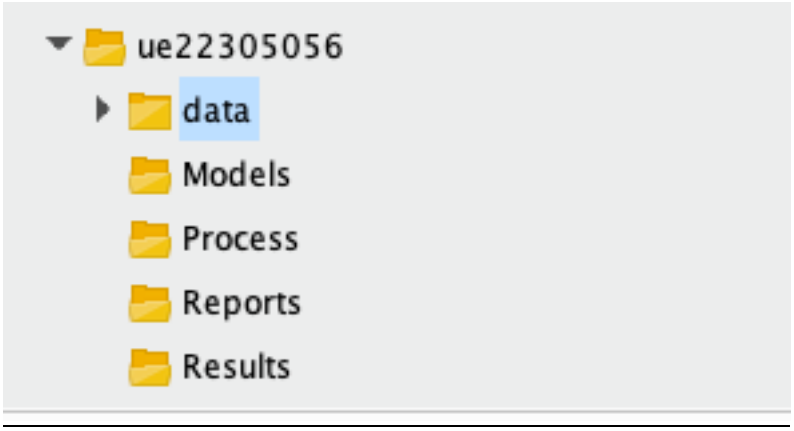
Master en Business Analytics
Módulo III: Fundamentos Tecnológicos en Data Science
Jerika Castillero
22305056
https://github.com/jerikacastillero08/M3Python.git

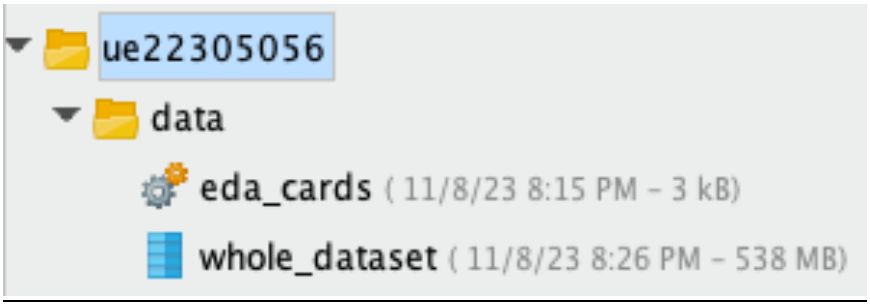
Práctica ETL + EDA + MODELING

Task 00



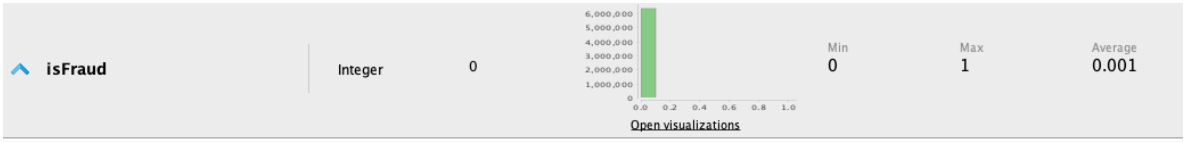
Se crearon en el repositorio local las diferentes carpetas que se estarán utilizando durante el proyecto.

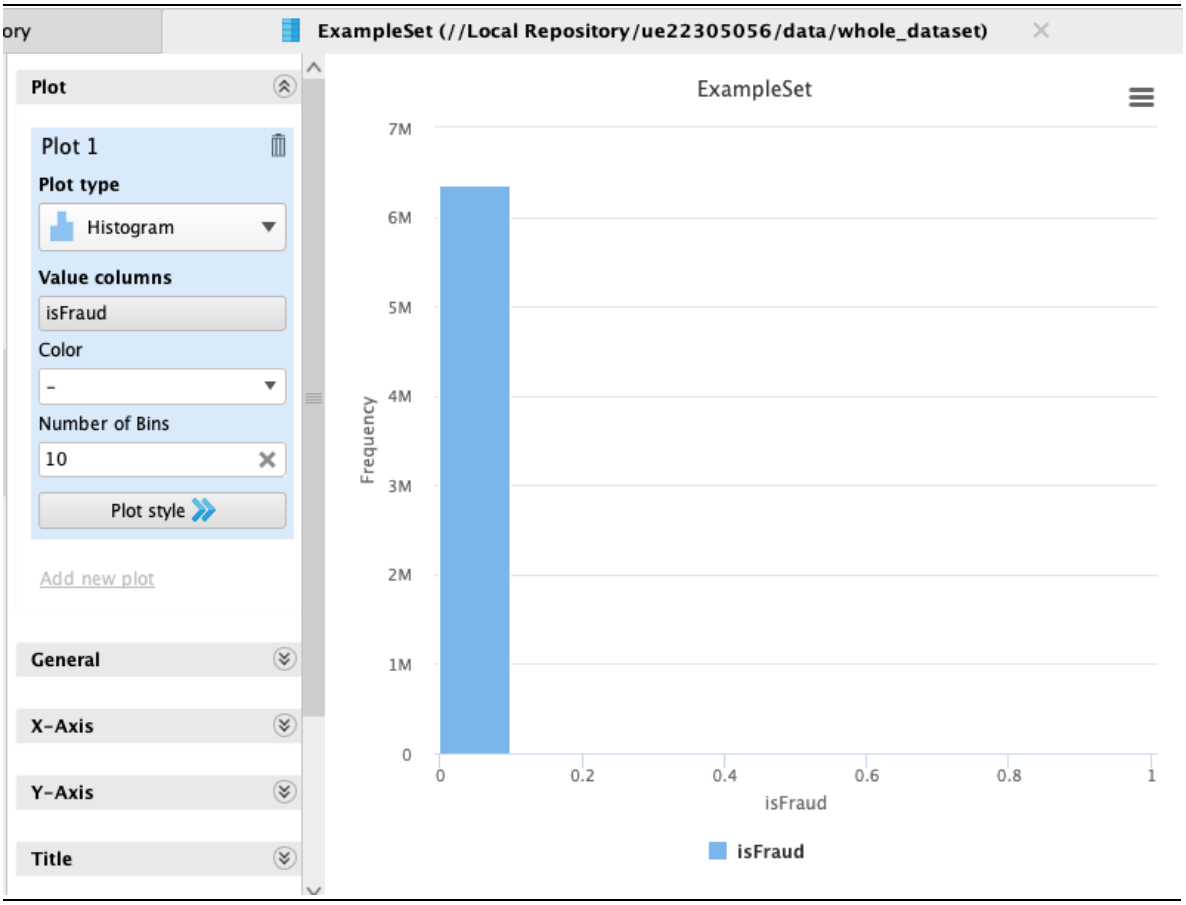
Task 01



Task 02

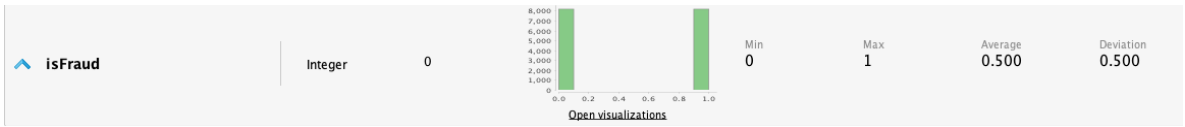
Se importó a la carpeta “data” el archivo csv llamado “whole_dataset” en el cual se basarán todos los procesos de la práctica.

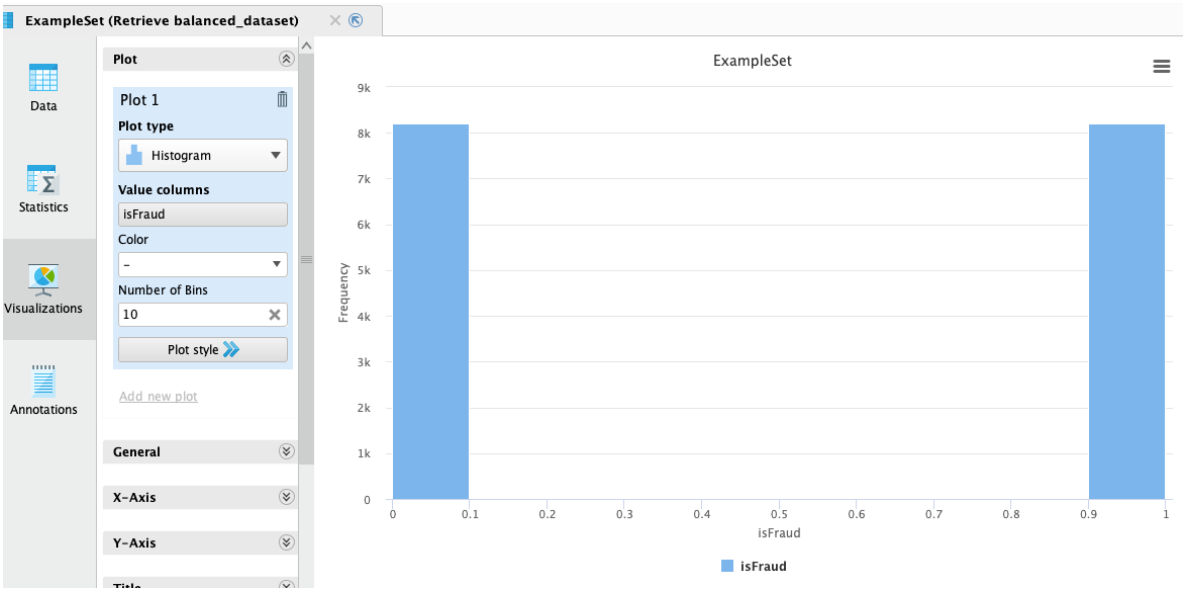




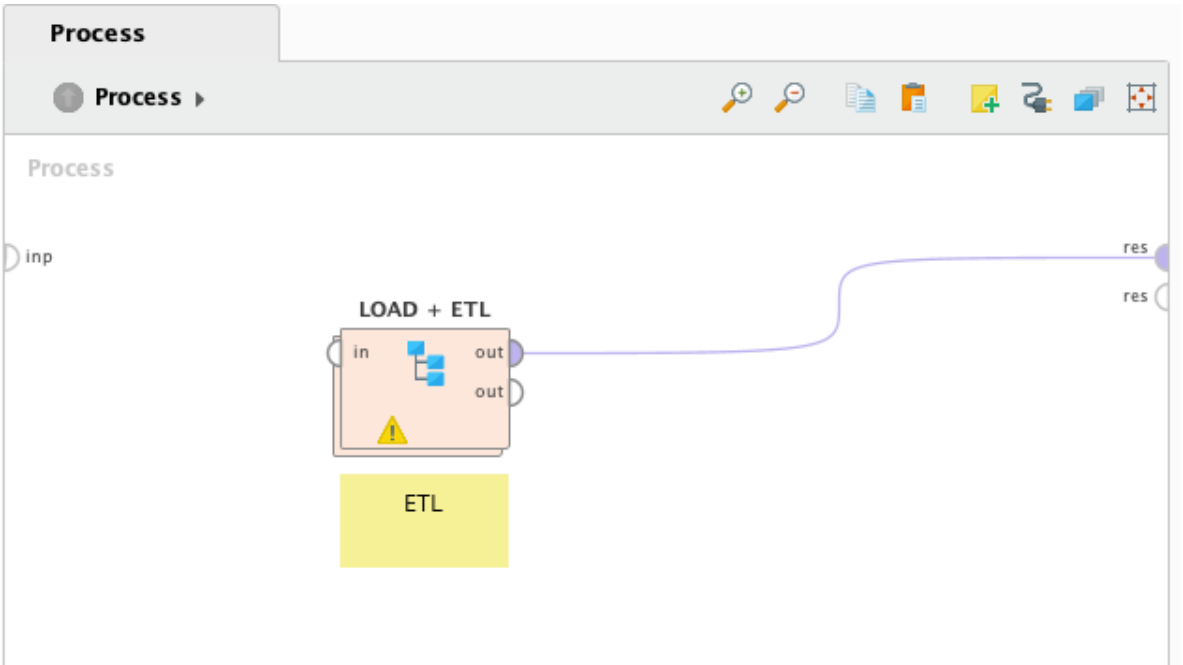
Después de leer los datos de “whole_dataset” se pudo observar que los datos de la variable clave para hacer el estudio de si una transacción es fraude o no “isFraud” no muestran ningun balance. El dataset se encuentra imbalanceado.

Task 03



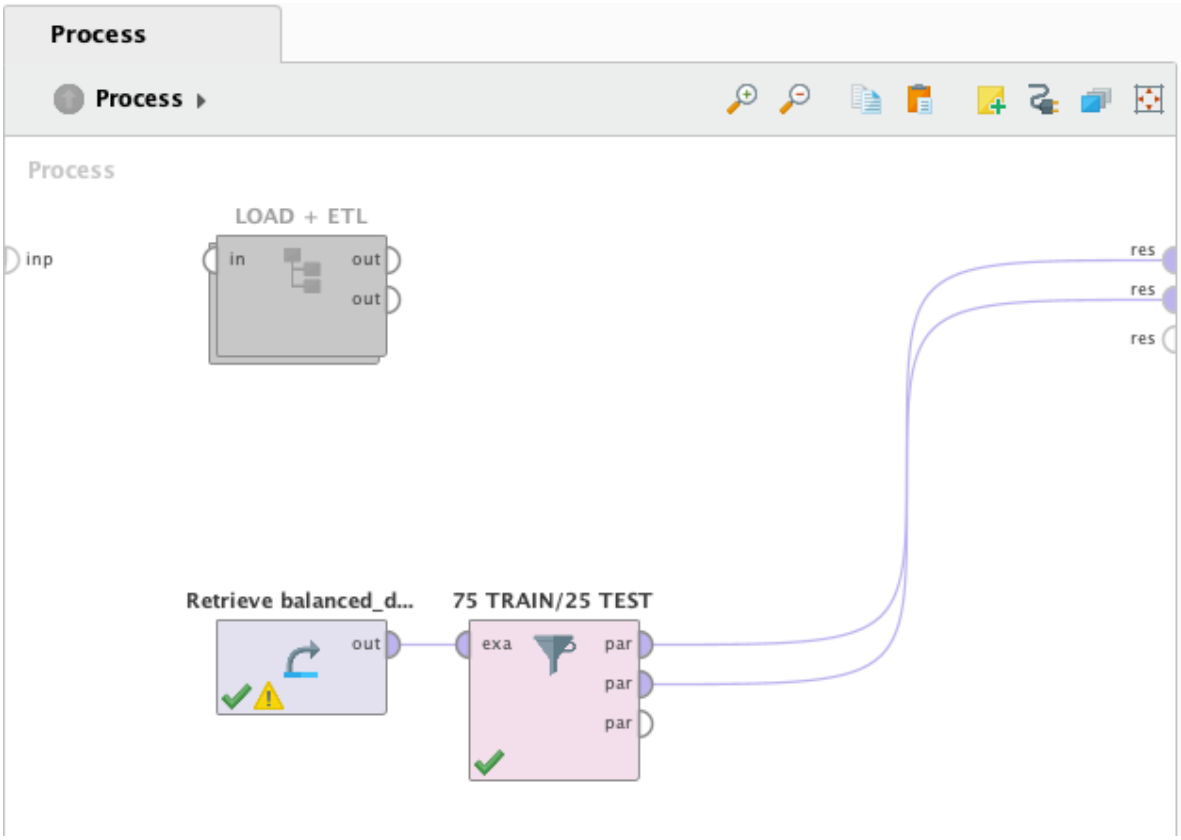


Se puede ver que por medio de un undersampling se obtuvo un dataset balanceado de 16,426 atributos teniendo 8,213 para cada posibilidad del atributo principal “isFraud”.



Se realiza el primer subproceso en el que se cargaron los datos y se balancearon.

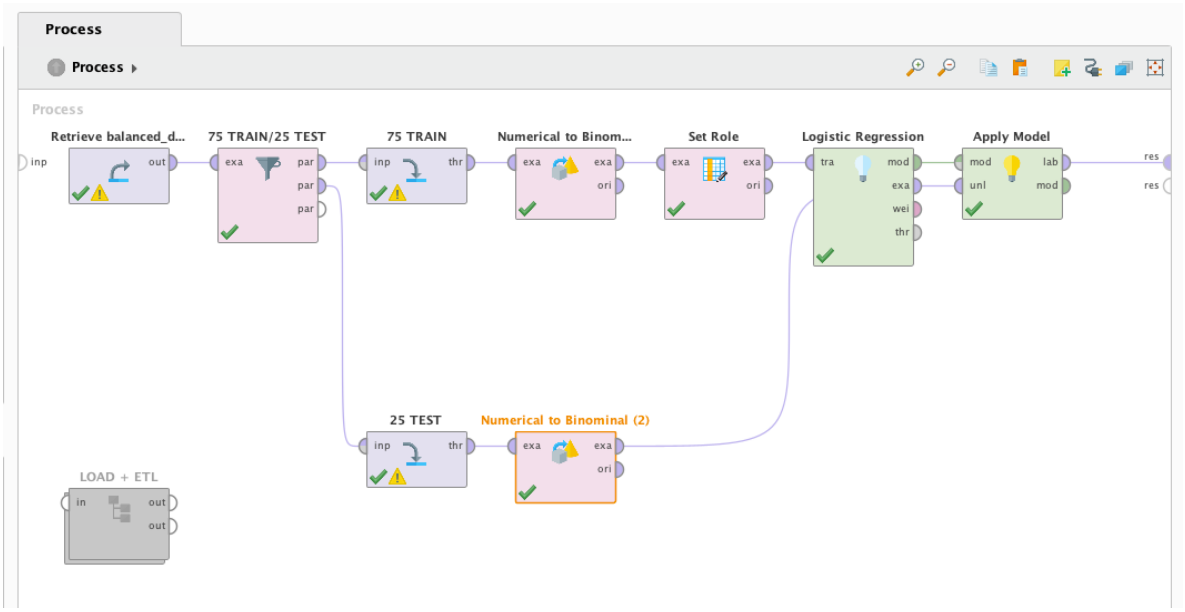
Task 04



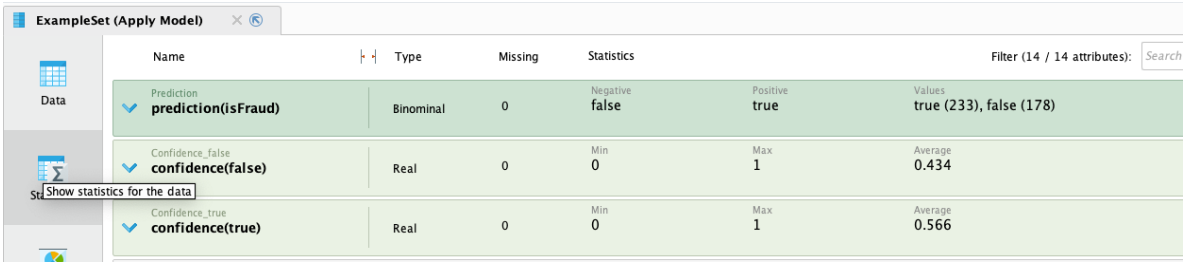
Se apaga el primer subproceso y luego se agarra la data balanceada y se divide 75% para entrenarla y 25% para probar para los modelos.

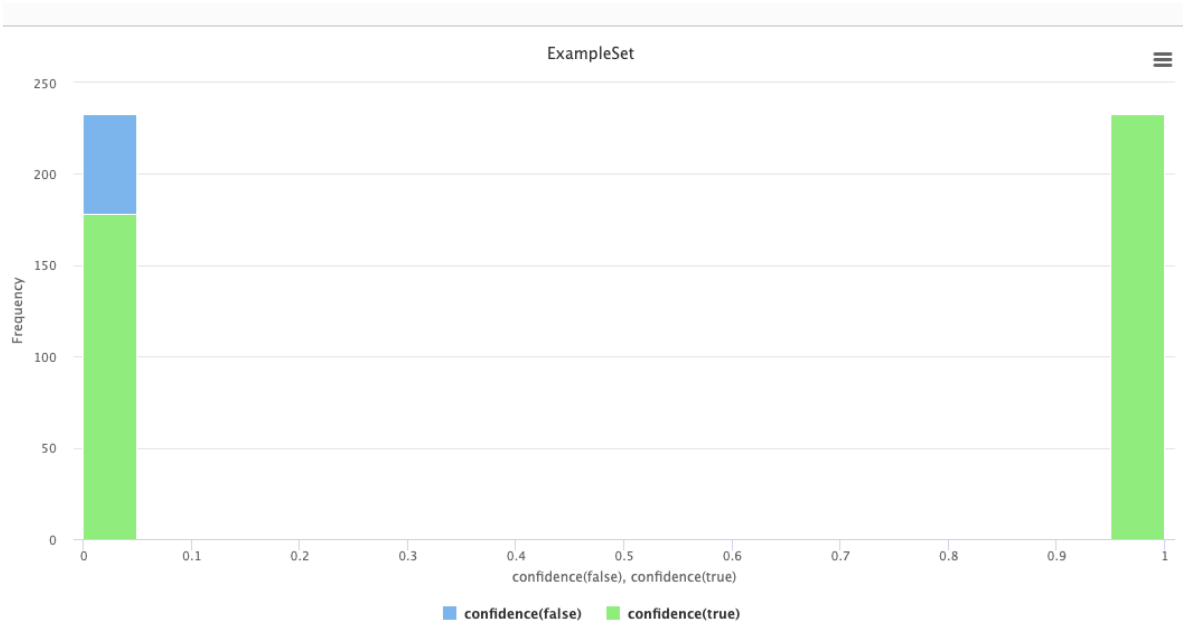
Los modelos escogidos fueron Regresión Logística, Naive Bayes y Decision Tree. Estos modelos fueron escogidos debido a que todos son modelos específicos para Problemas de Clasificación como lo es el problema de esta práctica. A continuación, se presentan los modelos junto con sus resultados. De estos modelos para el dataset presentado el que más accuracy presenta es el de Regresión Logística con un 93.15% de accuracy.

Modelo 1: Regresión logística

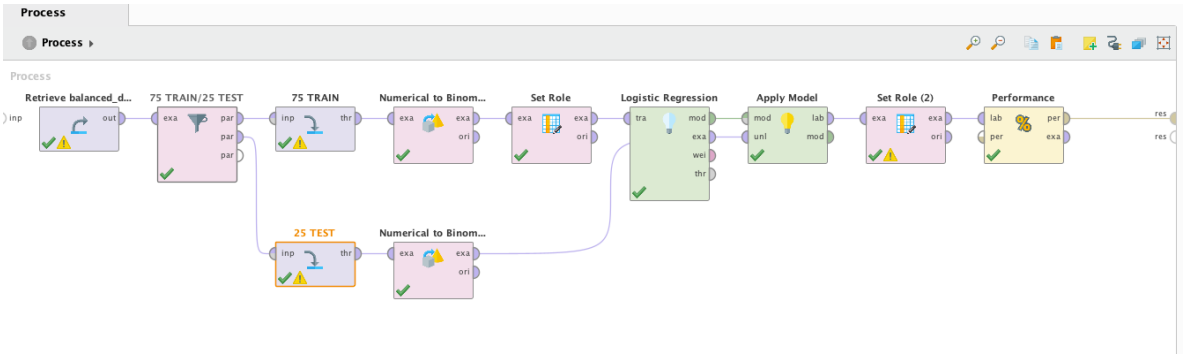


Para poder crear el modelo de Regresión Logísitica se necesitó transformar ambos sets de datos de Numerical a Binomial, así también como establecer el label (Set role) que utilizaría para realizarse. En este caso el label es “isFraud” dado que es el parámetro de clasificación.

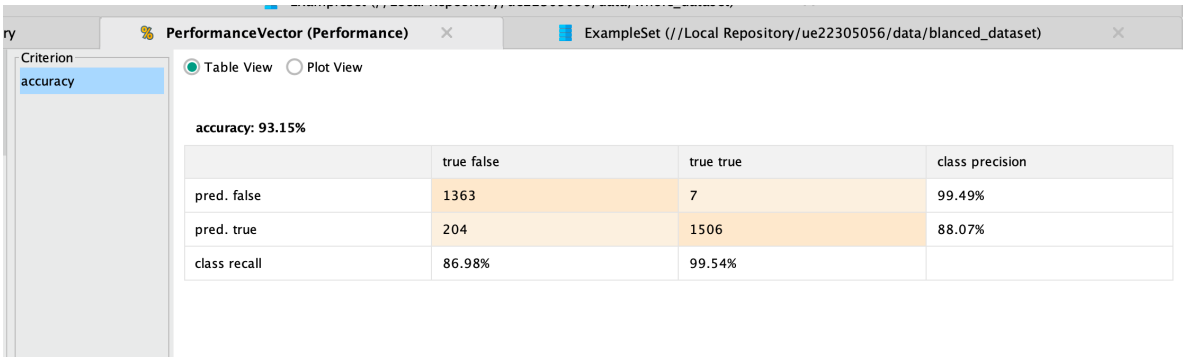




Aquí se pueden obtener los resultados del modelo.

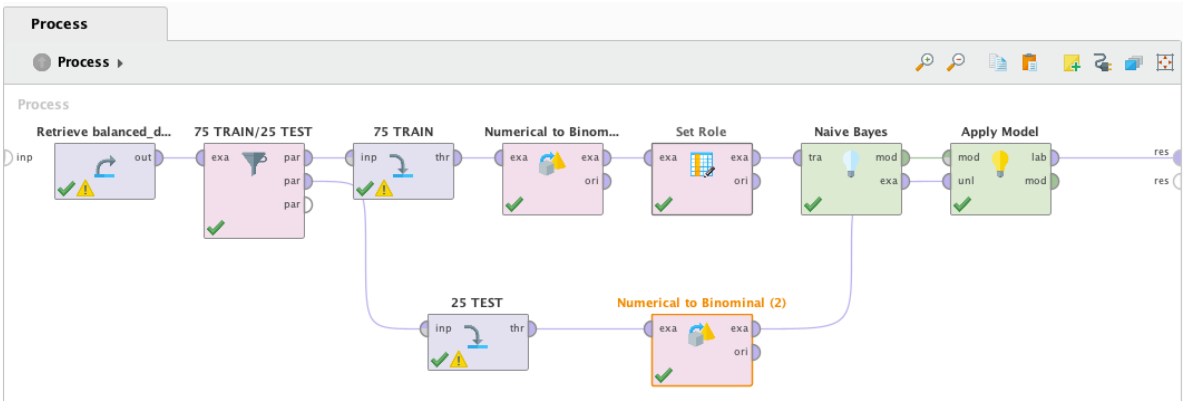


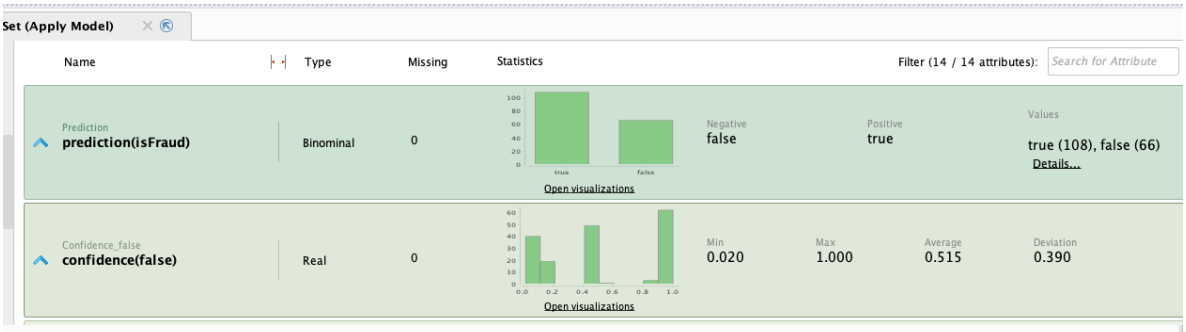
Como no me quedaban claros los resultados del modelo coloqué un operador de Rendimiento para saber el accuracy del modelo.

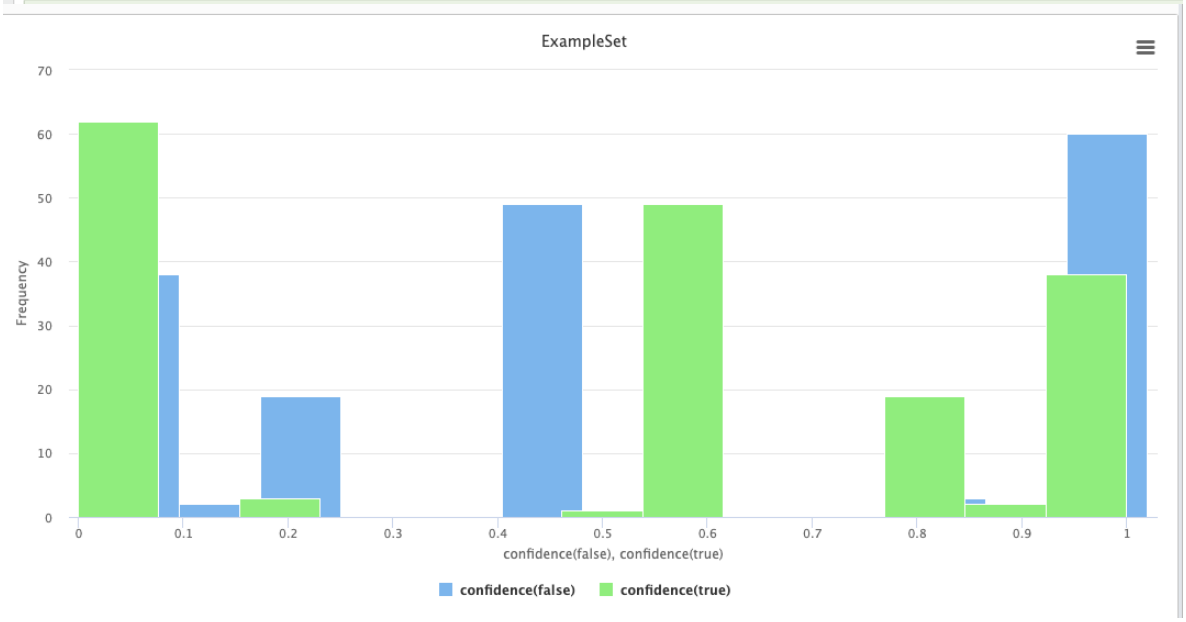


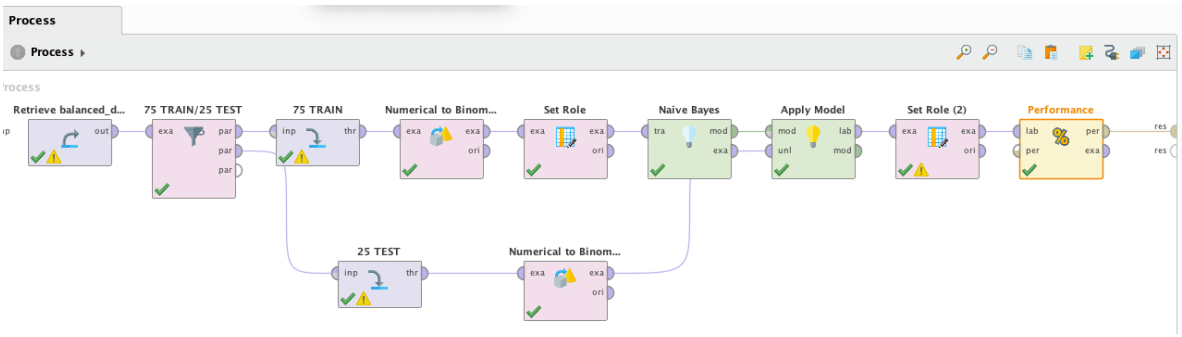
Como se puede observar, en términos generales, este modelo tiene un accuracy del 93.15%.

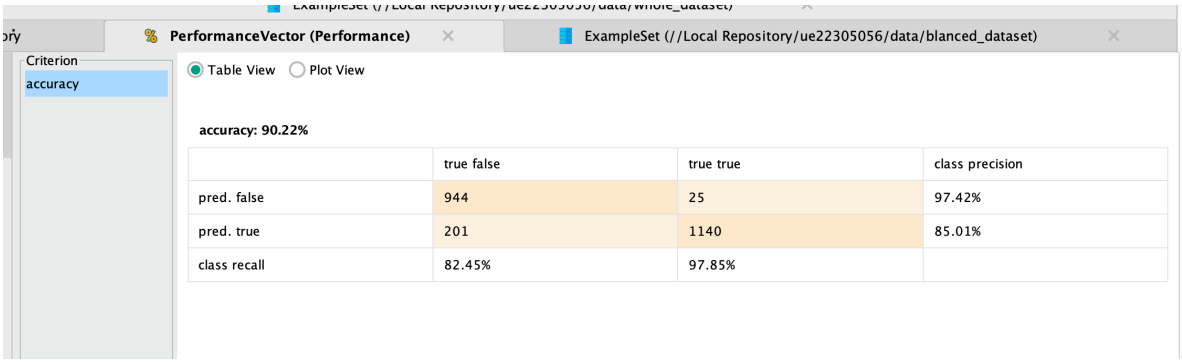
Modelo 2: Naive Bayes


Process
Process
Process
Retrieve balanced_d...
75 TRAIN/25 TEST
75 TRAIN
Numerical to Binom...
Set Role
Naive Bayes
Apply Model
25 TEST
Numerical to Binominal (2)


Set (Apply Model)
Name
Type
Missing
Statistics
Filter (14 / 14 attributes):
Search for Attribute
Prediction
prediction(isFraud)
Binominal
0
Negative
false
Positive
true
Values
true (108), false (66)
Details...
Open visualizations
Confidence_false
confidence(false)
Real
0
Min
0.020
Max
1.000
Average
0.515
Deviation
0.390
Open visualizations


ExampleSet
Frequency
confidence(false), confidence(true)
confidence(false)
confidence(true)

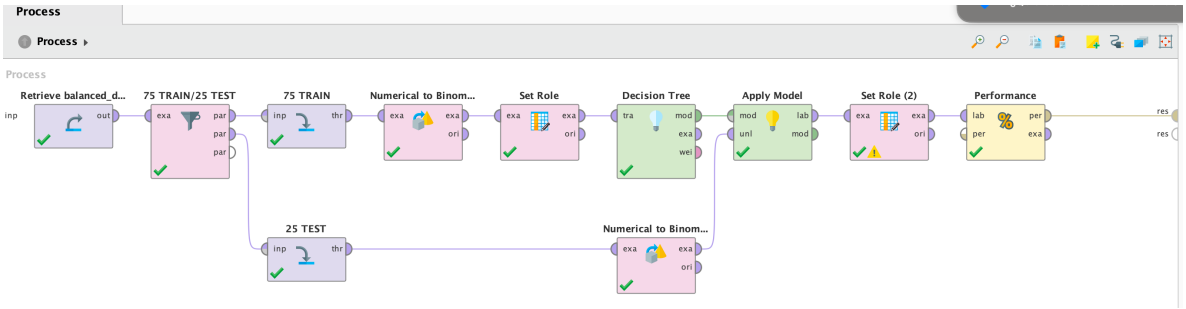


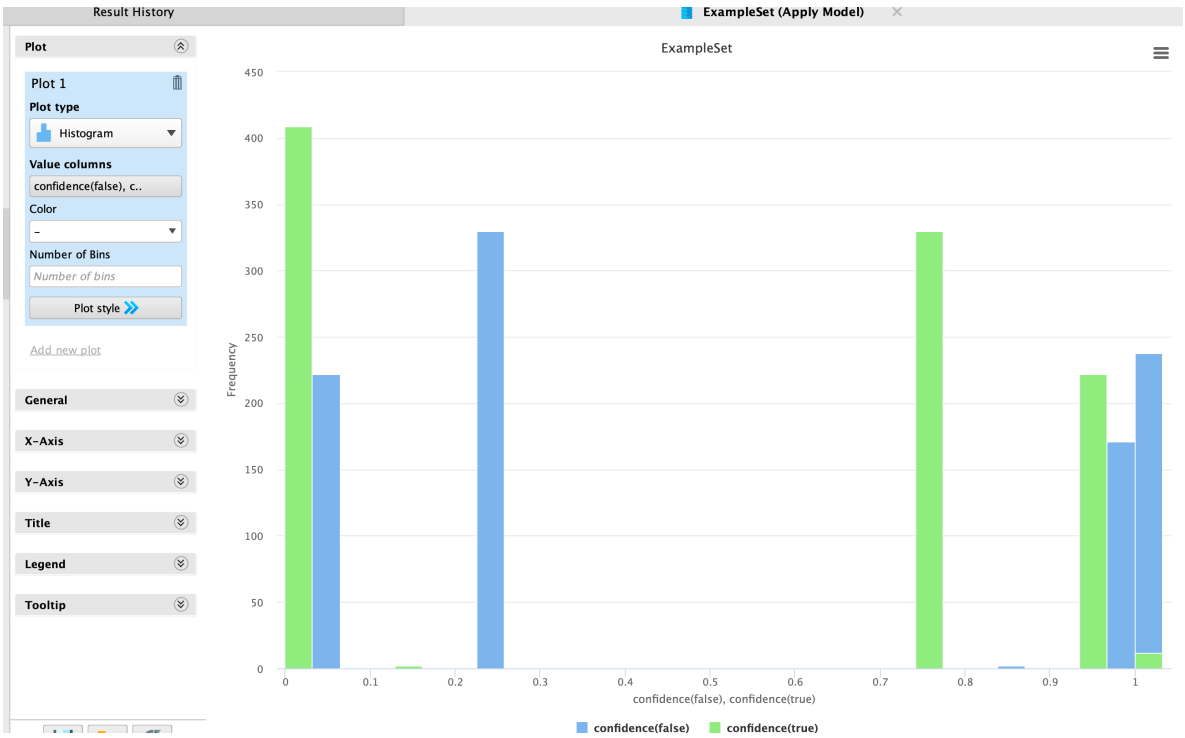


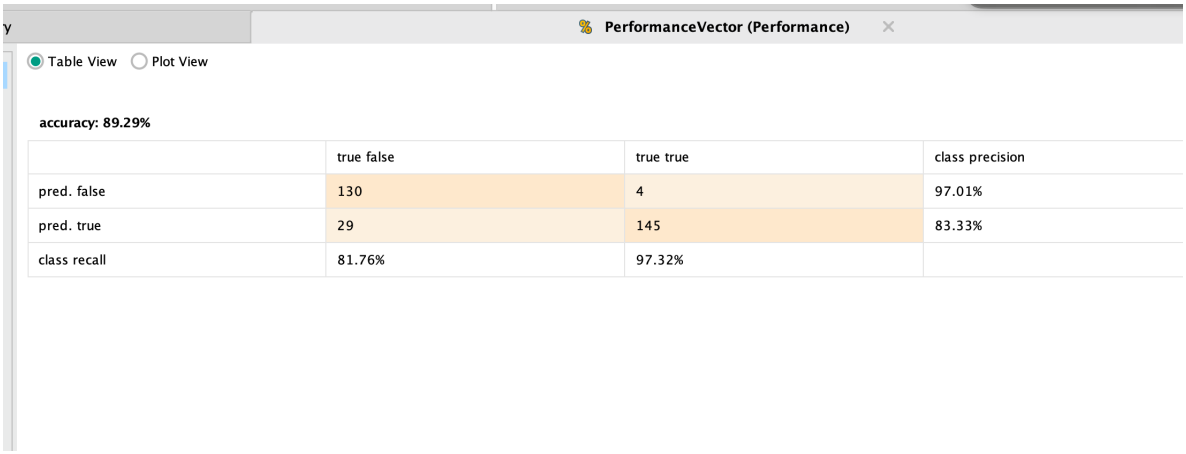
	true false	true true	class precision
pred. false	944	25	97.42%
pred. true	201	1140	85.01%
class recall	82.45%	97.85%	

Para este modelo se siguieron los mismos pasos que para el primero y se obtuvo un 90.22% de accuracy.

Modelo 3: Decision tree







	true false	true true	class precision
pred. false	130	4	97.01%
pred. true	29	145	83.33%
class recall	81.76%	97.32%	

Siguiendo los pasos para crear el modelo, entrenarlo y probarlo como en los dos ejemplos anteriores tenemos que este presenta el menor porcentaje de accuracy con un 89. 29% siendo este el modelo menos adecuado de los tres modelos probados.

Task 05

Load Data

```
[3]: import pandas as pd

[5]: df = pd.read_csv('Desktop/M3Python/PwC/credit_card_bal_1.csv')

[8]: df.describe()
```

[8]:

	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
count	16426.000000	1.642600e+04	1.642600e+04	1.642600e+04	1.642600e+04	1.642600e+04	16426.000000	16426.000000
mean	306.185194	8.236570e+05	1.252718e+06	5.348437e+05	8.341153e+05	1.264052e+06	0.500000	0.000974
std	192.704918	1.852158e+06	3.277629e+06	2.539971e+06	3.226697e+06	3.592841e+06	0.500015	0.031196
min	1.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000
25%	162.000000	3.662393e+04	1.067450e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000
50%	283.000000	1.706524e+05	1.202234e+05	0.000000e+00	0.000000e+00	1.244370e+05	0.500000	0.000000
75%	408.000000	5.482430e+05	7.965316e+05	0.000000e+00	5.214951e+05	1.115317e+06	1.000000	0.000000
max	743.000000	1.511569e+07	5.958504e+07	4.958504e+07	2.362305e+08	2.367265e+08	1.000000	1.000000

Para cargar el archivo de los datos balanceados con Python se utilizó el Jupyter Lab como IDLE. Se importó la librería de python Pandas debido a que es la más adecuada para hacer este tipo de problemas de análisis de datos como el presentado en esta práctica.

Una vez cargada la librería se definió que el data frame como “df” y se cargo el archívo csv por medio de su path.

Una vez cargado el archivo se paso a usar el comando describe para observar los datos.

```
        dtype='object')

[13]: df.corr

[13]: <bound method DataFrame.corr of        step      type      amount     nameOrig  oldbalanceOrg  newbalanceOrig  \
0        18   CASH_IN   180078.01  C1791832105     2796833.20      2976911.22
1       258   PAYMENT     2138.35   C634754263           0.00            0.00
2       183  TRANSFER   342675.47  C1778106585      342675.47            0.00
3       577  TRANSFER   141730.20  C1677850525      141730.20            0.00
4       742  TRANSFER  4009058.39  C1044665079     4009058.39            0.00
...     ...       ...         ...          ...            ...             ...
16421   141  CASH_OUT    14898.80  C1882828175      104665.00        89766.20
16422   188   CASH_IN   437986.25  C1072308575       50474.00       488460.25
16423   586  CASH_OUT        0.00  C1303719003           0.00            0.00
16424   355  CASH_OUT    42483.97  C1595793252       42483.97            0.00
16425   374   PAYMENT     6069.42  C1139940003           0.00            0.00

          nameDest  oldbalanceDest  newbalanceDest  isFraud  isFlaggedFraud
0      C1756248403      1169682.86       989604.85        0               0
1       M205383539            0.00            0.00        0               0
2       C144558690            0.00            0.00        1               0
3       C607738036            0.00            0.00        1               0
4       C750074708            0.00            0.00        1               0
...            ...             ...             ...      ...             ...
16421  C1045368778      2756474.78      2771373.59        0               0
16422  C1532905677        49843.00            0.00        0               0
16423   C900608348      1328472.86      1328472.86        1               0
16424   C636454309        51495.75        93979.72        1               0
16425    M17941836            0.00            0.00        0               0

[16426 rows x 11 columns]>
```

Luego se utilizó el comando corr para poder observar la correlación entre los datos del dataset.