UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN INGENIERÍA INFORMATICA

SISTEMAS INTELIGENTES Y REPRESENTACIÓN DEL CONOCIMIENTO

PRACTICA 01

JOSÉ RODRIGO LÓPEZ FLORES

CURSO 2023-2024

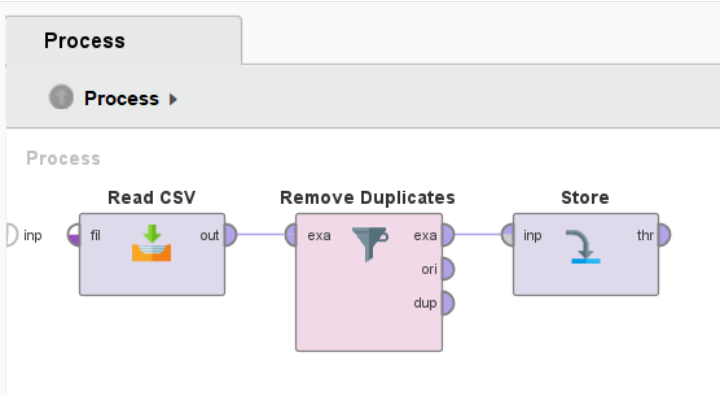
Índice

Capítulo 1. Ejercicio 1

1.1 Adquisición de Datos

Enlace al repositorio: https://github.com/JoseR200/ue22314990-Practica1

En esta parte del proceso se adquirieron los datos del repositorio que compartió el profesor. Siendo el archivo full-dataset.csv. Se removieron los duplicados para tener un dataser limpio y se guardo con el nombre de whole-dataset.



Se muestra la parte de estadísticas para verificar el tipo de dato y si existen variables faltantes.

Name	Type	Missing	Statistics		
step	Integer	0	Min 1	Max 743	Average 243.397
type	Nominal	0	Least DEBIT (41432)	Most CASH_OUT (2237500)	Values CASH_OUT (2237500), PAYMENT (2151495), ...[3 more]
amount	Real	0	Min 0	Max 92445516.640	Average 179861.904
nameOrig	Nominal	0	Least C999999784 (1)	Most C1065307291 (3)	Values C1065307291 (3), C1462946854 (3), ...[6353305 more]
oldbalanceOrg	Real	0	Min 0	Max 59585040.370	Average 833883.104
newbalanceOrig	Real	0	Min 0	Max 49585040.370	Average 855113.669
nameDest	Nominal	0	Least M999999784 (1)	Most C1286084959 (113)	Values C1286084959 (113), C985934102 (109), ...[2722360 more]

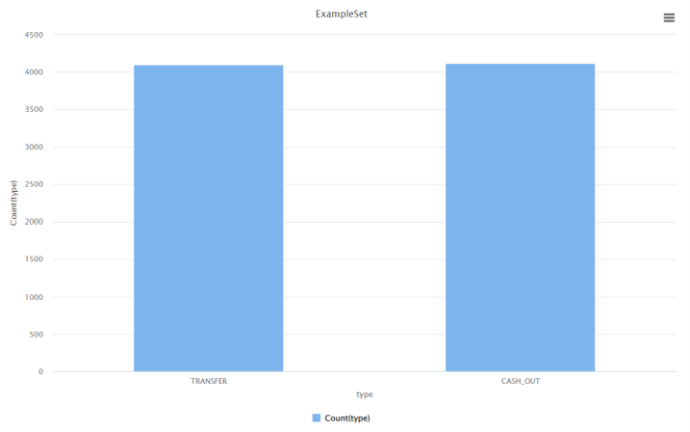
Podemos verificar que para el atributo type podemos realizar la normalización de variables categóricas.

1.2 Analisis Exploratorio de Datos

Enlace al repositorio: https://github.com/JoseR200/ue22314990-Practica1

La variable sobre la que vamos a trabajar va a ser la de si se ha cometido fraude bancario o no. Para ello hemos dividido el data set completo en base a dicho parámetro.

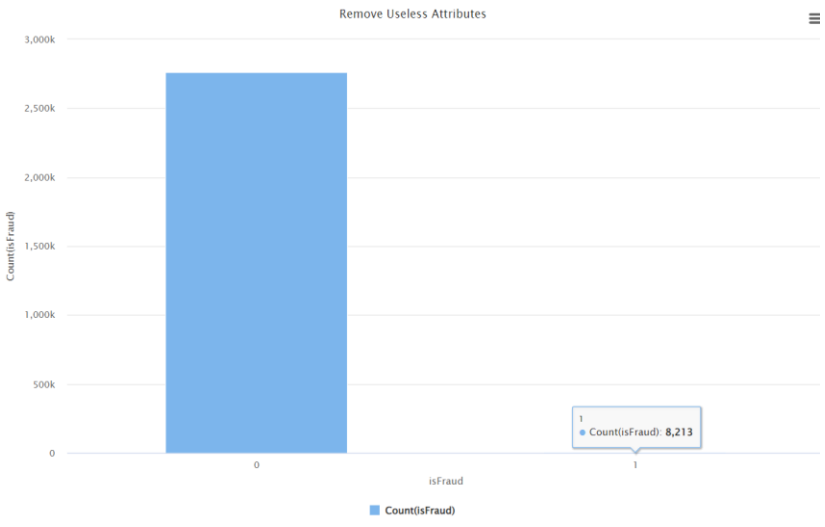
Viendo las estadísticas del dataset que solo contiene los fraudes, podemos verificar que solo hay fraudes en 2 de los 5 tipos que existen: En transferencia y cash-out.



Por lo tanto cuando trabajemos con el data set. Hay que dividirlo para solo trabajar con los tipos transfer y cash-out. Teniendo la data solo con lo que nos importa, vamos a retirarle las columnas que consideremos correlacionadas y solo hace que el data set sea más pesado.

Name	Type	Missing	Statistics		Filter (9 / 9 attributes): Search for Attributes
Label **isFraud**	Integer	0	Min 0	Max 1	Average 0.003
Metadata **type**	Polynominal	0	Least PAYMENT (0)	Most CASH_OUT (2237500)	Values CASH_OUT (2237500), TRANSFER (532909), ...[3 more]
step	Integer	0	Min 1	Max 743	Average 242.008
amount	Real	0	Min 0	Max 92445516.640	Average 317536.141
oldbalanceOrg	Real	0	Min 0	Max 59585040.370	Average 47643.079
newbalanceOrig	Real	0	Min 0	Max 49585040.370	Average 16091.905
oldbalanceDest	Real	0	Min 0	Max 356015889.350	Average 1703551.162
newbalanceDest	Real	0	Min 0	Max 356179278.920	Average 2049734.437
isFlaggedFraud	Integer	0	Min 0	Max 1	Average 0.000

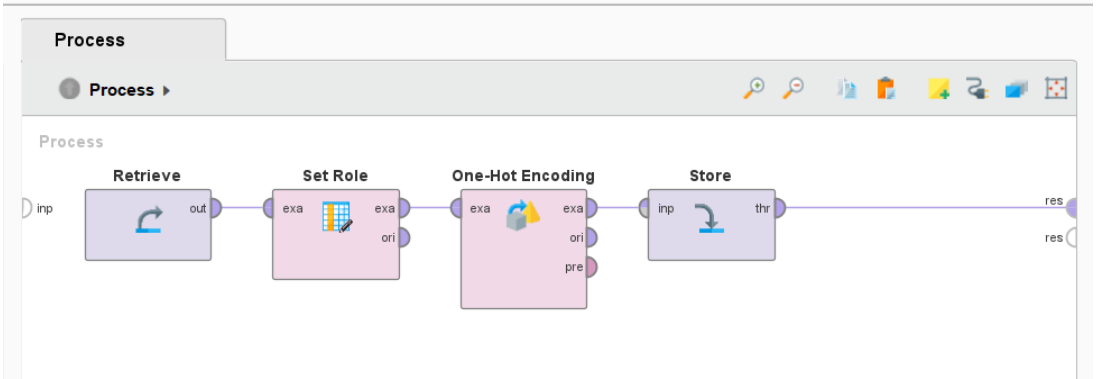
Finalmente vamos a realizar el sampling en base a igualar todo a isFraud false de 8213



1.3 Ingenieria de Caracteristicas

Enlace al repositorio: https://github.com/JoseR200/ue22314990-Practica1

Realizamos el one hot encoding para la variable type

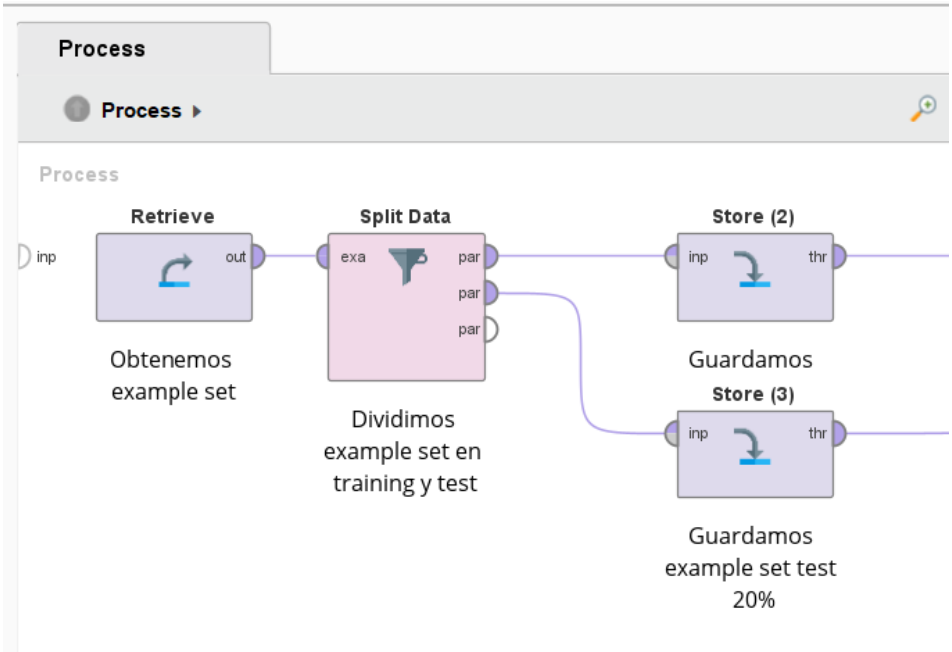


Row No.	isFraud	type = PAYM...	type = CASH...	type = DEBIT	type = CASH...
1	true	0	0	0	0
2	true	0	1	0	0

1.4 Preparacion de Datos para Modelo

Enlace al repositorio: https://github.com/JoseR200/ue22314990-Practica1

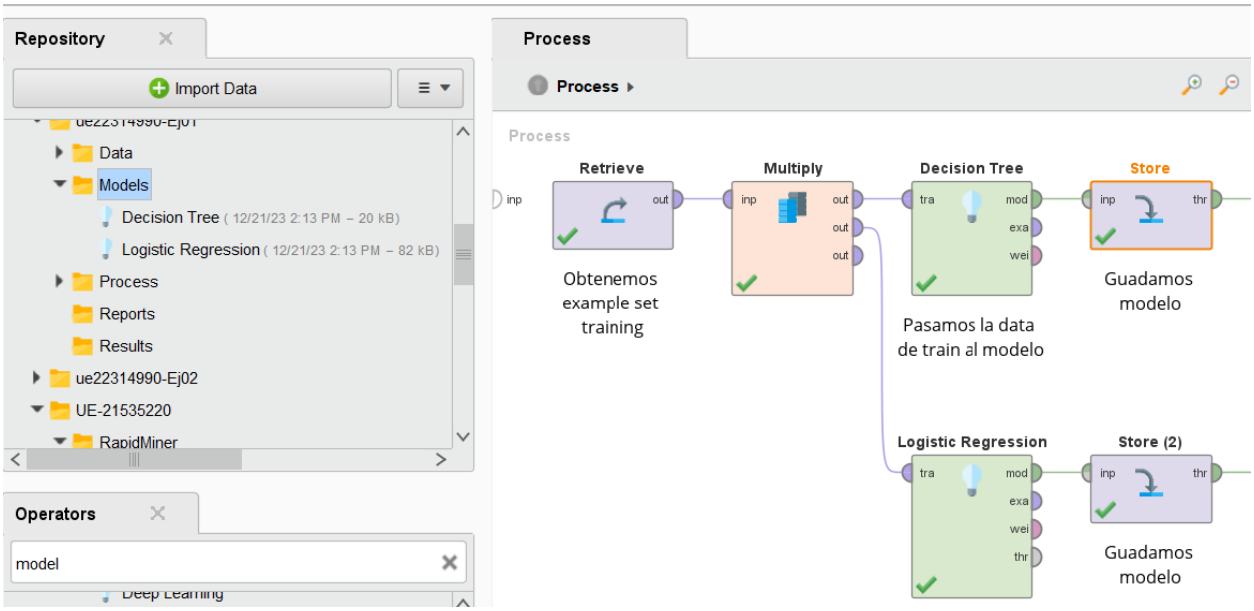
En este paso vamos a separar los 2 grupos de validación y testeo, siendo el primero un 80% y el segundo un 20%.



1.5 Selección y Entrenamiendo de Modelo

Enlace al repositorio: https://github.com/JoseR200/ue22314990-Practica1

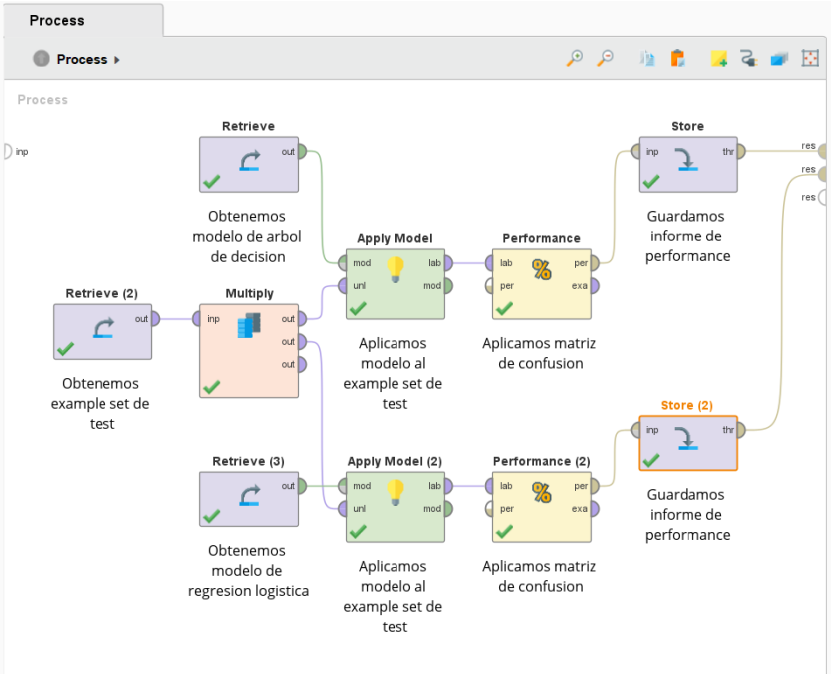
Procedemos a realizar las pruebas y la generación de modelos para regresión logística y árbol de decisión.



1.6 Evaluacion del modelo

Enlace al repositorio: https://github.com/JoseR200/ue22314990-Practica1

Procedemos a realizar la aplicación del modelo con el dataset de test para obtener el performance basado en una matriz de confusión



Resultados de la matriz de confusión:

Regresión logistica:

accuracy: 94.19%

	true false	true true	class precision
pred. false	1545	93	94.32%
pred. true	98	1550	94.05%
class recall	94.04%	94.34%	

Árbol de decisión:

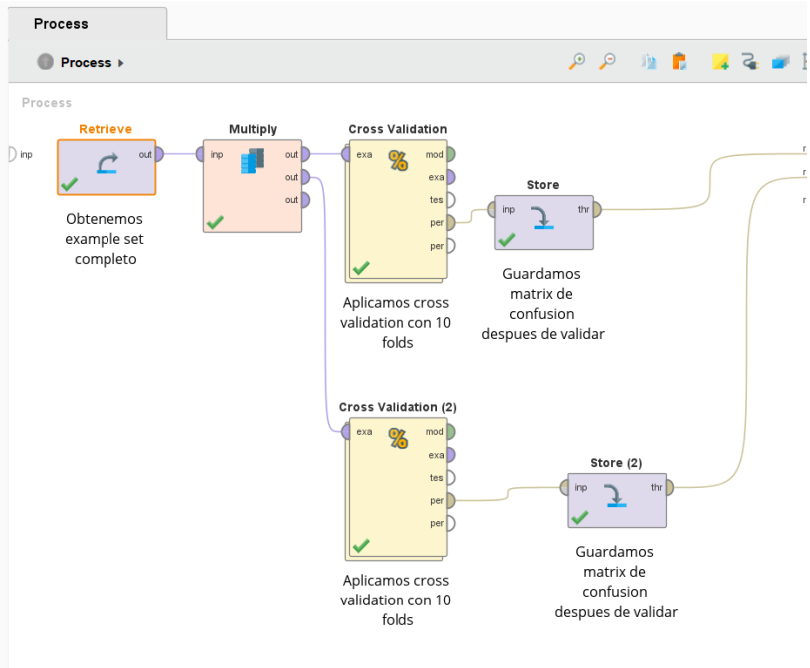
accuracy: 96.41%

	true false	true true	class precision
pred. false	1531	6	99.61%
pred. true	112	1637	93.60%
class recall	93.18%	99.63%	

1.7 Validacion y Optimización de Modelo

Enlace al repositorio: https://github.com/JoseR200/ue22314990-Practica1

Finalmente realizamos el cross validation de 10 folds para tener un acuraccy final



Teniendo como resultados finales, los siguientes:

Regresión logistica: 94.6%

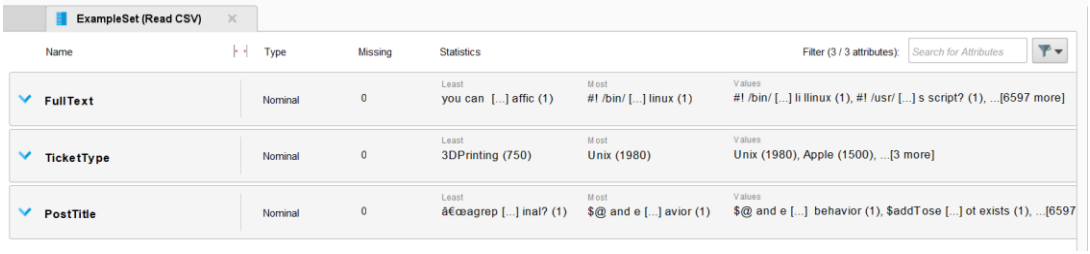
Árbol de decisión: 96.57%

Capítulo 2. Ejercicio 2

2.1 Adquisicion de datos

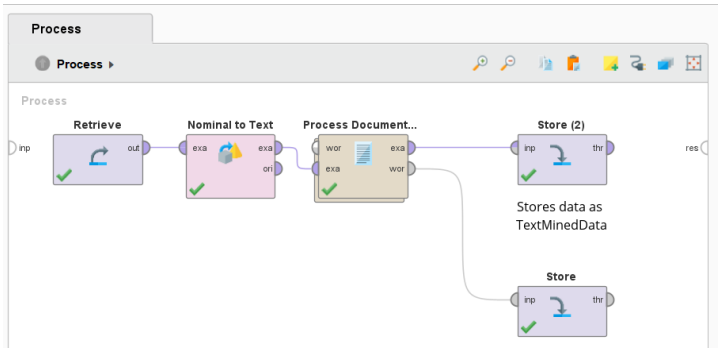
Enlace al repositorio: https://github.com/JoseR200/ue22314990-Practica1

Imagen de la estructura de dataset. Podemos observar que el atributo de Ticket Type se puede normalizar de tipo categórico.



2.2 Minado de datos

Enlace al repositorio: https://github.com/JoseR200/ue22314990-Practica1



Se ha realizado el minado de todos los ítems dentro del data set para obtener la lista de palabras y el texto minado que nos va a servir para entrenar el modelo y la generación de topicos

2.3 Entrenamiento de modelo

Enlace al repositorio: https://github.com/JoseR200/ue22314990-Practica1

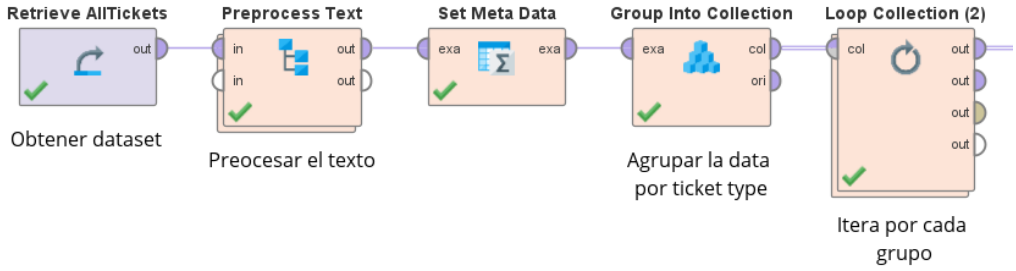
Una vez obtenido la data minada en el paso anterior. Se pasa por un cross validation de 10 folds para generar un modelo que pueda predecir si algún token este relacionado con el tycket type. Consiguiendo el siguiente performance.

accuracy: 85.74% +/- 1.30% (micro average: 85.74%)

	true 3DPrinting	true Android	true Apple	true Dba	true Unix	class precision
pred. 3DPrinting	707	2	1	3	8	98.06%
pred. Android	4	914	135	0	78	80.81%
pred. Apple	18	115	1215	10	158	80.15%
pred. Dba	4	5	20	1197	111	89.53%
pred. Unix	17	43	129	80	1625	85.80%
class recall	94.27%	84.71%	81.00%	92.79%	82.07%	

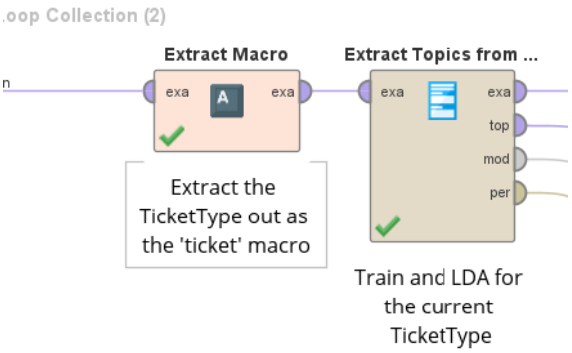
2.4 Generacion de Topicos

Enlace al repositorio: https://github.com/JoseR200/ue22314990-Practica1



Para la generación de los tópicos vamos a preprocesar cada una de las filas del dataset original. Para luego separarla en grupos por el ticket type que cada fila tiene.

Después tocaría aplicar el modelo de LDA para la generación de tópicos por cada una de las colecciones.



Finalmente se genera una carpeta por cada colección con el modelo de topico, la lista de la palabras y el performance.