	32	22	33	72
px_width	15	27	41	19	43	85
ram	20	36	74	20	36	75
sc_h	2	8	18	28	39	102
sc_w	4	11	13	16	36	79
talk_time	1	12	22	21	43	86
three_g	0	1	4	25	53	68
touch_screen	1	0	6	14	38	67
wifi	0	2	3	13	40	93

Observamos que la variable `ram` es la más usada, seguida de `px_width` y `battery_power`. De las 20 veces que `ram` ha sido candidata, todas las veces se ha elegido.

6.3 Bank Marketing

En la siguiente imagen podemos ver una parte del árbol de decisión del conjunto de datos Bank Marketing.

Observamos que la primera división que realiza el árbol es según el atributo `nr.employed`, esto significa que este atributo es el más determinante a la hora de clasificar la clase. Dependiendo del valor de esta variable, podemos ver los siguientes atributos relevantes, en este caso `duration`.

Figura 44: Vista del árbol de decisión de Bank Marketing

Veamos las estadísticas de los atributos de Random Forest.

Atributo	#splits (level 0)	#splits (level 1)	#splits (level 2)	#candidates (level 0)	#candidates (level 1)	#candidates (level 2)
age	2	2	9	23	41	78
job	1	3	15	27	43	62
marital	0	1	2	16	40	69
education	1	1	6	16	51	85
default	1	4	2	17	39	86
housing	0	1	0	20	38	69
loan	0	0	2	26	33	87
contact	1	9	15	20	44	85
month	3	18	42	12	47	88
day_of_week	0	2	8	16	39	75
duration	9	37	74	13	38	86
campaign	0	0	8	21	38	81
pdays	16	13	34	25	36	85
previous	0	7	10	19	44	93
poutcome	8	18	27	23	41	91
emp.var.rate	7	14	21	24	39	83
cons.price.idx	6	15	29	19	42	81
cons.conf.idx	5	10	25	18	27	73
euribor3m	13	24	36	18	39	72
nr.employed	27	21	34	27	41	71

Observamos que la variable `nr.employed` es la más usada, seguida de `pdays` y `euribor3m`. De las 27 veces que `nr.employed` ha sido candidata, todas las veces se ha elegido.

6.4 Tanzania Power Pump

En la siguiente imagen podemos ver una parte del árbol de decisión del conjunto de datos Tanzania Power Pump.

Observamos que la primera división que realiza el árbol es según el atributo `quantity_group`, esto significa que este atributo es el más determinante a la hora de clasificar la clase, es decir, a la hora de determinar el precio de un móvil. De esta forma, el árbol se subdivide según los valores de esta variable, `enough`, `insufficient`, `seasonal`, `dry` o `unknown`.

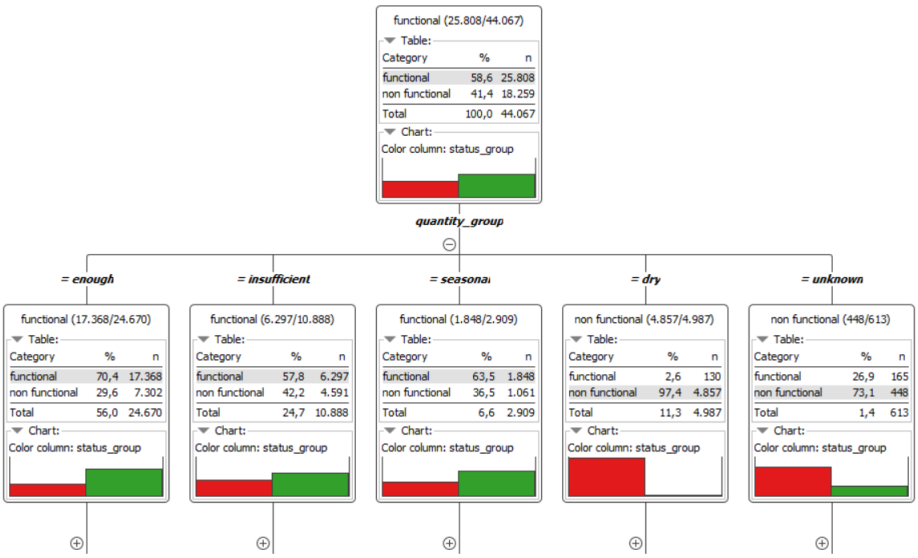

Figura 45: Vista del árbol de decisión de Tanzania Power Pump

Veamos las estadísticas de los atributos de Random Forest.

row ID	#splits (level 0)	#splits (level 1)	#splits (level 2)	#candidates (level 0)	#candidates (level 1)	#candidates (level 2)
amount_tsh	1	1	3	13	39	69
date_recorded	0	8	22	15	36	65
funder	4	6	19	10	23	54
gps_height	0	0	0	17	32	64
installer	3	10	33	14	35	66
longitude	0	0	0	19	16	69
latitude	0	0	0	17	35	68
wpt_name	13	25	62	13	25	63
num_private	0	0	0	16	34	63
basin	0	0	0	18	28	58
subvillage	11	22	62	12	22	68
region	0	1	10	10	33	43
region_code	0	0	0	13	27	76
district_code	0	0	0	11	40	71
lga	2	7	14	20	32	54
ward	16	25	33	17	32	50
population	0	0	1	11	28	61
public_meeting	0	0	2	17	37	58
recorded_by	0	0	0	12	32	67
scheme_management	0	1	3	17	32	64
scheme_name	8	14	31	21	29	68
permit	0	0	0	19	33	56
construction_year	0	0	4	14	24	65
extraction_type	8	9	9	21	29	59
extraction_type_group	1	5	9	10	29	61
extraction_type_class	4	5	6	14	35	76
management	0	2	1	17	31	57
management_group	0	0	0	18	29	62
payment	3	8	5	20	33	56
payment_type	0	2	2	10	34	60
water_quality	1	2	1	18	30	59
quality_group	0	1	3	18	27	66
quantity	8	15	16	15	33	62
quantity_group	7	16	15	12	38	41
source	0	1	6	15	40	54
source_type	0	0	3	17	31	57
source_class	0	0	1	17	24	56
waterpoint_type	7	10	10	16	25	64
waterpoint_type_group	3	4	14	16	28	70

Observamos que la variable `ward` es la más usada, seguida de `wpt_name` y `subvillage`. De las 17 veces que `ward` ha sido candidata, 16 veces se ha elegido.

7 Contenido adicional

A continuación se muestran los flujos de trabajo en cada conjunto de datos.

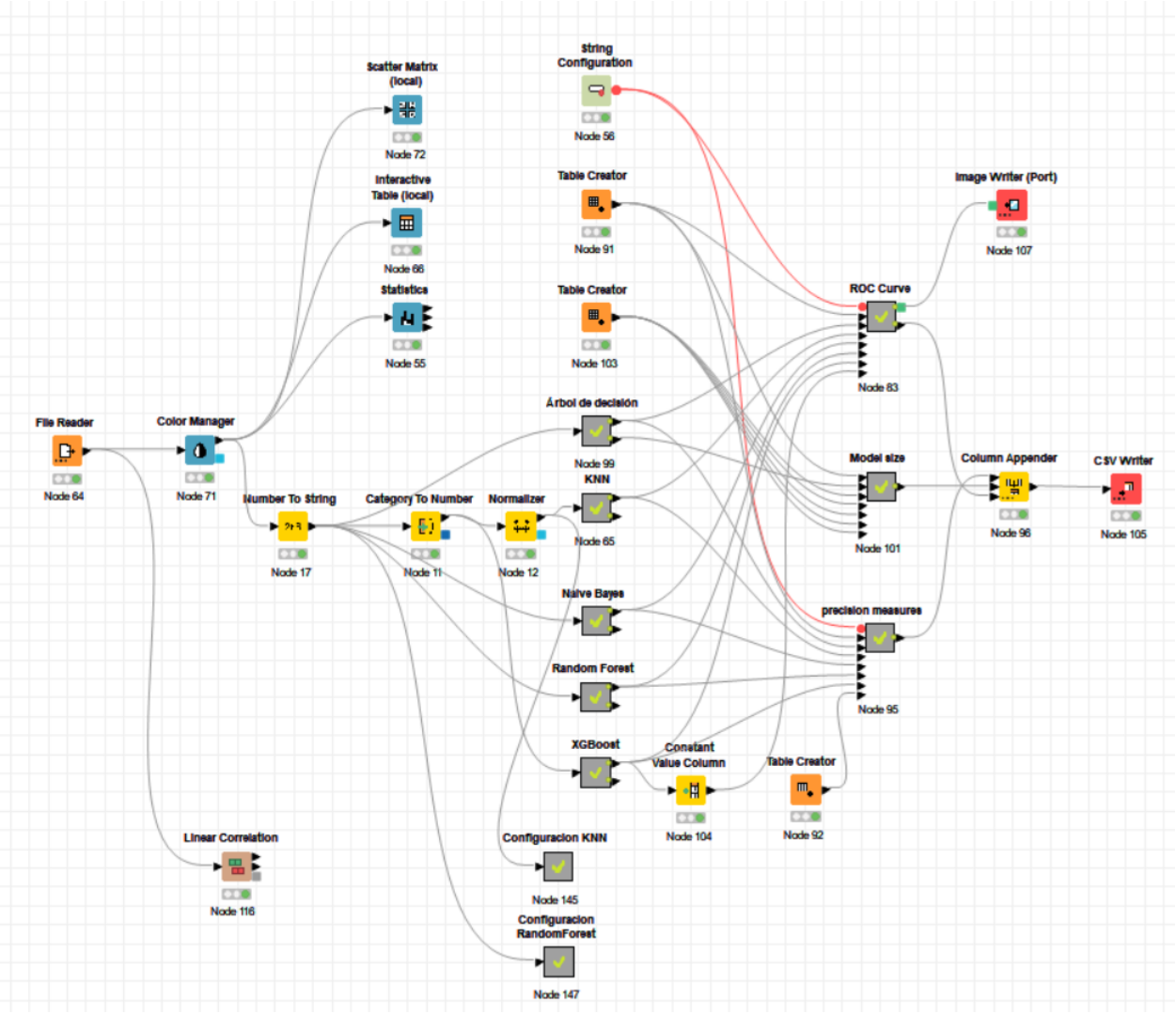

Figura 46: Flujo de trabajo de Heart Failure

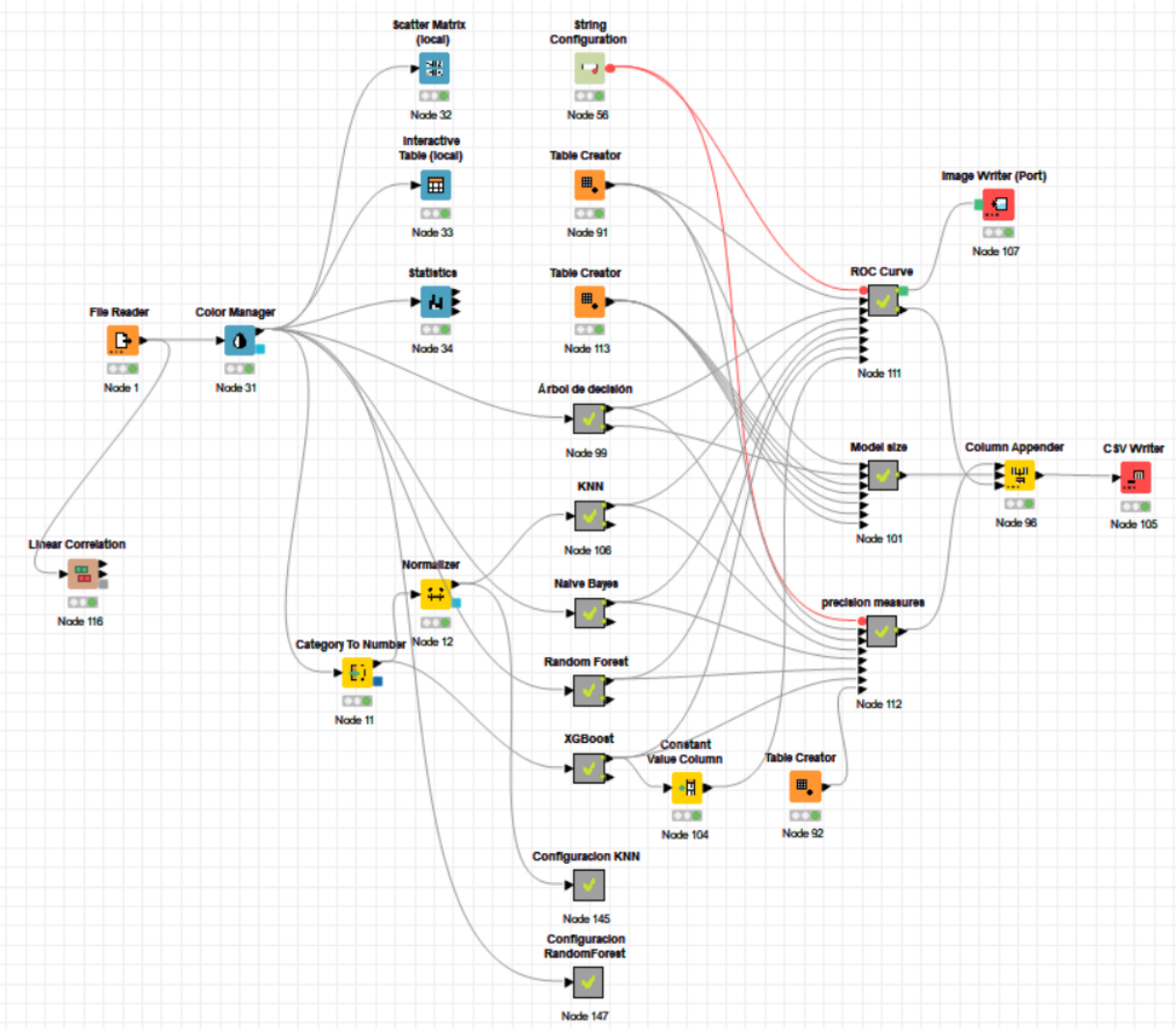

Figura 47: Flujo de trabajo de Bank Marketing

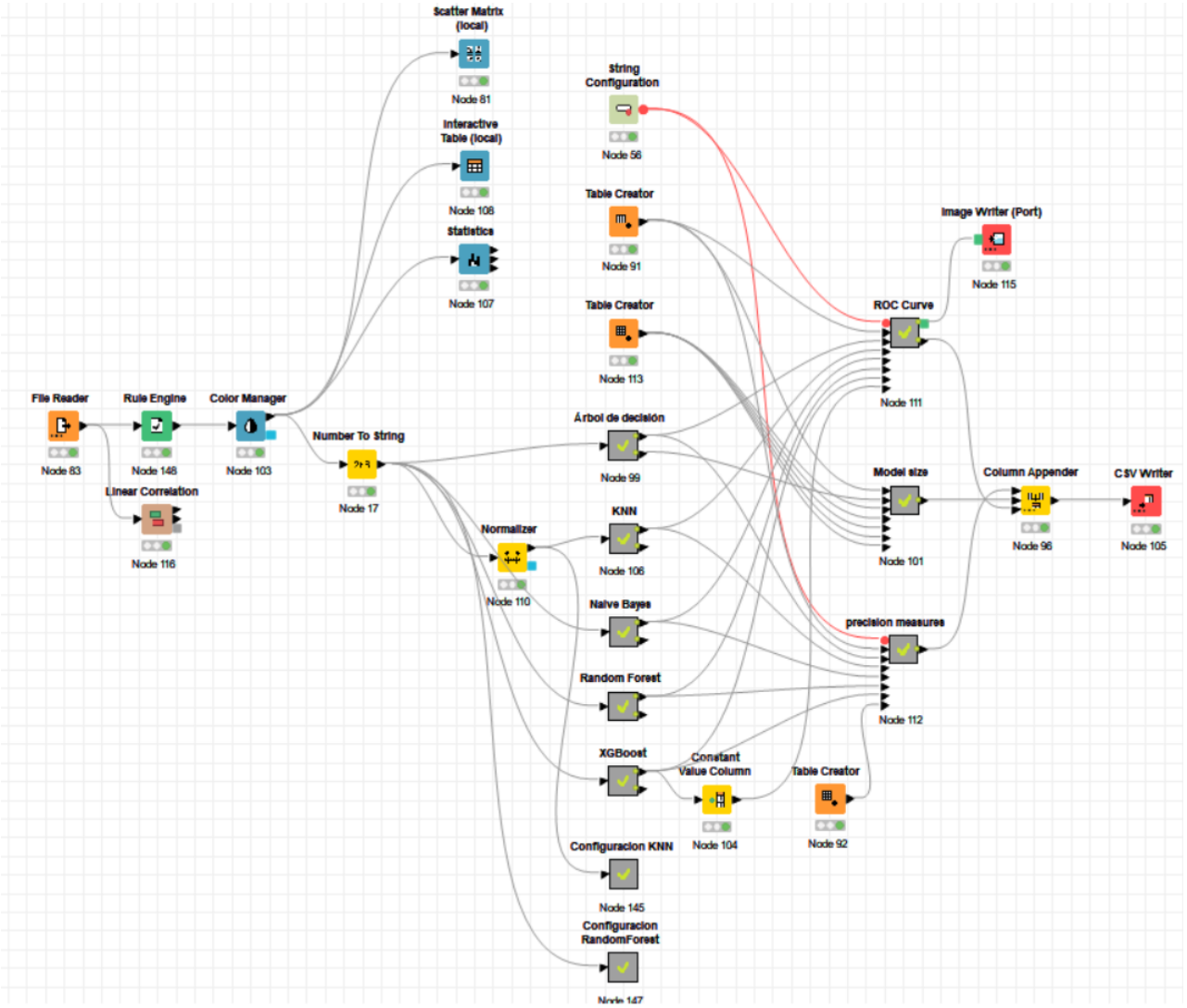

Figura 48: Flujo de trabajo de Mobile Price

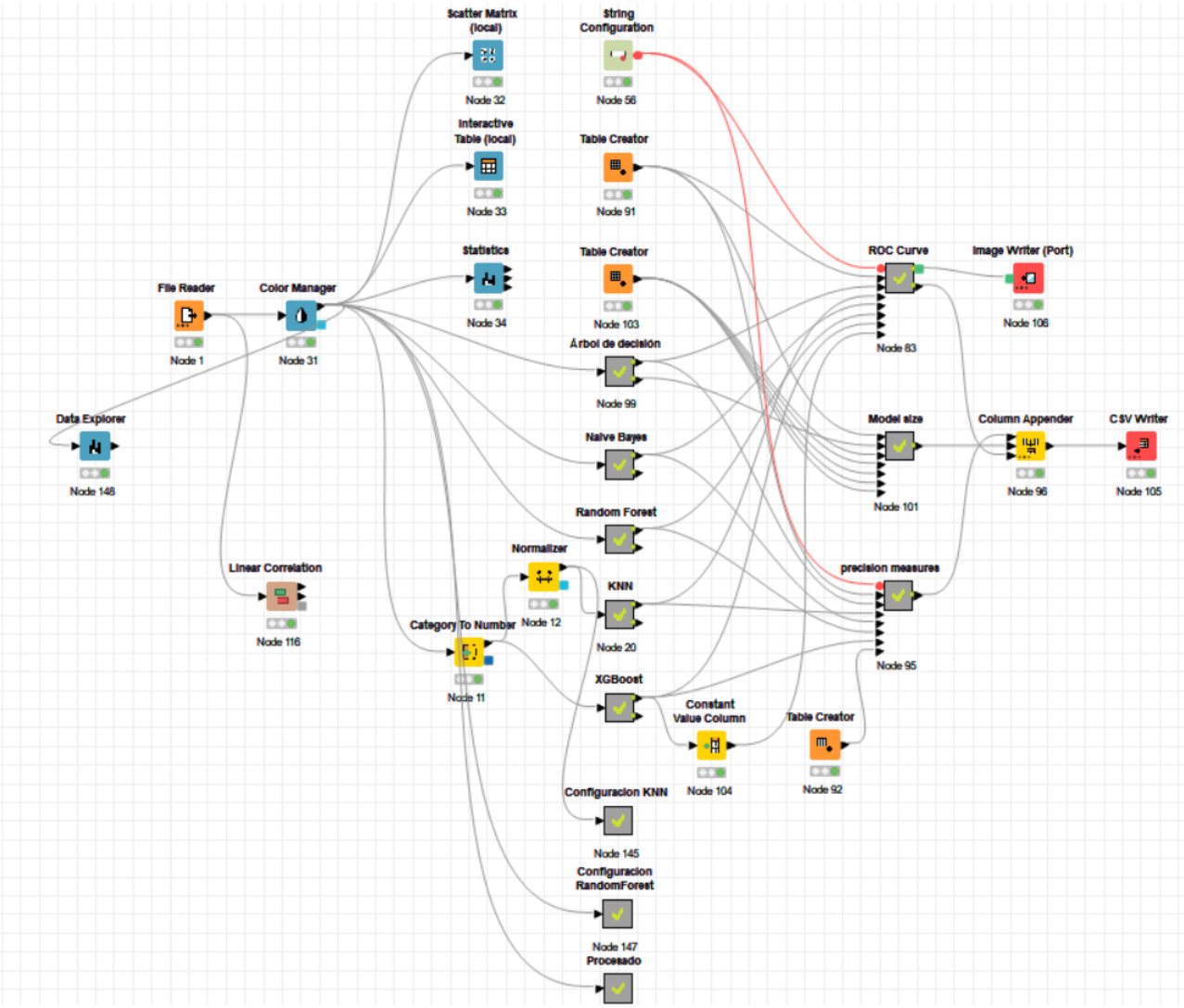

Figura 49: Flujo de trabajo de Tanzania Power Pump

8 Bibliografía

- Material proporcionado por los profesores sobre la asignatura.
  https://ccia.ugr.es/~casillas/knime.html
- Foro de KNIME
  https://forum.knime.com/