# Aprendizaje de máquina para determinar el incumplimiento de pago en tarjetas de crédito

Jose Alberto Arango Sánchez
*Ingeniería de Sistemas*
*Universidad de Antioquia*
Medellín, Colombia
jose.arangos@udea.edu.co

Leon Dario Arango Amaya
*Ingeniería de Sistemas*
*Universidad de Antioquia*
Medellín, Colombia
leon.arango@udea.edu.co

***Abstract*—Este estudio tiene como objetivo realizar una comparación entre modelos clásicos del Machine learning como: KNN, SVM, RNA, Random Forest y Naïve Bayes, usados como clasificadores para determinar el incumplimiento de pago de una tarjeta de crédito. Los modelos que presentaron mejores medidas de desempeño, fueron: Random Forest y Maquina de Soporte Vectorial, seguido de la Red Neuronal Artificial. Sin embargo el modelo Naïve Bayes, tiene mejores valores en la sensibilidad, lo que indica que logra predecir mejor los incumplimientos de pago frente a los demás. Finalmente, con el objetivo de reducir la dimensionalidad se realiza un proceso de selección y extracción de características buscando optimizar los modelos con mejores resultados.**

***Index Terms*—Aprendizaje de máquina, modelos fenomenológicos,incumplimiento de pago, selección de características, extracción de características, machine learning**

## I. Resumen

Este estudio tiene como objetivo realizar una comparación entre modelos clásicos del **Machine learning** como: KNN, SVM, RNA, Random Forest y Naïve Bayes, usados como clasificadores para determinar el incumplimiento de pago de una tarjeta de crédito. Los modelos que presentaron mejores medidas de desempeño, fueron: Random Forest y Maquina de Soporte Vectorial, seguido de la Red Neuronal Artificial. Sin embargo el modelo Naïve Bayes, tiene mejores valores en la sensibilidad, lo que indica que logra predecir mejor los incumplimientos de pago frente a los demás. Finalmente, con el objetivo de reducir la dimensionalidad se realiza un proceso de selección y extracción de características buscando optimizar los modelos con mejores resultados.

## II. Introducción

El mundo moderno cada día es más sorprendente, los avances en tecnología son cada vez mejores y la dificultad de los retos que surgen crece de forma exponencial. En este documento, pretende abordar una problemática del sector bancario, la cual ha sido objeto de estudio durante los últimos años para múltiples investigadores; se trata del incumplimiento de pagos con tarjetas de crédito y para ello se utilizó un conjunto de datos que tuvo origen en Taiwan a mediados del 2005 [4]. El objetivo de este estudio es evaluar cinco algoritmos diferentes de aprendizaje automático, sacar métricas para cada uno de ellos y así poder determinar cual de éstos es mejor. Posteriormente se realizara un proceso de selección y extracción de características, con el fin de optimizar los mejores modelos y determinar cual de éstos realiza mejores predicciones, para resolver la problemática en cuestión.

## III. Descripción del problema

### A. El problema - Análisis de riesgo bancario

Este es un problema supervisado de clasificación, y su propósito es determinar si una persona incumplirá con el pago de su tarjeta de crédito el próximo mes.

### B. Variables del problema

El conjunto de datos utilizado consta de veintitrés características y una variable a predecir, las cuales se definen a continuación:

- X1: Cupo del crédito otorgado en $ NT Dollar.
- X2: Genero (1: Male, 2: Female)
- X3: Nivel de educación ( 1:Postgrado, 2:Universidad, 3:Secundaria, 4:Otros)
- X4: Estado civil ( 1:Casado, 2:Soltero, 3:Otros)
- X5: Edad (años de la persona)
- X6-11: Historial de pago mes mes en orden descendente.
  - X6: Estado pagó SEPTIEMBRE de 2005.
  - X11:Estado pagó ABRIL de 2005.
    - -2: No se usó la tarjeta.
    - -1: Pagó debidamente.
    - 0: El cliente pagó el monto mínimo adeudado.
    - 1:Retraso de 1 mes.
    - 9:Retraso de 9 meses o mas.
- X12-17: Monto del estado de cuenta en de SEPTIEMBRE - ABRIL respectivamente en $ NT Dollar.
  - X12: Monto del estado de cuenta en de SEPTIEMBRE 2005.
  - X17: Monto del estado de cuenta en de ABRIL 2005.
  - Lo que se le debe al banco o si este es negativo es lo que el banco debe a la persona.
- X18-23: Monto del pago anterior $ NT Dollar.
  - X18: Monto pagado en SEPTIEMBRE de 2005.
  - X17: Monto pagado en ABRIL del 2005.

  - Pago que se le realizo al banco.
- Y: Incumplimiento de pago al siguiente mes (Si:1, No:0)

Es importante resaltar que el conjunto de datos no requiere imputación, ya que las muestras están completas.

## IV. Estado del arte

Determinar si un cliente incumplirá en el pago de su tarjeta, Representa un problema de iteres para el sector bancario, debido a que la minimización de riesgos genera mayor esperanza de retorno del dinero. Por lo tanto es importante dar solución a este problema de una manera objetiva y precisa, aprovechando la información con la que se cuenta de los clientes. Información que ha sido empleada por múltiples investigadores para abordar este problema, usando algoritmos de ***Machine Learning*** y obteniendo muy buenos resultados.

A continuación se presentan 4 artículos relevantes que abordan el problema mencionado anteriormente. estos emplean las misma base de datos usada en este artículo:

### *A. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients*

En este articulo se realiza [1] una comparación de 6 modelos clásicos del ***Machine Learning***, con el fin de determinar el incumplimiento del pago de una tarjeta el siguiente mes, por parte de un cliente asociado a un banco de Taiwán, con base en su información bancaria y personal. Además propone un método suavizado de clasificación, para determinar la probabilidad real del incumplimiento. A continuación se presentan los modelos implementados en este artículo:

*1) Modelos de aprendizaje automático implementados:*

- K-nearest neighbor classifiers (KNN)
- Logistic regression (LR)
- Discriminant analysis (DA)
- Naïve Bayesian classifier (NB)
- Artificial neural networks (ANNs)
- Classification trees (CTs)

*2) Metodología de validación:*
La metodología de validación empleada por estos investigadores fue bootstrapping [2]. Aunque no especifican la proporción de división del conjunto de datos.

*3) Resultados obtenidos:*
En la figura 1, se presenta una tabla con los resultados obtenidos para cada modelo, en la etapa de entrenamiento y validación:

Table 1
Classification accuracy

| Method | Error rate | | Area ratio | |
|---|---|---|---|---|
| | Training | Validation | Training | Validation |
| K-nearest neighbor | 0.18 | 0.16 | 0.68 | 0.45 |
| Logistic regression | 0.20 | 0.18 | 0.41 | 0.44 |
| Discriminant analysis | 0.29 | 0.26 | 0.40 | 0.43 |
| Naïve Bayesian | 0.21 | 0.21 | 0.47 | 0.53 |
| Neural networks | 0.19 | 0.17 | 0.55 | 0.54 |
| Classification trees | 0.18 | 0.17 | 0.48 | 0.536 |

Figure 1. Resultados algoritmos evaluados

En este gráfico se aprecia un apartado llamado área ratio la cual esta dada por la siguiente expresión:

$$\text{Relación de área} = \frac{\text{ACMCR}}{\text{AMCTCR}}$$

Donde ACMCR es el área entre la curva del modelo y la curva de referencia y AMCTCR es área entre la mejor curva teórica y la curva de referencia, como se observa en la figura 2.

Esta medida, es un criterio importante para la selección del mejor modelo, dado que un mayor valor representa una mayor similitud entre el modelo planteado y el teórico.

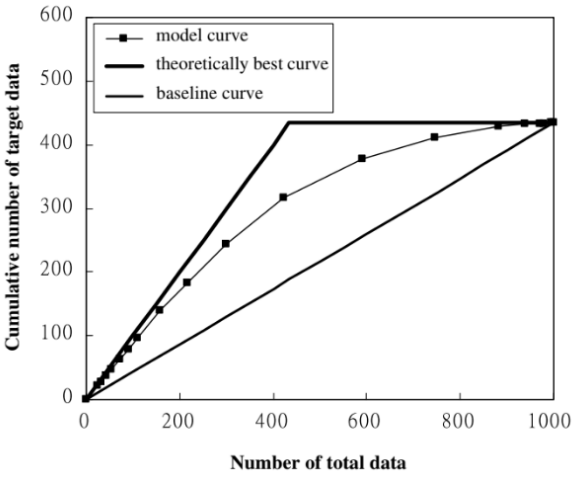



Figure 2. Relación de área

Finalmente de la figura 1 y 2 se concluye que:

- El método KNN tiene un mejor desempeño en la fase de entrenamiento ya que cuenta con la tasa de error más baja y la relación de área más alta en esta misma fase.
- La red neuronal tiene mejor desempeño en la fase de validación, dado que su relación de área es la más alta entre todos los modelos, y su tasa de error es relativamente baja (0.17).

Finalmente los autores de este articulo recomiendan implementar redes neuronales artificiales, para calificar a los clientes.

*B. An experimental comparison of classification algorithms for imbalanced credit scoring data sets*

En este articulo se [3] compara varias técnicas de clasificación del **Machine learning**, para conjuntos de datos desequilibrados, empleados en la calificación crediticia conocido comúnmente como **Scoring**. Además de utilizar técnicas de clasificación tradicionales, como la regresión logística, las redes neuronales y los árboles de decisión, este artículo explora el **gradient boosting**, máquina de vector de soporte de mínimos cuadrados y random forests para la predicción del incumplimiento de pago.

Los criterios de rendimiento elegidos en este artículo para medir el efecto del desbalanceo en los diferentes modelos son: el área bajo la curva (AUC), la estadística de Friedman y las pruebas post hoc de Nemenyi. A continuación se presentan los modelos implementados en este artículo:

*1) Modelos de aprendizaje automático implementados:*

- Logistic regression
- Linear and quadratic discriminant analysis
- Neural networks (Multi-layer perceptron)
- Least square support vector machines (LS-SVMs)
- C4.5. decision trees
- k-NN (memory based reasoning)
- Random forests
- Gradient boosting

*2) Metodología de validación:*

La metodología de validación empleada fue Bootstrapping, con una proporción de $\frac{2}{3}$ del *dataset* para el entrenamiento y $\frac{1}{3}$ para las pruebas.

*3) Resultados obtenidos:*

- Random forests y gradiente boosting fueron los de mejor resultado debido a que tiene un AUC mayor.
- Decision tree, quadratic discriminant analysis y KNN obtienen los peores resultados.
- Las técnicas de sobremuestreo tiene mejor resultado que las técnicas de submuestreo.
- El Random forests funciona sorprendentemente bien dado un gran desequilibrio de clases.

*C. Application of machine learning algorithms in credit card default payment prediction*

En este trabajo se realiza una comparación de ocho modelos diferentes de los cuales sacan la precisión, sensibilidad y especificidad como métricas para evaluar el rendimiento. Además con base en estos resultados obtener el mejor modelo.

*1) Modelos de aprendizaje automático implementados:*

- Naïve Bayes
- K-nearest neighbor
- support vector machine
- Logistic regression
- Classification trees
- Bagging
- Boosting
- Voting

*2) Metodología de validación:*

La metodología de validación utilizada fue validación cruzada con diez k-fold

*3) Resultados obtenidos:*

Según el estudio de Husejinovic, Keco y Masetić [6], se determinó que el mejor modelo es la regresión logística con una eficiencia de 0.820, seguido de la máquina de soporte vectorial con 0.819 Figura 3

| Model | Initial dataset | No outliers | Feature selection |
|---|---|---|---|
| Naive Bayes | 0.655 | 0.766 | 0.806 |
| K-NN | 0.812 | 0.795 | 0.792 |
| SVM | 0.819 | 0.803 | 0.804 |
| Logistic | 0.820 | 0.804 | 0.805 |
| C4.5 | 0.808 | 0.789 | 0.791 |
| Bagging | 0.816 | 0.801 | 0.797 |
| Boosting | 0.813 | 0.796 | 0.796 |
| Voting | 0.789 | 0.795 | 0.797 |

Figure 3. Eficiencia de los modelos

*D. Credit default mining using combined machine learning and heuristic approach*

La investigación realizada por Islam, Eberle y Ghafoor [7], buscaba encontrar la solución al problema desde dos enfoques diferentes: uno heurístico y a través del aprendizaje de Máquina. Los investigadores concluyen que el aprendizaje de máquina es mejor por aproximadamente 2% de exactitud. Para determinar esto, utilizan cinco modelos diferentes:

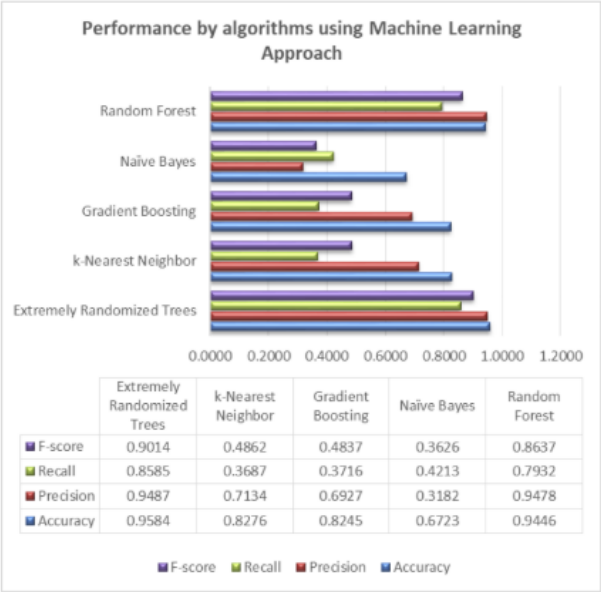
- Random Forest
- Naïve Bayes
- Gradient Boosting
- K-nearest neighbor
- Extremly Randomized Trees

*1) Metodología de validación:*

La metodología de validación utilizada fue validación cruzada con diez k-fold

*2) Resultados obtenidos:*

Según los investigadores, los mejores modelos para tratar este problema son árboles extremadamente aleatorios con una eficiencia de 0.958 y árboles aleatorios con 0.944 Figura 4




| | Extremely Randomized Trees | k-Nearest Neighbor | Gradient Boosting | Naïve Bayes | Random Forest |
|---|---|---|---|---|---|
| F-score | 0.9014 | 0.4862 | 0.4837 | 0.3626 | 0.8637 |
| Recall | 0.8585 | 0.3687 | 0.3716 | 0.4213 | 0.7932 |
| Precision | 0.9487 | 0.7134 | 0.6927 | 0.3182 | 0.9478 |
| Accuracy | 0.9584 | 0.8276 | 0.8245 | 0.6723 | 0.9446 |

Figure 4. Eficiencia de los modelos

Los resultados obtenidos por estos investigadores son buenos, pero no es posible saber cómo llegaron a ellos, ya que hicieron poco énfasis en los procedimientos.

## V. Experimentos

Para los siguientes experimentos se empleó la base de datos: *default of credit card clients Data Set* [4]. Esta base de datos cuenta con la información crediticia de 30.000 clientes, descrita por 23 características (ver sección 2B). Es importante especificar que la base de datos no contiene datos faltantes.

El problema se abordó como un modelo de clasificación biclase, en el cual las clases corresponden a:

- 0: La persona no incumplirá en el pago de su tarjeta en el siguiente mes.
- 1: La persona incumplirá en el pago de su tarjeta en el siguiente mes.

Se tiene un desbalanceo entre clases el cual se puede visualizar en la figura 5

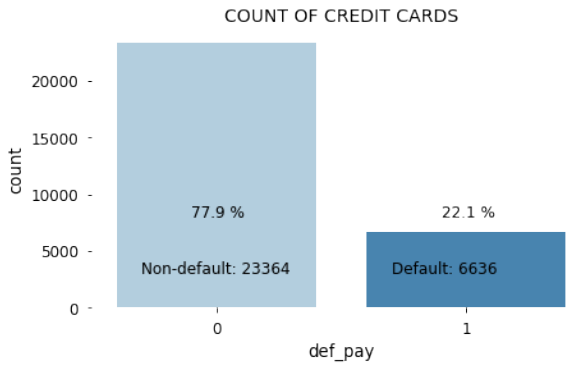



Figure 5. Frecuencia de clases

Además, las características se encuentran en diferentes escalas como se logra apreciar en la figura 6. Por lo tanto se realizo una normalización de todas estas con el criterio *Z-SCORE*.

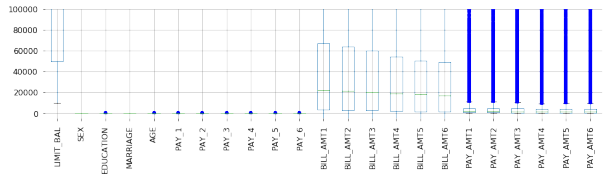



Figure 6. Escala de variables

La normalización ocasiona que la distribución tenga media 0 y desviación estándar 1, como se visualiza en la figura 7 mediante el Violin plot.

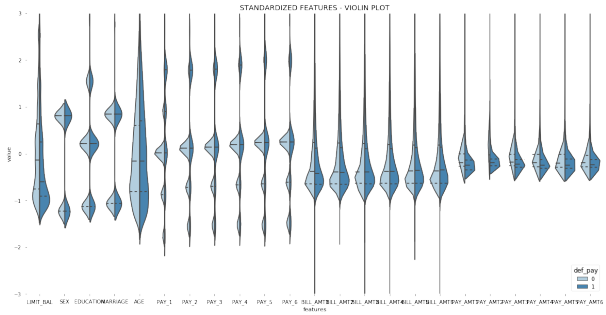



Figure 7. Violin Plot

Finalmente para atacar el problema de desbalanceo de clases, se empleo validación cruzada estratificada con 10 *folds* como lo sugiere el articulo 3. El conjunto de datos se dividió en tres partes, un subconjunto de entrenamiento y de validación que equivale al 80% del total de muestras y uno de pruebas con el 20%. Esto con el fin de determinar el comportamiento de los modelos entrenados frente a muestras que no conozcan, evaluando su capacidad de generalización. A continuación se presentaran los resultados obtenidos en 5 modelos diferentes:

### A. Naïve Bayes

En este modelo, se utilizaron los parámetros por defecto, incluidos en la librería sklearn [5]. Obteniendo un *Acurracy* del 67%. Otras medidas de desempeño utilizadas fueron: sensibilidad, especificidad, eficiencia y precisión las cuales se obtuvieron a partir de la figura 8 o bien llamada matriz de confusión y se aprecian en la tabla I

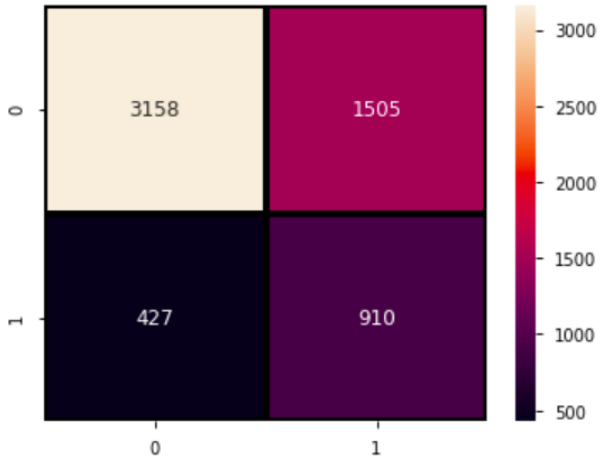



Figure 8. Matriz de confusión del modelo Naïve Bayes

| Sensibilidad | Especificidad | Eficiencia | Precisión |
|---|---|---|---|
| 0,68 | 0,68 | 0,68 | 0,38 |

Table I
MEDIDAS DE DESEMPEÑO DEL MODELO NAÏVE BAYES

Durante la fase de entrenamiento se logró una eficiencia de 0.6749 con un intervalo de confianza de +/- 0.0362. Mientras que en la fase de validación, una eficiencia de 0.6742 con un intervalo de confianza de +/- 0.0372. Siendo estas unas medidas de desempeño no tan buenas, sin embargo este modelo clasifica mucho mejor las muestras de la clase 1(personas que incumplirá en el pago de su tarjeta en el siguiente mes), que los otros modelos vistos, con un porcentaje del 68% de acierto, el cual es dado por la sensibilidad.

Finalmente a través de la figura 9, llamada curva ROC, la cual es una representación gráfica de la sensibilidad frente a la especificidad, para un sistema clasificador binario. llegamos a la conclusión de que este modelo se comporta de una manera muy regular, al momento de realizar la clasificación.

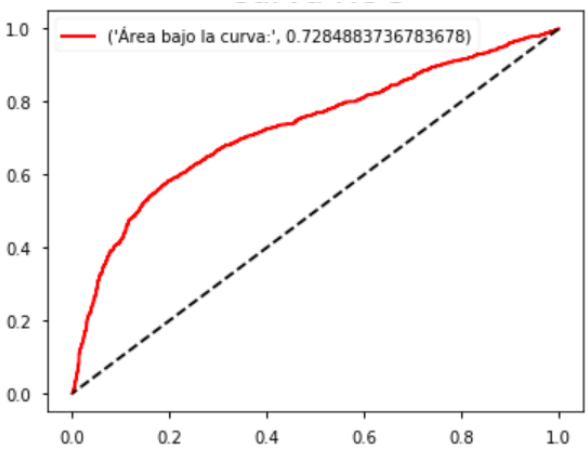



Figure 9. Curva ROC modelo Naïve Bayes

### B. K vecinos mas cercanos (KNN)

En este modelo se realizaron las siguiente combinaciones de parámetros que se pueden apreciar en la tabla II

| Parametro | Valores |
|---|---|
| n_neighbors | 10,20,30,40,50 |
| metric | minkowski, euclidean, manhattan |

Table II
COMBINACIÓN DE PARÁMETROS EMPLEADA PARA EL MODELO KNN

La mejor combinación de parámetros encontrada fue: metric: minkowski y n_neighbors: 11, la cual obtuvo un *Acurracy* del 81%. Durante la fase de entrenamiento se logró una eficiencia de 0.8271 con un intervalo de confianza de +/- 0.0010. Mientras que en la fase de validación, una eficiencia de 0.8058 con un intervalo de confianza de +/- 0.0054. Siendo estos unos resultados aceptables en comparación con el modelo Naïve Bayes. Como se han venido presentado los resultados en modelos previos, la figura 14 y figura 15 son la matriz de confusión y la curva ROC para este modelo.

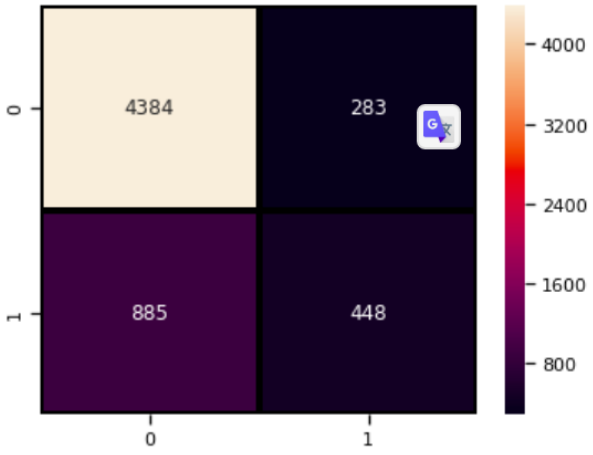



Figure 10. Matriz de confusión del modelo KNN

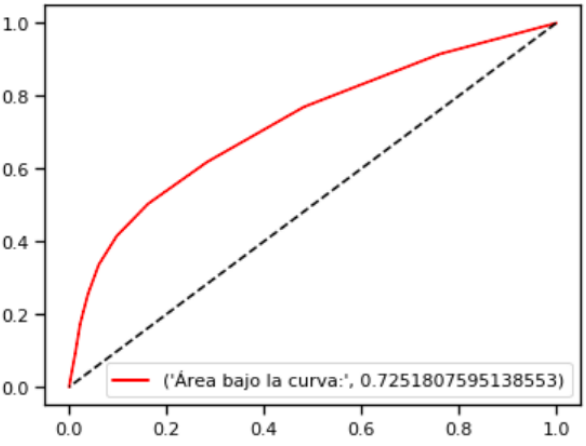



Figure 11. Curva ROC modelo KNN

De la matriz de confusión, se obtienen las medidas de desempeño de la tabla III

| Sensibilidad | Especificidad | Eficiencia | Precisión |
|---|---|---|---|
| 0,34 | 0,94 | 0,81 | 0,61 |

Table III
MEDIDAS DE DESEMPEÑO DEL MODELO KNN

Finalmente se concluye que este modelo:

- El 81% de las muestras predichas fueron correctamente clasificadas.
- No presenta sobre ajuste, ya que el error en entrenamiento no es cercano a cero y el de validación no es muy grande.
- Los errores son similares , sin embargo se tiene mayor error en la fase de validación que en entrenamiento, como es de esperar según la teoría.
- Según la sensibilidad solo el 34% de las muestras de la clase 1 (personas que incumplirá en el pago de su tarjeta en el siguiente mes), están siendo correctamente clasificadas.
- El modelo esta clasificando el 94% de las muestras de la clase 0 (persona que no incumplirá en el pago de su tarjeta en el siguiente mes), correctamente.
- Dado los valores de sensibilidad y especificidad, el modelo clasifica bien muestras de la clase 0 (la clase mayoritaria), caso contrario las muestras clase 1 (clase minoritaria). Por lo tanto se debe implementar otra manera de abordar el desabalanceo de clases al momento de entrenar este modelo.
- El área bajo la curva de este modelo es muy regular, incluso mas baja que el modelo Naïve Bayes por lo tanto, el modelo tiene capacidad de separación de clases pero no es muy bueno.

### *C. Redes Neuronales Artificiales (RNA)*

En este modelo se realizaron las siguiente combinaciones de parámetros la cual se puede apreciar en la tabla IV

| Parametro | Valores |
|---|---|
| solver | adam |
| max_iter | 300 |
| alpha | 0.1 , 0.01 , 0.001 |
| hidden_layer_sizes | 300 |
| activation | logistic, tanh |

Table IV
COMBINACIÓN DE PARÁMETROS EMPLEADA PARA EL MODELO DE REDES NEURONALES ARTIFICIALES

La mejor combinación de parámetros encontrada fue: Red neuronal artificial con función de activación logística, alpha = 0.01, numero de neuronas por capa = 72, máximo numero de iteraciones = 300 y el método para la optimización de los pesos = adam, que obtuvo un *Acurracy* del 82%. Durante la fase de entrenamiento se logró una eficiencia de 0.8228 con un intervalo de confianza de +/- 0.0010. Mientras que en la fase de validación, una eficiencia de 0.8197 con un intervalo de confianza de +/- 0.0044. Como se han venido presentado los resultados en modelos previos, la figura 12 y figura 13 son la matriz de confusión y la curva ROC respectivamente, para este modelo.

El resultado de las medidas de desempeño, expresado en la tabla IX fue:

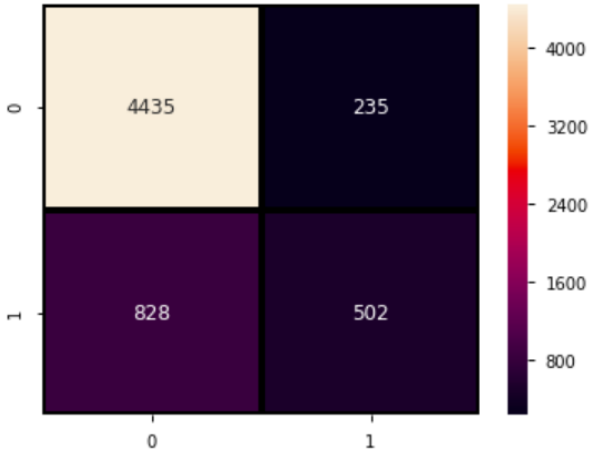



Figure 12. Matriz de confusión del modelo de Redes Neuronales Artificiales

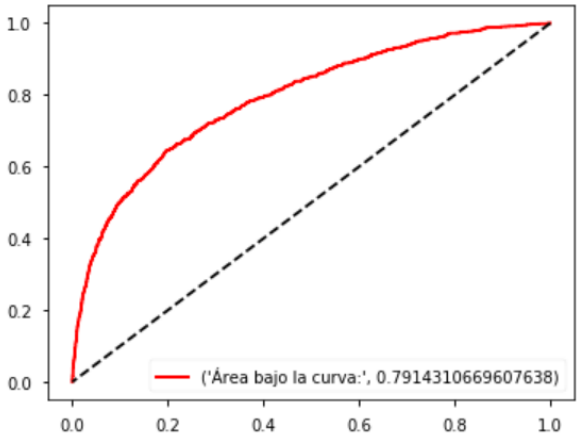



Figure 13. Curva ROC modelo de Redes Neuronales Artificiales

| Sensibilidad | Especificidad | Eficiencia | Precisión |
|---|---|---|---|
| 0,38 | 0,95 | 0,82 | 0,68 |

Table V
MEDIDAS DE DESEMPEÑO DEL MODELO DE REDES NEURONALES ARTIFICIALES

Finalmente se concluye que en este modelo:

- El 82% de las muestras fueron clasificadas correctamente, lo cual es un resultado muy bueno en comparación con los otros modelos.
- Al igual que el modelo KNN, este modelo no presenta sobreajuste.
- No se presenta subajuste, ya que para cumplir dicha condición el error durante el entrenamiento y validación debe ser elevado y similar para ambos casos; condición que no cumple este modelo.
- Este modelo presenta las misma falencias que el modelo KNN, respecto a la clasificación de muestras de la clase 1,sin embargo es un 4% mejor que este.
- 68% de precisión indica que hay una mediana proximidad entre los valores obtenidos en mediciones repetidas.
- El área bajo la curva de este modelo es mejor, que la de los modelo Naïve Bayes y KNN. Llegando casi al 0.8

valor levemente cercado a 1, que es el valor ideal. Por lo tanto, el modelo tiene una mayor capacidad de separación de clases respecto a los otros.

### *D. Random Forest*

En este modelo se realizaron las siguiente combinaciones de parámetros que se pueden apreciar en la tabla VI

| Parametro | Valores |
|---|---|
| max_features | 10,20,30,40,50 |
| max_features | 1,2,3,4,5 |
| max_depth | 1,2,3,4,5,8,10,15,20 |
| criterion | gini, entropy |

Table VI
COMBINACIÓN DE PARÁMETROS PARA EL MODELO RANDOM FOREST

La mejor combinación de parámetros encontrada fue: criterion: entropy, max_depth: 8, max_features: 5 y n_estimators: 50, la cual obtuvo un **Acurracy** del 83%

Durante la fase de entrenamiento se logró una eficiencia de 0.9514 con un intervalo de confianza de +/- 0.0012. Mientras que en la fase de validación, una eficiencia de 0.8263 con un intervalo de confianza de +/- 0.0051. Siendo estos unos resultados buenos en comparación a los del modelo Naïve Bayes y el KNN. Como se han venido presentado los resultados en modelos previos, la figura 14 y figura 15 corresponden a la matriz de confusión y la curva ROC respectivamente, para este modelo.

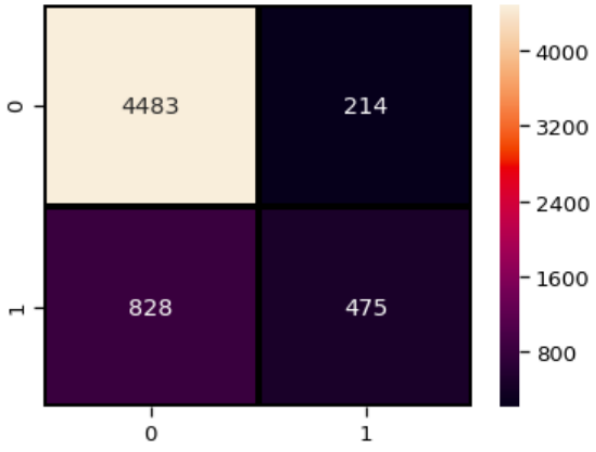



Figure 14. Matriz de confusión del modelo Random Forest

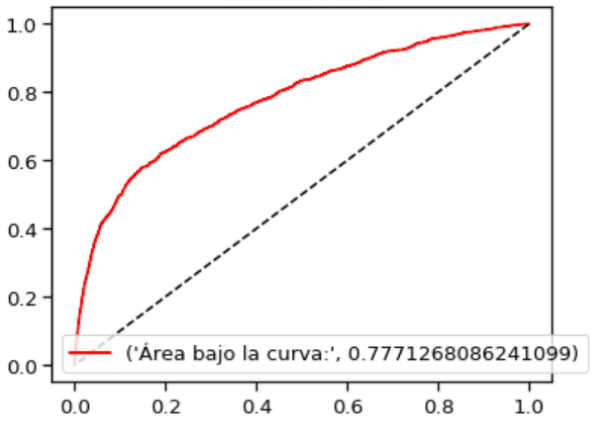



Figure 15. Curva ROC del modelo Random Forest

El resultado de las medidas de desempeño, son los presentados en la tabla VII:

| Sensibilidad | Especificidad | Eficiencia | Precisión |
|---|---|---|---|
| 0,36 | 0,95 | 0,83 | 0,69 |

Table VII
MEDIDAS DE DESEMPEÑO DEL MODELO RANDOM FOREST

Finalmente se concluye que en este modelo:

- El 83% de las muestras fueron clasificadas correctamente, lo cual es un resultado bueno en comparación a los otros modelos.
- Al igual que el modelo KNN, este modelo no presenta sobreajuste, sin embargo dada la alta eficiencia (95%) durante el entrenamiento se puede observar indicios de memorización de muestras.
- Este modelo presenta las misma falencias que el modelo KNN y la RNA, respecto a la clasificación de muestras de la clase 1.
- 69% de precisión indica que hay una mediana proximidad entre los valores obtenidos en mediciones repetidas, lo cual es un aumento leve respecto a la red neuronal artificial.
- El área bajo la curva de este modelo es mejor que el modelo KNN, pero menor que la RNA, cercano a 1 que es el valor de referencia. Por lo tanto, el modelo tiene una buena capacidad de separación.
- El costo computacional de este modelo es mucho menor que el empleado por la RNA, además obtiene resultados muy similares.

### *E. Maquinas de Soporte Vectorial*

En este modelo se realizaron las siguiente combinaciones de parámetros la cual se puede apreciar en la tabla VIII

| Parámetro | Valores |
|---|---|
| C | 0.001, 0.01, 0.1, 1, |
| gamma | 0.001, 0.01, 0.1, 1 |
| alpha | 0.1 , 0.01 , 0.001 |
| kernel | rbf,linear |

Table VIII
COMBINACIÓN DE PARÁMETROS PARA LA MAQUINAS DE SOPORTE VECTORIAL

La mejor combinación de parámetros encontrada fue: Maquina de soporte vectorial con C = 1, gamma = 0.1 y kernel rbf, obteniendo un **Acurracy** del 83%. Durante la fase de entrenamiento se logró una eficiencia de 0.8205 con un intervalo de confianza de +/- 0.000800. Mientras que en la fase de validación, una eficiencia de 0.832 con un intervalo de confianza de +/- 0.00479. Siendo estos unos buenos resultados. Como se han venido presentado los resultados en modelos previos, la figura 16 y figura 17 representan la matriz de confusión y la curva ROC respectivamente para este modelo.

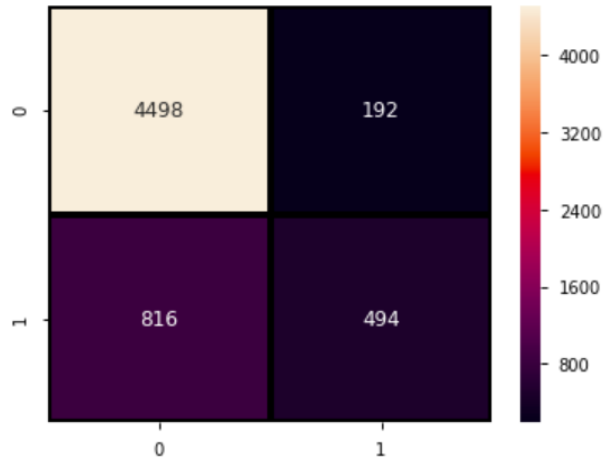



Figure 16. Matriz de confusión del modelo de las Maquinas de Soporte Vectorial

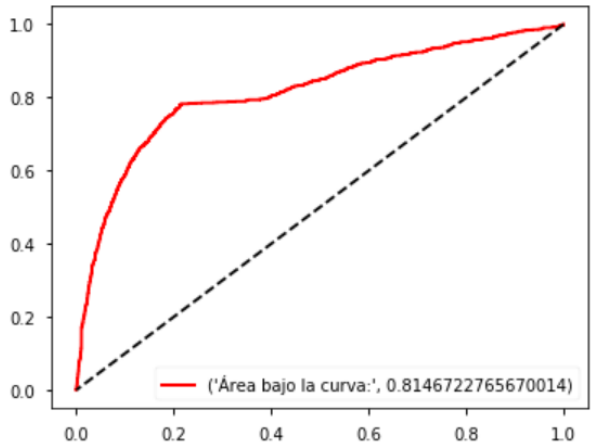



Figure 17. Curva ROC modelo de las Maquinas de Soporte Vectorial

El resultado de las medidas de desempeño, se expresa en la tabla IX:

| Sensibilidad | Especificidad | Eficiencia | Precisión |
|---|---|---|---|
| 0,38 | 0,96 | 0,83 | 0,72 |

Table IX
MEDIDAS DE DESEMPEÑO DEL MODELO DE MAQUINAS DE SOPORTE VECTORIAL

Finalmente se concluye que este modelo:

- Tiene la mayor área bajo la curva (0.81) de todos los modelos mencionados anteriormente. Por lo tanto, tiene una mayor capacidad de separación entre clases.
- Tiene las mejores medidas desempeño, en comparación a los modelos anteriormente vistos. estas medidas se expresan en la tabla IX.
- Es igual de bueno que el Random Forest para clasificar muestras correctamente, con una certeza del 83%
- Presenta las misma dificultades que el KNN, Random Forest Y RNA para la clasificación de muestra de la clase 1.

## VI. ANÁLISIS DE CARACTERÍSTICAS

Trabajar con las 23 características al tiempo, implica un costo computacional elevado. No obstante puede presentarse que algunas características no aporte mucha información a los modelos, siendo redundantes. También es importante cumplir el principio de parsimonia, el cual consiste en que el modelo mas simple, será siempre el escogido como solución. Por lo tanto es importante considerar la reducción de dimensionalidad, es decir, trabajar con menos variables.

Para lograr este fin, se plantearon dos métodos: Realizando selección y extracción de características. Sin embargo, para poder emplear estas estrategias se debe realizar una primera estimación de la cantidad de variables a eliminar, para el caso de selección; y una cantidad de componentes a seleccionar, para el caso de extracción. Dicha estimación se puede obtener de una manera experimental, mediante el análisis individual de cada una de las características, a partir de medidas de correlación y del índice de Fisher.

Empleando la matriz de correlación de la figura 18 y solo analizando la correlación entre variables, ya que las relaciones con la variable a predecir, no tiene ninguna interpretación valida en problemas de clasificación. Se puede determinar las características candidatas a ser eliminadas.

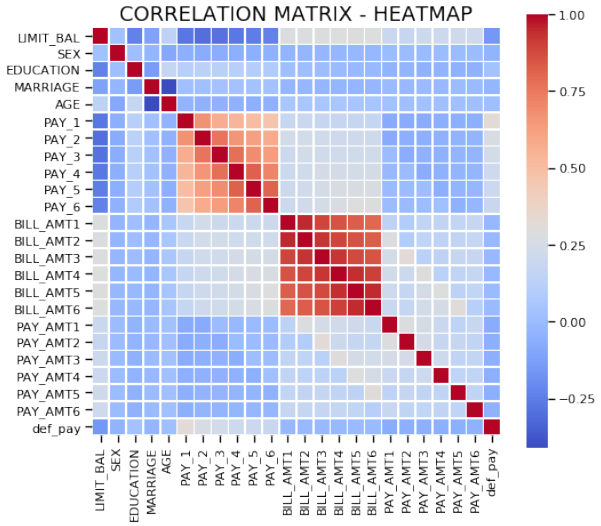



Figure 18. Matriz de correlación

Mayor intensidad de color rojo, entre las parejas de variables a evaluar, implica una correlación fuerte entre ellas. De la figura 18 se logra apreciar que dicha condición es cumplida por las variables: BILL_AMT1, BILL_AMT2, BILL_AMT3,

BILL_AMT4, BILL_AMT5, BILL_AMT5, en mayor proporción y en menor medida por las variables: PAY_1, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6, ya que estas presentan una menor intensidad. Por lo tanto las 12 variables mencionadas anteriormente son las cándidas a ser eliminadas.

Empleando el índice discriminante de **Fisher** se obtiene los siguientes resultados:

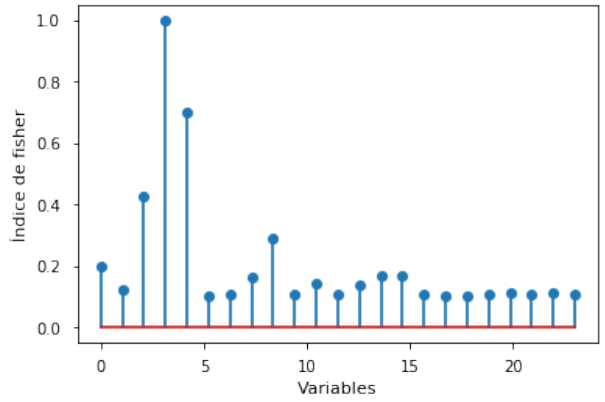



Figure 19. Índices de Fisher

En la gráfica 19, se encuentran los índices de Fisher normalizados, para cada uno de las variables. Solo se conservarán aquellas variables cuyo índice de Fisher sea mayor a 0.5, ya que tienen un mayor poder discriminante. Por lo tanto las únicas variables a conservar para el entrenamiento son: EDUCATION, MARRIAGE y AGE.

Finalmente, por la matriz de correlación existen 12 variables candidatas a eliminar y por el índice de Fisher 20. Es importante aclarar que una de la suposición del índice de Fisher, es que las variables de entrada son independientes entre ellas.

## VII. Selección de características

Para emplear la selección de características, se debe definir como parámetro el numero de características a seleccionar. Dicho valor se obtuvo con base en la cantidad de variables a eliminar, arrojada por la matriz de correlación. Se empleó una función tipo **Wrapper**, con los tres mejores modelos obtenidos en la sección de experimentos, iterando de 10 a 20 variables de dos en dos y usando selección secuencial hacia adelante y hacia atrás. A continuación se presentan los resultados alcanzados por cada modelo:

### *A. Random Forest Classifier*

El mejor resultado obtenidos, dada su eficiencia en la fase de validación es:

- Random Forest con 20 características, en el siguiente orden de importancia:
  LIMIT_BAL, SEX, AGE, PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6, BILL_AMT1, BILL_AMT2, BILL_AMT3, BILL_AMT4, BILL_AMT5, BILL_AMT6, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5, PAY_AMT6.

| **Random Forest Classifier** | | | | |
|---|---|---|---|---|
| N | Forward | Error V | IC (std) | Eficiencia V |
| 10 | TRUE | 0.1652 | 0.00090 | 83.45% |
| 13 | TRUE | 0.1648 | 0.00072 | 83.42% |
| 15 | TRUE | 0.1620 | 0.00083 | 83.73% |
| 18 | TRUE | 0.1619 | 0.00123 | 83.72% |
| 20 | TRUE | 0.1607 | 0.00100 | 83.76% |
| 10 | FALSE | 0.1640 | 0.00074 | 83.69% |
| 13 | FALSE | 0.1636 | 0.00080 | 83.67% |
| 15 | FALSE | 0.1617 | 0.00090 | 83.80% |
| 18 | FALSE | 0.1618 | 0.00082 | 83.79% |
| 20 | FALSE | 0.1605 | 0.00091 | 83.97% |

### *B. RNA Classifier*

| **RNA Classifier** | | | | |
|---|---|---|---|---|
| N | Forward | Error V | IC (std) | Eficiencia V |
| 10 | TRUE | 0.1774 | 0.00132 | 82.19% |
| 13 | TRUE | 0.1788 | 0.00072 | 82.12% |
| 15 | TRUE | 0.1755 | 0.00095 | 82.37% |
| 18 | TRUE | 0.1749 | 0.00060 | 82.39% |
| 20 | TRUE | 0.1745 | 0.00096 | 82.45% |
| 10 | FALSE | 0.1767 | 0.00088 | 82.25% |
| 13 | FALSE | 0.1760 | 0.00088 | 82.30% |
| 15 | FALSE | 0.1758 | 0.00053 | 82.38% |
| 18 | FALSE | 0.1748 | 0.00080 | 82.39% |
| 20 | FALSE | 0.1741 | 0.00099 | 82.44% |

El mejor resultado obtenidos, dada su eficiencia en la fase de validación es:

- Red neuronal artificial con 20 características, en el siguiente orden de importancia:
  LIMIT_BAL, SEX, EDUCATION, MARRIAGE, AGE, PAY_0, PAY_2, PAY_4, PAY_5, PAY_6, BILL_AMT1, BILL_AMT2, BILL_AMT3, BILL_AMT5, BILL_AMT6, PAY_AMT1, PAY_AMT2, PAY_AMT4, PAY_AMT5, PAY_AMT6

### *C. SVM Classifier*

| **SVM Classifier** | | | | |
|---|---|---|---|---|
| N | Forward | Error V | IC (std) | Eficiencia V |
| 10 | TRUE | 0.1724 | 0.00080 | 82.81% |
| 13 | TRUE | 0.1706 | 0.00076 | 82.98% |
| 15 | TRUE | 0.1691 | 0.00070 | 83.17% |
| 18 | TRUE | 0.1674 | 0.00078 | 83.25% |
| 10 | FALSE | 0.1722 | 0.00061 | 82.68% |
| 13 | FALSE | 0.1693 | 0.00056 | 83.03% |
| 15 | FALSE | 0.1678 | 0.00049 | 83.23% |
| 18 | FALSE | 0.1663 | 0.00055 | 83.32% |
| 20 | FALSE | 0.1655 | 0.00056 | 83.40% |

El mejor resultado obtenidos, dada su eficiencia en la fase de validación es:

- Maquina de soporte vectorial con 20 características, en el siguiente orden de importancia:
  LIMIT_BAL, SEX, EDUCATION, MARRIAGE, AGE, PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6, BILL_AMT1, BILL_AMT2, BILL_AMT3, BILL_AMT4, BILL_AMT5, BILL_AMT6, PAY_AMT2, PAY_AMT5, PAY_AMT6

La gráfica 20, indica la frecuencia de cada característica, de los tres mejores modelos. En esta se ve claramente que del total de 22 características los 3 modelos consideran 16 de suma importancia.

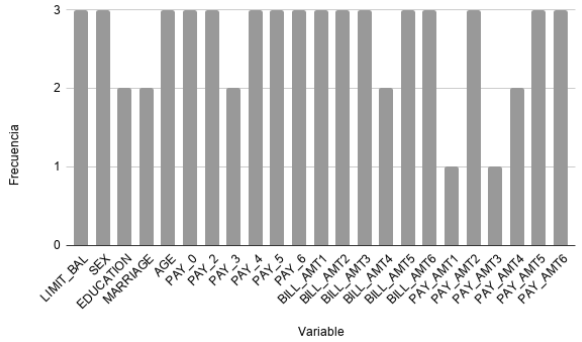



Figure 20. Frecuencia de características mas relevantes

Empleando 2 y 3 variables a seleccionar, las cuales fueron arrojadas por el análisis de Fisher. Se tiene los siguientes resultados por modelo: Donde N: es el numero de características implementadas, Error V:es el error de validación, IC: es el intervalo de confianza y Eficiencia V: es la eficiencia durante la etapa de validación.

### *D. Random Forest Classifier*

| **Random Forest Classifier** | | | | |
|---|---|---|---|---|
| N | Forward | Error V | IC (std) | Eficiencia V |
| 2 | TRUE | 0.17525 | 0.00050 | 82.49% |
| 3 | TRUE | 0.17068 | 0.00029 | 82.90 % |
| 2 | FALSE | 0.1772 | 0.00097 | 82.13 % |
| 3 | FALSE | 0.1733 | 0.00082 | 82.66 % |

### *E. RNA Classifier*

| **RNA Classifier** | | | | |
|---|---|---|---|---|
| N | Forward | Error V | IC (std) | Eficiencia V |
| 2 | TRUE | 0.1800 | 0.0005 | 81.973% |
| 3 | TRUE | 0.1793 | 0.0006 | 81.982% |
| 2 | FALSE | | | % |
| 3 | FALSE | | | % |

### *F. SVM Classifier*

| **SVM Classifier** | | | | |
|---|---|---|---|---|
| N | Forward | Error V | IC (std) | Eficiencia V |
| 2 | TRUE | 0.1789 | 0.0009 | 82.25 % |
| 3 | TRUE | 0.1783 | 0.00089 | 82.32% |
| 2 | FALSE | 0.1790 | 0.0010 | 82.255 % |
| 3 | FALSE | 0.1783 | 0.0008 | 82.32% |

## VIII. Extracción de características

La extracción de características es otro método para la reducción de la dimensionalidad. Este tiene como objetivo. Conservar la mayor cantidad de información posible; realizando transformaciones de los datos originales a un nuevo conjunto de variables. Para este caso se empleo el análisis de componente principal (PCA), el cual parte del supuesto de que mayor variación implica más información.

A continuación se presentan los resultados registrados por este método, empleando como número de componentes, la cantidad de características usadas en la sección de selección.

### *A. Random Forest Classifier*

| **Random Forest Classifier** | | | |
|---|---|---|---|
| N | Error V | IC (std) | Eficiencia V |
| 10 | 0.1964 | 0.00687 | 80.35% |
| 13 | 0.1934 | 0.00862 | 80.65% |
| 15 | 0.1869 | 0.00947 | 81.30% |
| 18 | 0.1874 | 0.00933 | 81.25% |
| 20 | 0.1860 | 0.00901 | 81.39% |

Si bien el mejor resultado es el obtenido con un total de 20 componentes, utilizar solo 15 componentes logra aproximarse mucho a este valor, teniendo un costo computacionalmente mejor.

### *B. RNA Classifier*

| **RNA Classifier** | | | |
|---|---|---|---|
| N | Error V | IC (std) | Eficiencia V |
| 10 | 0.1929 | 0.00787 | 80.70% |
| 13 | 0.1923 | 0.00873 | 80.76% |
| 15 | 0.1813 | 0.00998 | 81.86% |
| 18 | 0.1807 | 0.01070 | 81.92% |
| 20 | 0.1813 | 0.01110 | 81.86% |

En este caso un total de 18 componentes logra un mejor resultado, lo cual se sale de la tendencia impuesta por el total de 20 componentes, que en la mayoría de los casos alcanza un mejor resultado.

### *C. SVM Classifier*

| **SVM Classifier** | | | |
|---|---|---|---|
| N | Error V | IC (std) | Eficiencia V |
| 10 | 0.1932 | 0.00662 | 80.67% |
| 13 | 0.1935 | 0.00992 | 80.64% |
| 15 | 0.1834 | 0.01169 | 81.65% |
| 18 | 0.1813 | 0.01014 | 81.86% |
| 20 | 0.1813 | 0.00956 | 81.86% |

Se logra un empate de rendimiento, cuando se emplean 18 y 20 componentes, sin embargo las 18 componentes tiene un costo menor computacionalmente.

## IX. Comparación de resultados con el estado del arte

- Respecto al artículo [1], se lograron resultados muy similares, excepto por el modelo de Naïve Bayes, el cual tiene 12% de diferencia con el resultado del articulo. El modelo KNN y la red neuronal artificial, solo tienen un 1% de diferencia y respecto al maquina de soporte vectorial no se puede llegar a ninguna conclusión concreta, ya que no es un modelo de interés del articulo.
- Según el artículo [3], el Random forests funciona bien dado un gran desequilibrio de clases. Sin embargo esta afirmación no se logro constatar, debido que este modelo es igual de ineficiente a los otros, al momento de clasificar la clase minoritaria. Los motivos que pudieron incidir en estos resultados son: la metodología de validación usada en el articulo, la cual es diferente a la empleada en la

sección de experimentos y la no utilización de técnicas de submuestreo. Sin embargo la eficiencia de este modelo es la mas alta junto con la SVM, lo cual es una conclusión a la que llega el articulo.

- El articulo [6] emplea la misma cantidad de **Folds** en su metodología de validación. Las métricas de desempeño logradas por el método de Naïve Bayes, son muy similares a las obtenidas en ese artículo, la eficiencia tiene una diferencia de un 0.01, mientras que la precisión de un 0.07. Respecto al modelo KNN, se tiene un comportamiento similar en el cual la precisión y eficiencia tiene diferencias menores al 0.1, mientras que para la sensibilidad se presenta una diferencia de 0.02. Finalmente para el modelo de Random forest, no se logran tan buenos resultados como los mencionados en el artículo, en el cual alcanzan un **Acurracy** del 94%.
- Finalmente el articulo [7] tiene métricas de desempeño, muy similares a los resultados generados por los modelos: KNN, Naïve Bayes, Random forest y SVM. Siendo este ultimo modelo de gran interés, ya que es el único articulo del estado del arte que lo implementa. La diferencia entre el **Acurracy** obtenido en el articulo y el de la sección de experimentos, es de un 0.011.

## X. Conclusiones

El objetivo de este estudio fue determinar cuál de los modelos tenía un mejor comportamiento al momento de predecir el incumplimiento de pagos en tarjetas de crédito para el mes siguiente. Después de los experimentos, se pudo notar que los modelos de mayor eficiencia fueron: Random Forest con 0.83 y Maquina de Soporte Vectorial con 0.83; seguido de la Red Neuronal Artificial con 0.82. Por otro lado, al comparar la sensibilidad y la especificidad en los modelos, se pudo observar grandes similitudes en estas, ya que en casi todos los modelos la sensibilidad estaba alrededor de 0.34, mientras que la especificidad en un 0.95. Sin embargo, lo anterior no aplica para el modelo Naïve Bayes, por que este tiene mejores valores en la sensibilidad, lo que indica que este logra predecir mejor los incumplimientos de pago frente a los demás.

Después de analizar los experimentos se realizo el proceso de selección y extracción de características, con el cual se buscaba optimizar los modelos que dieron mejores resultados. De estos dos procesos, el que logro mejores resultados fue la selección de características, aumentando la eficiencia de los modelos en casi dos por ciento, como se muestra a continuación:

- Random Forest con una eficiencia del 83.97% empleando un total de veinte características.
- Red Neuronal Artificial con una eficiencia del 82.45% usando un total de veinte características.
- Maquina de Soporte Vectorial con una eficiencia del 83.40% manejando un total de veinte características.

Si se comparan estos valores con los obtenidos por otros investigadores, podemos notar que si hubo una mejora de casi un dos por ciento en algunos modelos, lo cual es una cifra buena en cuanto a riesgo financiero se refiere.

Por otro lado, es importante resaltar que este estudio aplica para el comportamiento de las personas en Taiwan, debido a la procedencia del conjunto de datos.

Además seria un practica interesante, realizar mas estudios en el tema, considerando otros lugares y un rango de fechas mas amplio, ya que en este caso el conjunto de datos solo posee información entre abril y septiembre de 2005.

## References


[1] Yeh, I. C., & Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.

[2] 8.3.1. sklearn.cross-validation.Bootstrap — scikit-learn 0.11-git documentation. (2019). Retrieved 13 October 2019, from https://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/modules/generated/sklearn.cross$_v alidation.Bootstrap.html$

[3] Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Systems with Applications, 39(3), 3446-3453.

[4] UCI Machine Learning Repository: default of credit card clients Data Set. (2019). Retrieved 13 October 2019, from https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[5] Scikit-learnorg. (2019). Scikit-learnorg. Retrieved 13 October, 2019, from http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

[6] Husejinovic, Admel & Kečo, Dino & Mašetić, Zerina. (2018). Application of Machine Learning Algorithms in Credit Card Default Payment Prediction. International Journal of Scientific Research. 7. 425. 10.15373/22778179#husejinovic.

[7] Islam, S.R., Eberle, W., & Ghafoor, S.K. (2018). Credit Default Mining Using Combined Machine Learning and Heuristic Approach. ArXiv, abs/1807.01176.