Mejorar la predicción de default para clientes con tarjetas de crédito

Alain Alejo Huarachi, Erison Mostacero Ramirez,
John E. Miller y Ricardo Linares Juarez
Facultad de Ingeniería, Pontificia Universidad Católica del Perú

16 Julio 2019

Contexto del Problema

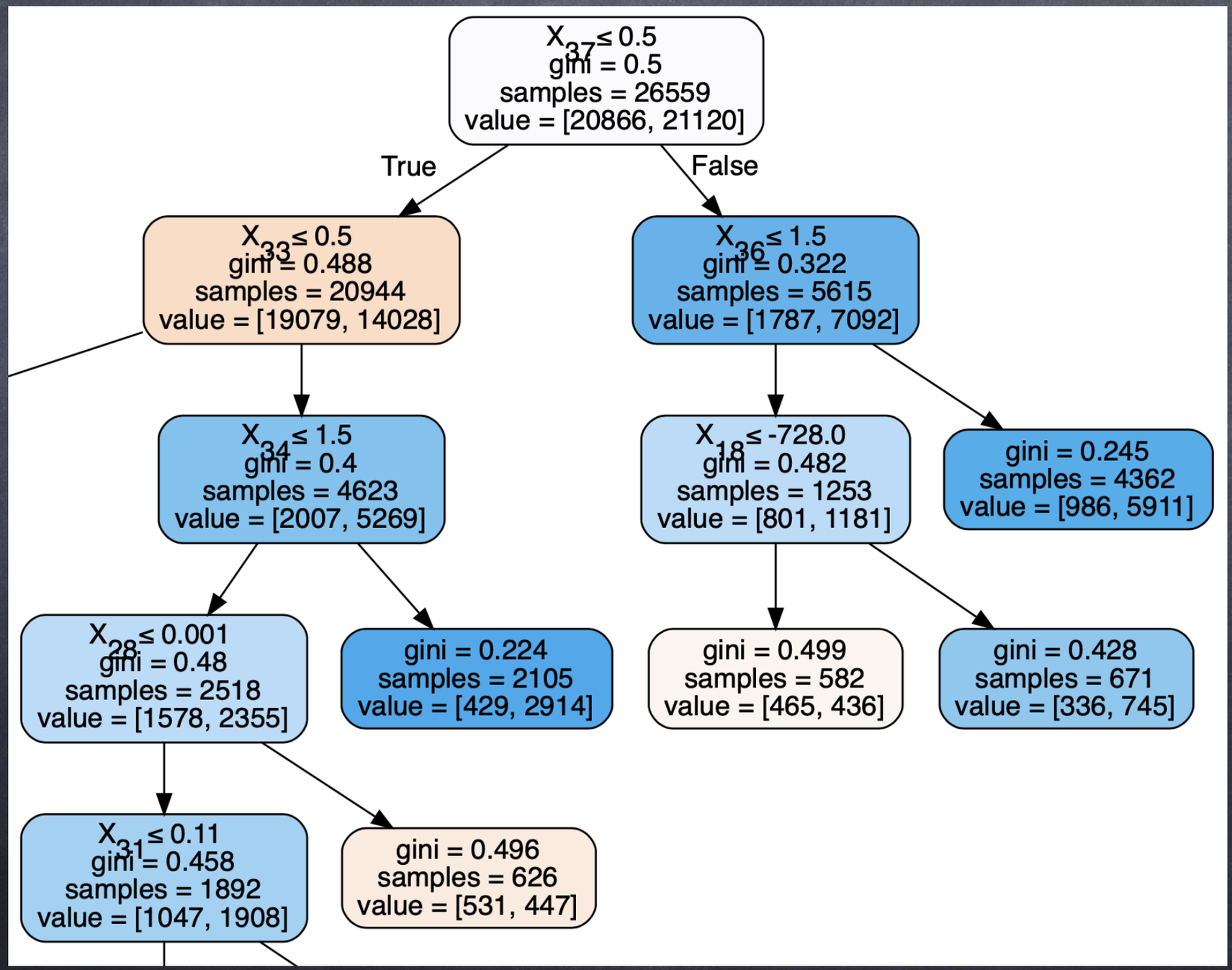
- Conjunto de Datos
  - Información del cliente, datos de la deuda y pagos realizados, clase binaria de «default» (0=no, 1=si)
  - Datos de 30,000 clientes (80% entren, 10% val, 10% prueba)
- Medida de Calidad
  - Costo adaptado: $J = -1TP + 5FN + 1FP + 0TN$
  - Exactitud: $Acc = (TP + TN)/N$
- Muestreo
  - Conjunto desbalanceado (22.1% «default», 77.9% OK)
  - «Oversampling» y transformaciones (raíz cuadrado de dinero, categorizaron de pago)

Metodología

- K Nearest Neighbor (KNN)
  - Vecinos más cercanos para predecir «default»
- Random Forests (RF)
  - Árboles multiples de decisiones para predecir «default»
  - Experimentar con: cantidad y profundidad de arboles, y decremento de impureza
- Support Vector Machines (SVM)
  - Busca la separación mas amplia del hiperplano de los datos para predecir «default»
  - Experimentar con kernels: «linear, poly, rbf, sigmoid»
- Neural Networks (NN)
  - Red neuronal de entrada de datos, capas escondidas de procesar, salida de probabilidad de «default»
  - Experimentar con configuraciones (32x16x8), (64x32x16x8), (64x16), y con la función de perdida $J = c_y y \log(h(x)) + (1 - y)\log(1 - h(x))$

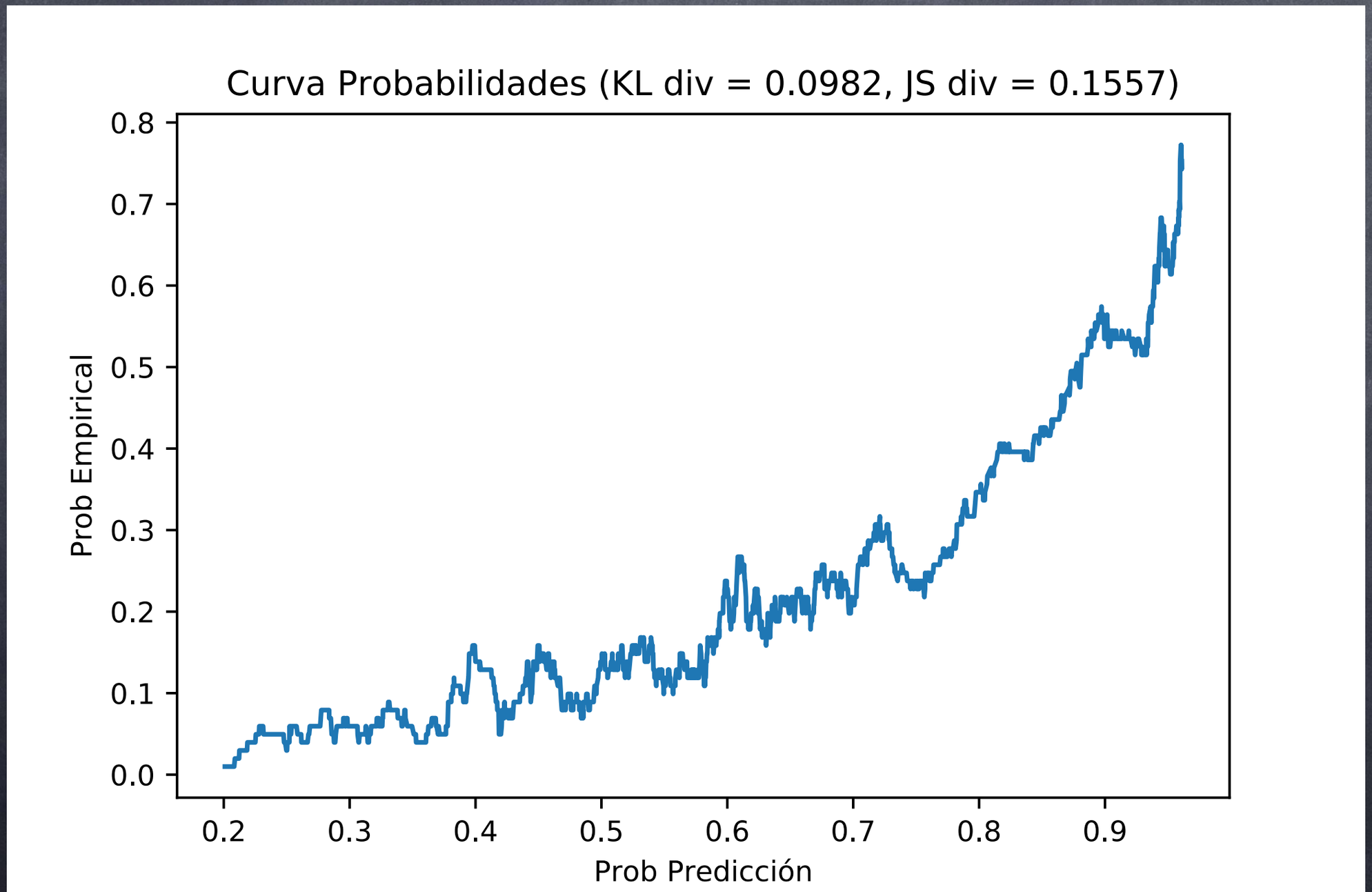
Experimentación y Resultados

- Random Forest (ejemplar)



Experimentación y Resultados

- Redes Neuronales (Ejemplar)



Experimentación y Resultados de todos los algoritmos

Algoritmo	Conjunto	Costo	Exactitud
Red neuronal	Validación	0.375	0.528
	Prueba	0.413	0.484
Random forests	Validación	0.453	0.742
	Prueba	0.451	0.762
Support vector machines	Validación	0.484	0.772
	Prueba	0.474	0.781
K nearest neighbor	Validación	0.594	0.675
	Prueba	0.668	0.616

Conclusiones

- El modelo red neuronal es el mejor basado en nuestra medida de costo adaptado.
- Aprendizajes:
  - Los datos desbalanceados tienen impacto negativo en todos los algoritmos, oversampling fue efectivo.
  - Una función de costo adaptado puede chocar con la de pérdida y tener otros efectos colaterales.
  - La transformación de datos con raíz cuadrada y categorización no ayudó a random forest ni a red neuronal.
  - SVM mantiene su exactitud cuando baja el costo. Podemos ahondar la investigación de esta técnica en un siguiente trabajo.