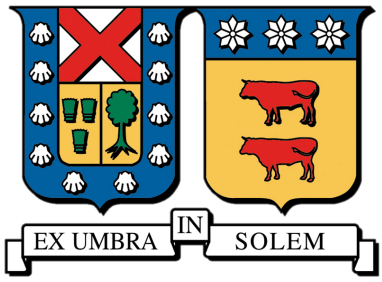


Universidad Técnica Federico Santa María

MAT-306

Detección de Transacciones Fraudulentas en Tarjetas de Crédito

---

Proyectos Estadísticos

Autores:
Bastián Aceitón
Vicente Frías

2024-2

Índice

Análisis del Problema

Contexto del Problema

El fraude en tarjetas de crédito es un problema grave que afecta tanto a las instituciones financieras como a los usuarios. Las compañías necesitan identificar de manera rápida y eficiente las transacciones fraudulentas, de manera que se minimicen las pérdidas y se proteja al cliente. Este informe se enfoca en la creación de un modelo capaz de detectar fraudes, utilizando una base de datos que contiene transacciones realizadas con tarjetas de crédito en septiembre de 2013 por titulares de tarjetas europeos. Este conjunto de datos presenta transacciones que ocurrieron en dos días, donde se tienen 492 fraudes de un total de 284.807 transacciones. El conjunto de datos está altamente desbalanceado, la clase positiva (fraudes) representa el 0,172 % de todas las transacciones.
El dataset contiene solo variables de entrada numéricas las cuales son el resultado de una transformación PCA. Desafortunadamente, debido a temas de confidencialidad, en la base de datos no están las características originales, ni más información de fondo sobre los datos. Las características V1, V2, ..., V28 son los componentes principales obtenidos con PCA, y las únicas características que no han sido transformadas son 'Time' y 'Amount'. La columna 'Time' contiene los segundos transcurridos entre cada transacción y la primera transacción en el conjunto de datos. La característica 'Amount' es el monto de la transacción, y esta característica puede usarse, por ejemplo, para aprendizaje sensible al costo dependiendo del ejemplo. La característica 'Class' es la variable de respuesta y toma el valor 1 en caso de fraude y 0 en caso contrario.

Características Deseadas

Para que el sistema sea efectivo en la detección de fraudes, se deben cumplir las siguientes características:

1. **Detección de fraudes en tiempo real con alta precisión:**
   Es ideal que el sistema logre identificar las transacciones fraudulentas a medida que ocurren, sin demoras que puedan afectar al cliente o que permitan que se materialice el fraude. Además, la precisión del modelo es clave, ya que es necesario evitar la no detección de fraudes (falsos negativos) y la identificación incorrecta de transacciones legítimas como fraudulentas (falsos positivos).

2. **Minimización de falsos positivos y falsos negativos:**
   Debe haber un equilibrio entre la detección de fraudes y las interrupciones innecesarias de las transacciones legítimas. Minimizar los falsos positivos es esencial para no generar molestias a los clientes y evitar bloqueos injustificados a sus cuentas. También, minimizar falsos negativos es vital para asegurar que la mayor cantidad posible de transacciones fraudulentas sea detectada.

3. **Capacidad para procesar grandes volúmenes de datos de manera eficiente:**
   Debido a que las instituciones financieras manejan cientos de miles de transacciones diariamente, el sistema debe estar optimizado de tal forma que procese grandes volúmenes de datos en forma eficiente. Esto implica poder realizar análisis precisos y rápidos, todo esto sin influir en la capacidad de identificar las actividades fraudulentas.

Criterios de Éxito

Para asegurar el éxito del modelo de detección de fraudes, se deben cumplir los siguientes criterios:

1. **Área bajo la curva de Precision-Recall (AUPRC):**
   Por el desbalance de las clases, el rendimiento del modelo se evaluará mediante la AUPRC, la cual es más adecuada para medir en contextos en donde las clases están desbalanceadas. Un alto AUPRC asegura que el modelo equilibre bien la precisión y la recuperación, minimizando tanto los falsos positivos como los falsos negativos.

2. **Tasa de detección de fraudes (Recall):**
   El modelo debe maximizar la detección de fraudes, reduciendo al mínimo los casos no detectados. Un Recall elevado es crucial para proteger tanto a la empresa como a los clientes de pérdidas financieras debido a fraudes no detectados.

3. **Exactitud (Accuracy):**
   La exactitud refleja la proporción de predicciones correctas realizadas por el modelo sobre todas las transacciones evaluadas. Esta métrica permite conocer el desempeño general del modelo y su capacidad para identificar correctamente tanto fraudes como transacciones legítimas.

4. **Puntuación F1 (F1 Score):**
   La puntuación F1 es la media armónica entre precisión y recuperación, especialmente útil en el caso de datos desbalanceados. Un F1 alto asegura que el modelo mantenga un buen balance entre identificar fraudes (recuperación) y minimizar los falsos positivos (precisión).

5. **Área bajo la curva ROC (AUC-ROC):**
   El AUC de la curva ROC es otro indicador importante para evaluar la capacidad del modelo de distinguir entre transacciones fraudulentas y legítimas, donde un valor elevado de AUC refleja una alta capacidad discriminativa del modelo.

6. **Eficiencia e integración en tiempo real:**
   El modelo debe integrarse en el flujo de transacciones en tiempo real, procesando los grandes volúmenes de datos que se manejan de forma rápida y eficiente, sin generar demoras que afecten la experiencia del cliente. Además, debe ser flexible y saber adaptarse a posibles nuevos patrones de fraude.

7. **Tiempo de entrenamiento y clasificación:**
   El tiempo de entrenamiento y el tiempo de clasificación son cruciales para la operatividad del modelo. Un tiempo de entrenamiento eficiente permite que el modelo se actualice rápidamente cuando se identifiquen nuevos patrones de fraude. Un tiempo de clasificación bajo asegura que las transacciones se procesen rápidamente sin afectar la experiencia del usuario.

Solución Propuesta

En esta sección se presenta la solución planteada para abordar el problema de la detección de fraudes. Se describen los requerimientos funcionales necesarios para que el sistema sea efectivo y las técnicas y modelos específicos que se utilizarán para su implementación.

Requerimientos Funcionales

Para la efectividad y eficiencia del modelo se deben cumplir una serie de requerimientos funcionales, los cuales aseguran que este no sólo detecte fraudes con precisión, sino que también se integre de forma adecuada en el entorno de producción y se mantenga un rendimiento óptimo a lo largo del tiempo.

1. **Automatización de la detección:**
   El sistema debe ser capaz de identificar fraudes en tiempo real sin intervención humana.

2. **Acceso y procesamiento de datos:**
   Se deben poder cargar y procesar grandes volúmenes de datos.

3. **Evaluación con métricas adecuadas:**
   Dado que los datos están desbalanceados, se recomienda el uso de la métrica AUPRC. De todos modos, se van a utilizar otra serie de métricas para comparar resultados.

4. **Monitoreo continuo:**
   El sistema debe ser monitoreado y ajustado dependiendo de las necesidades que vayan surgiendo, adaptándose a nuevos patrones de fraude y a cambios en los datos.

Técnicas y Modelos de la solución

El sistema de detección de fraudes requiere el uso de técnicas y modelos de machine learning que puedan manejar clases desbalanceadas y además asegurar un alto rendimiento.

1. **Modelos:**
   La bibliografía ([1], [2] y [3]) recomienda utilizar modelos robustos, tales como: *Random Forest, Gradient Boosting* y *Redes Neuronales*, ya que estos han demostrado ser eficaces en la detección de fraudes. Aún así, para comparar resultados se utilizarán otros modelos.

2. **Métricas:**
   Debido al contexto de desbalance de datos es que se utiliza la métrica AUPRC, ya que esta refleja mejor el balance entre la detección de fraudes y la minimización de falsos positivos. También se van a utilizar las más clásicas: **Recall**, **Accuracy**, **F1 Score**, **AUC-ROC**.

Planificación

Es esencial, para garantizar una correcta implementación del modelo, establecer un plan de trabajo claro que defina el enfoque, la metodología y los roles del equipo.

Enfoque de Trabajo

El enfoque va a ser iterativo, lo que permite ajustes continuos basados en el rendimiento del modelo y las necesidades que vayan surgiendo. Esto facilita la adaptación a nuevos patrones de fraude y a posibles cambios en el flujo de datos.

Metodología

Se aplicará una metodología ágil, con iteraciones cortas y reuniones regulares de seguimiento para evaluar avances y priorizar tareas. Esto garantiza flexibilidad en la toma de decisiones y permite al equipo responder a imprevistos en el análisis de datos o resultados de los modelos.

Roles

1. **Analista de datos:**
   Encargado del proceso de preparación y exploración del conjunto de datos. Esto incluye la limpieza, el procesamiento y gestión del desbalance de clases.

2. **Desarrollador de modelos:**
   Se encarga del desarrollo, entrenamiento y evaluación de los modelos de machine learning.

Análisis Exploratorio de los Datos

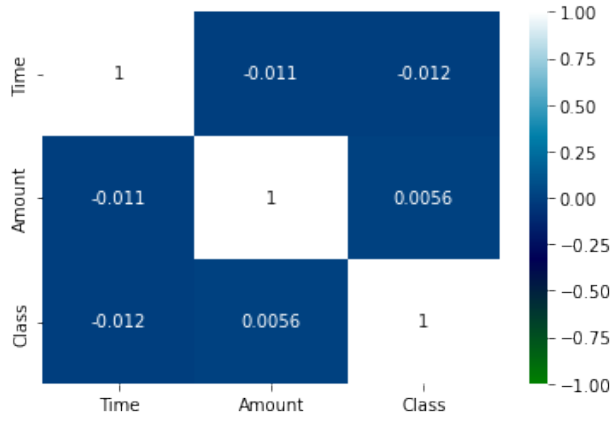
Algunos detalles del dataset:

- El Dataset contiene transacciones de tarjetas de créditos en Europa en el mes de septiembre del 2013.
- Dada la confidencialidad de los datos, es que no se tiene el background de todas las variables. Las 28 características en las cuales no se presenta información se obtuvieron mediante PCA (Análisis de Componentes Principales), es decir, los datos están transformados.
- Las componentes que no están transformadas son 'Time' y 'Amount'. La primera contiene los segundos transcurridos entre todas las transacciones y la primera transacción del dataset y la segunda es el monto de la transacción. Finalmente, se tiene la clase 'Class', el cual es un 1 si es que es un fraude y 0 si no lo es.

Aspectos más interesantes

Mapa de Calor

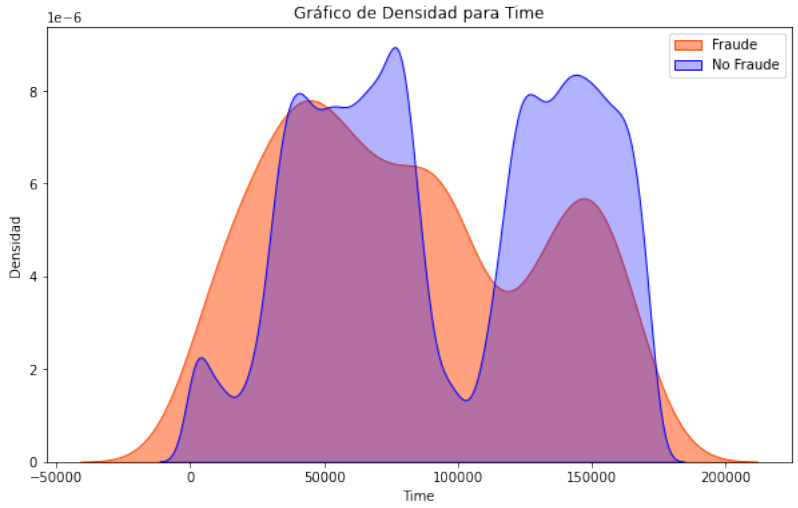
Debido al desconocimiento de la mayoría de las covariables es que se decidió hacer estadística con las columnas que si se conocen: 'Time', 'Amount' y 'Class'. Lo más relevante es el siguiente mapa de calor:

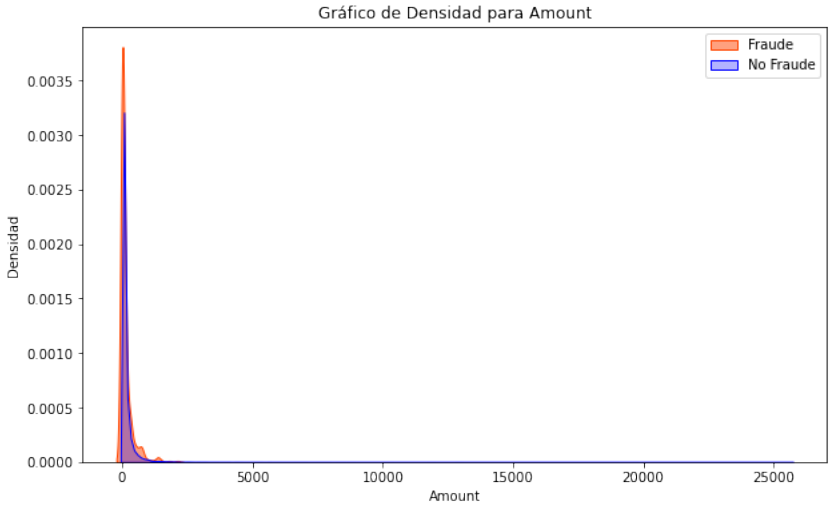


De esto se concluye la prácticamente nula relación entre las variables conocidas.

Distribución de los datos

Al haber dos clases (Fraude y No Fraude) es que se decidió separar los datos, ver cómo se distribuyen y comparar estas distribuciones entre clases.

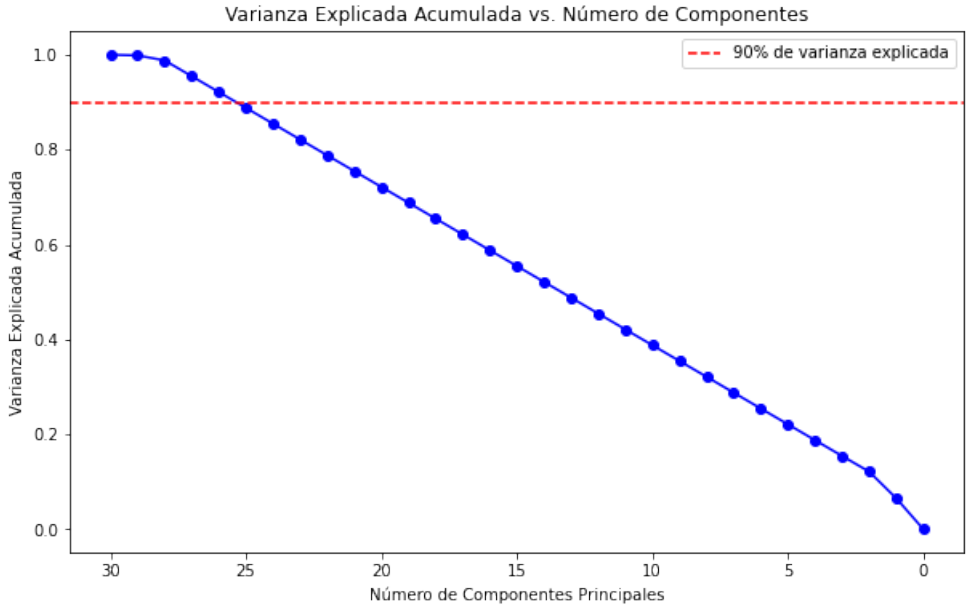




En el gráfico de distribución del tiempo, y especialmente para los datos no fraudulentos, resulta complicado ajustar la curva a alguna distribución conocida. Para los datos fraudulentos, podría decirse que es algo más uniforme, aunque eso aún podría ser exagerado. Sin embargo, ambas curvas presentan una forma similar, con la diferencia de que varían en la intensidad de los valles y picos.
Ahora, en el caso del monto, se puede ver una semejanza para ....
El mismo proceso se hizo con las covariables desconocidas, pero de esto es más difícil (o imposible) concluir cosas, ya que se no se sabe el contexto.

PCA

Para la reducción de dimensionalidad, para eliminar redundancia e inclusive para facilitar la visualización, es que se aplica el Análisis de Componentes Principales (PCA). La idea es identificar un valor para 'n', donde este representa la cantidad de componentes, tal que la suma acumulada de la varianza explicada sea mayor al 90 %. Utilizando librerías de python como **sklearn.decomposition**, **sklearn.preprocessing** y **numpy** es que se llegó a lo siguiente:



De donde se puede notar que el umbral de 90 % antes mencionado se cruza con 25 componentes. Es por esto que los modelos se van a trabajar con 26 componentes, lo cual implica un 92,14 % de varianza acumulada explicada, que resulta suficiente para conservar la mayoría de la información relevante y además obtener un conjunto de menos dimensionalidad.

Link a un notebook

Se adjunta un link de Github, en el cuál se encuentran los avances del informe, las presentaciones y los códigos de este proyecto: Proyecto Estadístico - MAT306

Experimentos

Definición de particiones de los datos

El dataset se particiona en una razón 80:20, en donde el 80 % será el conjunto de entrenamiento, mientras que el 20 % servirá como conjunto de prueba.

Definición de métricas

Área bajo la curva Precision-Recall (AUPRC)

El Área bajo la curva Precision-Recall (AUPRC) es una métrica adecuada para problemas con clases desbalanceadas. Captura el equilibrio entre precisión y recall para distintos umbrales de probabilidad.

$$\text{AUPRC} = \int_0^1 \text{Precision}(\text{Recall})\, d(\text{Recall})$$

Puntuación F1 (F1 Score)

La puntuación F1 es la media armónica entre precisión y recall. Se define como:

$$F1 = 2 \times \frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}}$$

donde:

$$\text{Precisión} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

donde:

- TP: Verdaderos Positivos
- FP: Falsos Positivos
- FN: Falsos Negativos

Área bajo la curva ROC (AUC-ROC)

El AUC-ROC mide la capacidad del modelo para distinguir entre clases. Se calcula integrando la curva ROC, que representa la Tasa de Verdaderos Positivos (TPR) frente a la Tasa de Falsos Positivos (FPR).

$$\text{AUC-ROC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR})$$

donde:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

Exactitud (Accuracy)

La exactitud es la proporción de predicciones correctas sobre el total de predicciones:

$$\text{Exactitud} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

donde:

- TP: Verdaderos Positivos
- TN: Verdaderos Negativos
- FP: Falsos Positivos
- FN: Falsos Negativos

Tiempo de Clasificación

El tiempo de clasificación mide el tiempo requerido para realizar una predicción. En aplicaciones en tiempo real, es crucial minimizar este tiempo para mejorar la experiencia del usuario.

Definición de baseline

Modelo Naive Bayes

El modelo de **Naive Bayes** es un clasificador probabilístico basado en el *teorema de Bayes*. Se llama 'naive' (ingenuo), porque asume que las características son independientes entre sí, una simplificación que rara vez es cierta en la práctica, pero que permite obtener resultados efectivos en muchos casos.

Formalmente, el modelo Naive Bayes estima la probabilidad de una clase C dado un conjunto de características $X = (x_1, x_2, \ldots, x_n)$ mediante el teorema de Bayes:

$$P(C \mid X) = \frac{P(X \mid C)\, P(C)}{P(X)}$$

donde:

- $P(C \mid X)$: Probabilidad posterior de la clase C dado el conjunto de características X.
- $P(X \mid C)$: Verosimilitud, o la probabilidad de observar X dada la clase C.
- $P(C)$: Probabilidad a priori de la clase C.
- $P(X)$: Probabilidad de observar el conjunto de características X en general.

Dado que $P(X)$ es constante para todas las clases, el cálculo se simplifica a:

$$P(C \mid X) \propto P(X \mid C)\, P(C)$$

Para la clasificación, el modelo asigna la clase con la mayor probabilidad posterior:

$$\hat{C} = \arg\max_{C} P(C \mid X)$$

En el contexto del problema, y evaluando en las métricas antes mencionadas, se obtuvieron los siguientes valores:

Métrica	**Valor**
AUPRC	0.0820
Exactitud	0.9797
Puntuación F1	0.1201
AUC	0.9568
Tiempo de entrenamiento (segundos)	0.0977
Tiempo de clasificación (segundos)	0.0289

Cuadro 1: Resultados del modelo Naive Bayes

Modelos

Definiciones

Gradient Boosting

Gradient Boosting es un método de ensamble que construye un modelo fuerte a partir de varios modelos débiles (generalmente árboles de decisión). En cada iteración, el modelo intenta corregir los errores del modelo anterior, ajustando los errores residuales. Formalmente, el modelo final se expresa como una suma de modelos individuales $f_m(x)$, en donde cada uno es un estimador del residuo de la iteración anterior:

$$F_M(x) = \sum_{m=1}^{M} f_m(x)$$

El objetivo es minimizar la función de pérdida en cada paso mediante el gradiente descendente.

Redes Neuronales

Las **Redes Neuronales** son modelos de aprendizaje profundo inspirados en el cerebro humano. Están compuestas de capas de nodos (neuronas) conectadas entre sí. Cada conexión tiene un peso asociado que se ajusta en el entrenamiento. Formalmente, una red neuronal produce una salida y dada una entrada X mediante una función no lineal de la forma:

$$y = f(W \cdot X + b)$$

donde W representa los pesos, X la entrada y b el sesgo. Las redes se entrenan mediante retropropagación para ajustar los pesos y minimizar el error.

Random Forest

Random Forest es un modelo de ensamble basado en múltiples árboles de decisión, cada uno entrenado en diferentes subconjuntos aleatorios de los datos y características. La predicción final se obtiene combinando las predicciones de cada árbol mediante votación (clasificación) o promedio (regresión):

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} f_t(x)$$

donde $f_t(x)$ representa la predicción de cada árbol t y T es el número total de árboles.

Árbol de Decisión (Decision Tree)

Un **Árbol de Decisión** es un modelo de clasificación y regresión que divide el espacio de características en regiones homogéneas basándose en un conjunto de condiciones lógicas. Cada nodo representa una característica, y cada rama representa una decisión basada en los valores de esa característica. El objetivo es minimizar la impureza en cada partición, medida por métricas como el *Gini* o la *Entropía*.

$$\text{Impureza} = \sum_{i=1}^{C} -p_i \log(p_i)$$

donde p_i es la proporción de muestras de la clase i en el nodo.

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) es un algoritmo de clasificación basado en la proximidad. Para clasificar una muestra, el modelo encuentra los k vecinos más cercanos en el espacio de características y asigna la clase más común entre ellos. La proximidad se calcula usando una métrica de distancia, como la distancia Euclidiana:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2}$$

donde x_i y x_j son vectores de características.

Quadratic Discriminant Analysis (QDA)

Quadratic Discriminant Analysis (QDA) es un clasificador lineal que asume que los datos siguen una distribución normal, pero a diferencia de LDA, permite que cada clase tenga su propia matriz de covarianza. El modelo clasifica los datos calculando la probabilidad de que una muestra pertenezca a una clase específica, basándose en la función discriminante cuadrática:

$$\delta_k(x) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \log P(C_k)$$

donde Σ_k es la matriz de covarianza de la clase k, μ_k es la media y $P(C_k)$ es la probabilidad a priori de la clase k.

Objetivos de cada modelo

No se define un objetivo específico para cada modelo de manera individual; en cambio, el propósito es evaluar el rendimiento de todos los modelos considerados y seleccionar el mejor de acuerdo con las métricas establecidas. De este modo, el objetivo general es identificar el modelo que ofrezca el desempeño más alto según los criterios de evaluación definidos.

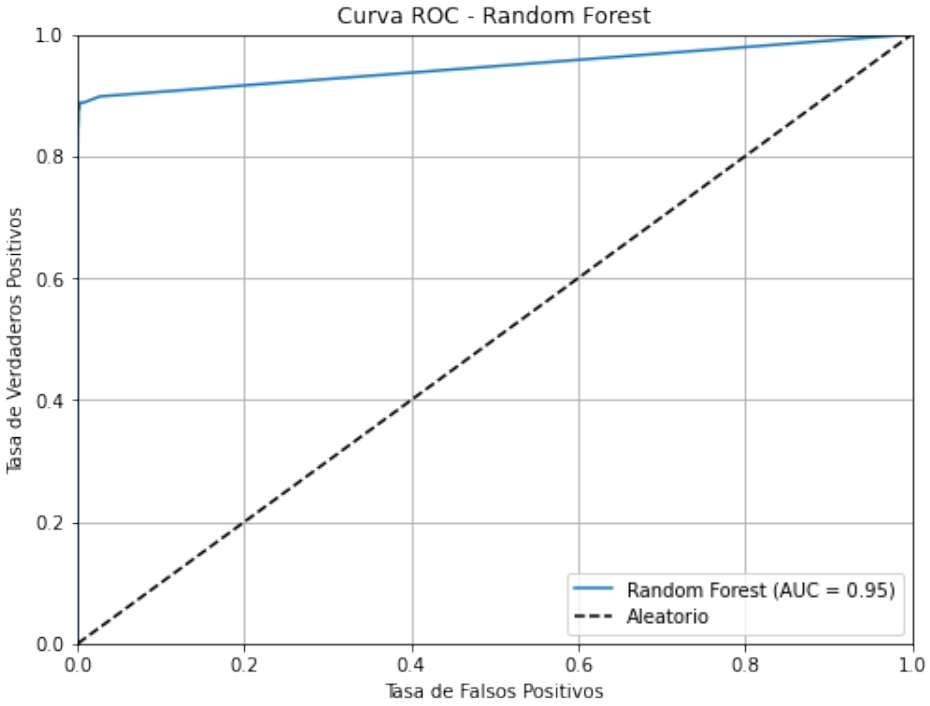
Resultados Actuales

Se presentan los resultados obtenidos separados por modelos:

Random Forest

Métrica	**Valor**
AUPRC	0.8587
Exactitud	0.9995
Puntuación F1	0.8391
AUC	0.9474
Tiempo de entrenamiento (segundos)	178.3891
Tiempo de clasificación (segundos)	0.4508

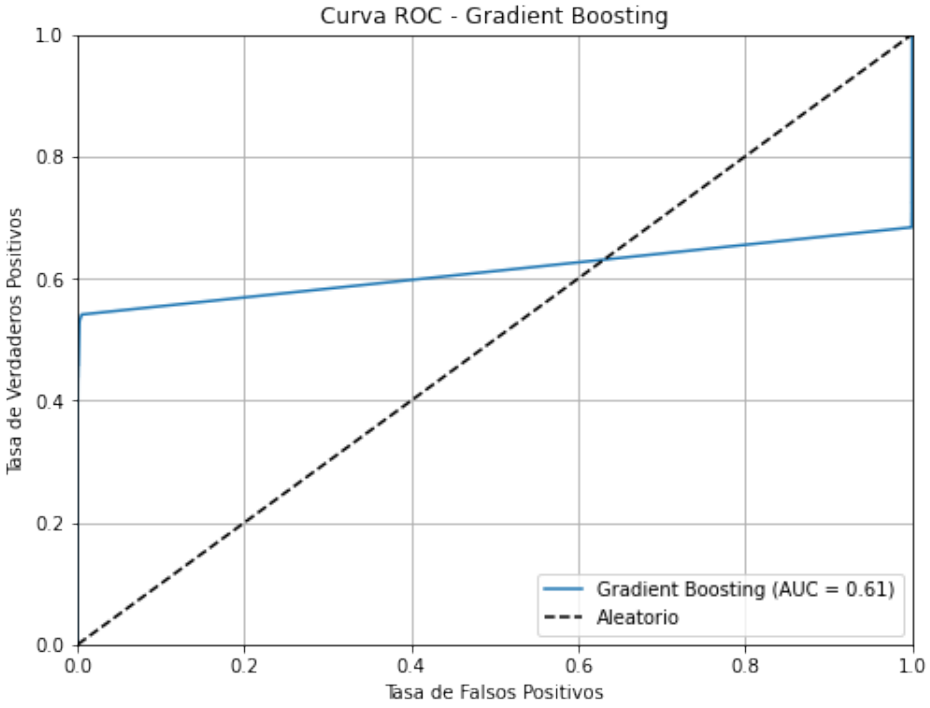
Cuadro 2: Resultados Random Forest



Gradient Boosting

Métrica	**Valor**
AUPRC	0.4131
Exactitud	0.9988
Puntuación F1	0.5352
AUC	0.6118
Tiempo de entrenamiento (segundos)	234.5868
Tiempo de clasificación (segundos)	0.0529

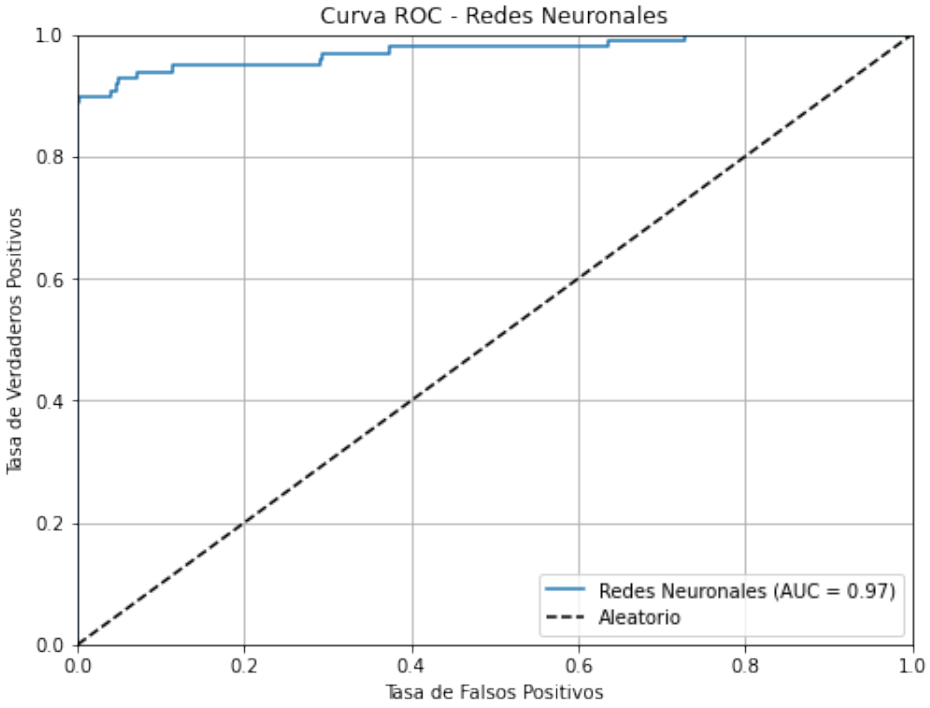
Cuadro 3: Resultados Gradient Boosting



Redes Neuronales

Métrica	**Valor**
AUPRC	0.8706
Exactitud	0.9995
Puntuación F1	0.8333
AUC	0.9730
Tiempo de entrenamiento (segundos)	21.6481
Tiempo de clasificación (segundos)	0.0469

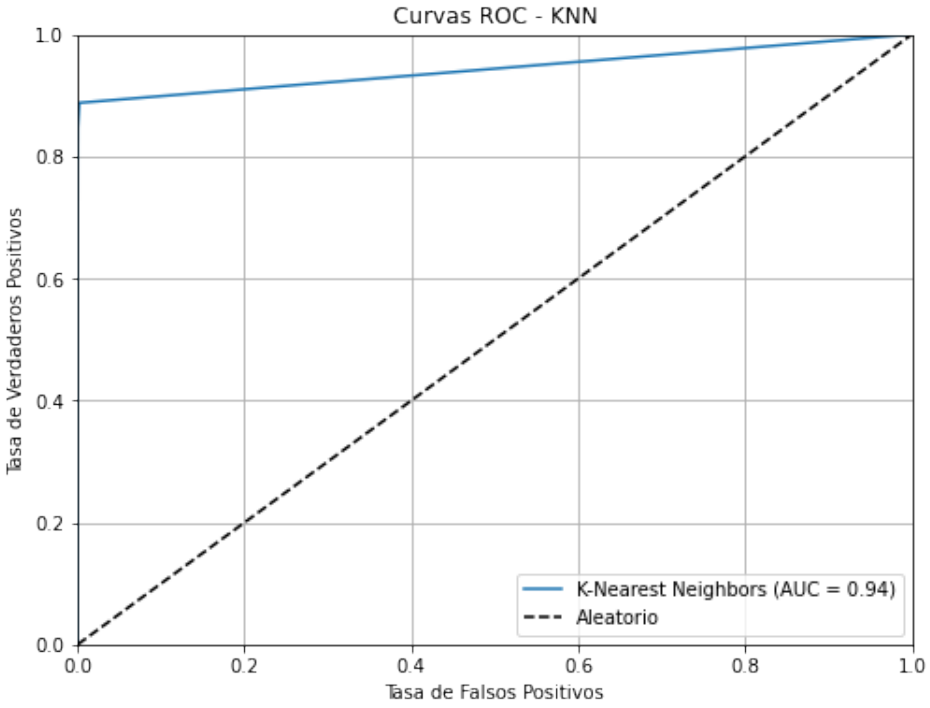
Cuadro 4: Resultados Redes Neuronales



KNN

Métrica	Valor
AUPRC	0.8280
Exactitud	0.9994
Puntuación F1	0.7882
AUC	0.9437
Tiempo de entrenamiento (segundos)	0.0189
Tiempo de clasificación (segundos)	193.2743

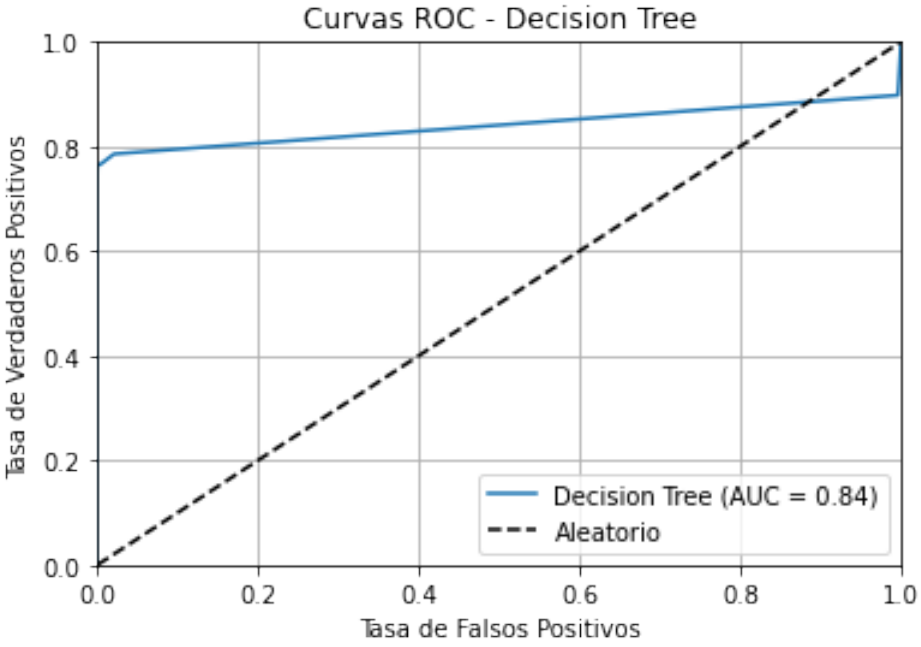
Cuadro 5: Resultados K-Nearest Neighbors (KNN)



Decision Tree

Métrica	Valor
AUPRC	0.7094
Exactitud	0.9995
Puntuación F1	0.8249
AUC	0.8408
Tiempo de entrenamiento (segundos)	7.4451
Tiempo de clasificación (segundos)	0.0060

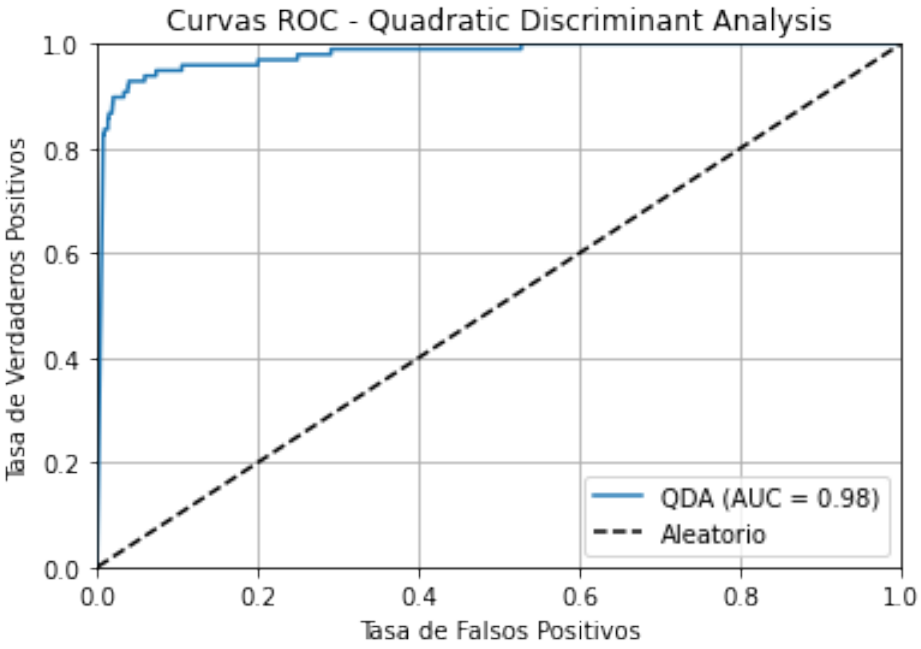
Cuadro 6: Resultados Decision Tree



Quadratic Discriminant Analysis

Métrica	Valor
AUPRC	0.1419
Exactitud	0.9743
Puntuación F1	0.1074
AUC	0.9793
Tiempo de entrenamiento (segundos)	0.2922
Tiempo de clasificación (segundos)	0.0259

Cuadro 7: Resultados Quadratic Discriminant Analysis (QDA)



Análisis de Resultados

Al observar los resultados se puede concluir lo siguiente:

AUPRC

- **Redes Neuronales:** Tiene el AUPRC más alto (0.8706), esto indica que es el mejor en términos de equilibrio entre precisión y recall para las clases desbalanceadas.
- **QDA:** Tiene el AUPRC más bajo (0.1419), esto nos dice que tiene un rendimiento deficiente en este aspecto, lo cual podría reflejar un mal manejo de las clases desbalanceadas.

Exactitud

- Todos los modelos tienen una exactitud por encima de 0.998, excepto para **QDA** (0.9743). Esto puede ser engañoso si el conjunto de datos está muy desbalanceado, ya que una exactitud alta no necesariamente refleja una buena detección de la clase minoritaria (fraudes).

Puntuación F1

- **Redes Neuronales:** Tiene una puntuación F1 alta (0.8333), esto indica un buen equilibrio entre precisión y recall.
- **QDA:** Otra vez se destaca negativamente con un F1 bajo (0.1074), lo que refleja un mal rendimiento en cuanto a la capacidad de detectar correctamente los fraudes.

AUC

- **Redes Neuronales:** Tiene el AUC más alto (0.9730), lo que indica que tiene una excelente capacidad para distinguir entre las clases.
- **QDA:** Tiene el AUC más alto (0.9793), indicando que es muy eficaz en separar las clases, pero no hay que dejarse llevar por esto, ya que AUC no siempre refleja el rendimiento real cuando las clases están desbalanceadas.

Conclusiones Generales

- **Redes Neuronales:** Mostró ser el modelo más equilibrado en cuanto a las métricas AUPRC, Exactitud, F1 y AUC. Su rendimiento es consistentemente alto, lo que lo convierte en una opción fuerte para la detección de fraudes.
- **QDA:** Tiene el peor desempeño en términos de AUPRC, Exactitud y F1, aunque su AUC es alto. Esto sugiere que el modelo tiene dificultades para manejar clases desbalanceadas, lo que es crucial en tareas de detección de fraudes.
- **KNN:** En términos de Exactitud y AUPRC muestra un buen desempeño, pero tiene un tiempo de clasificación significativamente más alto en comparación con otros modelos.
- **Decision Tree:** Muestra un rendimiento aceptable en la mayoría de las métricas, pero no destaca en ninguna de ellas de manera significativa.
- **Gradient Boosting:** Tiene un AUPRC más bajo que Redes Neuronales y KNN, lo que sugiere que no es tan efectivo en términos de precisión y recall en este caso específico.

Por lo tanto, se recomienda probar modelos como **Redes Neuronales** para obtener el mejor rendimiento general, aunque otros modelos como **KNN** o **Random Forest** pueden ser útiles dependiendo de los requisitos de tiempo de clasificación y eficiencia.

Bibliografía

[1] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[2] Simon Rogers y Mark Girolami. *A first course in machine learning*. Chapman y Hall/CRC, 2016.

[3] Stuart J Russell y Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, 2016.