# Workshop G2 TP3

Alvarez, Martha
Roa, Fernando
Tasat, Kevin

# Objetivo

**Entrenar** un **set de modelos** que puedan **predecir transacciones fraudulentas** partir del dataset de Credit Card de Kaggle el cual contiene transacciones hechas con tarjetas de crédito en Septiembre 2013 por europeos.

# Metodología

Dado que la data se encuentra completa y limpia, y las features anonimizadas, **empezaremos con un breve análisis exploratorio y selección de variables.**

En seguida, entrenaremos los siguientes tipos de modelo: **Regresión Logística, Naive Bayes Gaussian, KNN, árbol de decisión y Random Forest.**

**Compararemos los resultados** de los distintos modelos, seguido de la selección del mejor modelo y selección hiper-parámetros para obtener predicciones y métricas finales.

# Presentación del Dataset

# Datos Principales

**Shape** 284.807 filas x 31 Columnas

**Features** 'Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10','V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20', 'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount'.

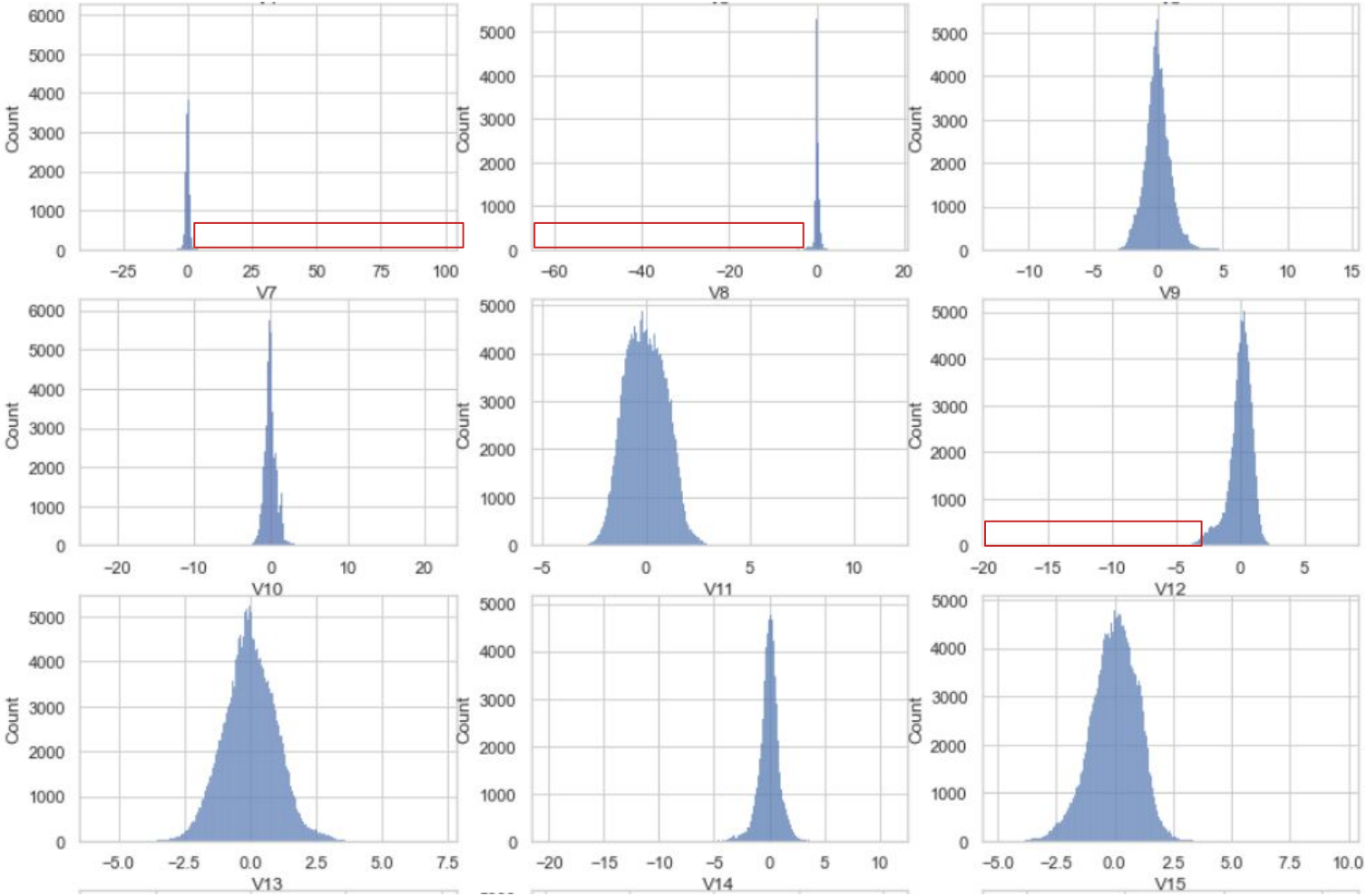
**Target** 'Class'

**Balance Clases**

| | | |
|---|---|---|
| 0 | 284.315 | 99.83% |
| 1 | 492 | 0.17% |

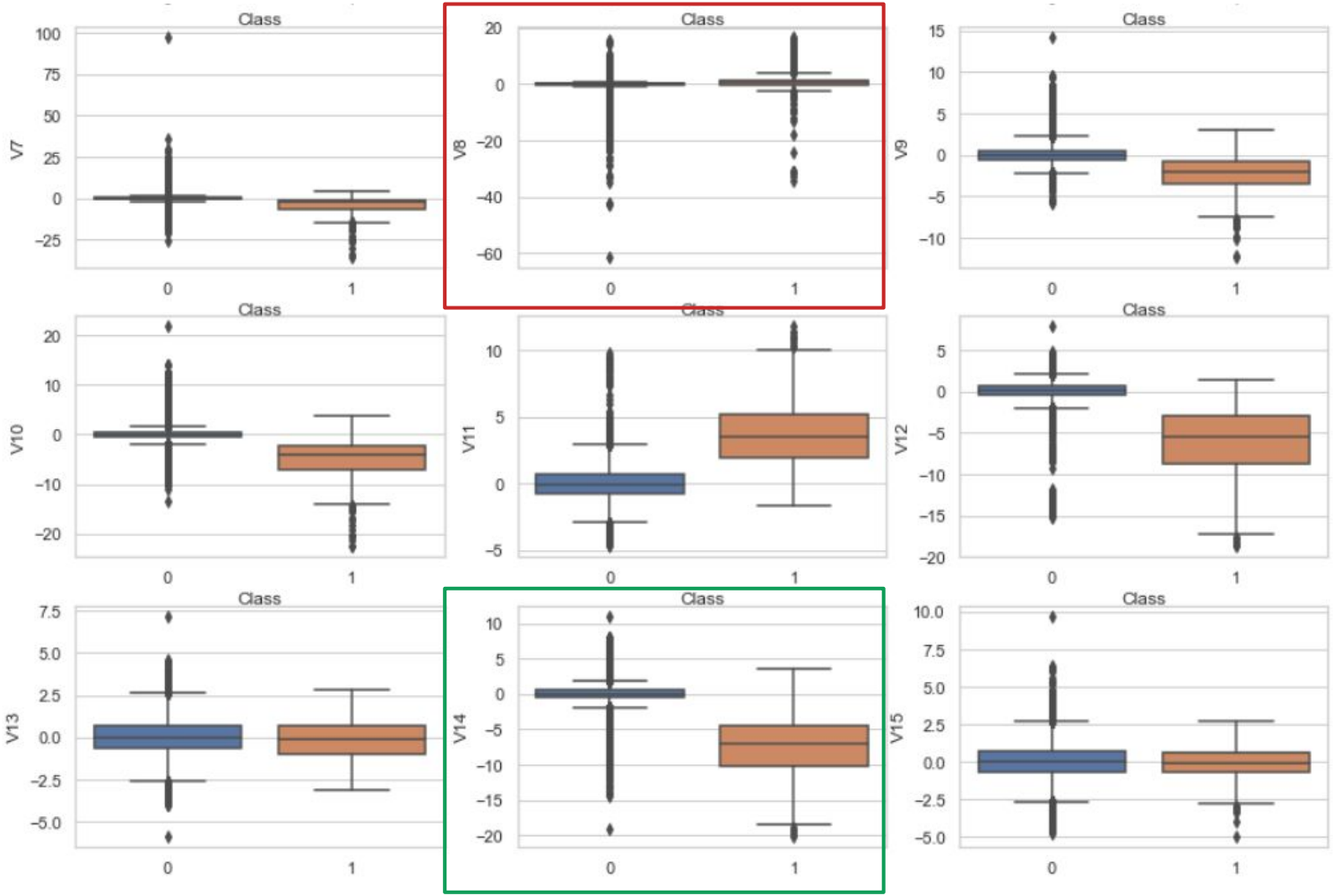
*0= Bueno , 1= Fraude*

**H0) Accuracy Base = 99.83%**




Como vemos **tenemos presencia de valores extremos** para distintas variables.

# Selección de Variables




**Estudiamos** variables de acuerdo a su **distribución por clase** para empezar a **detectar posibles features relevantes**.

**Problema**: Los **valores extremos** pueden evitar visualizar diferencias significativas entre ciertas variables.

# Regresión Logística con Stat Models

Results: Logit

| | | | |
|---|---|---|---|
| Model: | Logit | Pseudo R-squared: | 0.709 |
| Dependent Variable: | Class | AIC: | 1638.8927 |
| Date: | 2022-04-24 13:06 | BIC: | 1947.0493 |
| No. Observations: | 213605 | Log-Likelihood: | -789.45 |
| Df Model: | 29 | LL-Null: | -2715.9 |
| Df Residuals: | 213575 | LLR p-value: | 0.0000 |
| Converged: | 1.0000 | Scale: | 1.0000 |
| No. Iterations: | 16.0000 | | |

| | Coef. | Std.Err. | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 48.5657 | 19.9253 | 2.4374 | 0.0148 | 9.5129 | 87.6185 |
| V1 | 4.6347 | 2.9022 | 1.5970 | 0.1103 | -1.0534 | 10.3229 |
| V2 | 5.2127 | 7.3455 | 0.7096 | 0.4779 | -9.1842 | 19.6095 |
| V3 | 0.7794 | 3.0753 | 0.2534 | 0.7999 | -5.2480 | 6.8068 |
| V4 | 18.6266 | 2.3880 | 7.8001 | 0.0000 | 13.9462 | 23.3071 |

* Variables significativas:
Las variables **V4,V8,V10,V13,V14,V20,V21,V22,V27 son significativas con un nivel de confianza del 95%**.
* Se obtuvo un **Pseudo R-squared de 0.709**, valor que es muy bueno y nos indica un buen poder de predicción de la variable objetivo.

# Regresión Logística - Regularización Lasso L1

| | variable | coeficiente |
|---|---|---|
| **0** | V1 | 1.26745 |
| **3** | V4 | 13.06847 |
| **4** | V5 | 4.60432 |
| **9** | V10 | -13.16905 |
| **10** | V11 | 2.16794 |
| **11** | V12 | -2.17481 |
| **12** | V13 | -3.82533 |
| **13** | V14 | -19.48994 |
| **15** | V16 | -5.93675 |
| **16** | V17 | -0.07415 |
| **20** | V21 | 9.11522 |
| **21** | V22 | 6.40620 |

Los coeficientes de las demás variables fueron 0.

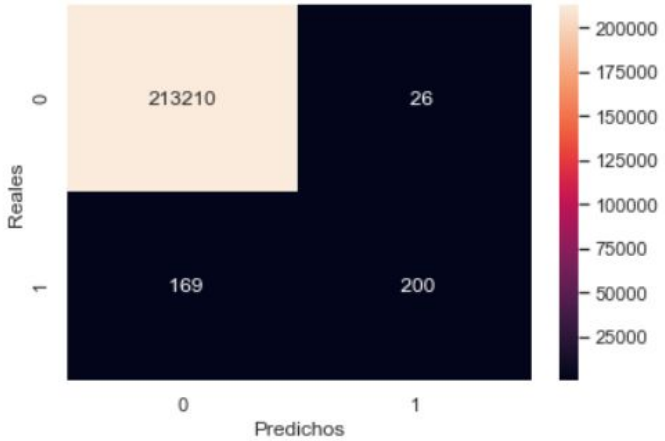
# Variables predictoras para modelar

Tendremos en cuenta la unión de los conjuntos de variables que fueron significativas en el modelo de stat models y aquellas con coeficiente diferente de 0 en la regresión logística con regularización lasso -l1. En total utilizaremos 15 variables predictoras.

**variables =**
**['V1','V4','V5','V8','V10','V11','V12','V13','V14','V16','V17','V20','V21','V22','V27']**

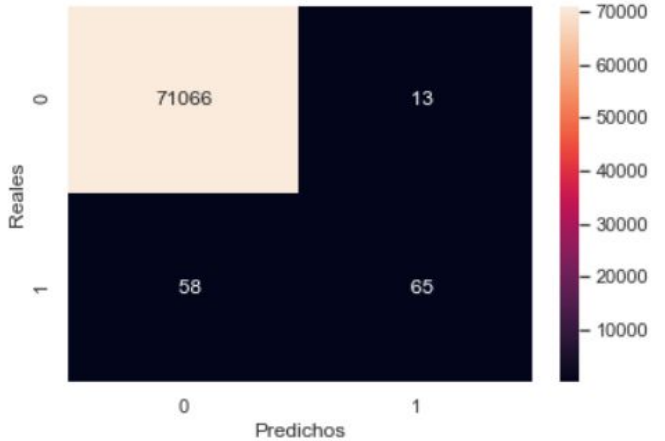
# Modelo: Regresión Logística

## Matriz de Confusión Train




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9992 | 0.9999 | 0.9995 | 213236 |
| 1 | 0.8850 | 0.5420 | 0.6723 | 369 |
| accuracy | | | 0.9991 | 213605 |
| macro avg | 0.9421 | 0.7709 | 0.8359 | 213605 |
| weighted avg | 0.9990 | 0.9991 | 0.9990 | 213605 |

## Matriz de Confusión Test

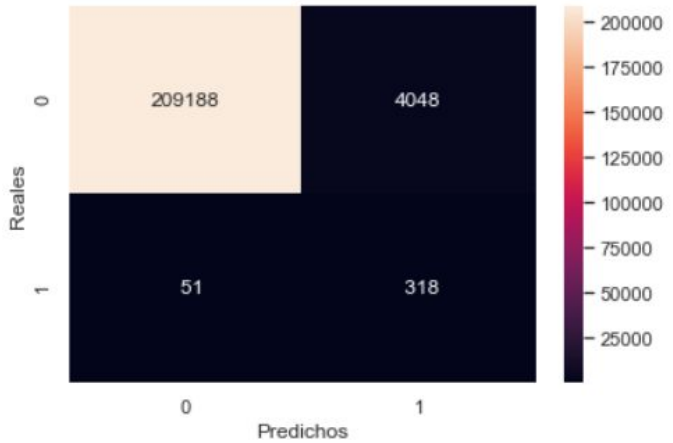



| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9992 | 0.9998 | 0.9995 | 71079 |
| 1 | 0.8333 | 0.5285 | 0.6468 | 123 |
| accuracy | | | 0.9990 | 71202 |
| macro avg | 0.9163 | 0.7641 | 0.8231 | 71202 |
| weighted avg | 0.9989 | 0.9990 | 0.9989 | 71202 |

Las métricas obtenidas sobre las predicciones en la data de entrenamiento y de prueba se comportan de forma similar, además el accuracy obtenido en ambos casos es mayor al obtenido por el modelo base de la hipótesis nula, el cual alcanza un valor de 0.9983 cuya predicción siempre será la clase 0.

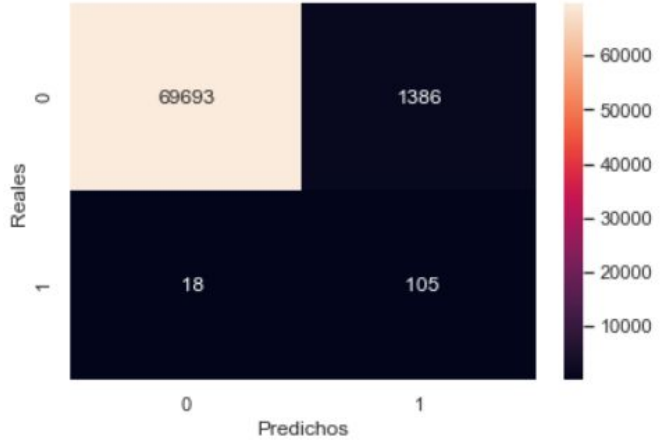
# Modelo:
# Naive Bayes

## Matriz de Confusión Train




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9998 | 0.9810 | 0.9903 | 213236 |
| 1 | 0.0728 | 0.8618 | 0.1343 | 369 |
| accuracy | | | 0.9808 | 213605 |
| macro avg | 0.5363 | 0.9214 | 0.5623 | 213605 |
| weighted avg | 0.9982 | 0.9808 | 0.9888 | 213605 |

## Matriz de Confusión Test

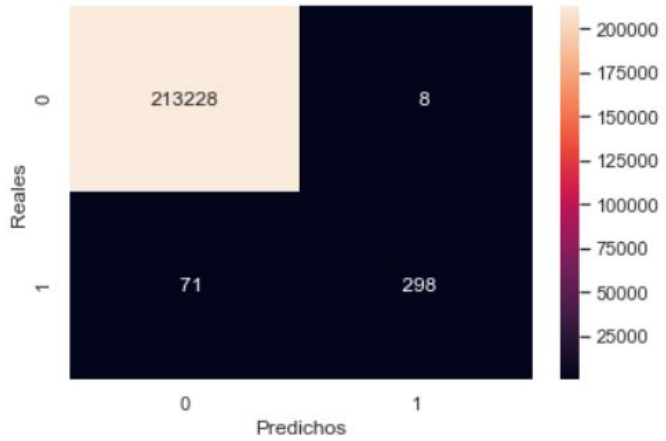



| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9997 | 0.9805 | 0.9900 | 71079 |
| 1 | 0.0704 | 0.8537 | 0.1301 | 123 |
| accuracy | | | 0.9803 | 71202 |
| macro avg | 0.5351 | 0.9171 | 0.5601 | 71202 |
| weighted avg | 0.9981 | 0.9803 | 0.9885 | 71202 |

Este modelo tiene un rendimiento menor que el modelo base lo cual se evidencia en un menor accuracy. A partir de las predicciones se obtiene una tasa de falsos positivos bastante alta y consecuentemente un precision score para la clase fraude - "1" de apenas 0.07. En conclusión por su rendimiento este modelo será descartado.

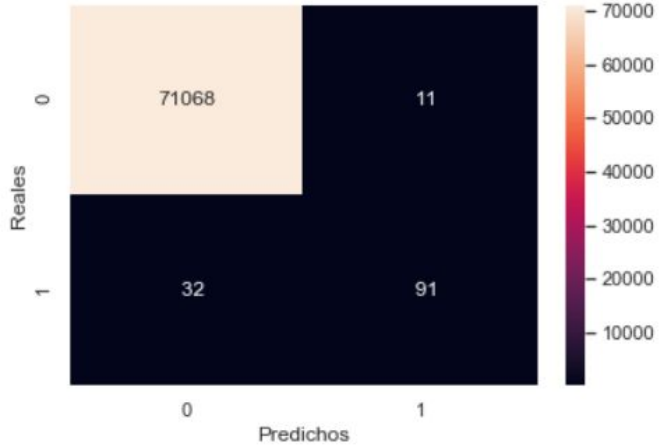
# Modelo:
# KNN

## Matriz de Confusión Train




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9997 | 1.0000 | 0.9998 | 213236 |
| 1 | 0.9739 | 0.8076 | 0.8830 | 369 |
| accuracy | | | 0.9996 | 213605 |
| macro avg | 0.9868 | 0.9038 | 0.9414 | 213605 |
| weighted avg | 0.9996 | 0.9996 | 0.9996 | 213605 |

## Matriz de Confusión Test

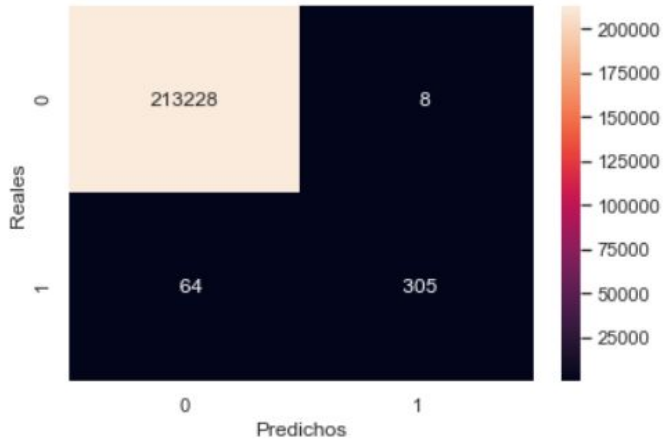



| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9995 | 0.9998 | 0.9997 | 71079 |
| 1 | 0.8922 | 0.7398 | 0.8089 | 123 |
| accuracy | | | 0.9994 | 71202 |
| macro avg | 0.9459 | 0.8698 | 0.9043 | 71202 |
| weighted avg | 0.9994 | 0.9994 | 0.9994 | 71202 |

El modelo knn tiene un rendimiento superior respecto a H0. En términos de accuracy y f1 score. sobre la data de entrenamiento y de prueba alcanza valores de 0.9996 y 0.9994 respectivamente y a la vez valores de f1 score superiores a 0.8 para la clase fraude.

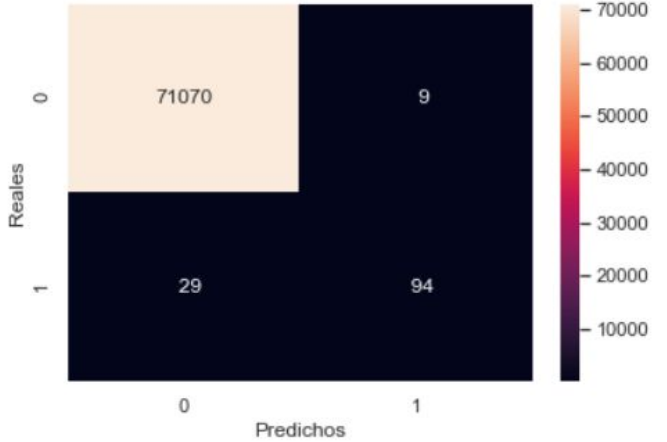
# Modelo: Decision Tree Classifier

## Matriz de Confusión Train




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9997 | 1.0000 | 0.9998 | 213236 |
| 1 | 0.9744 | 0.8266 | 0.8944 | 369 |
| accuracy | | | 0.9997 | 213605 |
| macro avg | 0.9871 | 0.9133 | 0.9471 | 213605 |
| weighted avg | 0.9997 | 0.9997 | 0.9996 | 213605 |

## Matriz de Confusión Test

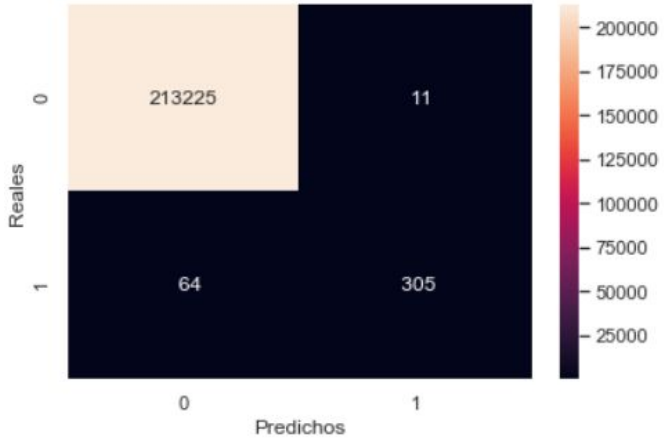



| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9996 | 0.9999 | 0.9997 | 71079 |
| 1 | 0.9126 | 0.7642 | 0.8319 | 123 |
| accuracy | | | 0.9995 | 71202 |
| macro avg | 0.9561 | 0.8821 | 0.9158 | 71202 |
| weighted avg | 0.9994 | 0.9995 | 0.9994 | 71202 |

El modelo de árbol de decisión nos arroja un rendimiento ligeramente superior al modelo de KNN y por consiguiente tiene resultados superiores a la hipótesis nula.

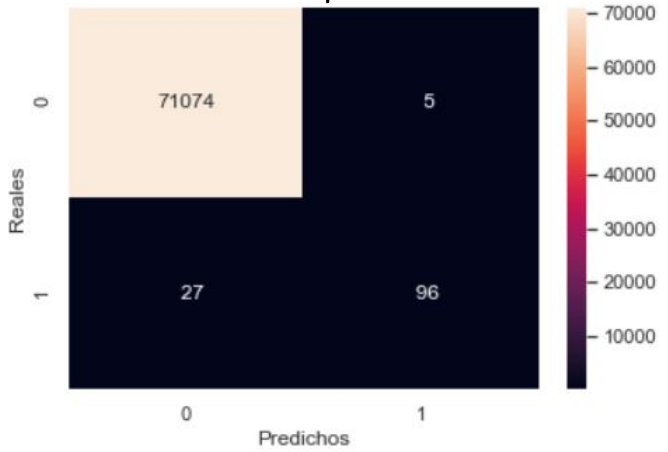
# Modelo:
# Random Forest Classifier

## Matriz de Confusión Train




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9997 | 0.9999 | 0.9998 | 213236 |
| 1 | 0.9652 | 0.8266 | 0.8905 | 369 |
| accuracy | | | 0.9996 | 213605 |
| macro avg | 0.9824 | 0.9133 | 0.9452 | 213605 |
| weighted avg | 0.9996 | 0.9996 | 0.9996 | 213605 |

## Matriz de Confusión Test

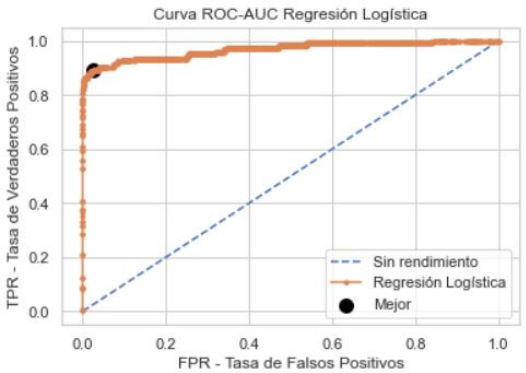



| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9996 | 0.9999 | 0.9998 | 71079 |
| 1 | 0.9505 | 0.7805 | 0.8571 | 123 |
| accuracy | | | 0.9996 | 71202 |
| macro avg | 0.9751 | 0.8902 | 0.9285 | 71202 |
| weighted avg | 0.9995 | 0.9996 | 0.9995 | 71202 |

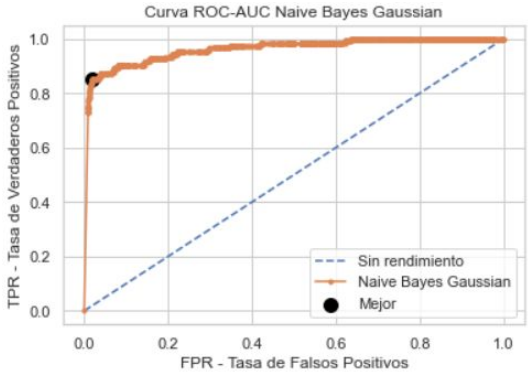
Los resultados obtenidos en el modelo random forest son muy buenos y son superiores a la hipótesis nula.

# Comparación de Modelos

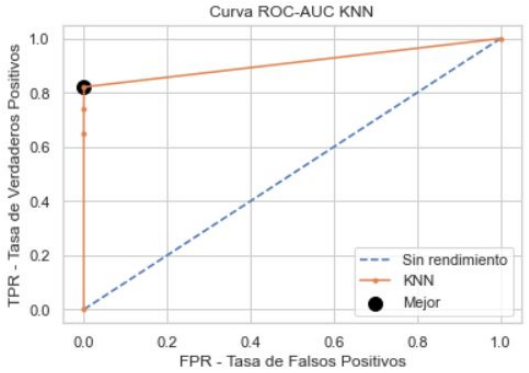
# Curva ROC-AUC



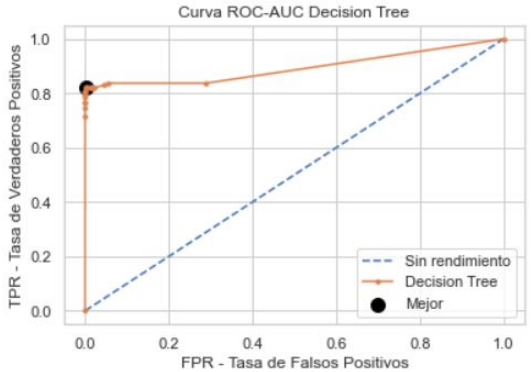

ROC AUC=0.969
Mejor threshold=0.002, G-Mean=0.934



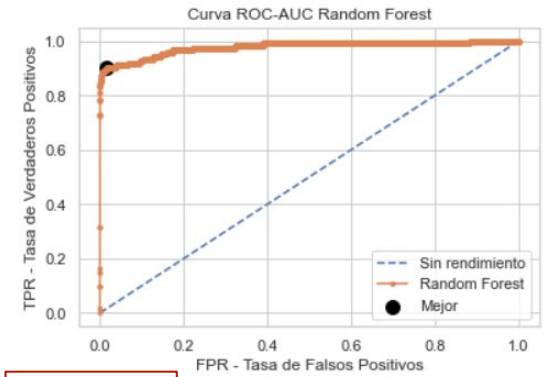

ROC AUC=0.962
Mejor threshold=0.512, G-Mean=0.915




ROC AUC=0.910
Mejor threshold=0.333, G-Mean=0.906




ROC AUC=0.894
Mejor threshold=0.064, G-Mean=0.906




ROC AUC=0.977
Mejor threshold=0.004, G-Mean=0.942

# Tabla Comparativa de Modelos

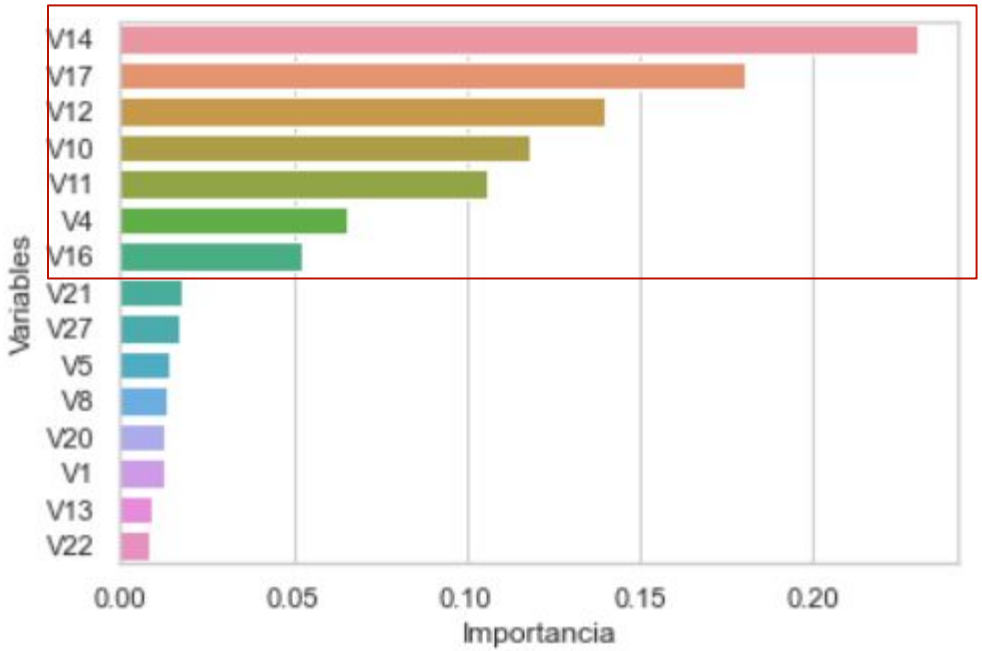
| | Modelo | Accuracy | Recall | Precision | F1 score | AUC |
|---|---|---|---|---|---|---|
| 0 | Regresión Logística | 0.99900 | 0.52846 | 0.83333 | 0.64677 | 0.96863 |
| 1 | Naive Bayes Gaussian | 0.98028 | 0.85366 | 0.07042 | 0.13011 | 0.96241 |
| 2 | KNN | 0.99940 | 0.73984 | 0.89216 | 0.80889 | 0.91047 |
| 3 | Decision Tree | 0.99947 | 0.76423 | 0.91262 | 0.83186 | 0.89446 |
| 4 | Random Forest | 0.99955 | 0.78049 | 0.95050 | 0.85714 | 0.97746 |

El Mejor modelo es el Random Forest, siendo superior en todas las métricas con solo una excepción respecto a los otros modelos (Recall). Sin embargo...

# Importancia de Variables Random Forest

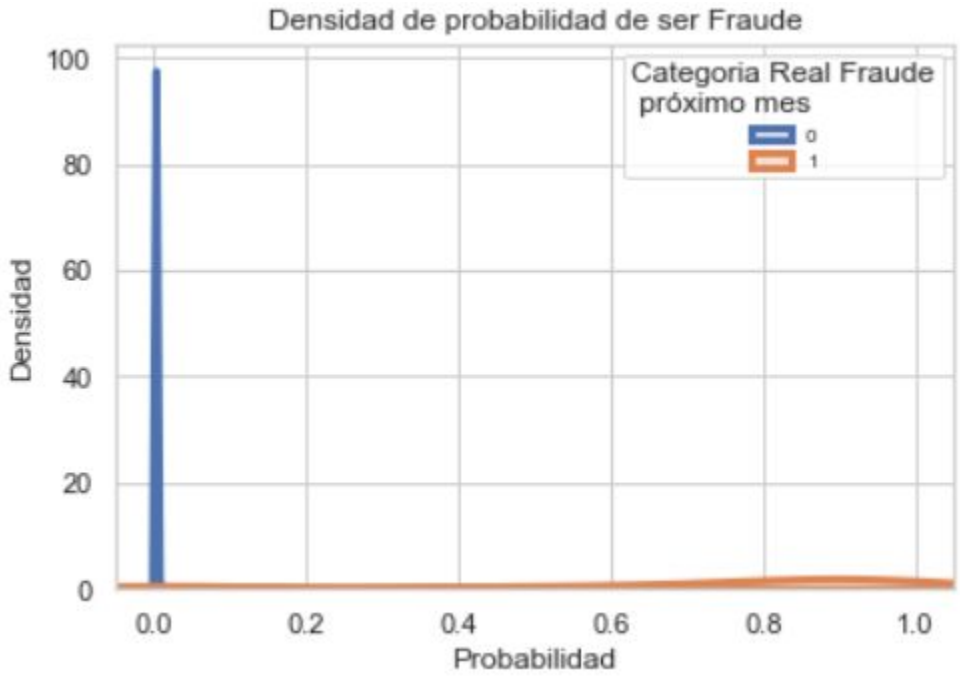
# Importancia de Variables

| | Variable | Importancia |
|---|---|---|
| 0 | V14 | 0.23036 |
| 1 | V17 | 0.18073 |
| 2 | V12 | 0.13980 |
| 3 | V10 | 0.11857 |
| 4 | V11 | 0.10606 |
| 5 | V4 | 0.06587 |
| 6 | V16 | 0.05274 |
| 7 | V21 | 0.01791 |
| 8 | V27 | 0.01720 |
| 9 | V5 | 0.01456 |
| 10 | V8 | 0.01355 |
| 11 | V20 | 0.01281 |
| 12 | V1 | 0.01264 |
| 13 | V13 | 0.00912 |
| 14 | V22 | 0.00808 |



# Mejor Threshold Random Forest

# Curva de densidad de probabilidades



# Comparativa de métricas variando Threshold

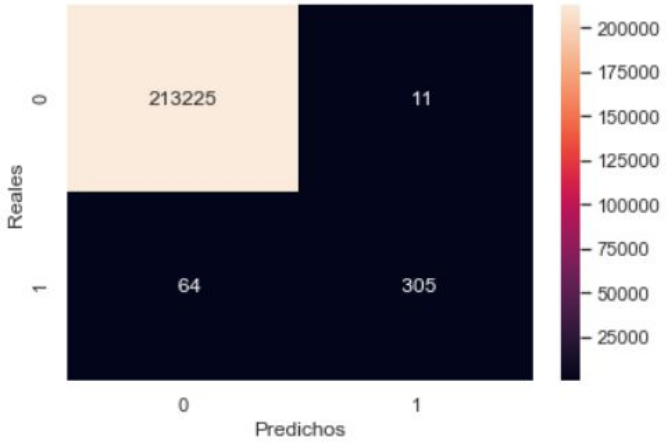
| | Threshold | Accuracy | Recall | Precision | F1 score |
|---|---|---|---|---|---|
| **0** | 0.05000 | 0.99878 | 0.85366 | 0.60345 | 0.70707 |
| **1** | 0.10000 | 0.99914 | 0.84553 | 0.71233 | 0.77323 |
| **2** | 0.15000 | 0.99927 | 0.84553 | 0.75912 | 0.80000 |
| **3** | 0.20000 | 0.99930 | 0.84553 | 0.77037 | 0.80620 |
| **4** | 0.25000 | 0.99938 | 0.84553 | 0.80620 | 0.82540 |
| **5** | 0.30000 | 0.99947 | 0.84553 | 0.84553 | 0.84553 |
| **6** | 0.35000 | 0.99947 | 0.82927 | 0.85714 | 0.84298 |
| **7** | 0.40000 | 0.99942 | 0.78862 | 0.86607 | 0.82553 |
| **8** | 0.45000 | 0.99952 | 0.78862 | 0.92381 | 0.85088 |
| **9** | 0.50000 | 0.99955 | 0.78049 | 0.95050 | 0.85714 |
| **10** | 0.55000 | 0.99952 | 0.75610 | 0.95876 | 0.84545 |
| **11** | 0.60000 | 0.99945 | 0.69919 | 0.97727 | 0.81517 |
| **12** | 0.65000 | 0.99944 | 0.69106 | 0.97701 | 0.80952 |
| **13** | 0.70000 | 0.99942 | 0.68293 | 0.97674 | 0.80383 |
| **14** | 0.75000 | 0.99934 | 0.63415 | 0.97500 | 0.76847 |
| **15** | 0.80000 | 0.99924 | 0.57724 | 0.97260 | 0.72449 |
| **16** | 0.85000 | 0.99909 | 0.48780 | 0.96774 | 0.64865 |
| **17** | 0.90000 | 0.99892 | 0.39024 | 0.96000 | 0.55491 |

# Comparativa de métricas variando Threshold - Zoom

| | Threshold | Accuracy | Recall | Precision | F1 score |
|---|---|---|---|---|---|
| **0** | 0.40000 | 0.99942 | 0.78862 | 0.86607 | 0.82553 |
| **1** | 0.41000 | 0.99945 | 0.78862 | 0.88182 | 0.83262 |
| **2** | 0.42000 | 0.99947 | 0.78862 | 0.88991 | 0.83621 |
| **3** | 0.43000 | 0.99948 | 0.78862 | 0.89815 | 0.83983 |
| **4** | 0.44000 | 0.99948 | 0.78862 | 0.89815 | 0.83983 |
| **5** | 0.45000 | 0.99952 | 0.78862 | 0.92381 | 0.85088 |
| **6** | 0.46000 | 0.99952 | 0.78862 | 0.92381 | 0.85088 |
| **7** | 0.47000 | 0.99954 | 0.78862 | 0.93269 | 0.85463 |
| **8** | 0.48000 | 0.99955 | 0.78862 | 0.94175 | 0.85841 |
| **9** | 0.49000 | 0.99956 | 0.78862 | 0.95098 | 0.86222 |
| **10** | 0.50000 | 0.99955 | 0.78049 | 0.95050 | 0.85714 |
| **11** | 0.51000 | 0.99956 | 0.78049 | 0.96000 | 0.86099 |
| **12** | 0.52000 | 0.99956 | 0.78049 | 0.96000 | 0.86099 |
| **13** | 0.53000 | 0.99956 | 0.78049 | 0.96000 | 0.86099 |
| **14** | 0.54000 | 0.99955 | 0.77236 | 0.95960 | 0.85586 |
| **15** | 0.55000 | 0.99952 | 0.75610 | 0.95876 | 0.84545 |
| **16** | 0.56000 | 0.99948 | 0.73171 | 0.95745 | 0.82949 |
| **17** | 0.57000 | 0.99948 | 0.72358 | 0.96739 | 0.82791 |
| **18** | 0.58000 | 0.99949 | 0.72358 | 0.97802 | 0.83178 |

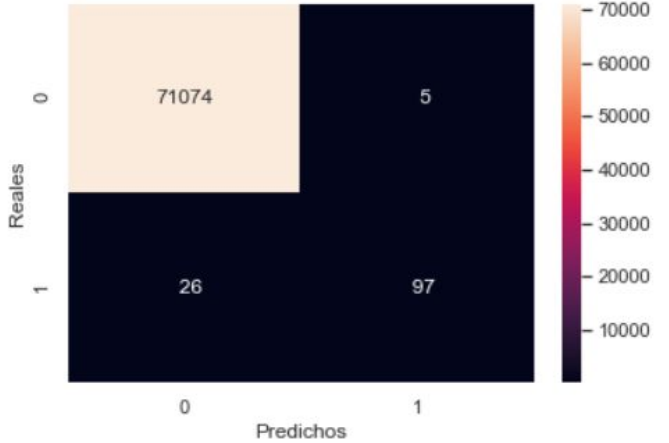
# Predicciones Finales Random Forest

## Matriz de Confusión Train




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9997 | 0.9999 | 0.9998 | 213236 |
| 1 | 0.9652 | 0.8266 | 0.8905 | 369 |
| accuracy | | | 0.9996 | 213605 |
| macro avg | 0.9824 | 0.9133 | 0.9452 | 213605 |
| weighted avg | 0.9996 | 0.9996 | 0.9996 | 213605 |

## Matriz de Confusión Test




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.9996 | 0.9999 | 0.9998 | 71079 |
| 1 | 0.9510 | 0.7886 | 0.8622 | 123 |
| accuracy | | | 0.9996 | 71202 |
| macro avg | 0.9753 | 0.8943 | 0.9310 | 71202 |
| weighted avg | 0.9996 | 0.9996 | 0.9995 | 71202 |

Rendimiento bastante bueno. Accuracy de 0.9996 en la data de entrenamiento y prueba y con un f1 score que alcanza un valor de 0.8611 para la clase fraude, lo que a su vez implica detección del 79% del total de casos de fraude (recall) a costa de una cantidad muy baja de falsos positivos y de falsos negativos.

# Conclusiones

# Conclusiones

- Al realizar la selección de variables para el modelo, se redujo la cantidad de predictores a la mitad, pasando de 30 a 15 variables.
- La aplicación del modelo de Naive Bayes Gaussian no es ideal para este tipo de casos, lo cual se evidencia en un bajo rendimiento que se evidencia en las métricas calculadas.
- Los modelos de regresión logística, árbol de decisión y random forest tienen un rendimiento bastante bueno en sus predicciones, por lo que la implementación de cualquiera de estos modelos sería de utilidad para el caso.
- El modelo de random forest fue el mejor modelo el cual junto a la elección de el mejor threshold tiene un poder de predicción que superó tanto a la hipótesis nula como a los demás modelos entrenados, alcanzando un accuracy de 0.9996 y un f1 score de 0.8611 en la clase fraude.
- Al inspeccionar la importancia de variables del random forest, se evidencia que es posible reducir la cantidad de predictores a tan solo 7, modelo con el que seguramente podemos tener buenas predicciones con un costo computacional menor.
- Los métodos de resampling no fueron considerados en este trabajo ya que en primera instancia no implica que el rendimiento mejore, además que superan los temas vistos durante las clases hasta el momento.

# MUCHAS GRACIAS