Ley de Benford y Machine Learning para detectar lavado de dinero

[1] *Department of Applied Economics, University of Valencia, 46022 Valencia, Spain*

Introducción

- **Objetivos:** Este trabajo se basa en el análisis de una base de datos sobre operaciones de un macrocaso de lavado de dinero orquestado entre una empresa principal y un grupo de sus proveedores, 26 de los cuales ya habían sido identificados por la policía como empresas fraudulentas.
- **Métodos:** Combinamos la Ley de Benford y algoritmos de aprendizaje automático (regresión logística, árboles de decisión, redes neuronales y bosques aleatorios) para encontrar patrones de criminales de lavado de dinero en el contexto de un caso real en España.

Ley de Benford

La ley de Benford (por el físico Frank Benford), también conocida como la ley del primer dígito, asegura que, en gran variedad de conjuntos de datos numéricos que existen en la vida real, la primera cifra es 1 con mucha más frecuencia que el resto de los números. Además, según crece este primer dígito, menos probable es que se encuentre en la primera posición. La ley también asegura cierta frecuencia para los siguientes dígitos. Una extensión de la ley de Benford predice la distribución de los primeros dígitos en otras bases además del decimal ; de hecho, cualquier base $b \geq 2$. La forma general es:

$$\begin{aligned} P(d) &= \log_{10}(d+1) - \log_{10}(d) \\ &= \log_{10}\left(\frac{d+1}{d}\right) \\ &= \log_{10}\left(1+\frac{1}{d}\right) \end{aligned}$$

d	$P(d)$	Relative size of $P(d)$
1	30.1%	
2	17.6%	
3	12.5%	
4	9.7%	
5	7.9%	
6	6.7%	
7	5.8%	
8	5.1%	
9	4.6%	

Esta ley se puede aplicar a muchos hechos relacionados con el mundo natural o con elementos sociales: facturas, artículos en revistas, números de puerta, precios, número de habitantes, tasas de mortalidad, longitud de los ríos, etcétera.
Un conjunto de datos que no cumpla la ley de Benford no precisamente indica que se hayan manipulado los datos, sin embargo, el que no se cumpla dicha ley es razón suficiente para realizar una investigación.

Ley de Benford y ML

Utilizaron la Ley de Benford como herramienta para caracterizar los registros contables de las operaciones comerciales entre la empresa principal y los proveedores y aplicaron cuatro modelos de clasificación para identificar otros posibles proveedores fraudulentos. Además los datos de dichas empresas fraudulentas fueron proporcionados por la policía donde el objetivo final es descubrir el mayor número posible de empresas fraudulenta y, al mismo tiempo, reducir la probabilidad de clasificar erróneamente a empresas que están operando correctamente.

Dataset

Se estudió el caso más grande de lavado de dinero en España, la red criminal consistía de un gran número de distribuidores, estructurados en clans y jerarquicamente organizados, todos estos coordinados por una compañía principal, esta compañía principal era parte de un grupo de negocios internacional que a su vez a partir de prácticas ilicitas financiaban a la compañia principal. La compañia le ofreció a los distribuidores (legales e ilegales) pagos por encima del mercado para sus servicios.
La investigación reunió una gran base de datos de 285774 operaciones llevadas a cabo por 643 distribuidores, por cada operación había que articulo se comercializó, dónde se comercializó, quién realizó la operación, el costo y la cantidad asociada con la operación.
A priori se sabe que el 4 % de los distribuidores eran fraudulentos, de los demás distribuidores es desconocido si eran fraudulentos o no.

Feature engineering

Las variables utilizadas de la base de datos para entrenar los modelos son: El número de operaciones registradas y un conjunto de 20 p-valores, correspondientes a Z-tests donde $F1 - F9$ corresponden a los p-valores asociados a los primeros digitos y $S0 - S9$ los p-valores de los segundos digitos, en adición se utiliza el p-valor del OverBenford test.

$$Z_{\mathrm{i}} = \frac{|\mathrm{n_{Oi}} - \mathrm{n_{Ti}}| - \frac{1}{2\ \mathrm{N}}}{\sqrt{\frac{\mathrm{n_{Ti}}(1-\mathrm{n_{Ti}}}{\mathrm{N}}}}$$

Resultados

Table 4
Confusion matrix of the models. Imbalanced dataset.

	LR		DT		NN		RF	
	No	Yes	No	Yes	No	Yes	No	Yes
No	311	1	302	10	301	11	312	0
Yes	20	3	17	6	15	8	19	4
Correctly classified	93.73%		91.94%		92.24%		94.33%	
Incorrectly classified	6.27%		8.06%		7.76%		5.67%	
TN rate (No)	99.68%		96.79%		96.47%		100.00%	
TP rate (Yes)	13.04%		26.09%		34.78%		17.39%	
FN rate (Yes)	86.96%		73.91%		65.22%		82.61%	
FP rate (No)	0.32%		3.21%		3.53%		0.00%	

Table 5
Confusion matrix of the models. Cost Matrix.

	LR		DT		NN		RF	
	No	Yes	No	Yes	No	Yes	No	Yes
No	234	78	290	22	285	27	309	3
Yes	10	13	16	7	15	8	17	6
Correctly classified	73.73%		88.66%		87.46%		94.03%	
Incorrectly classified	26.27%		11.34%		12.54%		5.97%	
TN rate (No)	75.00%		92.95%		91.35%		99.04%	
TP rate (Yes)	56.52%		30.43%		34.78%		26.09%	
FN rate (Yes)	43.48%		69.57%		65.22%		73.91%	
FP rate (No)	25.00%		7.05%		8.65%		0.96%	

Table 6
Confusion matrix of the models. SMOTE.

	LR		DT		NN		RF	
	No	Yes	No	Yes	No	Yes	No	Yes
No	239	73	269	43	252	60	300	12
Yes	53	246	31	268	38	261	15	284
Correctly classified	79.38%		87.89%		83.96%		95.58%	
Incorrectly classified	20.62%		12.11%		16.04%		4.42%	
TN rate (No)	76.60%		86.22%		80.77%		96.15%	
TP rate (Yes)	82.27%		89.63%		87.29%		94.98%	
FN rate (Yes)	17.73%		10.37%		12.71%		5.02%	
FP rate (No)	23.40%		13.78%		19.23%		3.85%	

Table 7
Measurements of precision of the procedures.

	Imbalanced dataset			Cost matrix			SMOTE		
	ROC	Kappa	RMSE	ROC	Kappa	RMSE	ROC	Kappa	RMSE
LG	0.747	0.2061	0.2360	0.711	0.3243	0.4227	0.844	0.5675	0.4012
DT	0.635	0.2664	0.2702	0.615	0.2086	0.3320	0.894	0.7348	0.3499
NN	0.765	0.3400	0.2578	0.630	0.2104	0.3306	0.926	0.7252	0.3392
RF	0.740	0.2817	0.2268	0.773	0.3499	0.2415	0.989	0.9116	0.2088

Table 8
Average of confusion matrix summaries of the models (training set).

	LR	DT	NN	RF
Correctly classified	78.06%	88.31%	86.60%	95.16%
Incorrectly classified	21.94%	11.69%	13.40%	4.84%
TN rate (No)	77.01%	86.55%	83.60%	93.87%
TP rate (Yes)	79.09%	90.03%	89.54%	96.16%
FN rate (Yes)	20.91%	9.97%	10.46%	3.84%
FP rate (No)	22.99%	13.45%	16.40%	6.13%

Conclusión

1. Las redes neuronales producen el mejor rendimiento cuando se trabajo con el dataset imbalanceado.
2. Random Forest produce mejor rendimiento cuando se trabaja con el dataset balanceado.
3. El algoritmo SMOTE de balanceo incrementa el rendimiento de todos los algoritmos.