Universidad Europea

UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN INGENIERÍA INFORMÁTICA

SISTEMAS INTELIGENTES

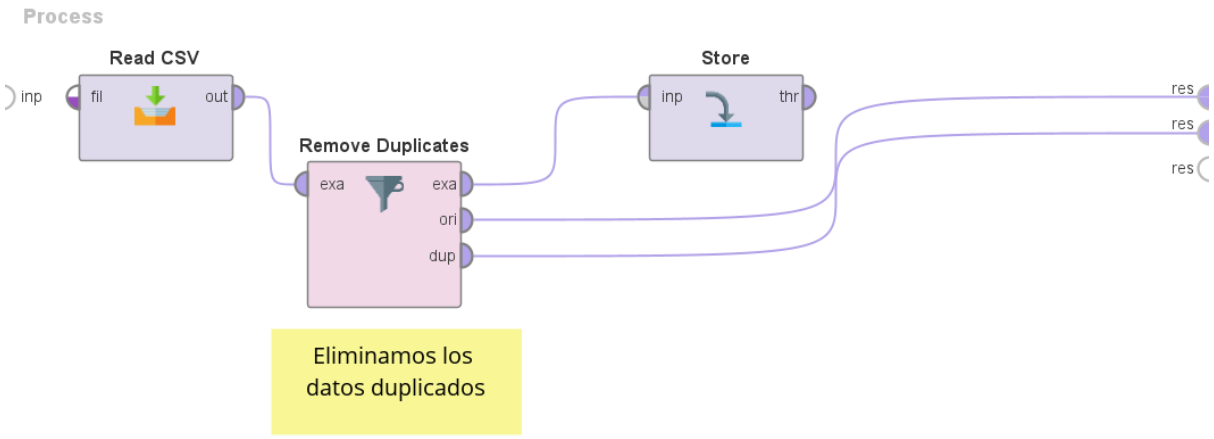
PRÁCTICA 1

GUILLERMO AÓS RODERA

UE222A197

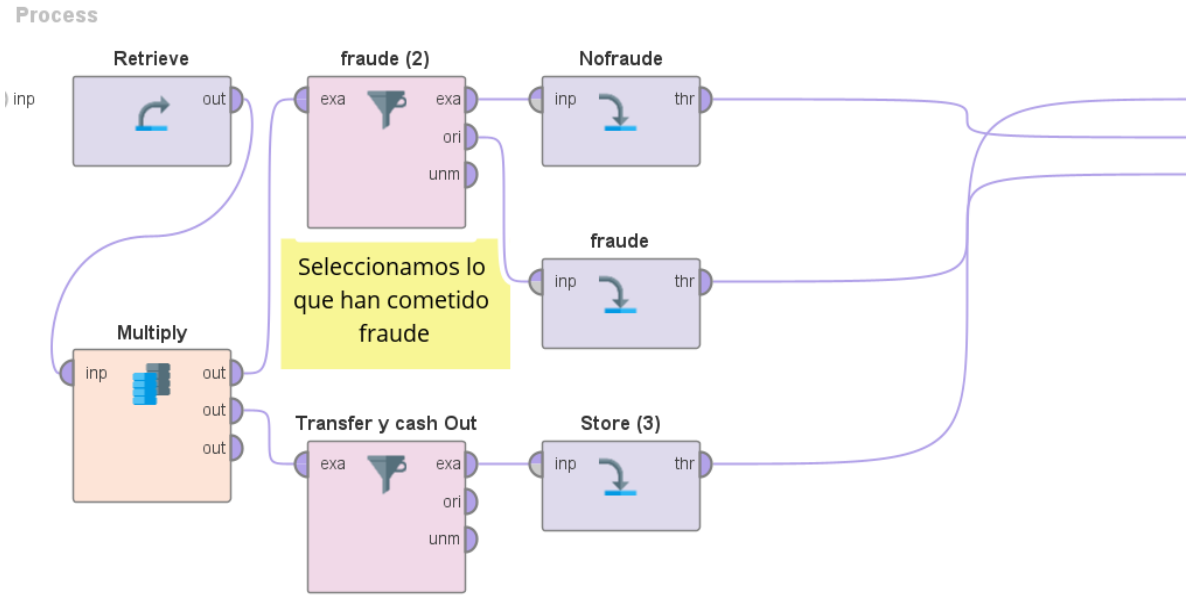
EJERCICIO 1

A) Limpieza y prepocesamiento de datos



- Leer el csv de las transacciones de la tarjeta de crédito
- Eliminamos los duplicamos
- Lo guardamos como un dataset nuevo

B) Análisis Exploratorio de Datos (EDA)

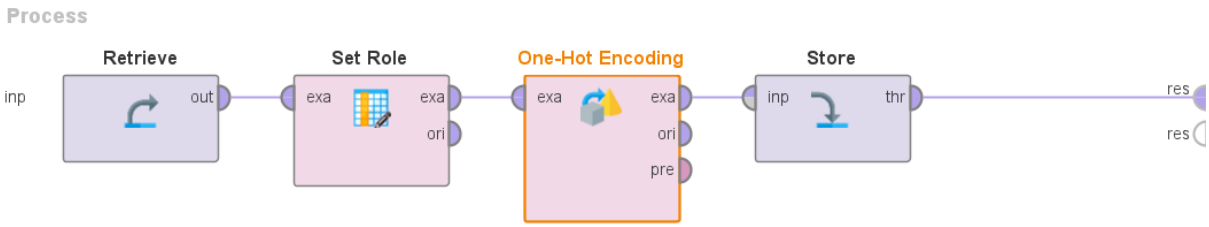


Con el dataset que sacamos en el apartado anterior hacemos lo siguiente:

- Lo divimos en los que han cometido fraude y no
- Seleccionamos también los types que sean con transfer y cash out ya que es mas probable que se cometa fraude con esa transacciones.

- Cargamos el dataset en el que solo aparecen el cashout y transfer que es lo que nos interesa
- Convertimos la columna de fraude de numérica categórica, para que sea mas fácil para el algoritmo aplique la lógica adecuada para la división de datos.
- Seleccionamos la columna de fraude que es la que vamos a querer predecir posteriormente y type(las declaramos como 'metadata', lo que significa que no se utilizará directamente para el modelado pero contiene información importante sobre los datos)
- Optimizamos el conjunto de datos previo al análisis de modelado
    -Establecido en 0.0, lo que significa que cualquier atributo numérico que no varíe será eliminado.
    -Fijado en 0.6, lo que indica que cualquier atributo nominal en el cual más del 60% de los registros compartan el mismo valor será descartado
    -Establecido en 0.4, significa que cualquier atributo nominal en el que menos del 40% de los registros compartan un valor específico también será eliminado
- Lo balanceamos para evitar sesgos en los modelos de aprendizaje automático al asegurar que todas las clases sean representadas equitativamente, mejorando así la generalización y la precisión en las predicciones sobre datos no vistos.
- Lo guardamos como balancedata

C)INGENIERIA DE CARACTERISTICAS

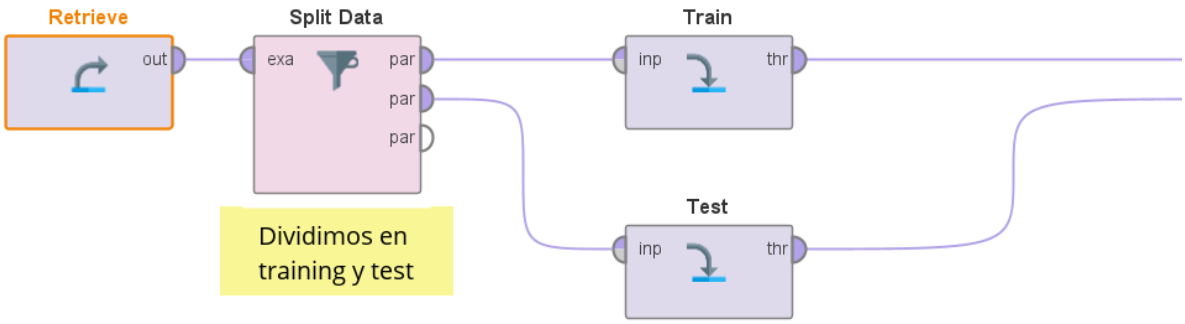


- Cogemos los datos que anteriormente hemos balanceado.
- Indicamos que la variable type es regular lo que indica debe ser tratada como un atributo predictivo regular en el análisis de datos y modelado. Esto significa que 'type' no será utilizada como una variable objetivo (label) ni

como un peso, identificador único (id) o alguna otra variable especial, sino como una característica estándar que puede influir en la predicción de un modelo sin ser la característica que se intenta predecir.

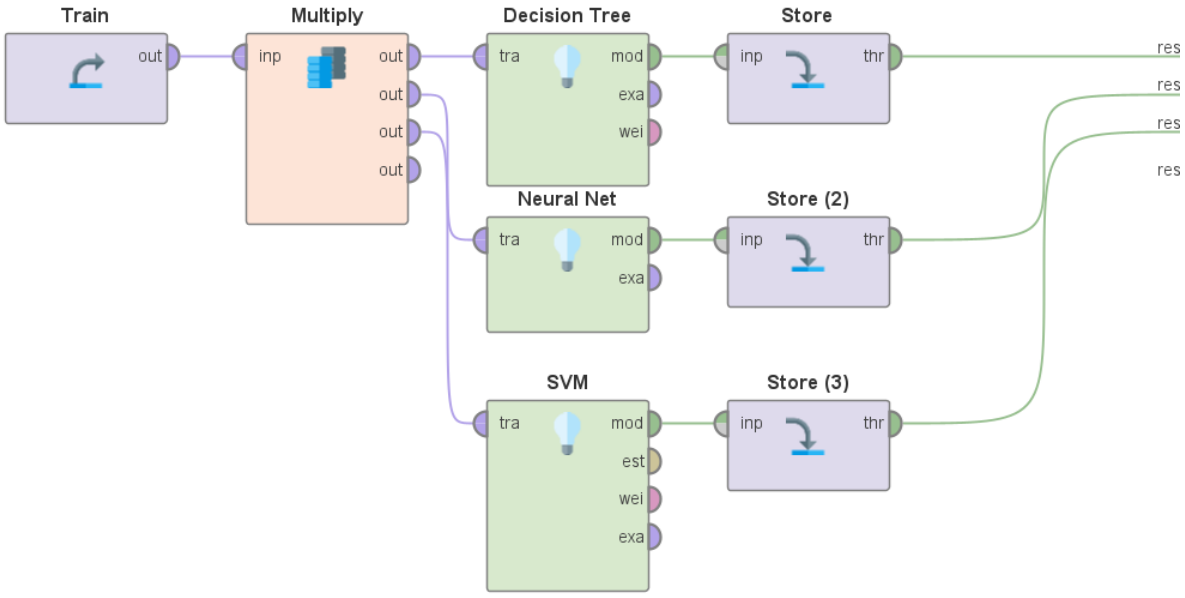
- transformamos la variable type de atributos categóricos a un conjunto de atributos binarios para que no sea mas útil para modelos de aprendizaje automático que requieren entradas numéricas, ya que permite representar datos categóricos de manera que puedan ser correctamente interpretados por dichos modelos.
- Lo guardamos como BalanceddataOHE

D) PREPARACION DE DATOS PARA EL MODELADO



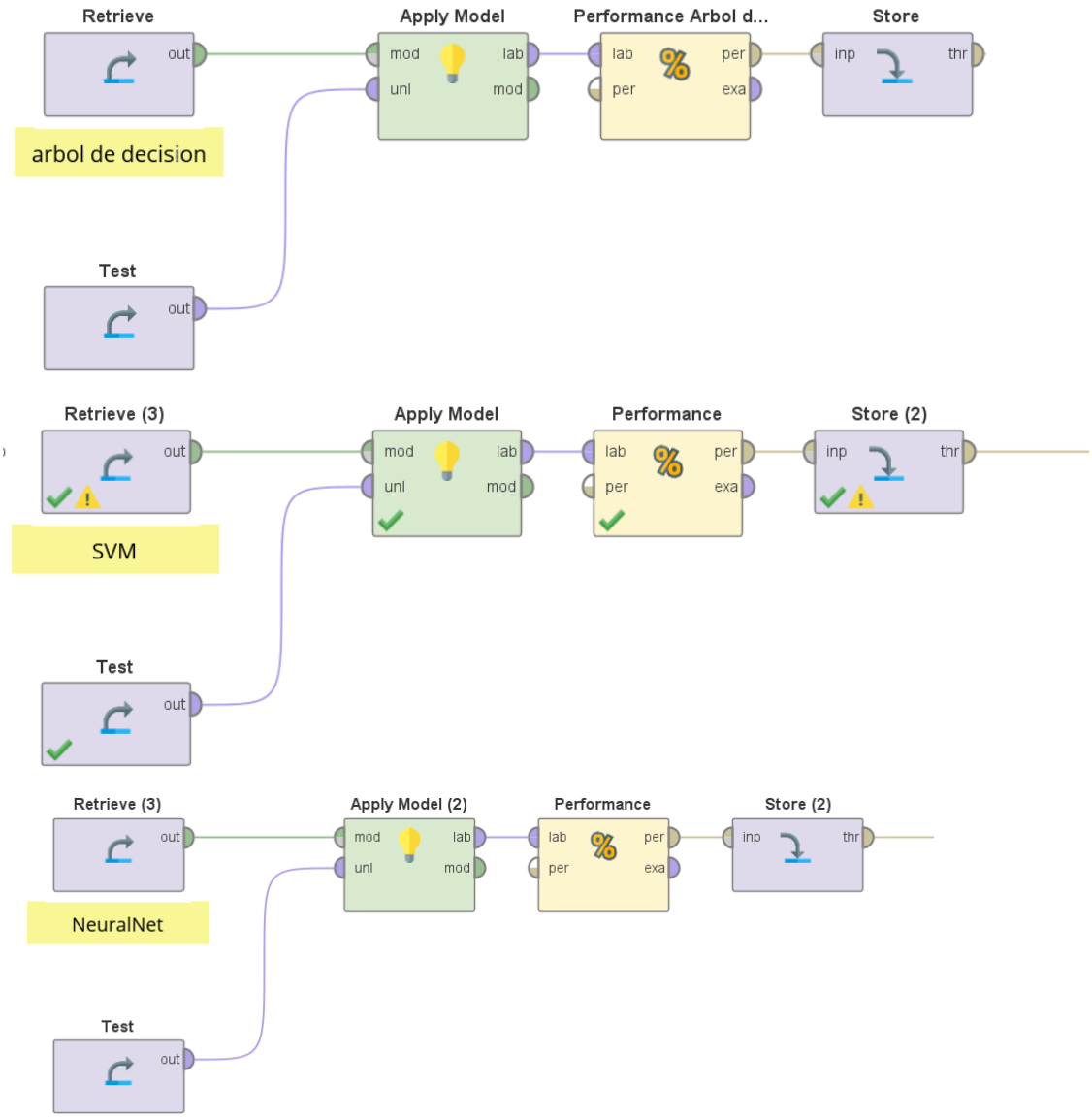
- Dividimos el BalancedDataOHE en Train(80%) y Test(20%)

E) SELECCIÓN Y ENTRENAMIENTIO DEL MODELO



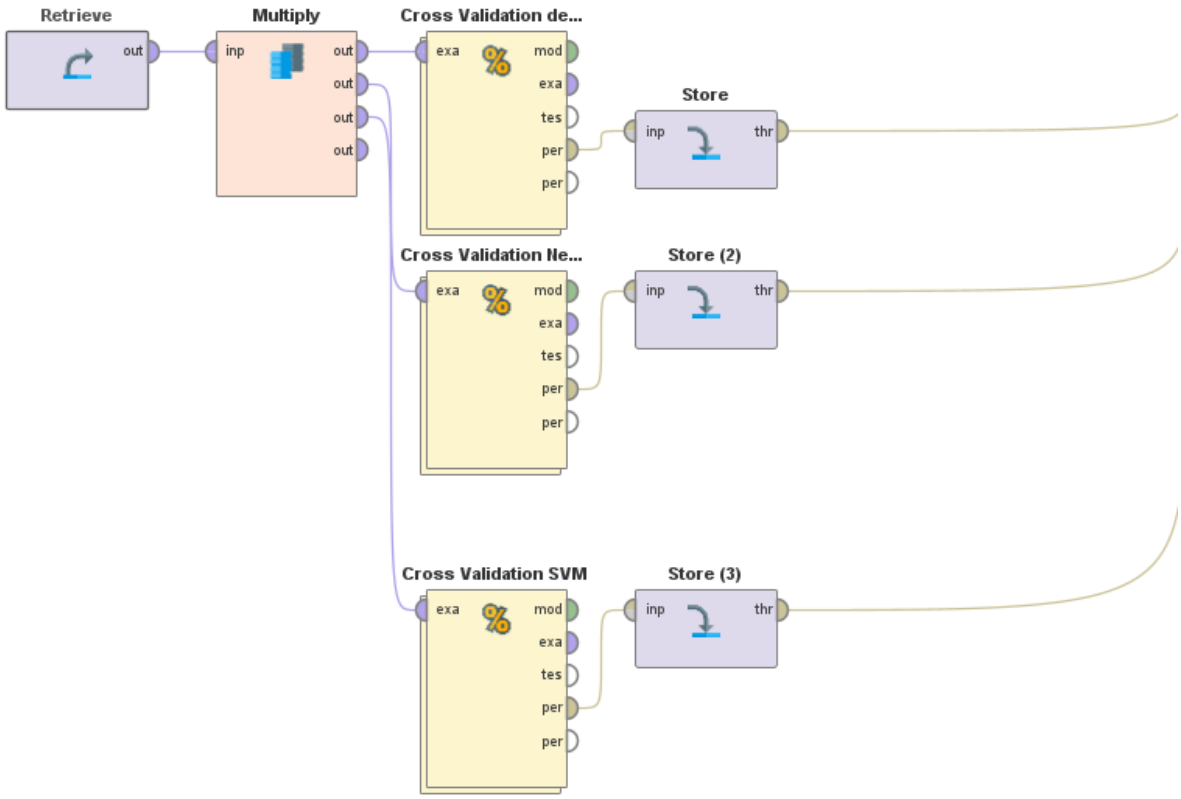
- Cargamos el dataset de train y aplicamos tres modelos que son el árbol de decisio, SVM y NeuraNet.
- Guardamos cada uno de los modelos

F) EVALUACIÓN DEL MODELO



- Evaluamos las predicciones del modelo con las etiquetas reales del conjunto de datos de prueba y vemos métricas de rendimiento.

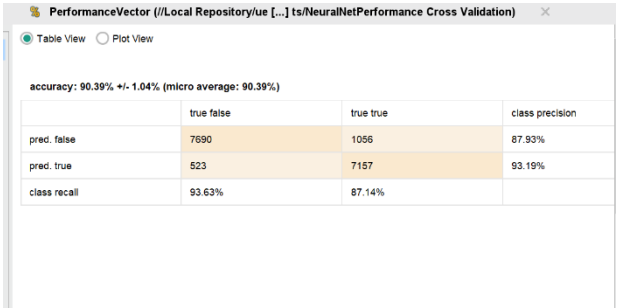
G) VALIDACION Y OPTIMIZACIÓN DEL MODELO



- cargamos el conjunto de datos de entrenamiento y luego aplicamos validación cruzada para evaluar tres modelos distintos: Decision Tree, Neural Net y SVM.
- Los resultados de cada modelo se almacenan individualmente para su análisis y comparación posterior.

E) RESULTADOS

1. NeuralNet: Tiene una precisión general del 90.39%, con un recall para la clase positiva (verdaderos positivos) del 87.14% y una precisión para la misma del 93.19%.



PerformanceVector (//Local Repository/ue [...] ts/NeuralNetPerformance Cross Validation)

Table View | Plot View

accuracy: 90.39% +/- 1.04% (micro average: 90.39%)

	true false	true true	class precision
pred. false	7690	1056	87.93%
pred. true	523	7157	93.19%
class recall	93.63%	87.14%	

2. SVM: La precisión general es más baja, del 77.26%, con un recall para la clase positiva del 71.15% y una precisión para la misma del 77.26%.

precision: 77.26% (positive class: true)

	true false	true true	class precision
pred. false	1299	474	73.27%
pred. true	344	1169	77.26%
class recall	79.06%	71.15%	

3. Decision Tree: Muestra la mejor precisión general del 93.96%, con un recall impresionante para la clase positiva del 99.54% y una precisión para la misma del 93.95%.

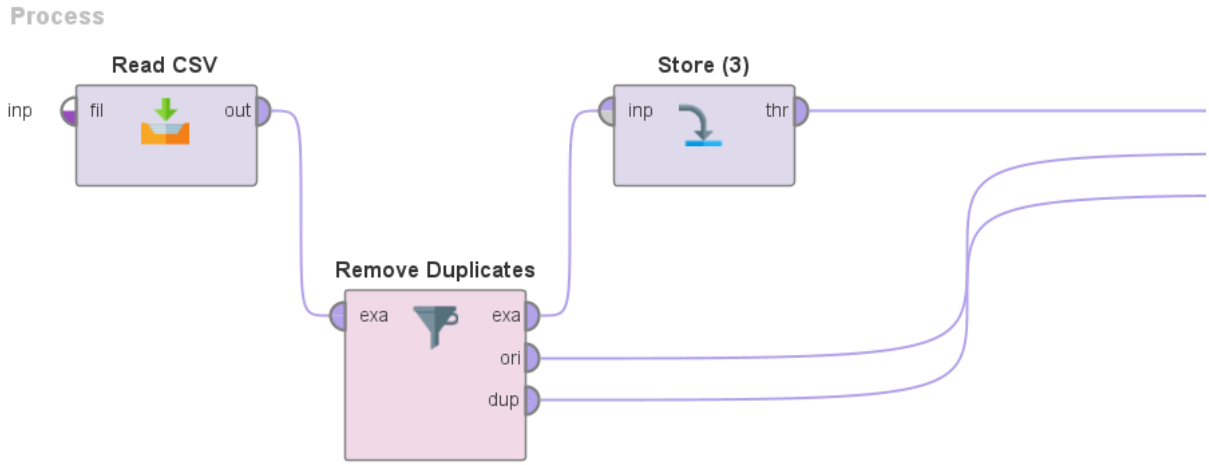
precision: 93.96% +/- 0.53% (micro average: 93.95%) (positive class: true)

	true false	true true	class precision
pred. false	7687	38	99.51%
pred. true	526	8175	93.95%
class recall	93.60%	99.54%	

El Decision Tree parece ser el mejor modelo para predecir el fraude de tarjetas debido a su alta precisión general y especialmente a su alta tasa de recall para la clase positiva, lo cual es crucial en contextos de fraude, ya que se quiere capturar la mayor cantidad posible de eventos fraudulentos (verdaderos positivos) mientras se minimiza la cantidad de transacciones legítimas clasificadas incorrectamente como fraude (falsos positivos).

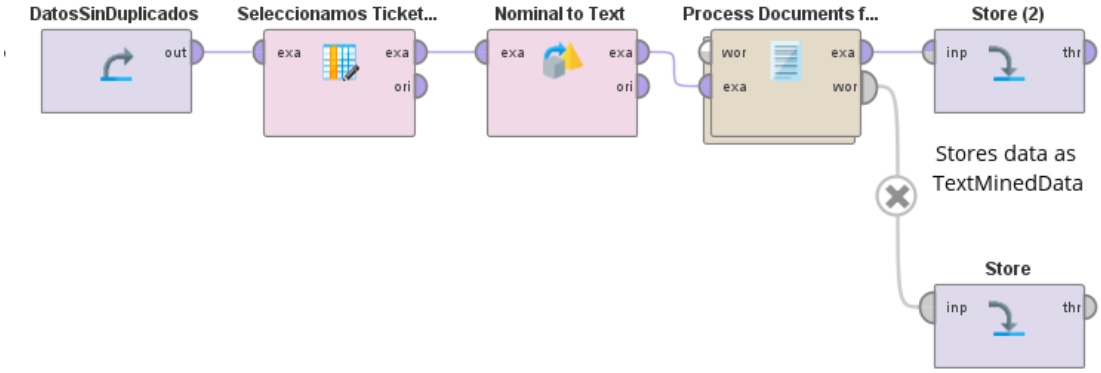
EJERCICIO 2

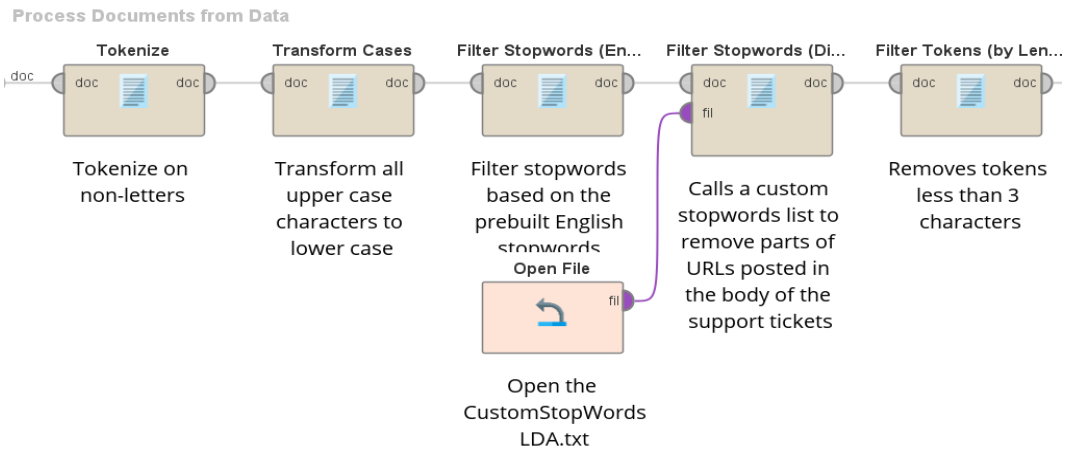
A)ADQUISICION DE DATOS



- Cargamos el dataset con toda la información
- Eliminamos los duplicados
- Lo guardamos de nuevo sin los duplicados

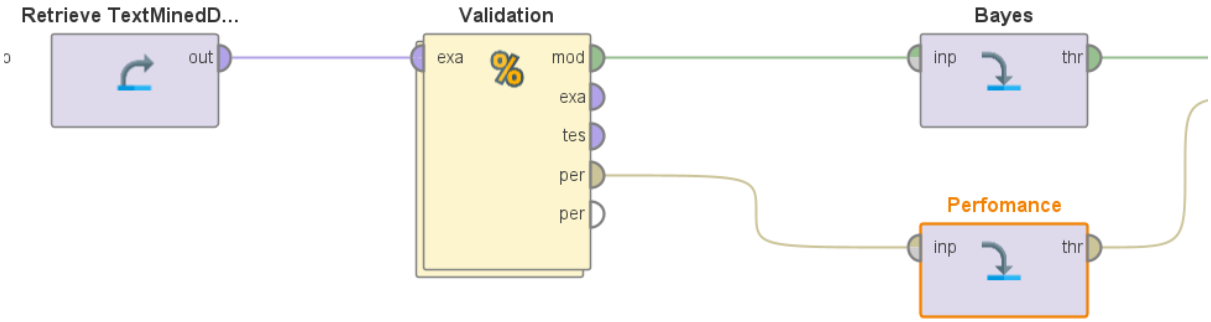
C) MINADO DE TEXTO

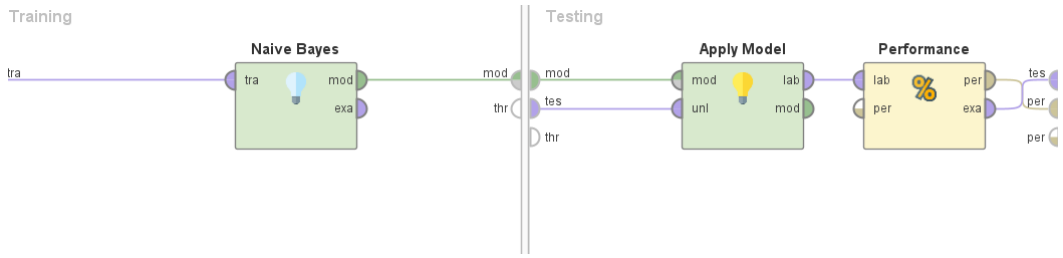




- **DatosSinDuplicados**: conjunto de datos ya limpio de duplicados, listo para el procesamiento adicional.
- **Seleccionamos TicketType**: seleccionamos la columna TicketType
- **Nominal to Text**: convertimos en categorías o etiquetas, en representaciones de texto para que puedan ser procesadas como datos de texto en etapas de preprocesamiento o modelado posteriores. Para facilitar la posterior Tokenizacion en otras cosa
- **Process Documents from Data**:
  - **Tokenize**: Divide el texto en tokens (palabras o términos) basados en caracteres no alfabéticos.
  - **Transform Cases**: Convierte todos los caracteres a minúsculas para uniformizar el texto.
  - **Filter Stopwords (English**): Elimina las palabras comunes en inglés que no aportan significado distintivo al texto.
  - Anadimos un txt con las palabras que queremos eliminar.
  - **Filter Stopwords (Dinámico):** Utiliza una lista personalizada de stopwords para eliminar términos irrelevantes que podrían estar presentes en los tickets de soporte, como partes de URLs.
  - **Filter Tokens (by Length):** Descarta tokens (palabras) que tengan menos de tres caracteres, ya que suelen aportar poco valor semántico.
- Guardamos tanto el texto minado como las wordlist.

D) ENTRENAMIENTO DEL MODELO





- Con el textoMinado hacemos un cross validation con el modelo de bayes y nos guardamos la performance y el modelo en si.

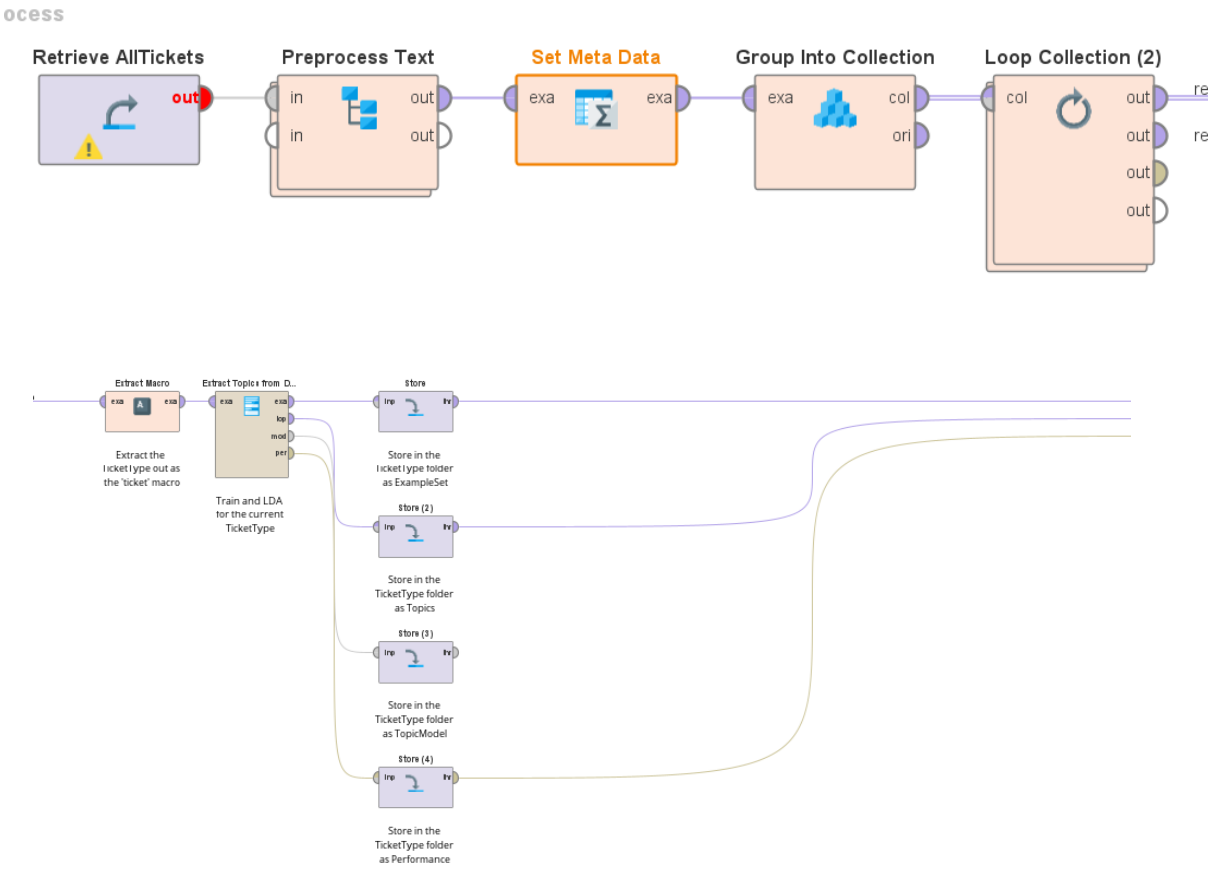
accuracy: 85.79% +/- 1.21% (micro average: 85.79%)

	true 3DPrinting	true Android	true Apple	true Dba	true Unix	class pr
pred. 3DPrinting	683	2	3	3	13	97.02%
pred. Android	5	905	138	2	66	81.09%
pred. Apple	15	109	1194	14	152	80.46%
pred. Dba	5	4	15	1109	88	90.83%
pred. Unix	17	34	112	65	1312	85.19%
class recall	94.21%	85.86%	81.67%	92.96%	80.44%	

La matriz proporciona información detallada sobre el rendimiento del modelo para cada clase, lo cual es crucial para entender dónde podría necesitar mejoras, como en la diferenciación entre ciertas clases o en la mejora de la captura de casos de clases específicas.

1. **Precisión General**: La precisión global del modelo es del 85.79%, lo que indica que, en promedio, el modelo hace una predicción correcta aproximadamente el 85.79% del tiempo.
2. **Variación en la Precisión**: Hay variabilidad en la precisión de las clases individuales, indicada por el "class precision". 3DPrinting tiene la precisión más alta con 97.02%, mientras que Apple tiene la precisión más baja con 80.46%.
3. **Recall de la Clase**: El "class recall" o sensibilidad de las clases también varía, con Dba teniendo el recall más alto (92.96%) y Unix el más bajo (80.44%). Esto indica que el modelo es mejor para identificar correctamente todos los casos relevantes de Dba y peor para Unix.
4. **Errores de Clasificación**: Hay cierta confusión entre las clases, por ejemplo, hay 138 instancias donde el modelo predijo Android en lugar de Apple y 152 instancias donde predijo Apple en lugar de Unix.
5. **Falsos Positivos y Negativos**: Los números fuera de la diagonal principal de la matriz (como los 15 casos de 3DPrinting predichos como Apple) son casos de falsos positivos y negativos, lo cual es importante para entender dónde el modelo se confunde.

E) GENERACION DE TÓPICOS



- **Preprocess Text**: Procesa el texto de los tickets para preparar los datos para el análisis.
    - ✓ Convertimos los datos estructurados en documentos
    - ✓ Realizamos un bucle sobre cada documento individual aplicando una serie de pasos de preprocesamiento de texto. Estos pasos incluyen la eliminación de stopwords, la tokenización, la lematización o la eliminación de palabras cortas y/o irrelevantes. Este bucle asegura que cada texto se procese de manera uniforme.
    - ✓ Recogemos los documentos procesados y los volvemos a convertir en un formato de datos estructurados, para que sea mas sencillo para el modelado
- **Set Meta Data**: Definimos los metadatos de los datos procesados indicando su tipo y su rol.
- **Group Into Collection**: Agrupa los datos procesados en una colección, que podría ser utilizada para realizar operaciones en lotes.

- **Loop Collection**: Itera a través de la colección de datos procesados, para aplicar una operación o análisis específico a cada grupo o ticket individualmente.
- **Extract Macro**: extraemos información específica de los datos, como categorías o tipos de tickets, para usarla en el análisis o para filtrar los datos.

- **Extract Topic from Data**: Se aplica LDA o un método similar de modelado de tópicos para identificar temas recurrentes en los textos de los tickets. Este proceso puede ayudar a entender las categorías principales de discusión o problemas reportados.
- **Train and LDA**: Entrena un modelo LDA con los datos de texto preprocesados para descubrir los tópicos latentes en el conjunto de datos.
- Guardamos los diferentes tipos de salida del análisis de tópicos.

Row No.	documentid	prediction(T...	confidence(...	confidence(...	confidence(...	confidence(...	con
1	0	Topic_2	0.077	0.085	0.187	0.125	0.06
2	1	Topic_4	0.092	0.043	0.082	0.132	0.15
3	2	Topic_1	0.088	0.152	0.099	0.099	0.07
4	3	Topic_6	0.052	0.133	0.053	0.185	0.07
5	4	Topic_3	0.073	0.080	0.080	0.189	0.12
6	5	Topic_4	0.066	0.115	0.083	0.079	0.17
7	6	Topic_5	0.112	0.092	0.139	0.057	0.07
8	7	Topic_4	0.082	0.109	0.109	0.116	0.12
9	8	Topic_7	0.094	0.058	0.121	0.118	0.06
10	9	Topic_3	0.077	0.101	0.108	0.190	0.08
11	10	Topic_7	0.065	0.054	0.054	0.063	0.10
12	11	Topic_8	0.071	0.058	0.074	0.122	0.05
13	12	Topic_2	0.093	0.064	0.260	0.068	0.06

- **Document ID**: Cada documento en el conjunto de datos está identificado por un 'documentid', que es identificador único para cada entrada.
- **Predicción de Tópicos**: La columna 'prediction(Topic_x)' indica el tópico más probable asignado a cada documento. Por ejemplo, al primer documento se le ha asignado el tópico 2 como el más relevante.
- **Confianza de la Predicción**: Las columnas 'confidence(...)' representan la probabilidad asignada a cada documento de pertenecer a cada uno de los tópicos posibles. Estos valores dan una medida de cuán confiado está el modelo en sus predicciones. Por ejemplo, para el primer documento, la mayor confianza es para el tópico 2 con un valor de aproximadamente 0.187.
- **Distribución de Tópicos**: La distribución de la confianza a través de diferentes tópicos para cada documento puede indicar la claridad o mezcla de tópicos dentro de ese documento. Por ejemplo, si un

documento tiene valores de confianza relativamente altos y cercanos entre varios tópicos, esto puede sugerir que el documento toca varios temas o que el modelo no está completamente seguro de su clasificación.

Este tipo de datos es útil para comprender cómo los documentos están distribuidos a través de diferentes tópicos y para evaluar la certeza del modelo en su clasificación. Para mejorar la interpretación y acción basada en estos resultados, uno podría considerar solo las predicciones con un nivel de confianza por encima de cierto umbral, o investigar documentos para los cuales el modelo muestra incertidumbre.