UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN INGENIERÍA INFORMÁTICA

SISTEMAS INTELIGENTES Y REPRESENTACIÓN DEL CONOCIMIENTO

PRÁCTICA 1

ÁLVARO CARRIZOSA PEÑA

Dirigido por:

Christian Vladimir Sucuzhanay Arévalo

Ejercicio 1: Predicción fraude en tarjetas de crédito

Introducción

La detección de fraude en transacciones con tarjetas de crédito es un desafío constante para las instituciones financieras. El uso de técnicas de machine learning puede ser fundamental para identificar patrones y anomalías en los datos que podrían indicar actividades fraudulentas.

La detección de fraude en transacciones con tarjetas de crédito representa uno de los desafíos más significativos en el sector financiero actual. Con el avance de la tecnología y el aumento en el uso de transacciones en línea, las técnicas de fraude se han vuelto cada vez más sofisticadas y difíciles de detectar. Los defraudadores constantemente adaptan sus métodos, empleando estrategias complejas que pueden pasar desapercibidas por los sistemas de seguridad tradicionales.

En este escenario, el machine learning emerge como una herramienta esencial para combatir el fraude. A través de la aplicación de algoritmos avanzados, es posible analizar grandes volúmenes de transacciones y detectar patrones que indican actividad fraudulenta. Los modelos de machine learning son capaces de aprender de los datos históricos y adaptarse a nuevos comportamientos fraudulentos, lo que los hace particularmente efectivos en la identificación de transacciones sospechosas.

Además, el machine learning no solo se limita a la detección de fraudes conocidos; su capacidad para explorar y aprender de los datos permite descubrir tácticas de fraude previamente desconocidas. Esto es crucial en un campo donde los defraudadores están en constante evolución, siempre buscando nuevas formas de eludir los sistemas de seguridad.

En este ejercicio, utilizaremos técnicas de machine learning para abordar el problema del fraude en tarjetas de crédito. Nuestro enfoque se centrará en el análisis exhaustivo del dataset proporcionado, empleando herramientas como RapidMiner, BigQuery y Cloud, para desarrollar un modelo predictivo robusto y confiable. El objetivo es crear un sistema que no solo identifique las transacciones fraudulentas con alta precisión, sino que también se adapte a las nuevas tendencias en métodos de fraude.

Objetivo

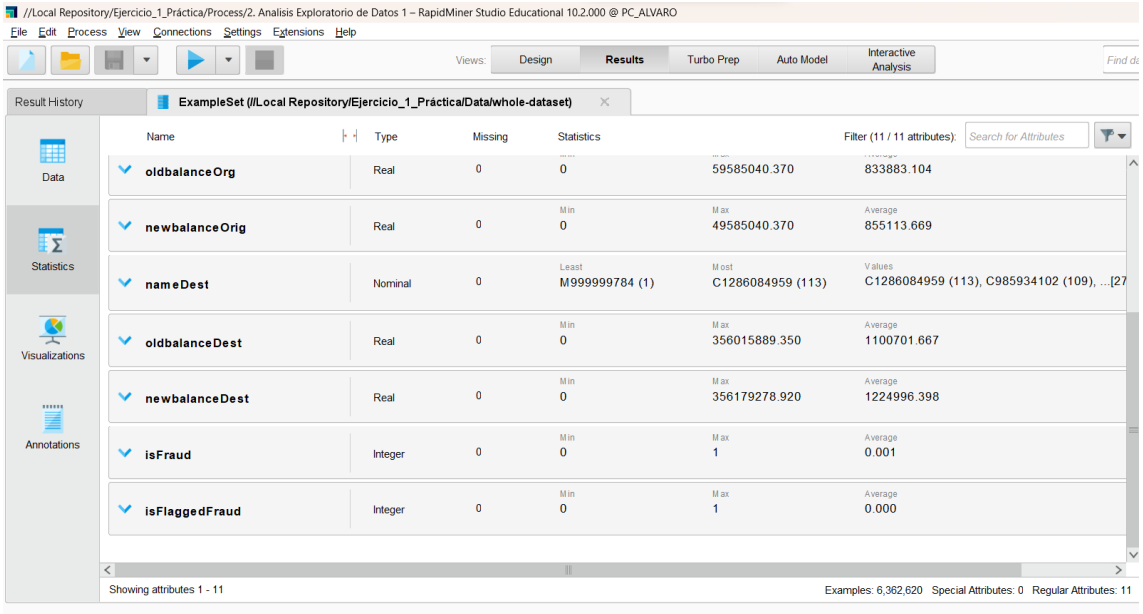
Este informe describe los pasos esenciales, que realizaremos sobre el dataset (ver punto A) para predecir el fraude en transacciones con tarjetas de crédito utilizando técnicas de machine learning con diferentes herramientas, como pueden ser :

1. RapidMiner
2. BigQuery
3. Cloud

A. Adquisición de Datos

Descripción del dataset

Nombre: fraud_cards.csv



oldbalanceOrg: Tipo Real. Este podría ser el saldo inicial en la cuenta del emisor antes de la transacción. No hay valores faltantes.

newbalanceOrig: Tipo Real. Este sería el saldo en la cuenta del emisor después de la transacción. No hay valores faltantes.

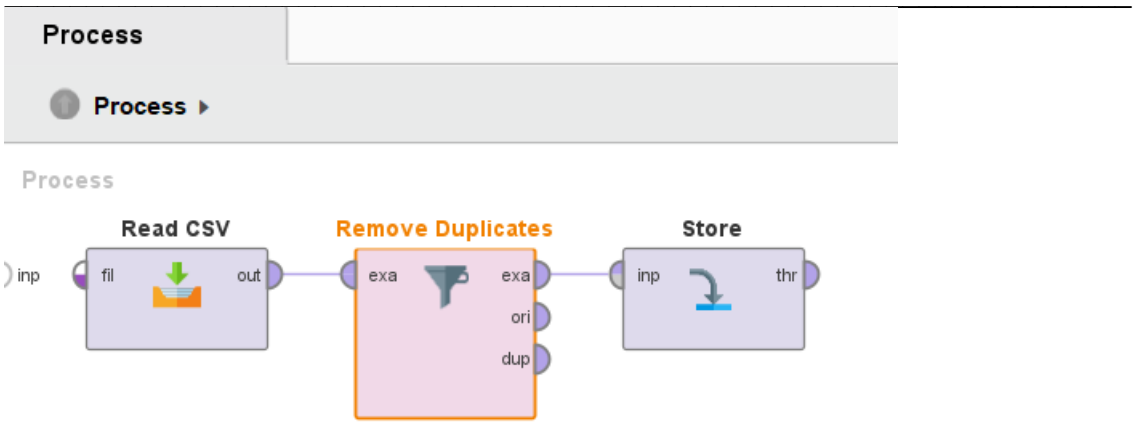
oldbalanceDest: Tipo Real. El saldo inicial en la cuenta del destinatario antes de la transacción. No hay valores faltantes.

newbalanceDest: Tipo Real. El saldo en la cuenta del destinatario después de la transacción. No hay valores faltantes.

step: Tipo Entero. Podría representar el tiempo en el que se registra la transacción, como un paso de tiempo en horas o días desde el inicio del conjunto de datos. No hay valores faltantes.

isFlaggedFraud: Tipo Entero. Es una variable que indica si la transacción ha sido marcada como fraudulenta por algún sistema o regla automática. No hay valores faltantes.

isFraud: Tipo Nominal. La etiqueta o clase objetivo que indica si una transacción es fraudulenta (1) o no (0). No hay valores faltantes.



Este es el proceso que he hecho en Rapid Miner para el apartado 1, donde primero hemos leído el dataset de "fraud_cards", hemos borrado los duplicados y hemos almacenado el resultado en un "Store", que a su vez lo guarda en la carpeta "Models" bajo el nombre "whole_dataset".

B. Análisis Exploratorio de Datos (EDA)

El Análisis Exploratorio de Datos (EDA) es un paso crucial en el proceso de modelado predictivo, que nos permite comprender mejor la naturaleza y las características de los datos con los que estamos trabajando. En el contexto de la detección de fraude con tarjetas de crédito, el EDA se enfoca en identificar patrones, anomalías, tendencias y relaciones en los datos que podrían ser indicativos de comportamiento fraudulento.

Al eliminar datos duplicados, nos aseguramos de que nuestro análisis no esté sesgado por información redundante que podría distorsionar las estadísticas del modelo. Esto es especialmente importante en los conjuntos de datos financieros, donde las transacciones duplicadas pueden ser un indicador de fraude o un error en la recopilación de datos.

El manejo de valores faltantes es esencial para preparar el conjunto de datos para el modelado. Dependiendo del contexto, los valores faltantes pueden imputarse, eliminarse o utilizarse para crear nuevas características que podrían sugerir un patrón de comportamiento, como la no inclusión de cierta información en transacciones fraudulentas.

La normalización de las variables cuantitativas, como el monto de la transacción y los saldos de las cuentas, es importante para garantizar que el modelo no esté injustamente influenciado por la escala de los datos. Esto permite comparar directamente los coeficientes de las características y determinar su importancia relativa.

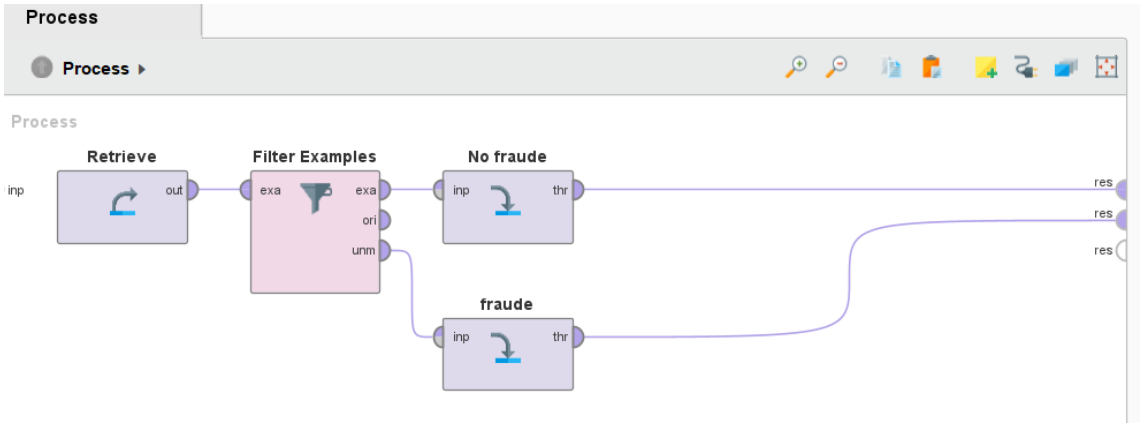
La codificación de variables categóricas, como el tipo de transacción o el identificador del cliente, permite que los algoritmos de aprendizaje automático interpreten y utilicen esta información para la predicción. Métodos como la codificación one-hot-encoding o la codificación de etiquetas se utilizan comúnmente, aunque deben elegirse cuidadosamente para evitar la introducción de sesgos o relaciones artificiales.

La exploración de características implica estadísticas descriptivas, como medias y desviaciones estándar, así como la búsqueda de correlaciones entre variables. En nuestro caso, podríamos estar interesados en cómo la cantidad de la transacción se correlaciona con la probabilidad de ser fraudulenta o cómo los patrones de balance antes y después de la

transacción se relacionan con las actividades fraudulentas.

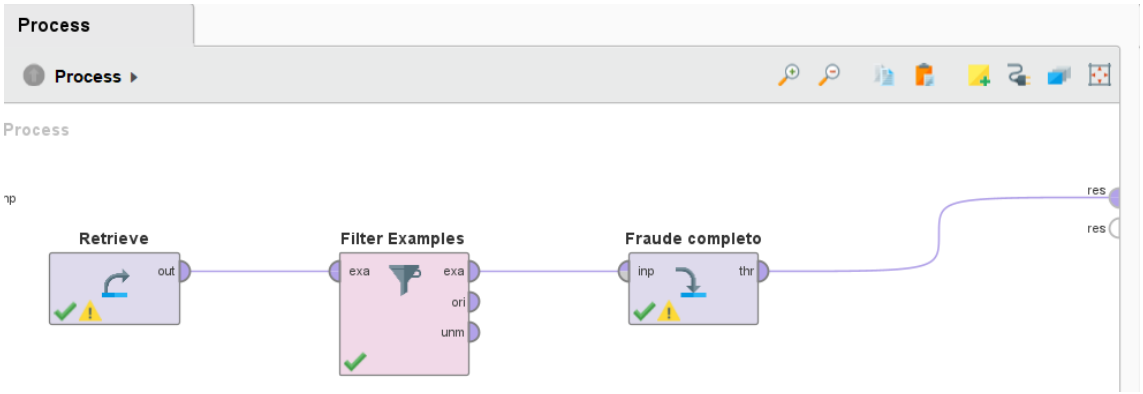
Finalmente, la visualización de datos es una herramienta poderosa para el EDA, ya que permite descubrir visualmente patrones y anomalías que pueden no ser evidentes en una revisión estadística. Histogramas, gráficos de cajas, mapas de calor y gráficos de dispersión son todos útiles para visualizar la distribución de las transacciones normales versus fraudulentas y pueden revelar información crítica sobre las características del fraude que se puede aprovechar para mejorar la precisión del modelo predictivo.

B.1



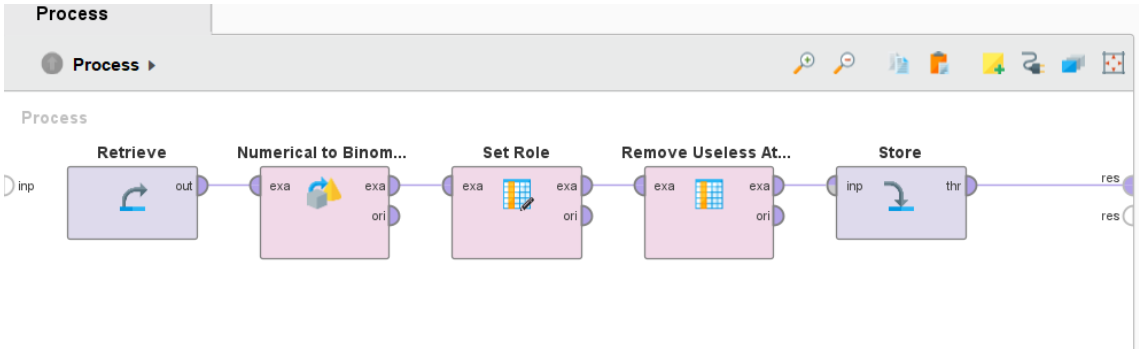
Whole Dataset: Este es probablemente el punto de partida, donde se carga el conjunto de datos completo.
Fraud y No Fraud: Aquí, el conjunto de datos se divide en dos partes, una conteniendo solo datos de transacciones fraudulentas y la otra conteniendo transacciones no fraudulentas.

B.2



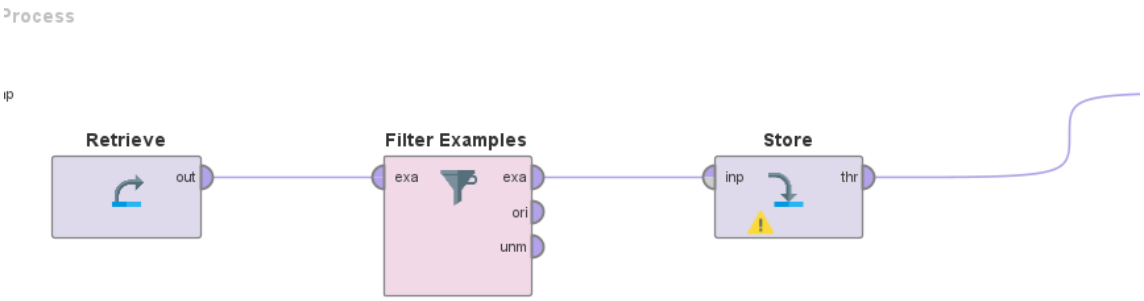
En este apartado guardamos todos los datos de fraude completo, tanto el no fraude como el fraude.

B.3

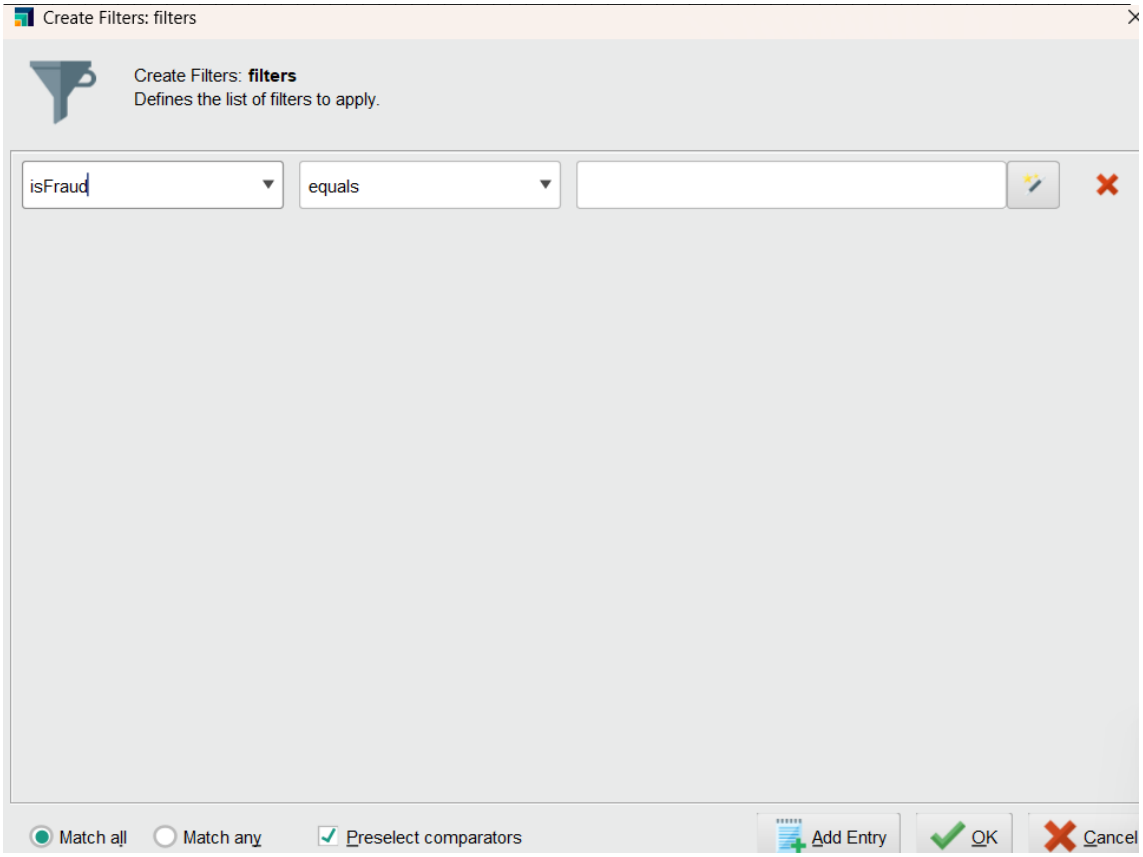


Con el operador **"Numerical to binomial"**convertimos atributos numéricos a binomiales. Esto se hace a menudo cuando los números representan categorías o cuando solo se están considerando dos estados.
Set role: Ponemos dentro de este operador "Fraud" y "Type". Los roles nos ayudarán con la variable objetivo o la etiqueta para tareas de modelado predictivo.
Además, eliminamos los atributos que creamos innecesarios y guardamos todo este proceso en un nuevo modelo de "Data".

B.4



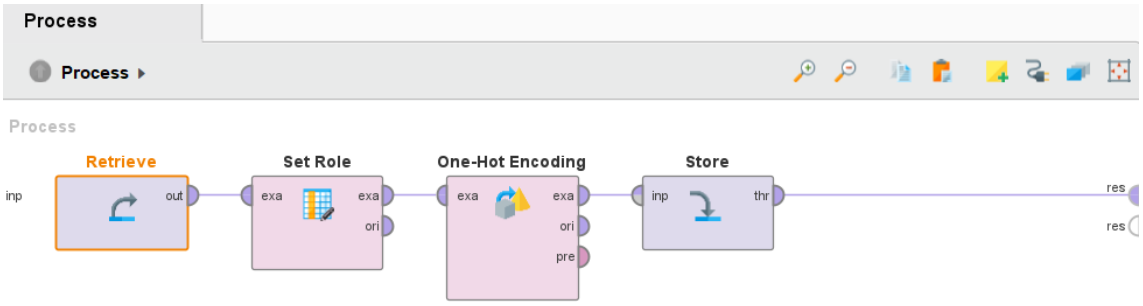
Cogemos de la carpeta "Data" el modelo de datos creado en el apartado anterior con el retrieve, le aplicamos un filter examples con estas características:



Y guardamos el resultado en la carpeta “Data” bajo el nombre de “sampled data”

C. Ingeniería de Características

Selección de características: Identificar las características más relevantes para el modelo predictivo.
Creación de nuevas características: Derivar características adicionales que puedan mejorar la capacidad predictiva del modelo

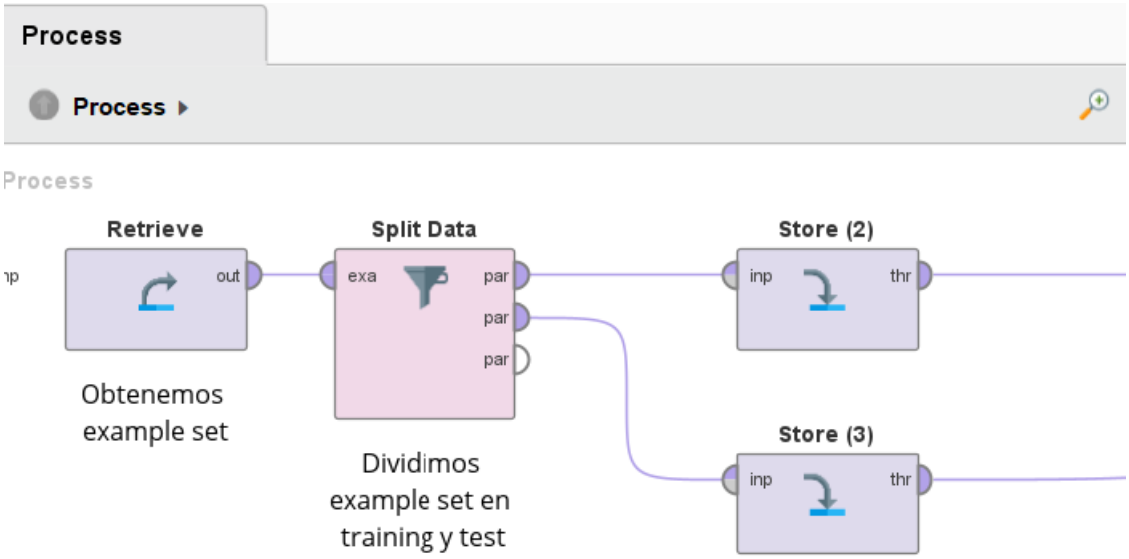


En este apartado, gracias al One-Hot Encoding, transformamos variables categóricas en una forma que pueda ser proporcionada a algoritmos de machine learning que requieren

entradas numéricas. Convierte cada categoría en una nueva columna binaria (1 o 0) para cada posible valor de la variable original.

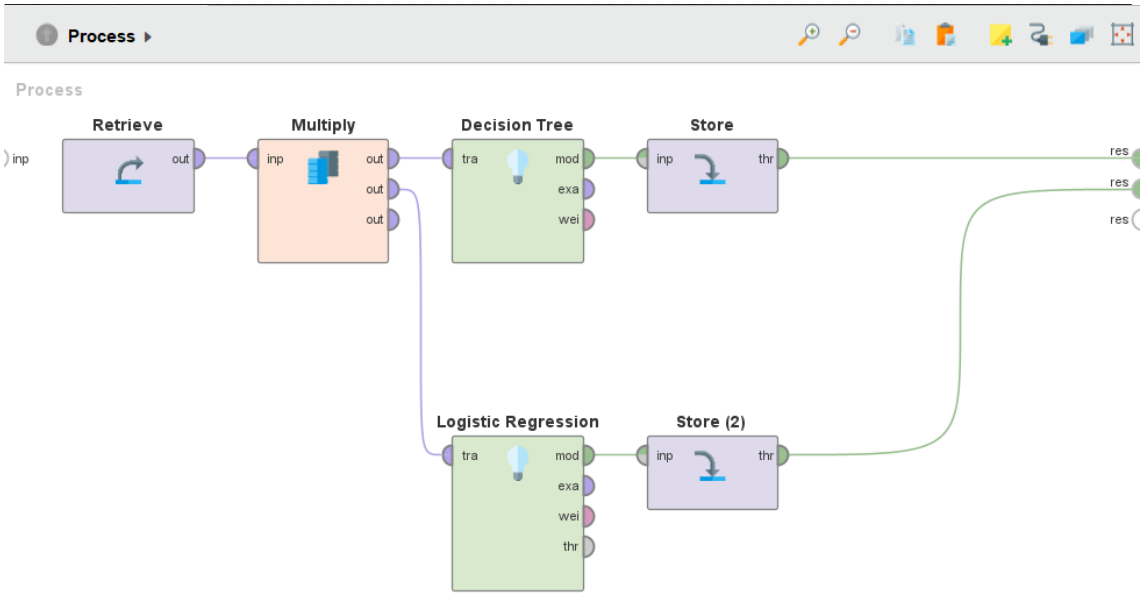
D. Preparación de Datos para Modelado

División de datos: Separar los datos en conjuntos de entrenamiento, validación y prueba.
Balanceo de datos: Si es necesario, aplicar técnicas de remuestreo para equilibrar clases (oversampling, undersampling, SMOTE, etc.).
Realizamos el under sampling.



E. Selección y Entrenamiento del Modelo

Elección del algoritmo: Seleccionar modelos de machine learning adecuados para la predicción de fraude (Random Forest, Support Vector Machines, Redes Neuronales, etc.).
Entrenamiento del modelo: Utilizar el conjunto de entrenamiento para entrenar el modelo seleccionado.

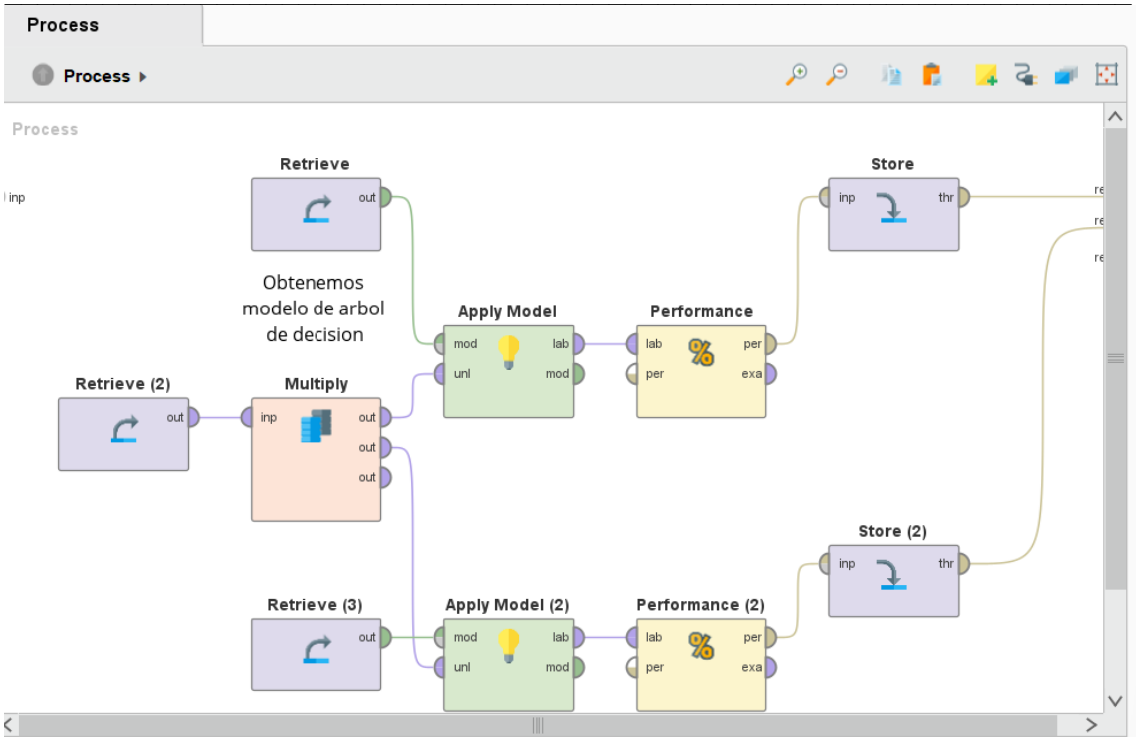


Decision Tree: Entrena un modelo de árbol de decisión utilizando el conjunto de datos. El árbol de decisión es un algoritmo de machine learning que divide los datos en subconjuntos basados en el valor de las características, lo que puede ser visualizado como un árbol.

Logistic Regression: Entrena un modelo de regresión logística en paralelo con el árbol de decisión. La regresión logística es otro algoritmo de machine learning, utilizado comúnmente para la clasificación binaria.

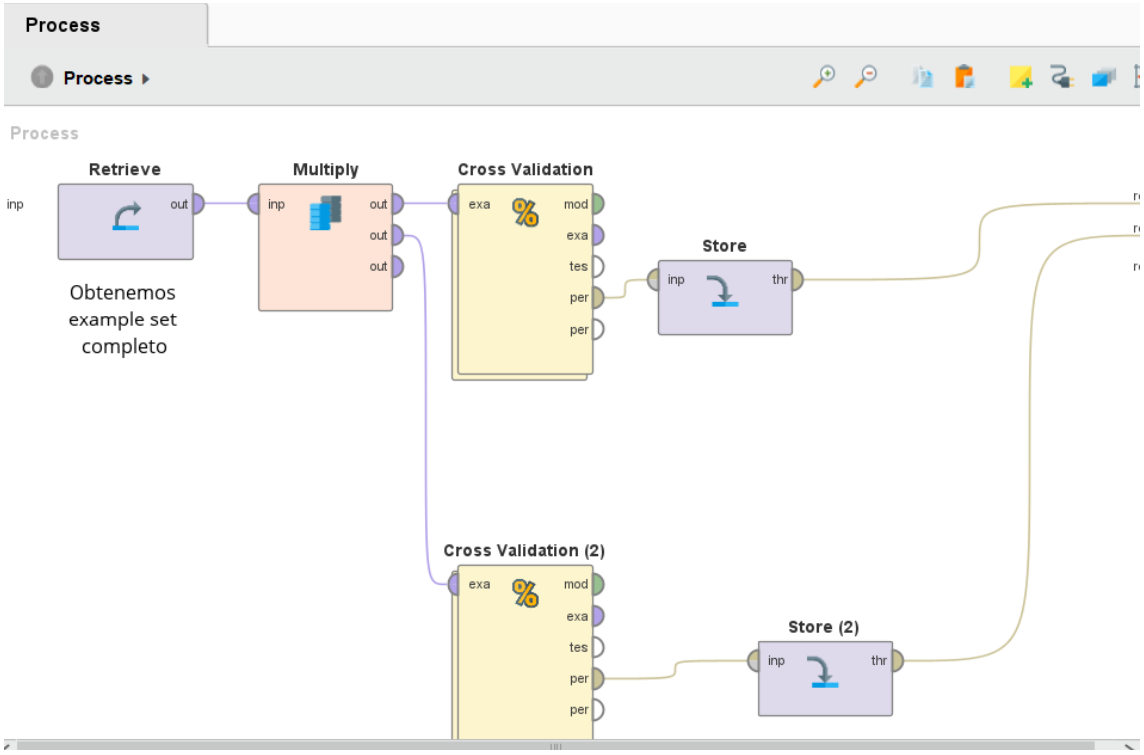
F. Evaluación del Modelo

Validación del modelo: Evaluar el rendimiento del modelo utilizando métricas como precisión, exhaustividad, F1-score, matriz de confusión, ROC-AUC, entre otras.
Ajuste de hiperparámetros: Optimizar los hiperparámetros del modelo para mejorar su desempeño.



G. Validación y Optimización del Modelo

Validación cruzada: Verificar la generalización del modelo utilizando técnicas de validación cruzada.
Optimización adicional: Realizar ajustes adicionales en el modelo para mejorar su capacidad predictiva.



Conclusiones

La predicción de fraude en tarjetas de crédito mediante técnicas de machine learning es fundamental para mitigar riesgos financieros. Los pasos mencionados constituyen un marco sólido para desarrollar un sistema efectivo de detección de fraudes.

Consideraciones Finales

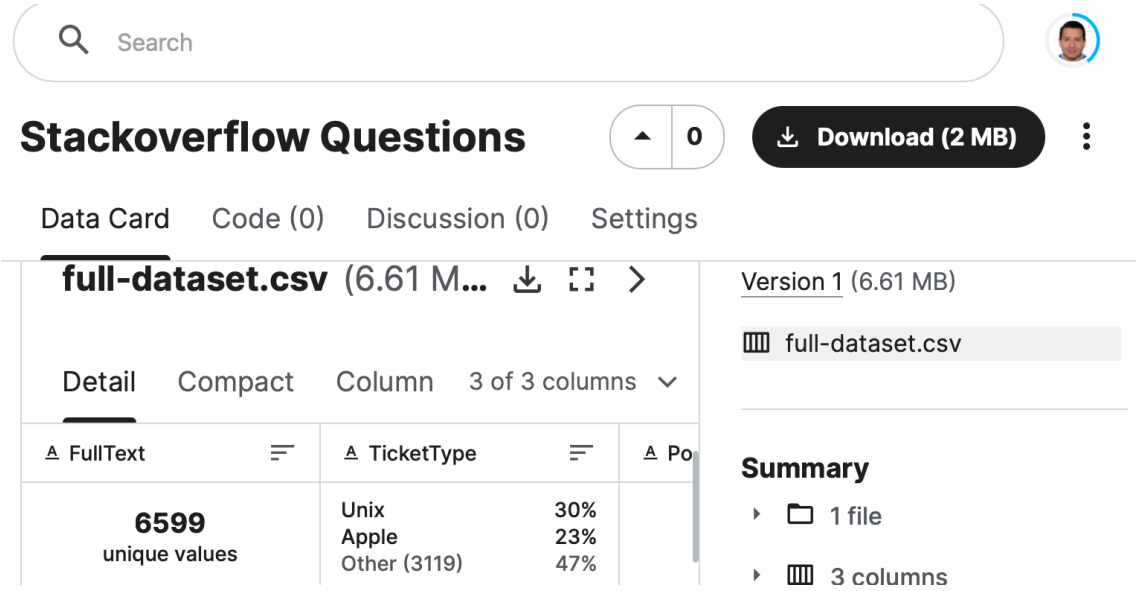
La actualización constante del modelo es crucial para adaptarse a nuevos patrones de fraude.
La colaboración con expertos en seguridad financiera es esencial para mejorar la precisión y eficacia del modelo.

Ejercicio 2

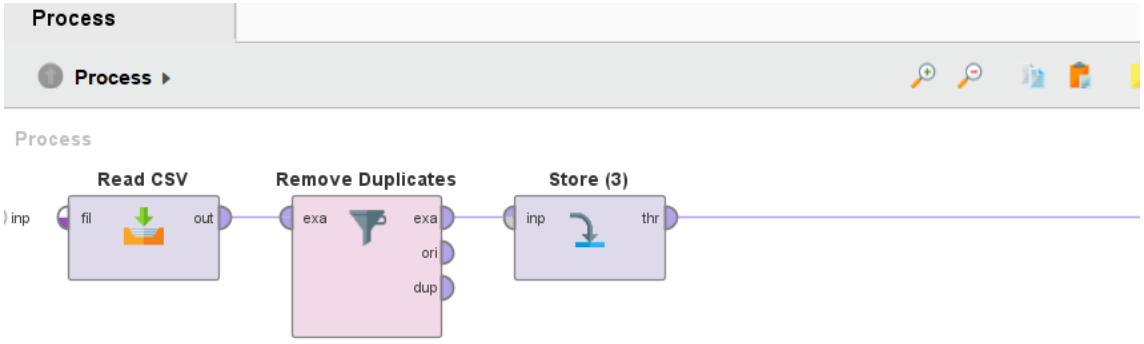
Clasificación de documentos

Introducción

Usaremos un dataset de Stackoverflow, donde hay diferentes categorías, las mismas albergan preguntas de usuarios sobre diferentes temas.

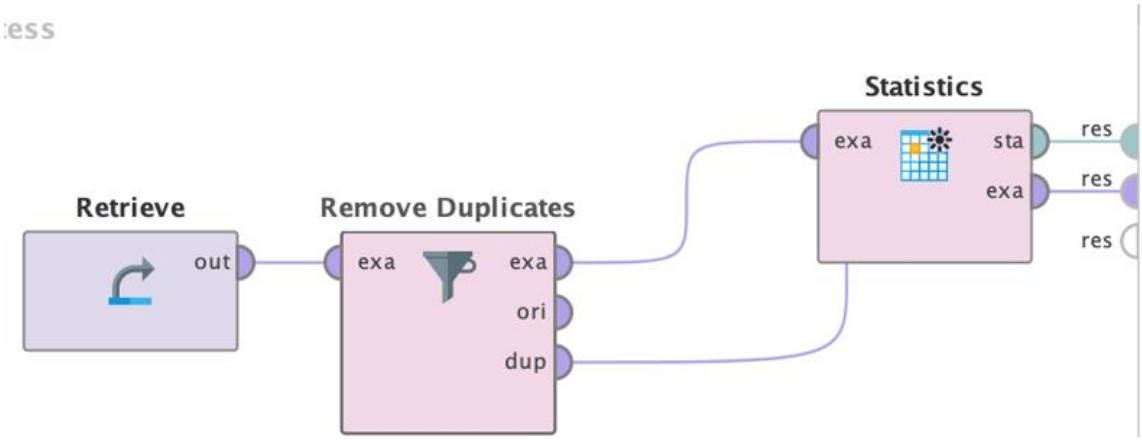


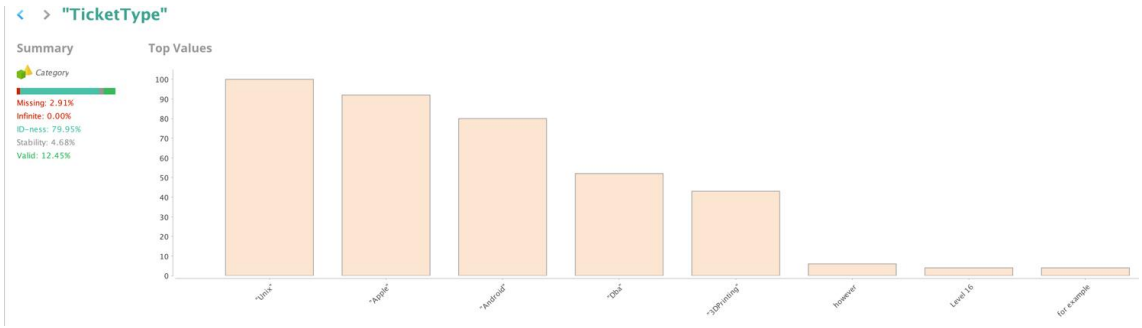
Leemos el dataset en Rapid Miner y este es el resultado:



A. Análisis Exploratorio de Datos (EDA)

Exploración de características: Analizar la distribución de variables, identificar correlaciones y buscar posibles patrones o anomalías en los datos.
Visualización de datos: Utilizar gráficos y visualizaciones para entender el dataset.

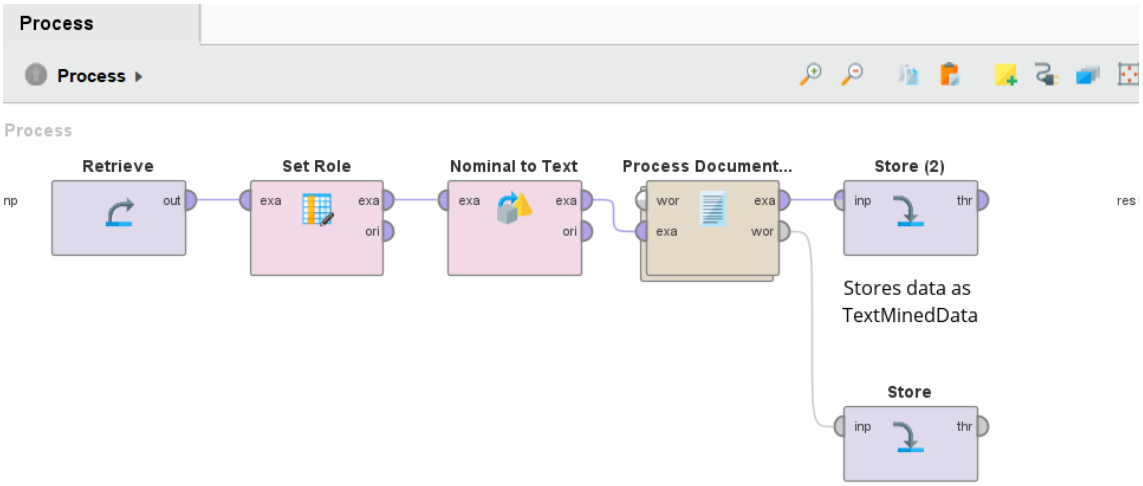




B. Ingeniería de Características

Selección de características: Identificar las características más relevantes para el modelo de clasificación
Creación de nuevas características: Derivar características adicionales que puedan mejorar la capacidad del modelo.
Como se puede apreciar, solo tenemos tres atributos que contienen texto por lo tanto no puedo sacar mayor provecho a las características con el datase actual, por lo tanto paso a la

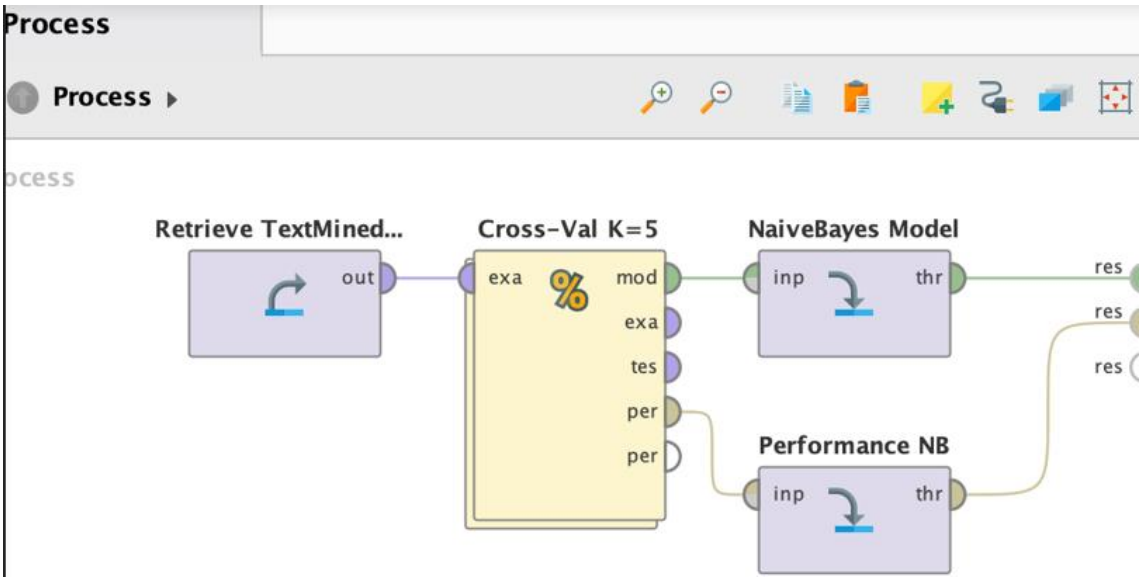
preparación de Dato para el modelo.



C. Preparación de Datos para Modelado y, E. Selección y Entrenamiento del Modelo

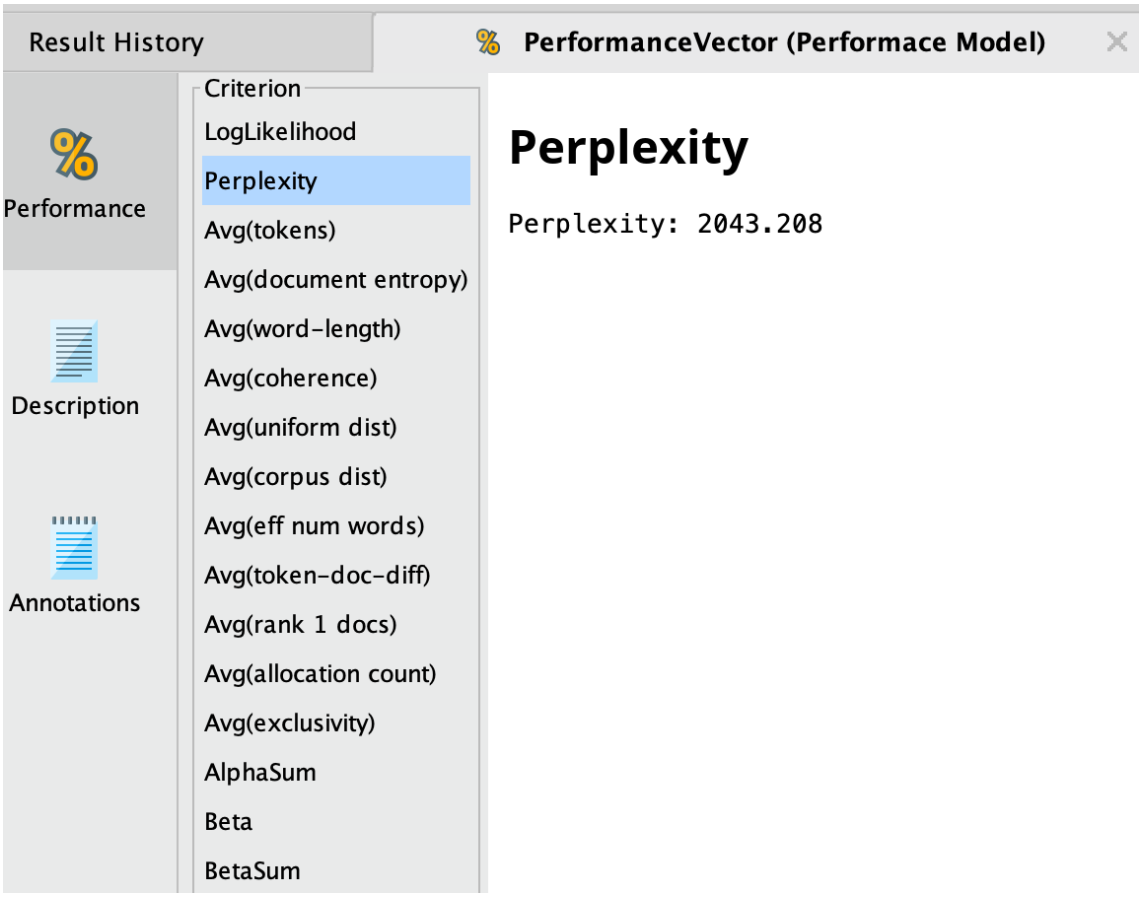
El tratamiento habitual del texto, igual a como hemos realizado en todas las clases, además de la división de datos: Separar los datos en conjuntos de entrenamiento, validación y prueba.

En este caso, en particular, conforme podemos observar que el texto ya está preprocesado. Por lo que puedo directamente pasar a la división para entrenar y textear en el modelo que en este caso utilizo NaiveBayes.
Utilizamos Cross-Validation con N-fold = 5 (por temas de velocidad, aunque lo último es 10) construyo el modelo y mido su rendimiento.
Ponemos cualquier modificación siempre dentro del operador Cross Validation no fuera de él.



D. Evaluación del Modelo

Validación del modelo: Evaluar el rendimiento del modelo utilizando métricas como precisión, exhaustividad, F1-score, matriz de confusión, ROC-AUC, entre otras.



Ajuste de hiperparámetros: Optimizar los hiperparámetros del modelo para mejorar su desempeño.

1. Perplejidad (Perplexity): Es una medida común para evaluar la calidad de un modelo de LDA. Se calcula utilizando la distribución de palabras en los documentos y mide qué tan bien predice el modelo un conjunto de datos. Un menor valor de perplejidad indica un mejor rendimiento del modelo.
2. Coherencia de los tópicos (Topic Coherence): Evalúa la interpretabilidad de los tópicos generados por el modelo LDA. Una puntuación alta de coherencia indica que los tópicos son más interpretables y representan mejor las relaciones entre las palabras.

3. Puntuación de similitud de documentos (Document Similarity Score): Mide cuán bien el modelo puede identificar la similitud entre diferentes documentos basados en la distribución de tópicos.
4. Precisión en tareas específicas: Si el modelo LDA se utiliza para clasificar documentos en categorías específicas, la precisión, recall o F1-score pueden ser métricas útiles para evaluar su rendimiento en estas tareas de clasificación.

5. Tiempo de entrenamiento y predicción: Evaluar el tiempo que lleva entrenar el modelo y realizar predicciones puede ser importante en aplicaciones en tiempo real o cuando se trabaja con grandes conjuntos de datos.

Estas métricas pueden variar según el contexto y el uso específico de LDA. Es común utilizar una combinación de estas métricas para tener una imagen más completa del rendimiento del modelo.

E. Validación y Optimización del Modelo

Validación cruzada: Verificar la generalización del modelo utilizando técnicas de validación cruzada.
Optimización adicional: Realizar ajustes adicionales en el modelo para mejorar su capacidad predictiva.

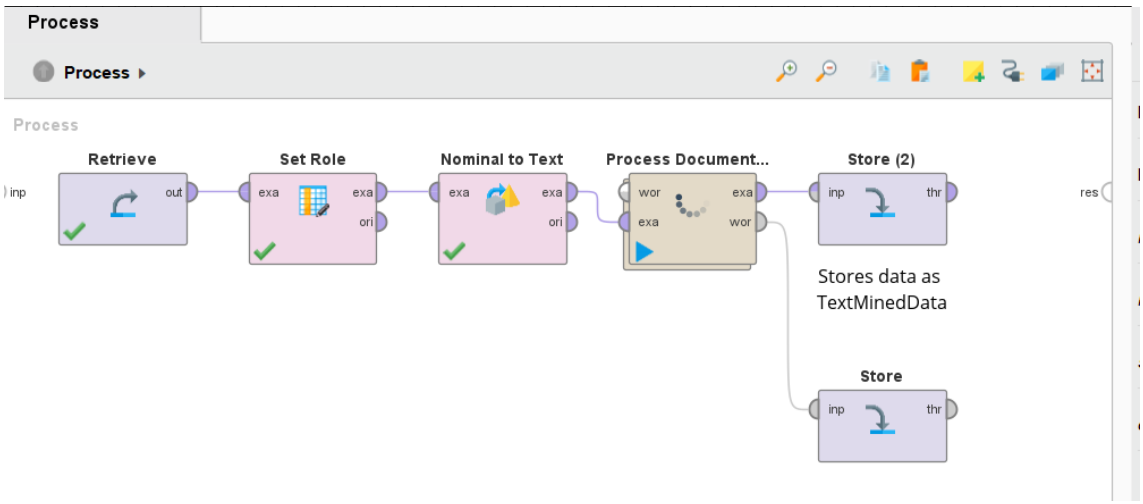
Dentro de la carpeta 'Process' se añaden los procesos con los que manipulamos la database. Hemos creado tres procesos principales.

1. TextMining

Retrieve Question-> con este operador cargamos los datos. En este caso guardamos las preguntas de stackoverflow, para posteriormente analizarlas.
Write CSV-> sirve para guardar los datos en un archivo .csv
Process Documents from Data->

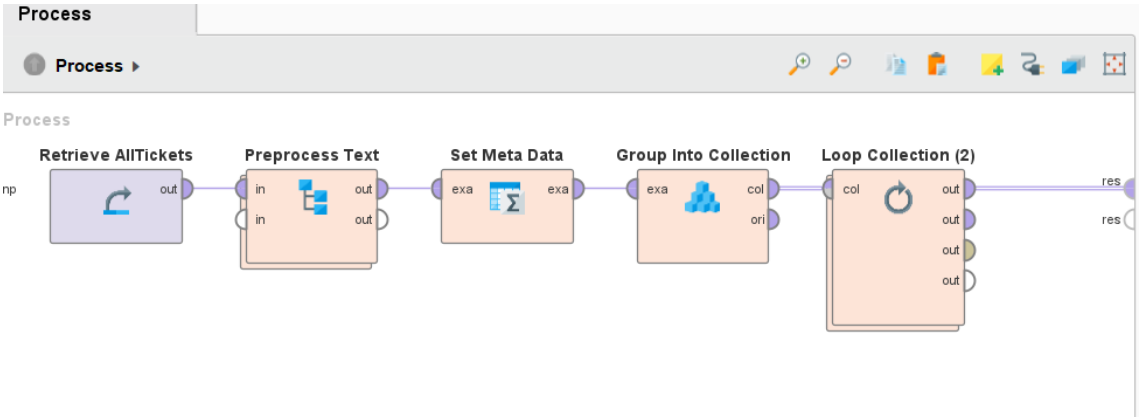
- Tokenize- divide el texto en unidades más pequeñas
- Transform Cases- sirve para transformar todo el texto a minusculas
- Filter StopWords- se utiliza para eliminar palabras comunes pero poco informativas de un conjunto de datos de texto
- Filter Tokens(by length)- este operador se utiliza para eliminar tokens que son demasiado cortos o demasiado largos dependiendo de las necesidades que tengamos en el analisis del texto
- Generet n-grams (Terms)- se utiliza para crear n-gramas a partir de un conjunto de datos de texto. Se utiliza para analizar un texto y capturar patrones de palabras y expresiones que pueden tener información importante.

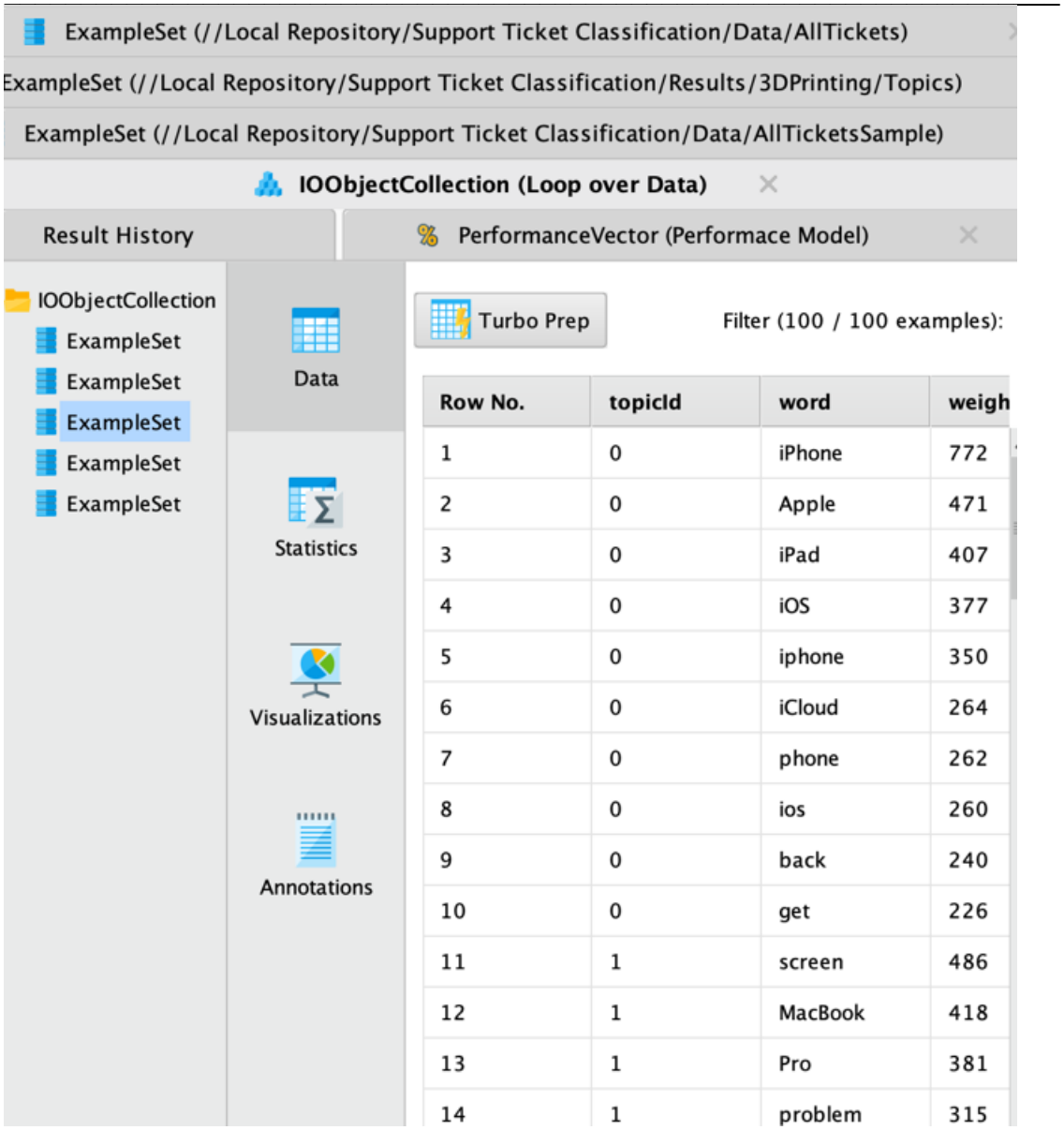
Store-> se utiliza para guardar el proceso en un sitio especifico



2. TopicModel

- Retrieve-> se utiliza para cargar datos de diferentes fuentes
- Preprocess Text-> se utiliza para realizar una serie de tareas de procesamiento de datos de texto antes de realizar el resto de tareas
- Set Meta Data-> se utiliza para definir o modificar los metadatos asociados con los ejemplos en un conjunto de datos.
- Group intro Collection-> agrupa los datos por un tipo de ticket para poder construir un modelo para cada tipo.



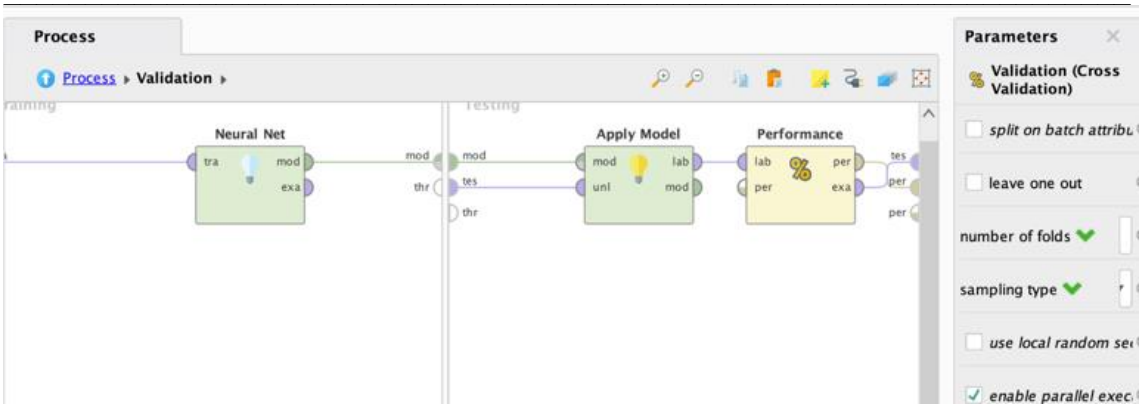


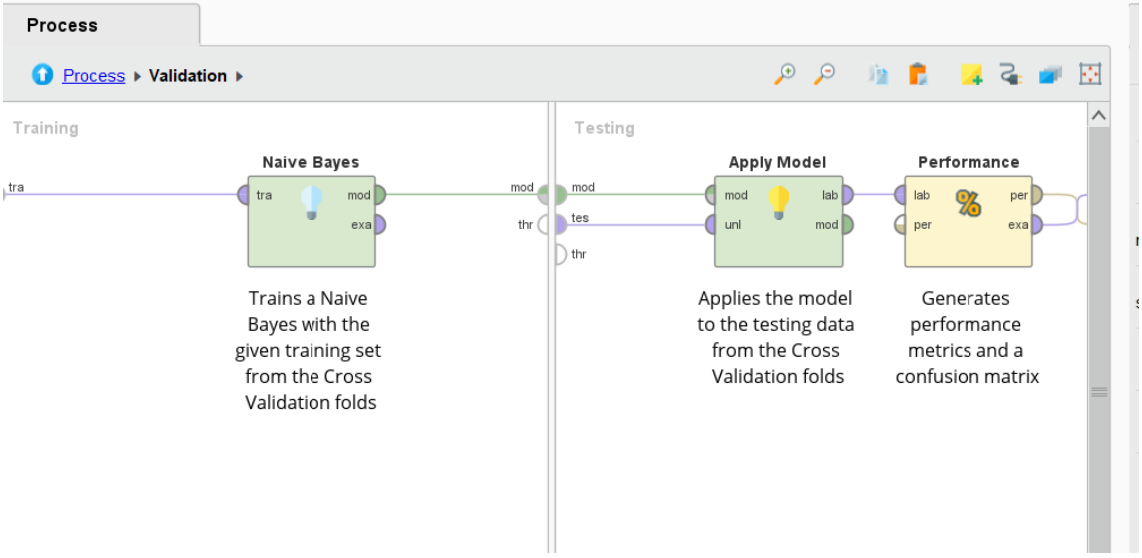
Row No.	topicId	word	weigh
1	0	iPhone	772
2	0	Apple	471
3	0	iPad	407
4	0	iOS	377
5	0	iphone	350
6	0	iCloud	264
7	0	phone	262
8	0	ios	260
9	0	back	240
10	0	get	226
11	1	screen	486
12	1	MacBook	418
13	1	Pro	381
14	1	problem	315

El objetivoprincipal es proporcionar una herramienta que automatice la clasificación de las preguntas de los ususarios de StackOverflow.
Debeis alcanzar minimo 93%.

Modelado

Necesitamos construir y validar una clasificación multinivel para n-clases. La idea básica aquí es que cada clase debe tener un conjunto único de palabras clave asociadas con ese tipo de clase. Por ej. Los términos utilizados para los dispositivos Apple deberían ser diferentes para los dispositivos Android y los algoritmos de aprendizaje automático deberán poder diferenciarlos.





Probamos este método.

Performance análisis

Con el Naive Bayes, nos salen los siguientes resultados:

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%	

accuracy: 86.71% +/- 0.87% (micro average: 86.71%)

	true 3DPrinting	true Android	true Apple	true Dba	true Unix	class precision
pred. 3DPrinting	1184	7	12	2	26	96.18%
pred. Android	12	1578	196	1	147	81.59%
pred. Apple	19	147	2076	13	242	83.14%
pred. Dba	9	6	29	2017	202	89.13%
pred. Unix	26	62	187	117	2683	87.25%
class recall	94.72%	87.67%	83.04%	93.81%	81.30%	

Modelo de temas

Conseguido lo anterior, debemos discernir cuál es el problema en la pregunta para poder darle una solución. Actualmente, solo tenemos las publicaciones en sí y no tenemos categorías para estas publicaciones. Necesitamos crear una lista de temas para cada pregunta, para que los equipos puedan asignar recursos, personal, técnicos de manera efectiva para resolverlas