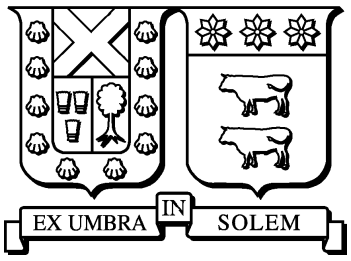
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INFORMÁTICA

SANTIAGO – CHILE



"REDISEÑO DE LA METODOLOGÍA DE DESARROLLO DE UN MODELO DE *SCORING* DE UNA INSTITUCIÓN BANCARIA"

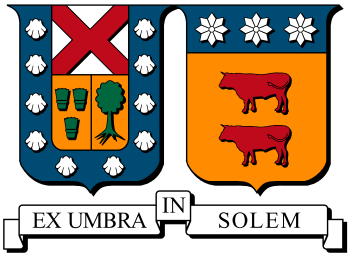
JUAN PABLO CASTILLO VERA

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INFORMÁTICO

PROFESOR GUÍA: JOSÉ LUIS MARTÍ LARA

DECIEMBRE 2017

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO – CHILE



"REDISEÑO DE LA METODOLOGÍA DE DESARROLLO DE UN MODELO DE *SCORING* DE UNA INSTITUCIÓN BANCARIA"

JUAN PABLO CASTILLO VERA

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INFORMÁTICO

PROFESOR GUÍA: JOSÉ LUIS MARTÍ LARA
PROFESOR CORREFERENTE: RICARDO ÑANCULEF ALEGRÍA

DECIEMBRE 2017

MATERIAL DE REFERENCIA, SU USO NO INVOLUCRA RESPONSABILIDAD DEL AUTOR O DE LA INSTITUCIÓN

Agradecimientos

A mi familia y a mis compañeros de trabajo. A mi profesor guía José Luis Martí L., gracias por toda la ayuda durante este largo proceso. A todos los profesores de informática que aportaron a formarme profesionalmente, en especial a Ricardo Ñanculef, Carlos Valle y Marcelo Mendoza, entre muchos otros.

A María Paz Carcamo,
Juan Castillo Armijo,
y Ricardo Muñoz Cancino,
Sin su constante apoyo, esto hubiera sido imposible.

Resumen

En esta investigación se busca mejorar la predicción de los modelos de *scoring* de una institución bancaria a través del uso de nuevas técnicas de modelamiento, de selección de variables y de transformación de variables. Durante el proceso se utiliza el lenguaje de programación Python para aplicar distintos algoritmos de máquinas de aprendizaje como *Logistic Regression*, *Random Forest*, *Extra Tree*, *Bagging*, *Gradient Boosting*, *AdaBoost* y *Naive Bayes*. Los paquetes utilizados fueron *pandas*, *matplotlib* y *scikit-learn*. En base a esta investigación se propone una solución para mejorar el desarrollo de modelos y con esto potenciar los futuros modelos que se usarán para producción dentro de los negocios de la institución bancaria.

Palabras clave: Máquinas de aprendizaje, selección de variables, transformación de variables, *logistic regression*, técnicas de ensamblaje, *Python*.

Abstract

This research seeks to improve the prediction of *scoring* models of a banking institution through the use of new techniques of modeling, variable selection and variable transformation. During the process the Python programming language is used to apply different algorithms of learning machines like *Logistic Regression*, *Random Forest*, *Extra Tree*, *Bagging*, *Gradient Boosting*, *AdaBoost* and *Naive Bayes*. The packages used were *pandas*, *matplotlib* and *scikit-learn*. Based on this research proposes a solution to improve the development of models and with this to enhance the future models that will be used for production within the business of the banking institution.

Keywords: Machine learning, Variables selection, Variables transformation, *logistic regression*, assembly techniques, *Python*.

Índice de Contenidos

Índice de cuadros

Índice de figuras

Capítulo 1

Definición del Problema

En este capítulo se abordará de forma general el contexto de la investigación realizada. En primera instancia se tratará el descubrimiento de conocimiento de clientes bancarios a través de modelos predictivos y el valor que tiene esto para el negocio de la institución bancaria. En segundo lugar, se abordará qué son los modelos de *Scoring* y los procesos asociados, junto al gran problema que trae desarrollar un modelo predictivo de este tipo. Finalmente se abordarán los objetivos principales de la memoria. Se espera durante este capítulo dar un marco general de la investigación realizada e informar sobre los beneficios de mejorar un modelo de *Scoring*.

1.1. Descubrimiento de Conocimiento con Modelos Predictivos

Para desarrollar estos modelos, la Institución Bancaria asociada al proyecto cuenta con una metodología que consta de varios procesos, los cuales apuntan a crear un modelo con un buen poder predictivo, con esto, se indica que todas las predicciones serán certeras y la probabilidad de errar es mínima. Cuando un modelo cumple los niveles esperados de predicción, se lleva a producción, es decir, se integra al negocio correspondiente. Con el modelo integrado se logra estimar la pérdida esperada asociada a una cartera de clientes.

Entre los modelos conocidos, los más utilizados son los modelos de *Scoring* ya que éstos entregan como salida un puntaje para un cliente, el cual es aplicable en varios aspectos del negocio y, por ende, facilita el uso en la gestión. Por ejemplo, un puntaje bajo del cliente, indica que éste tiene asociado un mayor riesgo y dependiendo del negocio donde se utiliza el modelo, el riesgo se traduce con el significado especifico para un negocio o contexto.

Para desarrollar un modelo de Scoring, se ejecuta una serie de etapas antes de construirlo; el primero es la definición del problema, el cual es la etapa inicial de un proyecto donde se desarrolla un modelo de *Scoring*, y donde participan los analistas y las personas asociadas al negocio. Luego se realiza la etapa denominada calidad de datos, en la cual se definen las fuentes de datos necesarias y se certifican que estos datos están aptos para reflejar el comportamiento del negocio estudiado, para así dejar claro que el futuro modelo a desarrollar aportará en la gestión del negocio respectivo con los datos que se tienen.

Cuando las fuentes ya están certificadas para el desarrollo del modelo, se ejecuta la etapa de muestreo, en la cual se busca generar un subconjunto de todos los datos, con el cual se trabajará posteriormente para el desarrollo del modelo. Esta muestra debe cumplir una serie de requisitos que garanticen que los resultados obtenidos sean extrapolables a todo conjunto de los datos. Ésta es una de las primeras etapas cruciales para asegurar un buen modelo predictivo.

Teniendo lista la muestra representativa de toda la población, se continua con una etapa clave para el desarrollo del modelo llamado transformación de las variables, cuyo objetivo es ampliar un conjunto de variables generada con las fuentes de datos, para reflejar comportamientos complementarios y facilitar la captura de patrones de acuerdo al modelo en desarrollo. Para lograr esto, se utilizan miradas estadísticas, políticas y de conocimiento del negocio, las cuales permiten pre-seleccionar la cantidad de variables estudiadas, para así centrar el análisis en las verdaderas candidatas a explicar cambios con el modelo.

Terminado la transformación de las variables, se ejecuta la etapa de selección de variables que tiene por objetivo reducir la cantidad de variables a analizar en la etapa posterior con el fin de enfocar los esfuerzos en las mejores candidatas, para así maximizar el poder predictivo de los modelos. Para lograr esto, se aplica una serie de análisis y filtros, los cuales

permiten certificar que se cumplan ciertos estándares mínimos y supuestos necesarios para el correcto desarrollo del modelo.

Por último, luego de todos las etapas anteriores, se procede con la etapa de modelamiento y validación, donde se ejecuta una metodología de modelamiento para construir un modelo de *Scoring* con un único indicador de riesgo que resuma la información de las variables que explican el comportamiento del negocio respectivo. Este indicador debe tener una interpretación simple y con sentido de negocio; además, debe permitir ordenar los clientes de acuerdo a su nivel de riesgo de una forma sencilla.

La tecnica tradicional de estimación corresponde a la regresión logística. Esta técnica de modelamiento es la más común para generar un puntaje asociado al riesgo del cliente; dado esto, la Institución Bancaria también la utiliza y las etapas de pre-entrenamiento de un modelo de *Scoring* se enfocan a que las variables (transformadas y seleccionadas) que entrarán al modelo de regresión logística, se ajusten a ésta para potenciar su funcionamiento y predicción.

El área que desarrolla los modelos ha crecido bastante con los años. Actualmente se desarrollan modelos en masa; por ejemplo, en un año, se producen alrededor de 20 modelos junto a un grupo grande de analistas. Las cantidades de horas invertidas en la metodología para desarrollar un buen modelo antes de que llegue a producción son grandes, ya que toman varios meses, incluso con los grandes proyectos pueden tomar hasta alrededor de dos años con varios analistas trabajando en paralelo para agilizar su desarrollo. Es por esto que la institución bancaria invierte muchos recursos en seguir manteniendo esta área, para así producir cada vez más modelos. Además, también busca que los tiempos de desarrollar un modelo se reduzcan, invirtiendo en mejores tecnologías para manejar los grandes volúmenes de datos que se necesitan para abordar los grandes proyectos.

1.2. El problema

La metodología utilizada en la Institución Bancaria está alcanzando los límites del poder predictivo en sus modelos de *Scoring*, ya que ésta tiene asociados ciertas etapas estandarizadas para desarrollar un modelo, de las cuales algunas están siendo poco efectivas en alcanzar un buen poder de predicción y, por ende, los modelos finales no cuentan con el suficiente poder predictivo que la Institución espera para que se estime bien la pérdida esperada en una cartera de clientes. Cuando un modelo no cumple los niveles esperados de predicción se descarta, provocando la pérdida de todas las horas invertidas por los analistas en el modelo malo. Dado que se debe alcanzar un modelo que prediga bien, el analista debe seguir iterando con las mismos etapas estandarizados hasta que el modelo aumente su poder predictivo. Las iteraciones se vuelven tediosas debido a que hay etapas que requieren muchas horas para volver a obtener un resultado, ya que no están optimizadas en tiempo de ejecución.

Entre algunos de los problemas que limitan el poder predictivo, está que la técnica de modelamiento de regresión logística no funciona en las mejores condiciones cuando las variables que explican el comportamiento del negocio no tienen monotonicidad creciente o decreciente; es por esto que en el proceso de transformación de variables se busca que todas las variables tengan atributos que reflejen alguna de dichas monotonicidades en el riesgo. Pero no se puede lograr esto para todas las variables, ya que algunas tienen un comportamiento no lineal para el negocio. Lamentablemente la mejor opción con esta técnica, es descartar las variables no lineales al menos que sea estrictamente necesario por el negocio que no sean filtradas.

Otro problema de la metodología, es que algunas etapas llevan un tiempo de ejecución muy alto. Esto se mitiga por una parte, corriendo la etapa dentro de un software fuera del horario de trabajo de los analistas, pero esto solo soluciona el problema de no invertir muchas horas de trabajo presencialmente en algunos procesos. No obstante, no existe ninguna optimización real de estas etapas o una mejora inteligente para su ejecución por lo que, cuando es necesario iterar variar veces en el mismo proceso, el analista consume mucho de su tiempo en una pequeña parte de la metodología, provocando el atraso del desarrollo completo del

modelo.

Cuando un modelo no es puesto en producción en las fechas establecidas al comienzo del proyecto, se produce una pérdida indirecta para la Institución Bancaria, ya que en el tiempo en que no se utiliza el modelo para su propósito específico, la gestión del riesgo no mejora en el negocio, por lo que se pierden millones de ganancias monetarias por la gestión no óptima en el manejo de recursos de la institución.

1.3. Objetivos

1.3.1. Objetivo principal

Mejorar la predicción de modelos de *scoring* de una institución bancaria, mediante nuevas técnicas de maquinas de aprendizaje, para apoyar la toma de decisiones automatizadas en los negocios de la institución.

1.3.2. Objetivos específicos

Para lograr el objetivo principal propuesto es necesario, cumplir los siguientes objetivos específicos:

- Diseñar una metodología nueva enfocada a los procesos de Transformación de Variables, Selección de Variables y Modelamiento, para generar mejoras en los modelos actuales en base a técnicas de modelamiento y no por la búsqueda de nuevas variables en etapas anteriores a la transformación.
- Implementar nuevas técnicas de modelamiento que aumenten el poder predictivo de los modelos de *scoring* dentro de los procesos actuales, con el fin de potenciar las decisiones automatizadas en los negocios financieros en que se ejecutan los modelos.
- Mejorar los indicadores asociados a la eficiencia de las etapas a rediseñar (Transformación de Variables, Selección de Variables y Modelamiento) y validar empíricamente el

valor agregado que entregan.

1.4. Beneficios para la Institución Bancaria

Con esta investigación se entregan las bases para aplicar una nueva metodología con las implementaciones necesarias para alcanzar el objetivo principal propuesto, con el fin de aumentar el poder predictivo de los futuros modelos y reducir su tiempo de desarrollo. El beneficio de un modelo con mejor poder predictivo se concreta en que el área donde se integre causará mayor impacto en la gestión del área y sus decisiones, lo cual se traduce en un uso eficiente de recursos y la correcta evaluación del riesgo de los clientes.

Con estos cambios se logrará reducir el trabajo de los analistas de desarrollo de modelos para alcanzar un modelo de *scoring* con buen poder predictivo. Entre estos beneficios, está la optimización de las etapas asociadas a pre-entrenamiento del modelo a través de etapas automatizadas, reduciendo los cambios manuales realizados por los analistas con las herramientas utilizadas hasta ahora. Logrando esto, las etapas de Transformación de las Variables y Selección de Variables serán más rápidas de ejecutar, dando el beneficio de invertir menos tiempo en una o varias iteraciones de ellas con el fin de enfocarse en otras partes de la metodología.

1.5. Metodología a utilizar y alcance

Para lograr el objetivo de esta investigación, se aplicarán las buenas prácticas de CRISP-DM; dado que todas las etapas mencionadas en este capítulo (presentes en la Institución Bancaria) están basadas en esta metodología, es recomendable seguir basándose en ella para construir los modelos de *scoring* y validar su poder predictivo. Por otro lado, la solución tomará en cuenta recomendaciones de otros autores que han investigado temas similares relacionados a construir buenos modelos predictivos para el mundo financiero. Por ultimo la Institución Bancaria necesita que la investigación comience con el proceso de Modelamiento

dado que es el objetivo especifico que más les interesa, dado que implementar nuevas técnicas de modelamiento es imposible actualmente debido a las limitaciones que trae el software SPSS Modeler respecto a nuevas técnicas.

Respecto al alcance, la metodología se desarrollará en el entorno de programación Python, ya que la Institución Bancaria necesita en el futuro una metodología que no dependa principalmente de un software (SPSS Modeler), si no que un entorno de programación (*Scripting*) con el fin de aprovechar las bibliotecas que con el tiempo se han potenciado para implementar soluciones de minería de datos y máquinas de aprendizaje. Esta restricción fue impuesta con la Institución para que la investigación aproveche el potencial de Python, y así, demostrar a los analistas el poder de estas bibliotecas.

Capítulo 2

Estado del Arte

En este capítulo se revisarán conceptos y conocimientos específicos que darán contexto al desarrollo de modelos de *scoring*, además de detallar las etapas en torno a la metodología actual para dejar claro cómo funciona cada una de ellas. Por último, se abarcará el estado del arte respecto a autores que trabajaron en temas de "*Credit Scoring*" y sus resultados de *benchmarking* con varias técnicas de modelamiento.

Se espera entonces informar sobre todos los conceptos necesarios para abarcar esta investigación dentro de la Institución Bancaria.

2.1. Minería de Datos

La minería de datos es un campo de la estadística y las ciencias de la computación, que se refiere a un proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos [7]. Utiliza los métodos de inteligencia artificial, máquinas de aprendizaje, estadística y sistemas de bases de datos. El objetivo general del proceso de minería de datos consiste en extraer conocimiento de un conjunto de datos y transformarlo en una estructura comprensible para su uso posterior. Este puede ser utilizado para mejorar procesos, disminuir costos, aumentar ganancias, etc.

Existen muchos software de minería de datos que permiten a sus usuarios analizar datos recopilados desde muchas dimensiones o ángulos diferentes, resumiendo todo en una serie de relaciones identificadas entre las variables estudiadas. Por lo general, la minería de datos se utiliza para encontrar correlaciones o patrones entre docenas de variables, o para encajar en un contexto ciertos campos de una gran base de datos relacional.

Un proyecto de minería de datos está compuesto de cinco etapas principales:

- Extraer, transformar y cargar datos en el *data warehouse* (ETL[1]).
- Almacenar y administrar los datos en un sistema de bases de datos relacional.
- Dar acceso a los datos a analistas del negocio y profesionales de TI.
- Analizar los datos con aplicaciones especializadas.
- Presentar la información en formatos útiles, como gráficos o tablas.

2.2. Procesos de Minería de Datos

En minería de datos existen varios procesos estándares para alcanzar el objetivo de la disciplina. Se revisarán los procesos más utilizados.

2.2.1. Proceso de descubrimiento del conocimiento (KDD)

Recibe este nombre el proceso que tiene por entrada la base de datos y como salida el subconjunto de patrones que se transformarán en conocimiento, luego de la aplicación de minería de datos. Este proceso cuenta con cinco fases fundamentales que se encuentran en la figura 2.1:

1. **Selección de Datos**: esta etapa consiste en definir un conjunto de datos, o enfocar los esfuerzos en una serie de variables de los mismos. En ésta, es fundamental contar con

[1]Por sus siglas en inglés: Extract, Transform, Load.

un conocimiento previo del negocio, que ayude a definir cuáles variables son relevantes para el estudio y cuáles no. Por ejemplo, si se desea descubrir qué clientes son más susceptibles a un esfuerzo de marketing, casi con certeza el nombre del cliente no será una variable importante para el estudio, pero sí el segmento económico o el nivel de ingresos del mismo.

2. **Preprocesamiento de Datos**: se busca limpiar los datos; esto quiere decir que se tomará una serie de acciones para que los datos no cuenten con inconsistencias u observaciones faltantes/inválidas. Durante esta etapa se realiza una limpieza de los datos:

    - **Faltantes**: en torno a esta situación se puede tomar una serie de acciones, como ignorar datos con observaciones faltantes, llenarlos manualmente, usar una variable global para llenarlos (como N/A, nulo, -inf, etc) o alguna solución con un estadístico, como poner la media del atributo con respecto a todos los datos, usar la media del atributo considerando sólo los datos de la misma clase o el valor más probable del dato.
    - **Inconsistentes**: se generan principalmente por variaciones al momento de ingresarlos, como el uso de diferentes capitalizaciones o faltas de ortografía. Una inconsistencia puede ser, por ejemplo, si en una observación de persona, su tipo de vivienda es “Departamento”, mientras que en otra es “Depar.” o “Dpt”. Se entiende que todas las observaciones hacen referencia a un departamento, pero por errores o decisiones humanas tienen un valor diferente.

3. **Transformación de Datos**: se realizan todas las transformaciones necesarias a los datos para que puedan ser interpretados de mejor manera por los algoritmos de minería de datos. Dependiendo de los algoritmos a aplicar, se requiere aplicar uno o más tipos de transformación, siendo algunas de ellas:

    - **Normalización**: consiste en representar los valores de las observaciones en un intervalo definido; por ejemplo, normalizar los datos para que sus valores estén dentro del rango [0,1]. Este método es de particular importancia cuando se planea utilizar técnicas de *clustering* basadas en distancia, ya que al no aplicarse, se desbalancea la importancia de diferentes variables por culpa de las unidades de

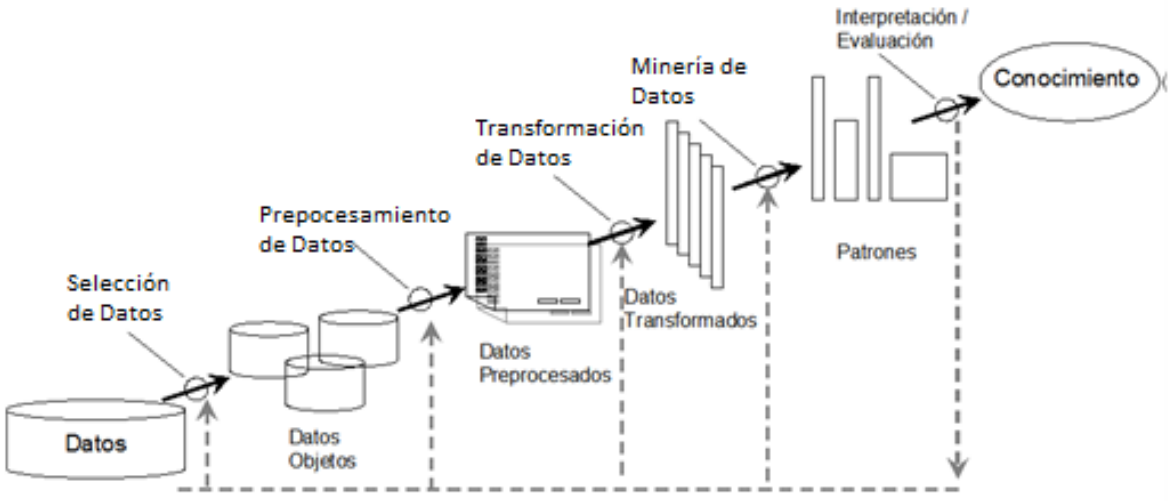
medidas usadas. Por ejemplo, de distorsiona/transforma la distancia, dándole más importancia a una variable de mayor magnitud, como podría ser el ingreso per cápita de una base de datos de clientes (orden de los cientos de miles y millones) respecto de la edad.

- **Agregación**: utilizada cuando se desea agrupar variables. Por ejemplo, pasar una serie de registros de ingreso mensual a una cantidad más reducida de registros de ingreso anual.

4. **Minería de datos**: consiste en la búsqueda de patrones de interés en alguna forma particular de representación, dependiendo del objetivo final de la minería de datos.

5. **Interpretación/Evaluación**: en esta etapa final, se interpretan y evalúan los patrones encontrados, con el fin de juzgar su utilidad para el objetivo final o negocio, además de su asertividad.

Figura 2.1: Fases del descubrimiento del conocimiento (KDD)



2.2.2. *Cross-Industry Standard Process for Data Mining (CRISP-DM)*

CRISP-DM [2] recibe su nombre del acrónimo en inglés y consiste en una metodología con un ciclo de vida de seis etapas, la cual se observa en el figura 2.2:

1. **Entendimiento del negocio**: se busca comprender los objetivos y requerimientos del proyecto desde el enfoque del negocio, para luego transformarlo en un problema de minería de datos y un plan preliminar para alcanzar los objetivos.

2. **Entendimiento de los datos**: empieza con un conjunto de datos inicial y se busca familiarizarse con ellos, identificar problemas de calidad, descubrir una primera mirada o subconjuntos interesantes con el fin formular una hipótesis para información escondida.

3. **Preparación de los datos**: comprende todas las actividades necesarias para generar el *set* de datos final a partir de los datos en bruto.

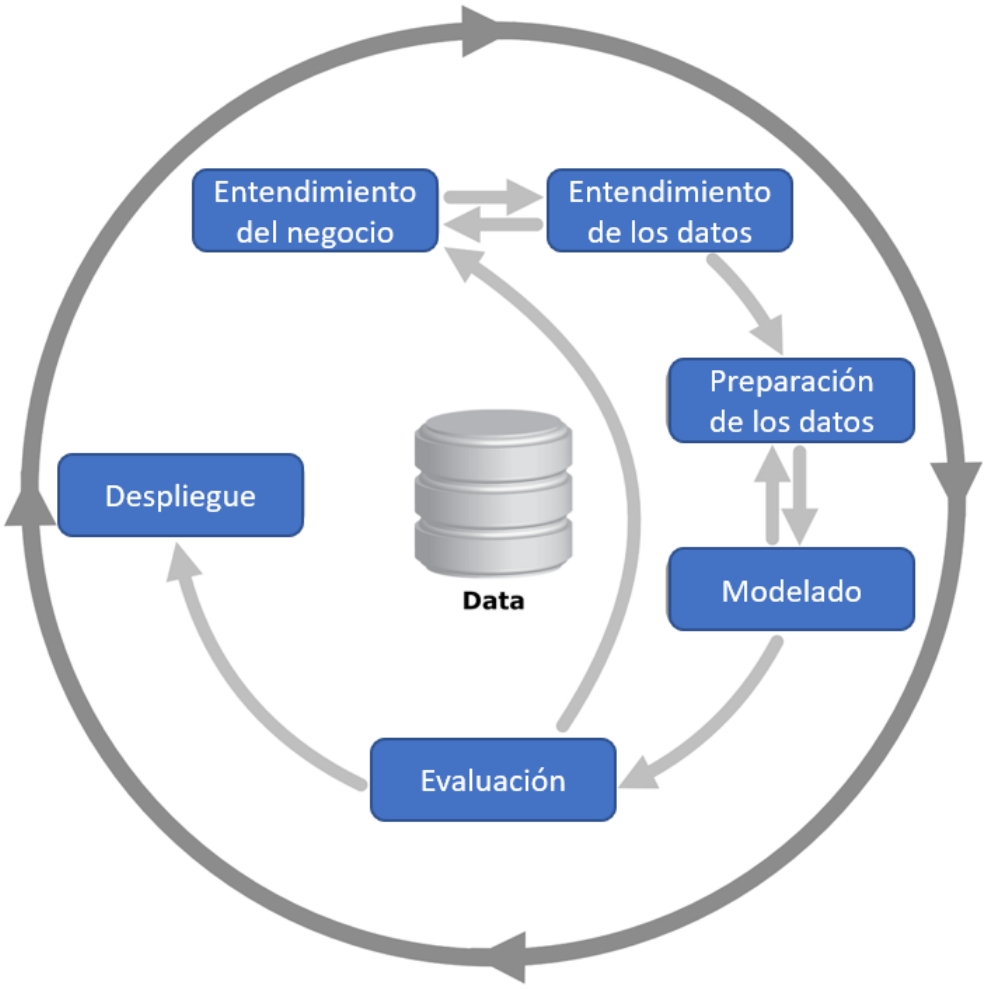
4. **Modelado**: es la aplicación de una o varias técnicas de modelamiento, calibrando sus parámetros a valores óptimos.

5. **Evaluación**: los modelos obtenidos son juzgados y los pasos para construirlos son evaluados con el fin de concluir con seguridad que efectivamente se cumple con los objetivos del negocio.

6. **Despliegue**: el término del modelo por lo general no significa el fin del proyecto. El conocimiento obtenido luego debe ser organizado y desplegado de forma que el cliente final (el negocio) pueda utilizarlo.

La ventaja de CRISP-DM sobre KDD es que está más aterrizado a la aplicación en la industria dado que todas las etapas están centralizadas en la iteración constante con los datos (para cada etapa) y por ende, al momento de volver a iterar en una de las etapas del proceso para corregir algo, se reduce el esfuerzo de ésta. Este es uno de los factores claves de porque las instituciones prefieren descartar KDD.

Figura 2.2: Fases de la metodología CRISP-DM



2.3. Algoritmos de Modelado para la Minería de Datos Predictiva

En esta sección se definirán, a nivel general, algunos algoritmos de aprendizaje supervisado, el cual consiste en una técnica para deducir una función a partir de datos de entrenamiento. La salida de la función puede ser un valor numérico (para algoritmos de regresión) o una etiqueta de clase (para algoritmos de clasificación).

El objetivo del aprendizaje supervisado es crear una función capaz de predecir el valor

correspondiente a cualquier objeto de entrada válido después de haber visto una serie de ejemplos, llamados datos de entrenamiento. Para esto, tiene que generalizar a partir de los datos presentados a las situaciones no vistas previamente.

A continuación, se presentan algunas técnicas que corresponden a los tipos de algoritmos más populares [3];

- **Regresión**: se ocupa de modelar la relación entre las variables que se refina iterativamente usando una medida de error en las predicciones hechas por el modelo. Los métodos de regresión son un caballo de batalla de las estadísticas y han sido confinados en el aprendizaje de la máquina estadística. Algunos de los algoritmos de regresión más populares son *Ordinary Least Squares Regression* (OLSR), *Linear Regression, Logistic Regression* y *Stepwise Regression*.

- **Árboles de decisión**: construyen un modelo de decisiones basadas en valores reales de atributos en los datos. Estos algoritmos generan bifurcaciones en estructuras de árbol hasta que se tome una decisión de predicción para un registro dado. Los algoritmos de árboles de decisión más populares son *Classification and Regression Tree* (CART), *Iterative Dichotomiser* 3 (ID3), C5.0, *Chi-squared Automatic Interaction Detection* (CHAID) y *Conditional Decision Trees*.

- **Bayesianos**: aplican explícitamente el teorema de Bayes para problemas como la clasificación y la regresión. Los algoritmos bayesianos más populares son *Naive Bayes*, *Gaussian Naive Bayes*, *Multinomial Naive Bayes* y *Bayesian Network* (BN).

- **Redes neuronales artificiales**: son modelos que se inspiran en la estructura y/o función de las redes neuronales biológicas. Son una clase de concordancia de patrones que se usan comúnmente para los problemas de regresión y clasificación, aunque realmente son un enorme sub-campo compuesto de cientos de algoritmos y variaciones para todo tipo de tipos de problemas. Los algoritmos más populares son Perceptrón, *Back-Propagation*, *Hopfield Network* y *Radial Basis Function Network* (RBFN).

- **Ensamblaje**: son modelos compuestos de muchos otros más débiles que son independientemente entrenados y cuyas predicciones se combinan de alguna manera para

hacer la predicción general. Se pone mucho empeño en qué tipos de modelos débiles se deben combinar y en las formas de combinarlos. Los algoritmos de ensamblaje más populares son *Boosting*, *Bootstrapped Aggregation (Bagging)*, *AdaBoost*, *Gradient Boosting Machines* (GBM), *Random Forest*, *Extra Tree*.

- **Máquinas de vectores de soporte (SVM)**: una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases lo más posible mediante un hiperplano de separación definido como el vector entre los dos puntos, de las dos clases, más cercanos al que se llama vector soporte. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de los espacios a los que pertenezcan, pueden ser clasificadas a una u otra clase. Más formalmente, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una clasificación correcta.

2.4. Regresión Logística

La regresión logística es un modelo estadístico de regresión que posee una variable dependiente categórica [5], la que puede ser de naturaleza dicotómica (regresión logística binaria o binomial) o con más valores (regresión logística multinomial). El modelo estima la variable de respuesta mediante una probabilidad, la cual se ajusta mediante la función logística [4].

Las variables explicativas, independientes o covariables, deben ser dicotómicas, por lo que en caso de que la naturaleza de algunas de éstas sea de tipo continua o presente más de una categoría, se debe aplicar una transformación de los datos, que permita codificar la información en variables binarias.

Por sus características, los modelos de regresión logística pueden ser utilizados para los siguientes objetivos:

- Cuantificar la importancia de la relación existente entre cada una de las covariables

y la variable dependiente, lo que lleva implícito también clarificar la existencia de interacción y confusión entre covariables respecto a la variable dependiente, es decir, conocer el *odds ratio* para cada variable explicativa.

- Clasificar individuos dentro de las categorías (presente/ausente) de la variable dependiente, según la probabilidad que tenga de pertenecer a una de ellas dada la presencia de determinadas covariables.

La regresión logística es una de las herramientas estadística más populares en la industria debido a su capacidad para el análisis de datos y su fácil explicación al negocio. El objetivo principal que resuelve esta técnica es el de modelar cómo influye en la probabilidad de aparición de un suceso, habitualmente dicotómico, la presencia o no de diversos factores y el valor o nivel de los mismos. También puede ser usada para estimar la probabilidad de aparición de cada una de las posibilidades de un suceso con más de dos categorías (politómico).

La regresión logística tiene ciertos supuestos que es recomendable tener en cuenta para desarrollar un buen modelo en base a esta técnica, los que se exponen a continuación:

- Puede manejar cualquier tipo de relación no necesariamente lineal, ya que aplica una transformación logarítmica no lineal.
- Las variables explicativas pueden ser continuas o discretas (categóricas u ordinales) y no necesitan ser independientes, pero en caso de serlo, la regresión da una solución estable.
- Se debe tener especial consideración en que la relación entre la variable independiente y la probabilidad del suceso no cambie de sentido, ya que en este caso el modelo logístico deja de tener la interpretabilidad deseada.
- Si las variables que intervienen están muy correlacionadas, el modelo logístico estará desprovisto de sentido y los valores de sus coeficientes no serán interpretables.

2.5. Preprocesamiento y Selección de variables

Para la preparación de los datos que se utilizarán en un modelo, es necesario realizar un proceso exhaustivo para elegir los mejores datos, transformarlos a una forma de fácil interpretación para el modelo y seleccionar a los mejores candidatos, con el fin de que la técnica de modelamiento se utilice con las mejores variables y se reduzca su dimensionalidad.

Las herramientas utilizadas se centran en tres etapas: Muestreo, Transformación de las variables y Selección de variables. A continuación, se detallarán algunas herramientas utilizadas en estas últimas dos etapas de preparación de los datos.

2.5.1. Transformación de las Variables

El tratamiento de variables es una etapa clave dentro del proceso de desarrollo de un nuevo modelo. Ésta tiene por objetivo ampliar un conjunto de variables generada con las fuentes de datos, para reflejar comportamientos complementarios y facilitar la captura de patrones de acuerdo al modelo en desarrollo. Esta etapa tiene, además, varias actividades asociadas a filtros preliminares de variables en función de su poder predictivo. Para esto se utilizan miradas estadísticas, políticas y de conocimiento del negocio, las cuales permiten disminuir la cantidad de variables estudiadas y centrar el análisis en las verdaderas candidatas a explicar cambios en la variable dependiente.

Para lograr esto, la transformación de variables debe pasar por tres actividades, la primera es la creación de nuevas variables, la segunda es la discretización de las variables y la tercera es el análisis bivariado, las que se detallarán brevemente a continuación:

Creación de nuevas variables

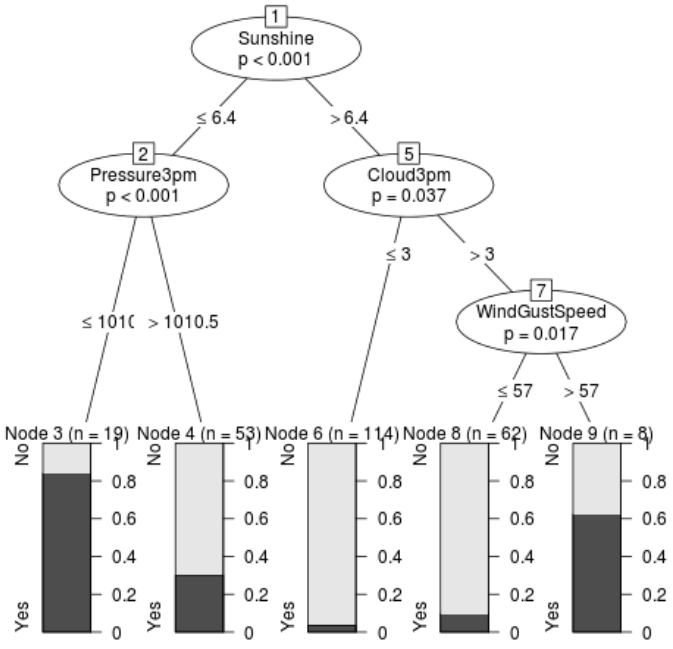
Respecto al trabajo realizado junto al negocio, se construyen nuevas variables explicativas del negocio para una posterior evaluación del poder predictivo luego del análisis bivariado.

Discretización

Esta actividad consiste en la transformación de las variables a tramos o categorías con el fin de poder aplicar un análisis bivariado posteriormente. Para asignar un puntaje en base a cada variable, es necesario discretizar tanto las variables continuas como categóricas; para esto, se aplican técnicas de tramificación como, por ejemplo, arboles de clasificación y regresión, que se explican brevemente a continuación (en los cuadros 2.1 a 2.2 se ejemplifican resultados de discretización):

Arboles de Clasificación y Regresión: uso de algoritmos de árbol con el fin de generar y/o identificar grupos de individuos que tengan características similares. La ventaja de estos algoritmos es que la noción de similitud es calculada mediante indicadores estadísticos. Los arboles de regresión son técnicas heurísticas que utilizan herramientas estadísticas para generar submodelos para explicar una variable de respuesta (ver figura 2.3). Algunas de estas técnicas son:

Figura 2.3: Ejemplo de árbol de clasificación para generar tramos/categorías



2.3 [Fuente: https://www.stat.auckland.ac.nz/paul/RGraphics/chapter1]

- **CHAID**: es una técnica heurística que permite generar segmentos en las variables de análisis. La técnica CHAID mezcla las técnicas de árbol AID con el test de *Chi Cuadrado* para fundir clases. La técnica se basa en tres pasos: fusión, división y detención, donde la profundidad del árbol se obtiene repitiendo estos pasos de manera recursiva. Los pasos se detallan a continuación:

  - Fusión: el objetivo de este paso es fusionar todas aquellas categorías que no sean estadísticamente significativas. Cada categoría considerada como diferente pasa a ser un nodo de división para el árbol que se entregara.

  - División: la "mejor" división para cada predictor es encontrada en la etapa de fusión. La etapa de división sirve para seleccionar el mejor predictor. La selección se basa en la comparación de los p-valores obtenidos en última etapa de fusión.

  - Detención: este paso de detención verifica si el proceso de crecimiento del árbol debería parar de acuerdo a una regla de detención preestablecida.

- **CRT** (*Classification and Regresion Tree*): los arboles de clasificación y regresión son técnicas que, al igual que las técnicas CHAID, buscan generar segmentos en las variables explicativas para describir a una variable de respuesta. La gran diferencia entre ambas es el algoritmo que se utiliza para la construcción de estos. La diferencia es que CHAID tiene tres etapas y CRT, en cambio, utiliza una división recursiva para seleccionar la mejor partición sobre las variables explicativas, la cual permite explicar a la variable de respuesta.

Cuadro 2.1: Ejemplo de discretización de una variable continua

Categoría	**Buenos**	**Indeterminados**	**Malos**	**Total**
(-inf, -99.0]	211	243	1113	1567
(-99.0, 10.7]	645	378	2865	3888
(10.7, 365.5]	95	32	248	375
(365.5, +inf)	46	16	79	141
Total	**997**	**669**	**4305**	**5971**

Cuadro 2.2: Ejemplo de discretización de una variable categórica

Categoría	**Buenos**	**Indeterminados**	**Malos**	**Total**
Media	332	152	1870	2354
Técnica	553	436	2186	3175
Universitaria	112	81	249	442
Total	**997**	**669**	**4305**	**5971**

Análisis Bivariado

El análisis bivariado es una actividad donde se calculan varios indicadores estadísticos en los tramos/categorías para analizar el poder predictivo de cada variable. Los indicadores son los siguientes:

- Recuento de registros por tramo y desempeño por categoría.
- Porcentaje de registros por tramo y desempeño por categoría.
- Porcentaje acumulado de registros por tramo y desempeño por categoría.
- Tasas de Malos (*Bad Rate*)
- *Odds Ratio* por tramo
- *Weight of Evidence* (WoE)
- *Information Value* (IV)
- Diferencias absolutas
- KS

A continuacíon, se definen los más importantes para el análisis bivariado.

- ***Bad Rate***: determina la tasa de clientes malos dentro de una categoría, calculada entre el total de fracasos sobre los fracasos más los éxitos. En una correcta segmentación el orden de los *Bad Rates*, deben presentar una tendencia lineal. Se calcula de la siguiente forma:

$$BadRate = \frac{Fracasos}{Fracasos - Exitos} \tag{2.1}$$

- ***Weights of Evidence (WoE)***: proporciona herramientas flexibles para recodificar los valores en las variables predictoras continuas y categóricas en categorías discretas de forma automática, y asignar a cada una un valor WoE único. Esta recodificación se lleva a cabo de manera que produzca las mayores diferencias entre los grupos recodificados con respecto a los valores de WoE a través de algoritmos de clasificación óptimo. Se obtiene de la siguiente forma:

$$WoE = \ln(\frac{CNE}{CE}) \tag{2.2}$$

donde:

CNE: es la proporción de fracaso en la categoría (casos no exitosos).

CE: es la proporción de éxito en la categoría (casos exitosos).

- ***Information Value (IV)***: es un indicador para determinar el grado de vinculación entre una variable y el éxito de respuesta. El valor calculado de este estadístico entrega una medida de fuerza de asociación entre la variable y la respuesta [1]. Está basado en la medida de divergencia de Kullback, la cual es utilizada en teoría de probabilidad y teoría de la información. En el contexto de desarrollar un modelo, es natural que algunas variables sean menos importantes que otras frente a una variable de respuesta, por lo que el IV permite comparar la potencia de dos o más variables. Se calcula de la siguiente manera:

$$IV = \sum_i (CNE_i - CE_i) * WoE_i \tag{2.3}$$

Estos tres indicadores son expuestos en una tabla bivariada, la cual muestra la información para cada variable de forma ordenada y apropiada para que los analistas verifiquen que la variable esta apta para continuar en el proceso. En el caso de que la variable no cumpla con ciertas cualidades, ésta debe ser retramificada hasta que cumpla todos los requisitos; en caso contrario, debe ser filtrada.

Los requisitos de una buena variable están asociados a que esta sea monotónica para la regresión logística; para esto, cada tramo de la variable debe tener un *Bad Rate* y un *WoE* que sean monotonicamente crecientes o decrecientes. Cuando una variable no es de esta forma, se puede corregir mediante una retramificación, es decir, la unión entre dos tramos vecinos. El análisis bivariado es una actividad iterativa, ya que es frecuente que los analistas deban iterar varias veces para lograr que todas las variables se comporten de forma monotónica.

2.5.2. Selección de Variables

Para seleccionar variables se aplican ciertos filtros iniciales antes de aplicar un algoritmo de selección; los filtros son los siguientes:

- **Análisis de la concentración de datos dentro de las categorías**: existe un estándar asociado a la proporción mínima de casos por tramo, el cual establece un filtro a las variables que presenten menos de un 1 % de los casos agrupados en una misma categoría. Este aspecto por lo general se configura en los software estadísticos, lo cual impide una categorización con estas características para las variables continuas; sin embargo, puede existir la posibilidad de que se genere este caso para las variables categóricas.
- **Análisis del Poder predictivo de la Variable**: existe un estándar de modelamiento, el cual establece un valor mínimo de 2 % de *IV* para que la variable sea considerada predictiva.
- **Análisis de la Estabilidad de la Variable**: tiene por objetivo validar que los tramos definidos en el Análisis Bivariado se mantienen estables en la población. Para esto, se calcula el Índice de Estabilidad Poblacional (IEP) para cada variable.

- **Filtro por índice de Estabilidad Poblacional (IEP)**: es un indicador que mide la similitud entre dos poblaciones distintas. Su uso permite saber si la categorización óptima aplicada a una variable se mantiene estable en un instante de tiempo determinado. La fórmula que determina este indicador es el mismo que el IV, donde la principal diferencia entre estos dos indicadores es la interpretación que se le da a cada uno. Se puede obtener de la siguiente manera:

$$IEP = \sum (PE_i - PR_i) * ln(\frac{PE_i}{PR_i}) \tag{2.4}$$

donde:

PE: es la proporción esperada de la categoría.

PR: es la proporción real de la categoría.

La forma de interpretar este indicador se puede observar del cuadro 2.3. Cuando el IEP es mayor a 0,25, la variable debe ser filtrada debido a que no representa a la población. Para el caso de un IEP de [0,1; 0,25] debe ser decidido por criterio experto del analista para filtrar o no.

Cuadro 2.3: Significado del valor de IEP

IEP	Población
<0.1	Sin cambios en la población
0.1 a 0.25	Cambio menor en la población
>0.25	La población cambió

Para garantizar la aplicabilidad de los tramos, una vez puesto en producción el modelo, se debe maximizar la cantidad de periodos donde se realiza el análisis de estabilidad. Para esto se utilizan en conjunto las bases de desarrollo y desempeño, además de toda la información nueva obtenida para periodos posteriores a los considerados en la muestra. Esto permite analizar la estabilidad por periodo de cada variable, y estudiar posibles comportamientos anómalos en base al estándar definido.

Luego de los filtros iniciales, se pueden seguir aplicando técnicas de selección de variables. Algunas de estas son las siguientes:

- **Selección de Variables por Componentes Principales**: una forma de agrupar variables en base a un criterio estadístico, es mediante la aplicación de un Análisis de Componentes Principales (PCA) en el cual se asocian variables a diferentes factores y se interpretan como familias de variables. Un análisis de componentes principales tiene sentido si existen altas correlaciones entre las variables, ya que esto es indicativo de que hay información redundante y, por lo tanto, pocos factores explicarán gran parte de la variabilidad total. La elección de los factores se realiza de tal forma que el primero recoja la mayor proporción posible de la variabilidad original, luego el segundo factor debe recoger la máxima variabilidad posible no recogida por el primero, y así sucesivamente. Del total de factores se elegirán aquellos que recojan el porcentaje de variabilidad que se considere suficiente. A éstos se les denominará componentes principales.

- **Selección por Correlación**: se llama correlación al grado de dependencia mutua entre dos variables; por su parte, el coeficiente de correlación intenta medir la intensidad con que dos variables están relacionadas. Este concepto está directamente relacionado con el concepto de curva de regresión. Mediante la regresión, se expresa la estructura funcional de la relación existente entre las variables, ajustando a la nube de puntos dada por los pares de valores de las dos variables a una curva de la mejor forma posible. El ajuste será de la forma $Y = f(X)+e$ o $X = f(XY)+e$, donde **e** es el error. El coeficiente de correlación mide la calidad de este ajuste. Cuando la curva es recta, la regresión es lineal; en este caso el coeficiente de correlación se llamará coeficiente de correlación lineal y mide el grado de asociación lineal entre las variables. Algunos coeficientes de correlación útiles son los siguientes:

    - **Correlación de Pearson**: este coeficiente, pensado para variables cuantitativas (escala mínima de intervalo), es un índice que mide el grado de covariación entre distintas variables relacionadas linealmente, que poseen una distribución normal bivariada conjunta. Esto significa que puede haber variables fuertemente relacionadas, pero no de forma lineal, en cuyo caso se sugiere no proceder a aplicarse la correlación de Pearson.
    - **Correlación Rho de Spearman**: es la versión no paramétrica del coeficiente de

correlación de Pearson, que se basa en los rangos de los datos en lugar de hacerlo en los valores reales. Resulta apropiado para datos ordinales o los de intervalo que no satisfagan el supuesto de normalidad. La ventaja de Spearman es que, al ser una técnica no paramétrica, es de libre distribución probabilística. Además, los supuestos son menos estrictos y es robusto a la presencia de *outliers* (es decir, permite ciertos desvíos del patrón normal). La manifestación de una relación causa-efecto es posible sólo a través de la comprensión de la relación natural que existe entre las variables y no debe manifestarse solamente por la existencia de una fuerte correlación.

- **Correlación Tau-b de Kendall**: es una medida no paramétrica de asociación para variables ordinales o de rangos que tiene en consideración los empates. El signo del coeficiente indica la dirección de relación y su valor absoluto indica la magnitud de la misma, de tal modo que los mayores valores absolutos indican relaciones más fuertes. Los valores posibles van de -1 a 1, pero un valor de -1 o +1 sólo se puede obtener a partir de tablas cuadradas.

Cuando dos o más variables explicativas de un modelo están altamente correlacionadas en la muestra, es muy difícil separar el efecto parcial de cada una de estas variables sobre la variable dependiente. La información muestral que incorpora una de estas variables es casi la misma que el resto de las correlacionadas con ella. En el caso extremo de multicolinealidad exacta, no es posible estimar separadamente estos efectos sino una combinación lineal de ellos. La selección de variables por correlación, es aplicada a las variables con el fin de quitar los atributos redundantes, que pueden estar repitiendo la misma información a los modelos.

Existe un estándar asociado a la interpretación de los coeficientes de correlación de Pearson, Spearman y Tau-b de Kendall, los cuales permiten establecer, si el conjunto de variables a utilizar en el modelo presenta problemas de alta correlación. Para las variables cuantitativas simétricas, normalmente distribuidas, se utiliza el coeficiente de correlación de Pearson; por su parte, si los datos no están normalmente distribuidos o tienen categorías ordenadas, se utilizan la Tau-b de Kendall o de Spearman, las cuales miden la asociación entre órdenes de los rangos. Los coeficientes pueden tomar valores entre -1 y +1, lo cual se interpreta como una relación negativa y positiva perfecta,

respectivamente.

Existe un estándar establecido para la inclusión de una variable en un modelo, el cual establece un valor máximo de 0.5 en el índice para una correlación positiva, y un estándar mínimo de -0.5 para el caso de una correlación negativa, valores que se encuentran definidos en el cuadro 2.4.

Cuadro 2.4: Significado del valor del coeficiente de correlación

Correlación		
Valor Rango	**Fuerza y Dirección**	
r = +1	Perfecta	Positiva
0.9 <r <1	Fuerte	
0.5 <r <= 0.9	Moderada	
0 <r <= 0.5	Débil	
r = 0	Sin Correlación	
-0.5 <= r <0	Débil	Negativa
-0.9 <= r <-0.5	Moderada	
-1 <r <-0.9	Fuerte	
r = -1	Perfecta	

El estándar establece como requisito que las variables candidatas sólo puedan poseer una correlación débil. Es posible que, para casos de borde y en base a las características particulares de cada variable, los analistas y el negocio decidan mantener una variable a pesar que posee una correlación definida como Moderada. Sin embargo, la regla general es que éstas deben ser eliminadas del conjunto de variables a evaluar en el modelo.

Este selección busca eliminar problemas de correlación, asignando prioridad a las variables con **mejor poder predictivo**, de forma que se realice una **selección inteligente**, que elimine problemas por correlación, manteniendo las mejores variables candidatas. El proceso puede ser realizado tanto sobre las variables brutas, es decir en su forma original, como con los tramos definidos en el Análisis Bivariado. El criterio de decisión sobre el tipo de variables aplicadas, depende de la cantidad de variables candidatas a

evaluar en el modelo, ya que en caso de que no se filtren muchas variables, y en consecuencia queden muchas a evaluar, se realizan dos pasos: donde en primer lugar se eliminan las correlacionadas por variables brutas, y luego se aplica la selección sobre las variables tramificadas.

2.6. Descubrimiento de conocimiento de clientes bancarios a través de Modelos

Actualmente en Chile, las instituciones bancarias buscan averiguar el riesgo asociado a sus clientes en torno a los productos que ofrecen (créditos) para estimar un perfil de deudor para una cartera de clientes. Con esta información se clasifica al cliente en un perfil y se estima la pérdida esperada en conjunto a varios clientes. Para lograr esta predicción, se desarrollan modelos de provisiones para calcular ciertos indicadores de pérdida, los cuales están normados [8] por la SBIF[2] con reglas asociadas al tipo de modelo de provisión que se construirá con analistas de desarrollo de modelos en la Institución Bancaria.

El objetivo de que estos modelos estén regulados por la SBIF, es para que cumplan ciertos estándares básicos del mundo financiero y no se salgan de una predicción "lógica" para el negocio específico que impactará él modelo. Por otra parte, el gran aporte de un modelo al negocio es la estimación de valores para el futuro, con el fin de mejorar la gestión realizada en la toma de decisiones, y así, no subestimar o sobreestimar el gasto asociado a los riesgos de pérdida; mientras mejor sea la estimación, menor será el riesgo asociado al negocio y, por ende, habrá menos pérdidas monetarias.

Los modelos se pueden aplican en distintas partes del negocio dentro de una institución bancaria, ya sea para otorgar un crédito a un cliente, ver el comportamiento de pago de un cliente o la forma de llegar a cobrarle a un cliente, entre otros.

[2]SBIF: Superintendencia de Bancos e Instituciones Financieras, es el organismo encargado de supervisar a las empresas bancarias, así como de otras entidades, en resguardo de los depositantes u otros acreedores y del interés público y su misión es velar por el buen funcionamiento del sistema financiero [Fuente: sbif.cl].

2.6.1. Modelos de *Scoring* y la predicción del riesgo

Una institución bancaria, al tener modelos para predecir el riesgo asociado a sus clientes, puede tomar decisiones inteligentes y certeras para disminuir su riesgo al mejorar su gestión para reducir la cartera de los distintos deudores. Para esto se desarrollan ciertos modelos, como los de *Scoring*, los cuales reciben como entrada información relevante del cliente y como salida entregan un puntaje el cual se utiliza para clasificar a un cliente en una de las categorías de deudor establecida por la SBIF (Figura 2.4). Conociendo esta categoría asociada al cliente, se obtienen indicadores para estimar la perdida esperada por él cliente.

Figura 2.4: Indicadores para los perfiles de deudores en 2 tipos de carteras

Tipo de Cartera	Categoría del Deudor	Probabilidades de Incumplimiento (%)	Pérdida dado el Incumplimiento (%)	Pérdida Esperada (%)
Cartera Normal	A1	0,04	90,0	0,03600
	A2	0,10	82,5	0,08250
	A3	0,25	87,5	0.21875
	A4	2,00	87,5	1,75000
	A5	4,75	90,0	4,27500
	A6	10,00	90,0	9,00000
Cartera Subestándar	B1	15,00	92,5	13,87500
	B2	22,00	92,5	20,35000
	B3	33,00	97,5	32,17500
	B4	45,00	97,5	43,87500

Gracias a la utilización de los modelos de *Scoring* dentro de las gestiones de la institución bancaria, se logra constituir oportunamente las provisiones necesarias y suficientes para cubrir las pérdidas esperadas asociadas a las características de los deudores y sus créditos, que determinan el comportamiento de pago y la posterior recuperación de las perdidas.

Para desarrollar un modelo de *Scoring*, las instituciones utilizan sus propias metodologías de minería de datos para lograr extraer el conocimiento de las bases de datos de sus clientes. Dependiendo de la efectividad de la metodología utilizada por los analistas, se logra entrenar un modelo que tiene asociado un **poder predictivo**, el cual es el indicador más relevante para decir con precisión si el modelo logrará predecir bien o no. Para llegar a calcular este indicador se utilizan varios métodos estadísticos, los cuales se aplican para validar el modelo.

2.6.2. *Benchmarking* de técnicas de modelamiento en *Credit Scoring*

En la comunidad científica, existen varios investigadores que se han dedicado a comparar el rendimiento y la capacidad de predicción de varias técnicas de modelamiento. La técnica que siempre se compara con el resto es la regresión logística, ya que esta técnica es la más utilizada en la industria para desarrollar modelos de *Scoring* para clasificar clientes por riesgo.

Stefan Lessmann, Bart Besens, Hsin-VonnSeow y Lyn C. Thomas [6], a través de un largo trabajo de varios años, recopilaron los resultados de varios autores expuestos en el siglo XXI en artículos científicos (Figura 2.5). Estos resultados exponen la comparación entre 41 técnicas de modelamiento, expuestas en la figura 2.6. Estas 41 técnicas se agrupan por tres tipos; el primero corresponde a 16 clasificadores individuales, el segundo a 8 técnicas de ensamblaje homogéneas, y el ultimo tipo a 17 técnicas de ensamblaje heterogéneas.

En base a las métricas de desempeño de AUC, PCC, BS, H, PG y KS, se observa que las mejores técnicas son las que mantienen estos indicadores al mínimo. Dado estos resultados, los algoritmos con mejor desempeño son los de ensamblaje heterogéneo directamente estático, siendo el mejor *Hill-climbing Ensemble Selection with Bootstrap Sampling (HCES-Bag)*.

Finalmente, los autores escogen las técnicas de *Multilayer Perceptron Artificial Neural Network* (ANN), *LogisticRegression* (LR), *Random Forest* (RF) y *HCES-Bag* para realizar una comparación completa respecto a estas buenas técnicas para *credit scoring*. El resultado se expone en la figura 2.7. Como puede observarse en ésta, la regresión logística es superada por todas estas técnicas en desempeño, evidenciando que las nuevas técnicas de este siglo han logrado superar a la regresión logistica para desarrollar un modelo de *scoring*. Lamentablemente las técnicas más nuevas como HCES-Bag y RF no se encuentran en los software de minería de datos debido a lo reciente y complicado de estandarizar estas técnicas computacionalmente.

Figura 2.5: Técnicas de modelamiento utilizadas para el estudio de *benchmarking*

	BM selection	Classification algorithm	Acronym	Models
Individual classifier (16 algorithms and 933 models in total)	n.a.	Bayesian Network	B-Net	4
		CART	CART	10
		Extreme learning machine	ELM	120
		Kernalized ELM	ELM-K	200
		k-nearest neighbor	kNN	22
		J4.8	J4.8	36
		Linear discriminant analysis[1]	LDA	1
		Linear support vector machine	SVM-L	29
		Logistic regression[1]	LR	1
		Multilayer perceptron artificial neural network	ANN	171
		Naive Bayes	NB	1
		Quadratic discriminant analysis[1]	QDA	1
		Radial basis function neural network	RbfNN	5
		Regularized logistic regression	LR-R	27
		SVM with radial basis kernel function	SVM- Rbf	300
		Voted perceptron	VP	5
	Classification models from individual classifiers		**16**	**933**
Homogenous ensembles	n.a.	Alternating decision tree	ADT	5
		Bagged decision trees	Bag	9
		Bagged MLP	BagNN	4
		Boosted decision trees	Boost	48
		Logistic model tree	LMT	1
		Random forest	RF	30
		Rotation forest	RotFor	25
		Stochastic gradient boosting	SGB	9
	Classification models from homogeneous ensembles		**8**	**131**
Heterogeneous ensembles	n.a.	Simple average ensemble	AvgS	1
		Weighted average ensemble	AvgW	1
		Stacking	Stack	6
	Static direct	Complementary measure	CompM	4
		Ensemble pruning via reinforcement learning	EPVRL	4
		GASEN	GASEN	4
		Hill-climbing ensemble selection	HCES	12
		HCES with bootstrap sampling	HCES-Bag	16
		Matchting pursuit optimization ensemble	MPOE	1
		Top-*T* ensemble	Top-*T*	12
	Static indirect	Clustering using compound error	CuCE	1
		k-Means clustering	k-Means	1
		Kappa pruning	KaPru	4
		Margin distance minimization	MDM	4
		Uncertainty weighted accuracy	UWA	4
	Dynamic	Probabilistic model for classifier competence	PMCC	1
		k-nearest oracle	kNORA	1
	Classification models from heterogeneous ensembles		**17**	**77**
Overall number of classification algorithms and models			**41**	**1141**

2.5 Fuente: [Stefan Lessmanna, H Seowb, Bart Baesenscd, and Lyn C Thomas D. Benchmarkingstate-of-the-art classification algorithms for credit scoring: A ten-year update. In Credit Research Centre, Conference Archive, 2013]

Figura 2.6: Resultados técnicas utilizadas para el estudio de *benchmarking*

Classifier family	BM selection	Classifier	AUC	PCC	BS	H	PG	KS	AvgR	High score
Individual classifier	n.a.	ANN	*16.2* (.000)	*18.6* (.000)	27.5 (.000)	*17.9* (.000)	*14.9* (.020)	*17.6* (.000)	*18.8*	*14*
		B-Net	27.8 (.000)	26.8 (.000)	20.4 (.000)	28.3 (.000)	23.7 (.000)	26.2 (.000)	25.5	30
		CART	36.5 (.000)	32.8 (.000)	35.9 (.000)	36.3 (.000)	25.7 (.000)	34.1 (.000)	33.6	38
		ELM	30.1 (.000)	29.8 (.000)	35.9 (.000)	30.6 (.000)	27.0 (.000)	27.9 (.000)	30.2	36
		ELM-K	20.6 (.000)	19.9 (.000)	36.8 (.000)	19.0 (.000)	23.0 (.000)	20.6 (.000)	23.3	26
		J4.8	36.9 (.000)	34.2 (.000)	34.3 (.000)	35.4 (.000)	35.7 (.000)	32.5 (.000)	34.8	39
		k-NN	29.3 (.000)	30.1 (.000)	27.2 (.000)	30.0 (.000)	26.6 (.000)	30.5 (.000)	29.0	34
		LDA	21.8 (.000)	20.9 (.000)	16.7 (.000)	20.5 (.000)	24.8 (.000)	21.9 (.000)	21.1	20
		LR	20.1 (.000)	19.9 (.000)	*13.3* (.000)	19.0 (.000)	23.1 (.000)	20.4 (.000)	19.3	16
		LR-R	22.5 (.000)	22.0 (.000)	34.6 (.000)	22.5 (.000)	21.4 (.000)	21.4 (.000)	24.1	28
		NB	30.1 (.000)	29.9 (.000)	23.8 (.000)	29.3 (.000)	22.2 (.000)	29.1 (.000)	27.4	33
		RbfNN	31.4 (.000)	31.7 (.000)	28.0 (.000)	31.9 (.000)	24.1 (.000)	31.7 (.000)	29.8	35
		QDA	27.0 (.000)	26.4 (.000)	22.6 (.000)	26.4 (.000)	23.6 (.000)	27.3 (.000)	25.5	31
		SVM-L	21.7 (.000)	23.0 (.000)	31.8 (.000)	22.6 (.000)	19.7 (.000)	21.7 (.000)	23.4	27
		SVM-Rbf	20.5 (.000)	22.2 (.000)	31.8 (.000)	22.0 (.000)	21.7 (.000)	21.3 (.000)	23.2	25
		VP	37.8 (.000)	36.4 (.000)	31.4 (.000)	37.8 (.000)	34.6 (.000)	37.6 (.000)	35.9	40
Homogeneous ensemble	n.a.	ADT	22.0 (.000)	18.8 (.000)	19.0 (.000)	21.7 (.000)	19.4 (.000)	20.0 (.000)	20.2	17
		Bag	25.1 (.000)	22.6 (.000)	18.3 (.000)	23.5 (.000)	25.2 (.000)	24.7 (.000)	23.2	24
		BagNN	15.4 (.000)	17.3 (.000)	12.6 (.000)	16.5 (.000)	*15.0* (.020)	16.6 (.000)	15.6	13
		Boost	16.9 (.000)	16.7 (.000)	25.2 (.000)	18.2 (.000)	19.2 (.000)	18.1 (.000)	19.0	15
		LMT	22.9 (.000)	23.4 (.000)	15.6 (.000)	25.1 (.000)	20.1 (.000)	22.9 (.000)	21.7	22
		RF	*14.7* (.000)	*14.3* (.039)	*12.6* (.000)	*12.8* (.004)	19.4 (.000)	*15.3* (.000)	*14.8*	*12*
		RotFor	22.8 (.000)	21.9 (.000)	23.0 (.000)	21.1 (.000)	21.6 (.000)	22.9 (.000)	22.2	23
		SGB	21.0 (.000)	19.9 (.000)	20.8 (.000)	21.2 (.000)	22.5 (.000)	20.8 (.000)	21.0	19
Heterogeneous ensemble	none	AvgS	8.7 (.795)	*10.8* (.812)	*6.6* (.628)	9.2 (.556)	12.0 (.420)	9.2 (.513)	9.4	4
		AvgW	*7.3* (/)	12.6 (.578)	7.9 (.628)	**7.3** (/)	**10.2** (/)	**7.9** (/)	*8.9*	*2*
		Stack	30.6 (.000)	26.6 (.000)	37.4 (.000)	29.6 (.000)	30.7 (.000)	29.5 (.000)	30.7	37
	Static direct	CompM	18.3 (.000)	15.3 (.004)	36.5 (.000)	17.2 (.000)	20.0 (.000)	18.2 (.000)	20.9	18
		EPVRL	8.2 (.795)	10.8 (.812)	6.8 (.628)	9.3 (.556)	13.7 (.125)	11.0 (.226)	10.0	5
		GASEN	8.6 (.795)	10.6 (.812)	6.5 (.628)	9.0 (.556)	*11.4* (.420)	*9.0* (.513)	9.2	3
		HCES	10.9 (.191)	11.7 (.812)	7.5 (.628)	10.2 (.449)	14.8 (.020)	13.1 (.010)	11.4	9
		HCES-Bag	*7.7* (.795)	**9.7** (/)	**5.8** (/)	*8.2* (.559)	12.5 (.420)	9.2 (.513)	**8.8**	*1*
		MPOE	9.9 (.637)	10.1 (.812)	9.4 (.126)	9.9 (.524)	15.1 (.018)	10.9 (.226)	10.9	6
		Top-T	8.7 (.795)	11.3 (.812)	10.0 (.055)	9.8 (.524)	14.8 (.020)	12.3 (.048)	11.2	8
	Static indirect	CuCE	10.0 (.637)	12.0 (.812)	10.1 (.050)	10.8 (.220)	*12.1* (.420)	11.2 (.226)	*11.0*	*7*
		k-Means	12.6 (.008)	13.6 (.118)	*9.8* (.073)	11.2 (.109)	14.9 (.020)	12.0 (.077)	12.4	10
		KaPru	27.7 (.000)	25.3 (.000)	15.7 (.000)	28.1 (.000)	25.1 (.000)	25.4 (.000)	24.5	29
		MDM	24.4 (.000)	24.0 (.000)	11.6 (.002)	23.7 (.000)	21.7 (.000)	23.7 (.000)	21.5	21
		UWA	*9.3* (.795)	*11.8* (.812)	19.5 (.000)	*10.1* (.453)	14.3 (.049)	*10.9* (.226)	12.7	11
	Dyna-mic	kNORA	*27.1* (.000)	*26.7* (.000)	*28.1* (.000)	*28.1* (.000)	*23.4* (.000)	*25.9* (.000)	*26.6*	*32*
		PMCC	40.1 (.000)	38.6 (.000)	32.9 (.000)	39.5 (.000)	39.9 (.000)	38.8 (.000)	38.3	41
Friedman χ^2_{40}			2775.1 (.000)	2076.3 (.000)	3514.4 (.000)	2671.7 (.000)	1462.3 (.000)	2202.6 (.000)		

2.6 Fuente: [Stefan Lessmanna, H Seowb, Bart Baesenscd, and Lyn C Thomas D. Benchmarkingstate-of-the-art classification algorithms for credit scoring: A ten-year update. In Credit Research Centre, Conference Archive, 2013]

Figura 2.7: Comparación entre las 4 mejores técnicas del estudio de *benchmarking*

	AvgR	Adjusted p-values of pairwise comparisons		
		ANN	LR	RF
ANN	2.44			
LR	3.02	.000		
RF	2.53	.167	.000	
HCES-Bag	**2.01**	.000	.000	.000
Friedman χ_3^2	216.2	.000		

2.7 Fuente: [Stefan Lessmanna, H Seowb, Bart Baesenscd, and Lyn C Thomas D. Benchmarkingstate-of-the-art classification algorithms for credit scoring: A ten-year update. In Credit Research Centre, Conference Archive, 2013]

Capítulo 3

Propuesta de Solución

En este capítulo se aborda como será el diseño de la solución. En primera instancia se habla sobre una metodología adaptada de CRISP-DM, luego las etapas asociadas a la metodologia, las métricas de evaluación que se utilizarán en cada etapa, y finalmente las técnicas de modelamiento que se utilizarán junto con sus parámetros.

3.1. CRISP-DM con enfoque inverso

La **metodología actual** basada en CRISP-DM de la institución bancaria está compuesta por las siguientes etapas:

- Entendimiento del negocio
- Calidad de Datos
- Muestreo
- Transformación de Variables
- Selección de Variables
- Modelamiento y Validación

- Puesta en producción
- Integración a la Gestión

La diferencia entre la metodología de la Institución bancaria y CRISP-DM está en las etapas de Muestreo, Transformación de Variables y Selección de Variables, ya que estas fueron adaptadas en base a buenas prácticas que fueron descubriendo con los años para desarrollar sus modelos y con innovaciones de sus analistas; por lo tanto, se volvieron a la medida para su contexto de modelos de *credit scoring*.

El enfoque de esta investigación es rediseñar, implementar y evaluar los procesos de Transformación de Variables, Selección de Variables y Modelamiento. La investigación aborda las etapas en el orden inverso de su ejecución con el fin de basar los resultados en la evaluación del modelo a medida que se avanza con las implementaciones.

Figura 3.1: Orden de ejecución natural de los procesos según la metodología actual

Es necesario mejorar estas tres etapas para generar modelos con mejor predicción para los negocios de la institución, ya que en ellas se generan y escojen las variables que van a explicar el resultado de un modelo; por ende, si el modelo cambia (en técnica), se vuelve vital que las variables sean las mejores para este modelo y además, vengan en una forma que potencie el descubrimiento de conocimiento.

3.2. Etapas de implementación de la solución

Para lograr los objetivos de esta investigación en las etapas de Transformación de Variables, Selección de Variables y Modelamiento, se implementarán secuencialmente en las siguientes 3 etapas:

- **Etapa 1**: modificar la técnica de Modelamiento y evaluar con distintas métricas.
- **Etapa 2**: modificar el proceso de Selección de Variables + Modelamiento y evaluar con distintas métricas.
- **Etapa 3**: modificar el proceso de Transformación de Variables + Selección de Variables + Modelamiento y evaluar con distintas métricas.

En la figura 3.2 se puede observar que la etapa 1 esta encerrada por el color verde (solo aborda Modelamiento), la etapa 2 por el color azul (solo aborda Modelamiento y Selección de Variables) y la etapa 3 por el color rojo (aborda las 3 etapas juntas).

Figura 3.2: Etapas desarrolladas en la investigación

Estas etapas siguen un enfoque en reversa para abordar el problema, es decir, desde el final hacia el comienzo, con el fin de abordar las tres etapas de forma progresiva para comparar en cada etapa lo siguiente:

- **Etapa 1**: comparar la técnica de Regresión Logística vs nuevas técnicas, y observar las ganancias.

- **Etapa 2**: comparar los resultados de la etapa vs etapa 1 al implementar nuevas técnicas de selección de variables y observar las ganancias.
- **Etapa 3**: comparar los resultados de la etapa con resultados anteriores al modificar la transformación de variables existente actualmente.

3.3. Métricas de evaluación para las tres etapas

Para los resultados de la implementación, se evalúa el desempeño de los modelos utilizando métricas tradicionales como *Accuracy*, *AUC Score*, *KS*, *Precision*, *Recall* y *F1-Score*. Cabe destacar que las tres primeras se evalúan para dos instancias; la primera es para el conjunto de datos que se utilizan para entrenar el modelo (datos de entrenamiento) y la segunda para el conjunto restante (datos de prueba). Todos los indicadores que se utilizan tienen una escala porcentual, por lo que su dominio es [0,1], donde 0 equivale a 0 % y 1 a 100 % respectivamente.

3.3.1. *Accuracy*

El *accuracy* se utiliza para evaluar dos instancias, los datos de entrenamiento y de prueba. Mediante este indicador se puede saber qué tanto acierta o se equivoca el modelo en ambos *datasets*; además, al tener ambos indicadores midiendo lo mismo, se puede saber si el modelo tiene sobreajuste o no, viendo que tanta diferencia hay entre los dos valores.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} = \frac{Predicciones\ correctas}{Todas\ las\ predicciones} \tag{3.1}$$

donde:

- **VP**: Verdaderos positivos
- **VN**: Verdaderos negativos

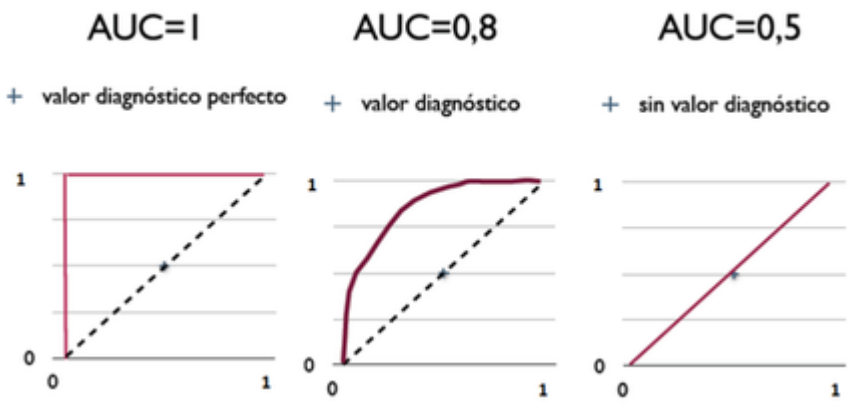
- **FP**: Falsos positivos
- **FN**: Falsos negativos

Esta métrica se incluye debido a que es la métrica estándar para ver si un modelo está sobreajustado o no.

3.3.2. *AUC Score*

Es un indicador que mide la razón de los verdaderos positivos y falsos positivos de un modelo; este valor indica el área bajo la curva y puede estar entre [0.5,1.0], donde 0.5 indica que el modelo no aporta nada y el 1.0 que el modelo discrimina sin errores entre clientes ”Buenos” y ”Malos”. Este indicador puede aplicarse para los datos de entrenamiento y de prueba; además, también mide sobreajuste al comparar los 2 valores.

Figura 3.3: Ejemplos de curva ROC

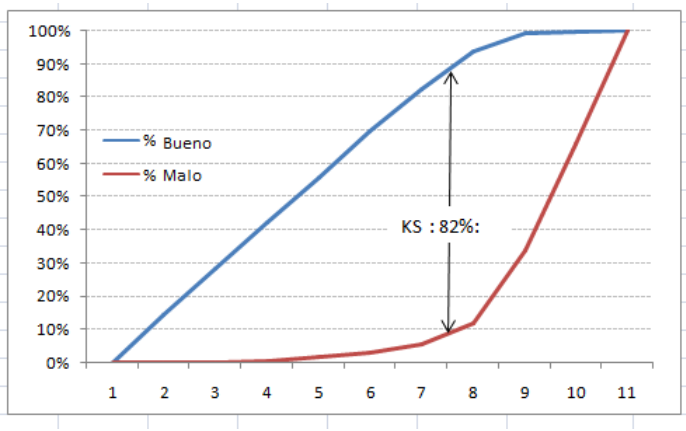


3.3.3. KS

El indicador KS (Kolmogorov-Smirnov) mide si el modelo tiene buena capacidad de discriminación para la variable dependiente (el *target*). Para el caso de *credit scoring* evalúa si los clientes “Buenos” y “Malos” se acumulan en los extremos opuestos del puntaje [10].

KS corresponde a la distancia máxima que existe entre la distribución acumulada de clientes "Buenos" y "Malos". Al igual que el *accuracy* y el *AUC Score*, este indicador puede aplicarse en los 2 *datasets* y medir sobreajuste.

Figura 3.4: Ejemplo de medición de KS



Esta métrica sera la principal a analizar en los trabajos a realizar, debido a que es la más importante para el área de desarrollo de modelos de la institución bancaria para decidir si el modelo está siendo mejor que una versión antigua o similar, por lo que permite comparar fácilmente modelos entrenados con las mismas variables.

3.3.4. Precision, Recall y F1-Score

La ***precision*** mide la proporción de todas las predicciones positivas, con el fin de medir cuántas predicciones positivas son observaciones positivas. El **recall** mide la proporción de los verdaderos positivos y el ***F1-Score*** corresponde a una media armónica entre la *precision* y el *recall*. Las ecuaciones para calcular estos indicadores son las siguientes:

$$Precision = \frac{TP}{TP + FP} = \frac{Predicciones\ correctas\ positivas}{Todas\ las\ predicciones\ positivas} \tag{3.2}$$

$$Recall = \frac{TP}{TP + FN} = \frac{Predicciones\ correctas\ positivas}{Todas\ las\ observaciones\ positivas} \tag{3.3}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.4}$$

Estas tres métricas son importantes de monitorear en los resultados debido a que provienen de la matriz de confusión (al igual que el accuracy) y permiten descartar modelos mal entrenados rápidamente si las tres métricas entregan una mala evaluación en sus indicadores.

3.4. Parámetros de las técnicas de modelamiento

Para el desarrollo de la solución en las etapas, se utilizan las siguientes técnicas de modelamiento para comparar con la *Logistic Regression*.

- *Random Forest* (ensamblaje)
- *Extra Tree* (ensamblaje)
- *Bagging* (ensamblaje)
- *Gradient Boosting* (ensamblaje)
- *AdaBoost* (ensamblaje)
- *Decision Tree*
- *Bernoulli Naive Bayes*
- *Gaussian Naive Bayes*

El motivo de usar estas ocho técnicas se basa en una preselección realizada junto a los analistas, ya que ellos están interesados principalmente por las técnicas de ensamblaje, ya que la literatura de los últimos años ha demostrado beneficios en este tipo de técnicas y esto se complementa con lo encontrado en el estado del arte. Para los arboles de decisión, es una técnica de modelado que siguen utilizando y aconsejaron tenerla en cuenta. Por ultimo,

para *Naive Bayes* recomendaron mantenerla porque respeta varios supuestos de la regresión logística.

Los parámetros utilizados en todos experimentos son:

- *Logistic Regression*:
  - *Penalty*: ”l2”; se usa para especificar la norma utilizada en la penalización.
  - *Dual*: False; formulación dual o primaria.
  - *Tol*: 0.0001; tolerancia para detener los criterios.
  - *C*: 1.0; inverso de la fuerza de regularización.
  - *Solver*: liblinear; algoritmo para usar en el problema de optimización.
  - *Random state*: 1
- *Random Forest*:
  - *N estimators*: 100; la cantidad de árboles en el bosque.
  - *Criterion*: ”gini”; la función para medir la calidad de una división.
  - *Min samples split*: 2; el número mínimo de muestras requeridas para dividir un nodo interno.
  - *Min samples leaf*: 1; el número mínimo de muestras requeridas para estar en un nodo hoja.
  - *Random state*: 1
- *Extra Tree*:
  - *N estimators*: 100; la cantidad de árboles en el bosque.
  - *Criterion*: ”gini”; la función para medir la calidad de una división.
  - *Min samples split*: 2; el número mínimo de muestras requeridas para dividir un nodo interno.
  - *Min samples leaf*: 1; el número mínimo de muestras requeridas para estar en un nodo hoja.

    - *Random state*: 1

- *Bagging*:

    - *Base estimator*: DecisionTreeClassifier; el estimador base para encajar en subconjuntos aleatorios del conjunto de datos.
    - *N estimators*: 100; el número de estimadores base en el conjunto.
    - *Max samples*: 1.0; el número de muestras para extraer de X para entrenar a cada estimador base.
    - *Max features*: 1.0; la cantidad de características que se deben extraer de X para entrenar a cada estimador base.
    - *Random state*: 1

- *Gradient Boosting*:

    - *Loss*: ”deviance”; función de pérdida para ser optimizado.
    - *Learning rate*: 0.1; la tasa de aprendizaje reduce la contribución de cada árbol según el índice de aprendizaje.
    - *N estimators*: 100; el número de etapas de refuerzo para realizar. El aumento de gradiente es bastante robusto para un ajuste excesivo, por lo que un número grande generalmente da como resultado un mejor rendimiento.
    - *Max depth*: 3; profundidad máxima de los estimadores de regresión individuales.
    - *Criterion*: “friendman mse”; la función para medir la calidad de una división.
    - *Random state*: 1

- *AdaBoost*:

    - *Base estimator*: DecisionTreeClassifier; el estimador base a partir del cual se construye el conjunto potenciado.
    - *N estimators*: 100; el número máximo de estimadores en los que se finaliza la potenciación.
    - *Learning rate*: 0.1; la tasa de aprendizaje reduce la contribución de cada árbol según el índice de aprendizaje.

    - *Algorithm*: SAMME.R; algoritmo real de impulso.
    - *Random state*: 1

- *Decision Tree*:
    - *Criterion*: “gini”; la función para medir la calidad de una división.
    - *Splitter*: “best”; la estrategia utilizada para elegir la división en cada nodo.
    - *Min samples split*: 2; el número mínimo de muestras requeridas para dividir un nodo interno.
    - *Min samples leaf*: 1; el número mínimo de muestras requeridas para estar en un nodo hoja.
    - *Random state*: 1

- *Bernoulli Naive Bayes*:
    - *Alpha*: 1.0; parámetro de suavizado de aditivos (Laplace / Lidstone).
    - *Binarize*: 0; umbral para binarización (mapeo a booleanos) de características de muestra.

- *Gaussian Naive Bayes*:
    - *Priors*: None; probabilidades previas de las clases.

Cabe destacar que estos parámetros de entrada fueron probados antes de comenzar los experimentos, luego de ser considerados óptimos para los experimentos de esta investigación. Respecto al *Random state* (presente en las técnicas de ensamblaje, *Logistic Regression* y *Decision Tree*), este parámetro se utiliza como semilla para poder replicar los experimentos, es decir, se controla la aleatoriedad del algoritmo. El *N estimators* (presente en las técnicas de ensamblaje) fue decidido junto a los analistas para que las técnicas de ensamblaje tengan un numero grande de estimadores. En los demás parámetros se toma la decisión en conjunto con los analistas de **confiar en los predeterminados por la API de Scikit-learn** [9]; dado que el objetivo de la investigación no busca centrar los esfuerzos en mejorar los hiper parámetros.

Capítulo 4

Implementación y Validación

En este capitulo se abordan las tres etapas desarrolladas para encontrar beneficios con las implementaciones realizadas para los procesos de Modelamiento, Selección de Variables y Transformación de Variables. El fin de separar por las etapas es encontrar individualmente los beneficios para cada proceso, para finalmente proponer variantes para cada una de ellas con herramientas que generen una ventaja en el desarrollo de modelos de *scoring* para los analistas.

Para comenzar este capitulo se detallan, a grandes rasgos, los *datasets* a usar; luego se presentan las evaluaciones con las métricas para cada etapa y finalmente se resumirá los beneficios encontrados en conjunto.

4.1. Descripción de los *Datasets* entregados por la Institución Bancaria

La institución, para aportar al logro de los objetivos del experimento, proporciona datos anonimizados en dos versiones; la primera consiste en *dataset* con todas las variables obtenidas de un proyecto real luego de la transformación de variables realizada. La segunda versión corresponde a las variables seleccionadas para *Logistic Regression* en el mismo

proyecto. El propósito de esto último es utilizar la versión 2 para la etapa 1, con el fin de emplear una selección de variables previa antes de entrenar el modelo (la selección viene realizada por los analistas con su metodología actual). La versión 1 se utiliza en la etapa 2 con el motivo de aplicar una técnica de selección de variables implementada con todas las variables. El detalle de los *datasets* se observan en el cuadro 4.1.

Cuadro 4.1: Información *datasets* utilizados en sus dos versiones

Dataset		**Número de variables**	
Número	**Volumen**	**Versión 1**	**Versión 2**
1	118.307	233	5
2	54.773	206	9
3	31.205	225	9
4	29.244	189	11
5	23.192	159	9
6	65.539	117	14

4.2. Etapa 1: Modelamiento

La técnica de modelamiento usada actualmente es la **Regresión Logística**, la cual se utiliza punto de referencia para comparar con otras técnicas. Al existir una gran variedad de técnicas de modelamiento, se decide escoger 8 técnicas para comparar con la regresión logística, de las cuales 5 son de ensamblaje, 1 de árbol de decisión y 2 de *Naive Bayes*. Las técnicas utilizadas para esta etapa son las siguientes:

- *Logistic Regression*
- *Random Forest* (ensamblaje)
- *Extra Tree* (ensamblaje)
- *Bagging* (ensamblaje)

- *Gradient Boosting* (ensamblaje)
- *AdaBoost* (ensamblaje)
- *Decision Tree*
- *Bernoulli Naive Bayes*
- *Gaussian Naive Bayes*

El experimento consiste en entrenar un modelo mediante regresión logística para tenerlo como punto de referencia en esta etapa; luego, se entrenan las 8 técnicas restantes con el fin de competir con la regresión. Para comparar se utilizan las métricas definidas luego de entrenar cada modelo; los resultados para cada *dataset* (versión 2) se encuentran en los cuadros 4.2 a 4.7.

Cuadro 4.2: Evaluación de técnicas de modelamiento en el *Dataset* 1 (5 variables)

Techniques / Metrics	**Train Acc**	**Test Acc**	**Train AUC**	**Test AUC**	**Train KS**	**Test KS**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0,8658	0,8648	0,8646	0,8632	**0,597**	**0,594**	0,85	0,86	0,85
Random Forest	0,8667	0,8654	0,8734	0,8705	0,6097	0,6056	0,85	0,87	0,85
Extra Tree	0,8667	0,8655	0,8734	0,8705	0,6097	0,6056	0,85	0,87	0,85
Bagging	0,8667	0,8654	0,8734	0,8705	0,6097	0,6056	0,85	0,87	0,85
Gradient Boosting	0,8661	0,8652	0,8696	0,8679	0,6026	0,5995	0,85	0,87	0,85
AdaBoost	0,8655	0,8646	0,8648	0,8635	0,5978	0,5947	0,85	0,86	0,85
Decision Tree	0,8667	0,8655	0,8734	0,8705	0,6097	0,6056	0,85	0,87	0,85
Bernoulli Naive Bayes	0,8191	0,8173	0,821	0,8176	0,557	0,5513	0,83	0,82	0,82
Gaussian Naive Bayes	0,815	0,8114	0,846	0,8427	0,5872	0,5824	0,84	0,81	0,82

Cuadro 4.3: Evaluación de técnicas de modelamiento en el *Dataset* 2 (9 variables)

Techniques / Metrics	**Train Acc**	**Test Acc**	**Train AUC**	**Test AUC**	**Train KS**	**Test KS**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0,907	0,9063	0,8271	0,8252	**0,5189**	**0,5165**	0,9	0,91	0,88
Random Forest	0,9262	0,9275	0,9231	0,9177	0,6628	0,6521	0,92	0,93	0,92
Extra Tree	0,9263	0,9271	0,9233	0,9179	0,6632	0,6527	0,92	0,93	0,92
Bagging	0,9262	0,9275	0,9231	0,9174	0,6627	0,6519	0,92	0,93	0,92
Gradient Boosting	0,9114	0,9115	0,8521	0,8466	0,5493	0,5412	0,91	0,91	0,89
AdaBoost	0,9075	0,908	0,8324	0,8301	0,5174	0,5134	0,9	0,91	0,89
Decision Tree	0,9263	0,927	0,9233	0,9173	0,6632	0,6526	0,92	0,93	0,92
Bernoulli Naive Bayes	0,884	0,8869	0,8092	0,8032	0,4869	0,4759	0,88	0,89	0,88
Gaussian Naive Bayes	0,8594	0,8597	0,8061	0,8031	0,4963	0,4813	0,88	0,86	0,87

Cuadro 4.4: Evaluación de técnicas de modelamiento en el *Dataset* 3 (9 variables)

Techniques / Metrics	**Train Acc**	**Test Acc**	**Train AUC**	**Test AUC**	**Train KS**	**Test KS**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0,8991	0,8982	0,9212	0,9147	**0,7165**	**0,6979**	0,89	0,9	0,89
Random Forest	0,9504	0,9425	0,9813	0,9721	0,8552	0,8192	0,94	0,94	0,94
Extra Tree	0,9504	0,9433	0,9818	0,9698	0,8586	0,8193	0,94	0,94	0,94
Bagging	0,9504	0,9418	0,9812	0,9716	0,8546	0,8187	0,94	0,94	0,94
Gradient Boosting	0,9089	0,9042	0,936	0,9273	0,7295	0,7102	0,89	0,9	0,9
AdaBoost	0,8989	0,8959	0,9213	0,9144	0,7082	0,6877	0,89	0,9	0,89
Decision Tree	0,9504	0,9422	0,9818	0,9675	0,8586	0,8209	0,94	0,94	0,94
Bernoulli Naive Bayes	0,8912	0,886	0,9075	0,8989	0,6627	0,6494	0,89	0,89	0,89
Gaussian Naive Bayes	0,8798	0,8733	0,9097	0,9019	0,6854	0,6721	0,89	0,87	0,88

Cuadro 4.5: Evaluación de técnicas de modelamiento en el *Dataset* 4 (11 variables)

Techniques / Metrics	**Train Acc**	**Test Acc**	**Train AUC**	**Test AUC**	**Train KS**	**Test KS**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0,6882	0,6949	0,7289	0,7356	**0,3396**	**0,3568**	0,68	0,69	0,68
Random Forest	0,8531	0,8346	0,9346	0,9147	0,6794	0,6502	0,83	0,83	0,83
Extra Tree	0,8532	0,8335	0,9355	0,9138	0,6798	0,6536	0,83	0,83	0,83
Bagging	0,8531	0,8343	0,9345	0,9137	0,6793	0,6471	0,83	0,83	0,83
Gradient Boosting	0,7122	0,712	0,7682	0,767	0,4011	0,4033	0,7	0,71	0,7
AdaBoost	0,6956	0,6995	0,7363	0,7394	0,3453	0,3628	0,69	0,7	0,68
Decision Tree	0,8532	0,8326	0,9355	0,9053	0,6798	0,6448	0,83	0,83	0,83
Bernoulli Naive Bayes	0,6694	0,6757	0,7139	0,7222	0,3111	0,3276	0,68	0,68	0,68
Gaussian Naive Bayes	0,6594	0,6665	0,7145	0,7184	0,3265	0,3389	0,69	0,67	0,67

Cuadro 4.6: Evaluación de técnicas de modelamiento en el *Dataset* 5 (9 variables)

Techniques / Metrics	**Train Acc**	**Test Acc**	**Train AUC**	**Test AUC**	**Train KS**	**Test KS**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0,7238	0,7314	0,7334	0,7446	**0,3458**	**0,3575**	0,72	0,73	0,71
Random Forest	0,7843	0,7876	0,844	0,8429	0,4998	0,5039	0,79	0,79	0,77
Extra Tree	0,7843	0,7872	0,8441	0,8432	0,4998	0,503	0,79	0,79	0,77
Bagging	0,7843	0,7876	0,844	0,8428	0,4998	0,503	0,79	0,79	0,77
Gradient Boosting	0,7462	0,7521	0,7765	0,7834	0,4291	0,4365	0,75	0,75	0,73
AdaBoost	0,7218	0,7288	0,7375	0,7465	0,3558	0,3684	0,72	0,73	0,71
Decision Tree	0,7843	0,7872	0,8441	0,8431	0,4998	0,503	0,79	0,79	0,77
Bernoulli Naive Bayes	0,7276	0,7287	0,7242	0,7362	0,3378	0,3533	0,72	0,73	0,71
Gaussian Naive Bayes	0,691	0,6946	0,7205	0,7301	0,3458	0,3561	0,69	0,69	0,69

Cuadro 4.7: Evaluación de técnicas de modelamiento en el *Dataset* 6 (9 variables)

Techniques / Metrics	**Train Acc**	**Test Acc**	**Train AUC**	**Test AUC**	**Train KS**	**Test KS**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0,9377	0,9385	0,8589	0,8604	**0,5882**	**0,5787**	0,92	0,94	0,92
Random Forest	0,9532	0,9534	0,9467	0,9412	0,7378	0,7334	0,95	0,95	0,94
Extra Tree	0,9532	0,954	0,9469	0,9411	0,7382	0,7331	0,95	0,95	0,94
Bagging	0,9532	0,9535	0,9467	0,9411	0,7377	0,7335	0,95	0,95	0,94
Gradient Boosting	0,942	0,9426	0,8843	0,8818	0,6175	0,6127	0,94	0,94	0,92
AdaBoost	0,9366	0,938	0,8617	0,8614	0,5811	0,5804	0,92	0,94	0,92
Decision Tree	0,9532	0,9539	0,9469	0,9407	0,7382	0,7327	0,95	0,95	0,94
Bernoulli Naive Bayes	0,9117	0,9095	0,8274	0,8273	0,5174	0,5201	0,91	0,91	0,91
Gaussian Naive Bayes	0,8873	0,8853	0,8369	0,8405	0,5454	0,5459	0,91	0,89	0,9

Como se puede observar en los Cuadros 4.2, 4.5 y 4.6, existe una mejora de desempeño de todas las métricas para las técnicas de ensamblaje (*Random Forest*, *Extra Tree*, *Bagging*, *Gradient Boosting* y *AdaBoost*) y en *Decision Tree* al comparar con *Logistic Regression*; esta ganancia es más significativa en *Random Forest*, *Extra Tree* y *Bagging*, además se puede notar que a mayor cantidad de variables (mayor dimensionalidad), más significativa es la diferencia de las métricas de desempeño de estas tres técnicas respecto de *Logistic Regression*.

En los Cuadros 4.3, 4.4 y 4.7 se puede observar resultados similares en mejora de desempeño, excepto para *AdaBoost*, donde esta técnica obtiene un desempeño equivalente a la regresión (no hay mejora significativa). Por ultimo, se destaca que las dos técnicas de *Naive Bayes* obtienen peor desempeño en todas las métricas para todos los experimentos.

Para el análisis cuantitativo de las métricas, se comparan las técnicas en base a la métrica más importante considerada por los analistas de modelos, la cual es el KS; los resultados se pueden ver en las figuras 4.1 a 4.6.

Para facilitar el análisis cuantitativo, se fija como punto de referencia (línea de *benchmark* roja) el **Test KS** de la regresión logística, dado que representa el poder de discriminación entre ”buenos” y ”malos” del modelo a comparar por el resto de las técnicas.

Con los resultados de las figuras 4.1 a 4.6 se puede corroborrar visualmente el análisis realizado con los cuadros 4.2 a 4.7, es decir, las técnicas de *Random Forest*, *Extra Tree*, *Bagging* y *Decision Tree* son las que entregan un mejor desempeño, ya que la diferencia en

ganancia de desempeño es significativa. Además se puede observar, que a mayor complejidad del problema a resolver (más variables), mayor llega a ser la diferencia de desempeño entre la regresión logística y las técnicas con alto desempeño; en particular, las técnicas basadas en árboles logran segmentar mejor los comportamientos que tienen los datos. Por otro lado, se puede observar que ninguna técnica presenta sobreajuste, dado que la diferencia entre *Train KS* y *Test KS*, en particular, es poco significativa; por lo tanto, los modelos aprendieron a clasificar los clientes ”buenos” y ”malos” a nivel general y no sólo para los datos utilizados.

Respecto al área bajo la curva ROC, las diferencias de desempeño se pueden observar en las figuras 4.7 a 4.12. Como puede observarse, en general para los 6 *datasets* hay ganancia de desempeño en *AUC Score* para *Random Forest*, *Extra Tree*, *Bagging* y *Decision Tree*, pero no tanta diferencia como el KS; pero se corrobora que estas técnicas están discriminando mejor a los clientes ”buenos” y ”malos”, debido a que a mayor *AUC Score*, mayor es la tasa entre verdaderos positivos y falsos positivos.

Figura 4.1: Desempeño del KS para *Logistic Regression* y otras técnicas en el *Dataset* 1

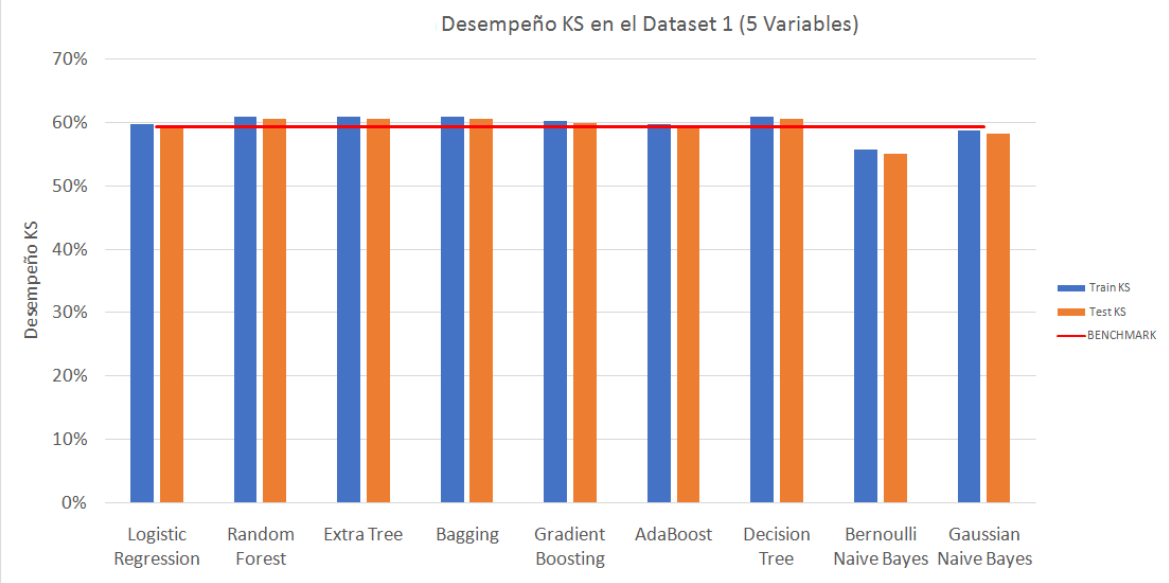


Figura 4.2: Desempeño del KS para *Logistic Regression* y otras técnicas en el *Dataset* 2

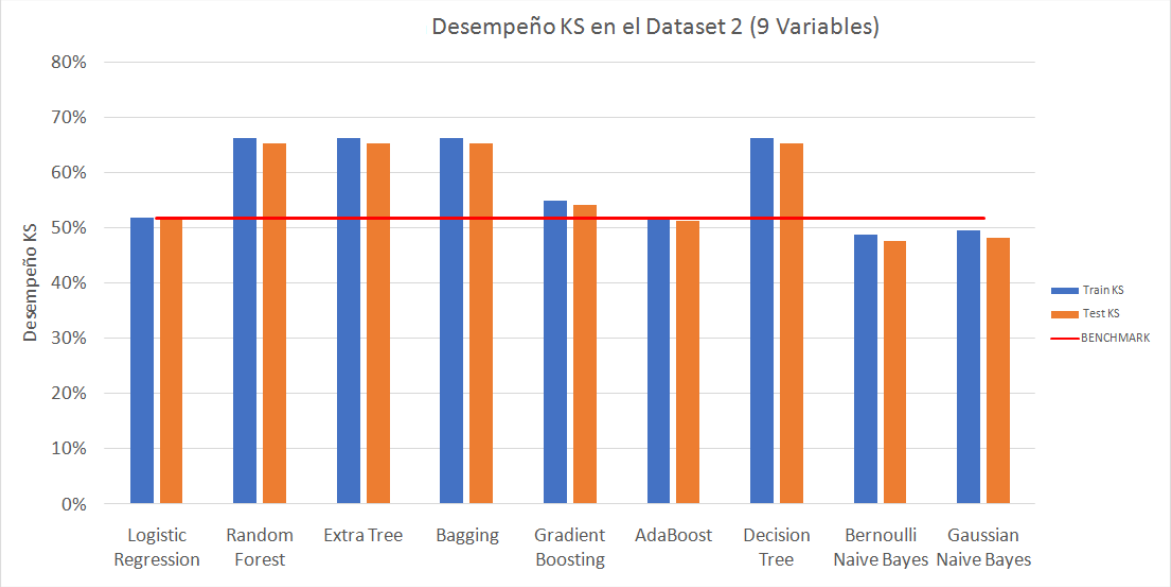


Figura 4.3: Desempeño del KS para *Logistic Regression* y otras técnicas en el *Dataset* 3

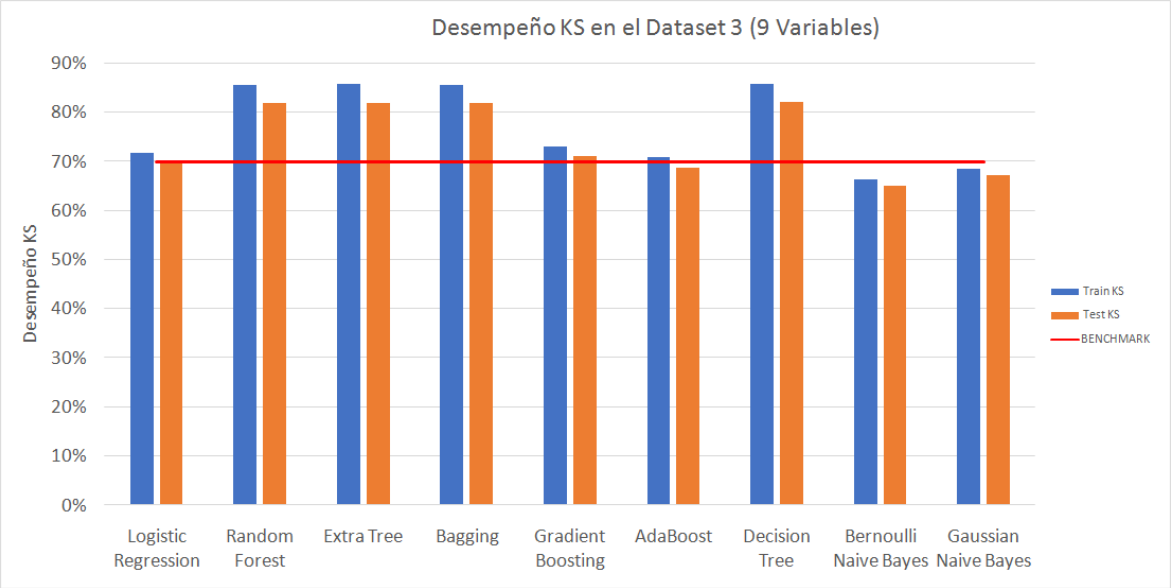


Figura 4.4: Desempeño del KS para *Logistic Regression* y otras técnicas en el *Dataset* 4

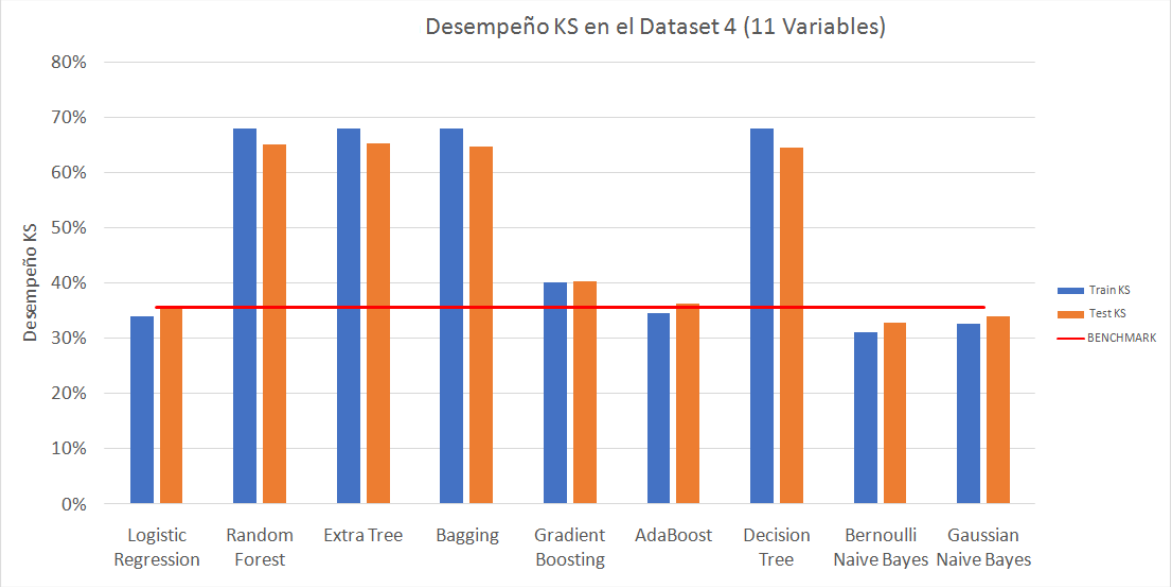


Figura 4.5: Desempeño del KS para *Logistic Regression* y otras técnicas en el *Dataset* 5

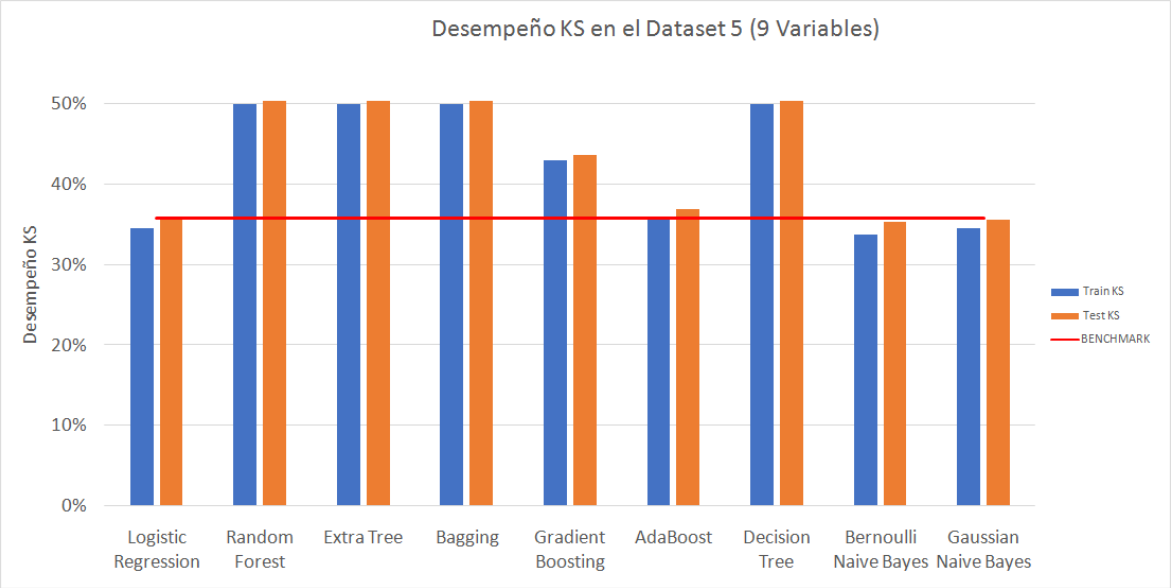


Figura 4.6: Desempeño del KS para *Logistic Regression* y otras técnicas en el *Dataset* 6

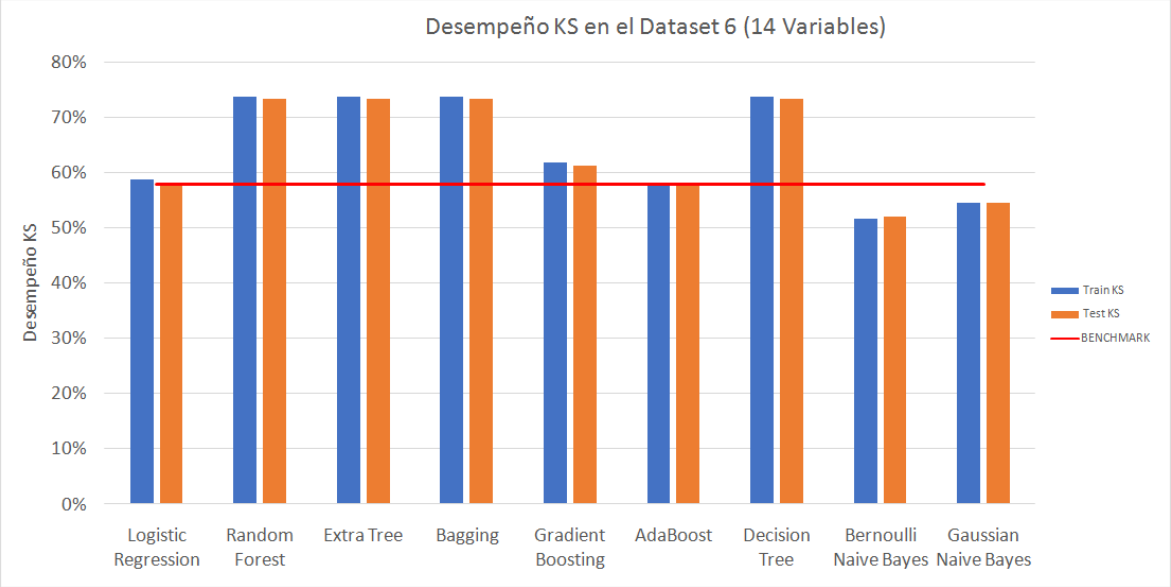


Figura 4.7: Desempeño del AUC *Score* en el Dataset 1 entre las técnicas

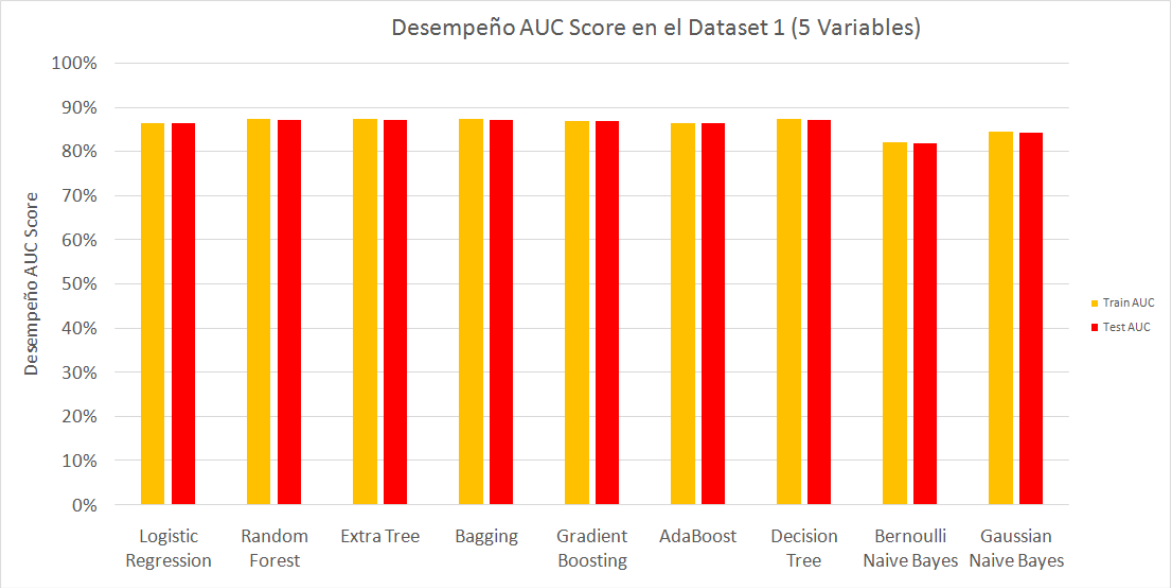


Figura 4.8: Desempeño del AUC *Score* en el Dataset 2 entre las técnicas

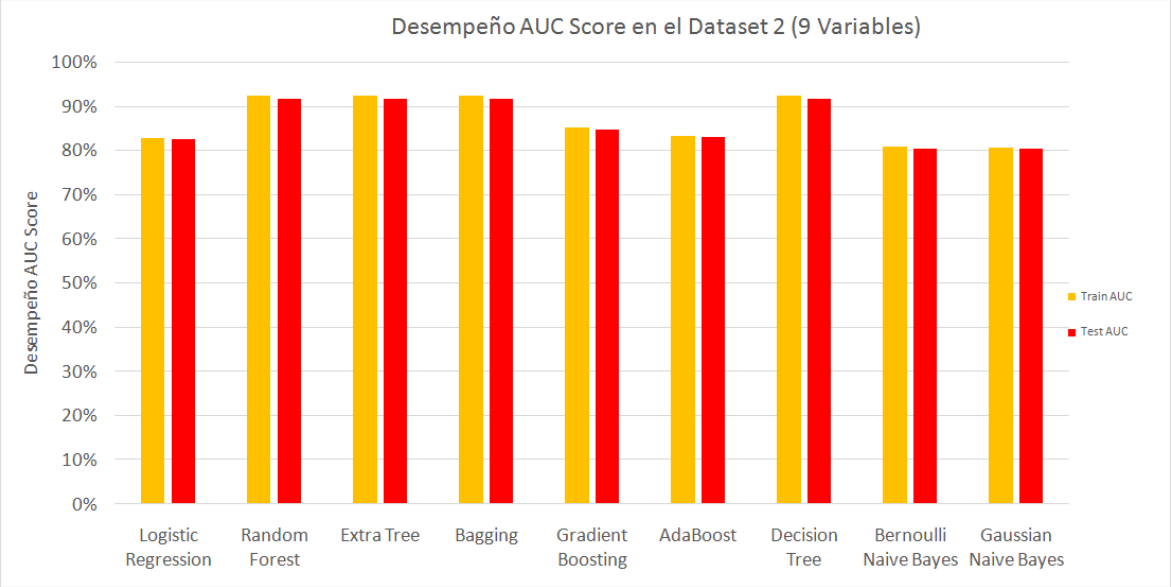


Figura 4.9: Desempeño del AUC *Score* en el Dataset 3 entre las técnicas

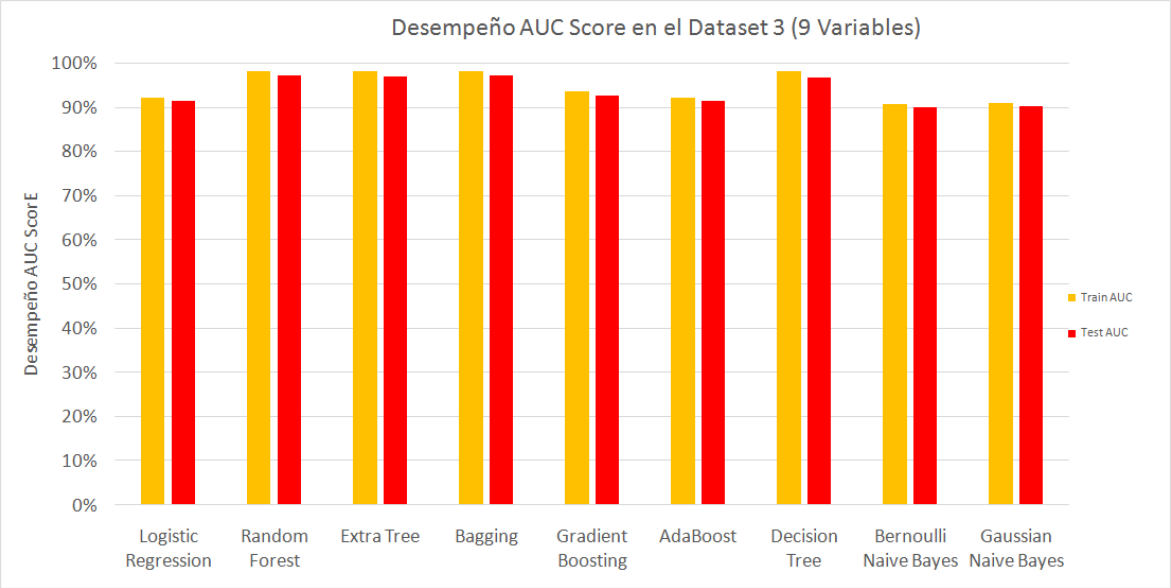


Figura 4.10: Desempeño del AUC *Score* en el Dataset 4 entre las técnicas

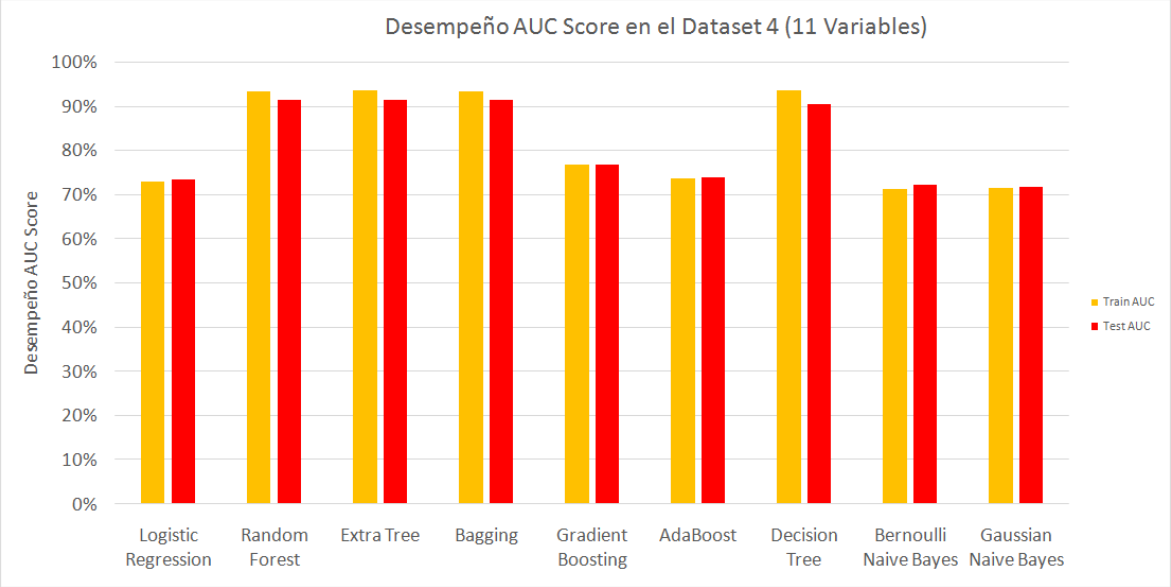


Figura 4.11: Desempeño del AUC *Score* en el Dataset 5 entre las técnicas

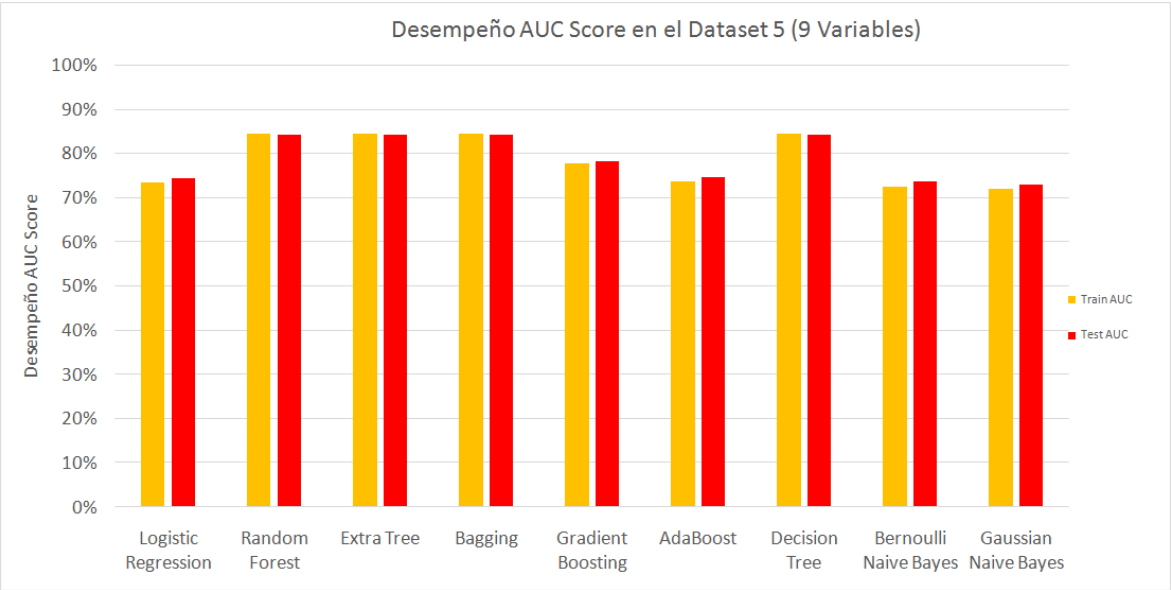
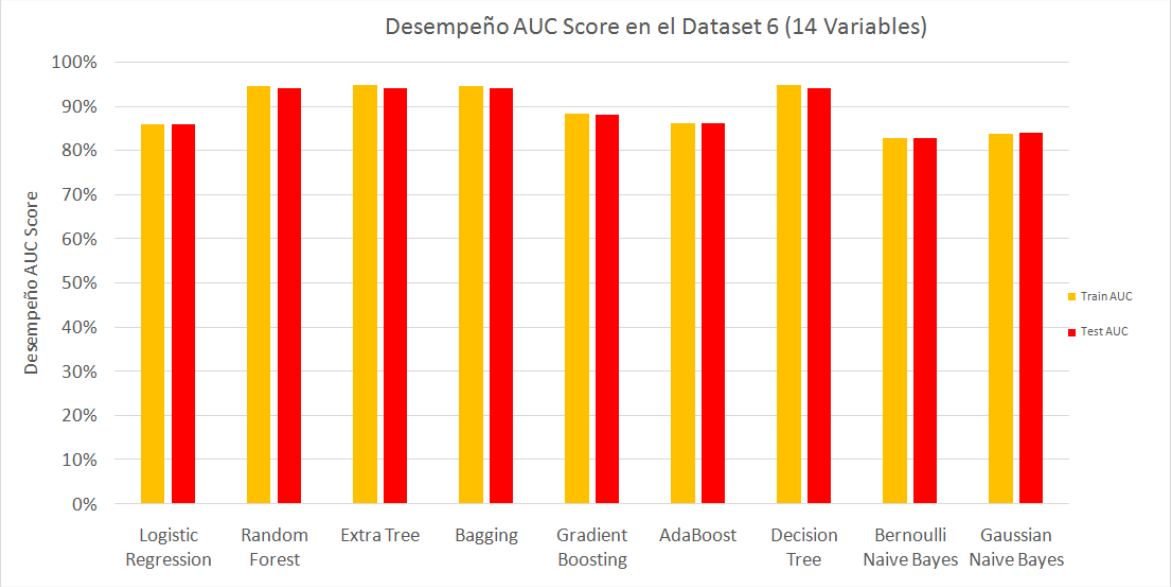


Figura 4.12: Desempeño del AUC *Score* en el Dataset 6 entre las técnicas



Finalmente se comprueba, al igual que en las figuras del KS, que no existe sobreajuste debido a la diferencia poca significativa entre *Train AUC* y *Test AUC* para cada técnica de modelamiento.

En esta etapa se comprueba empíricamente que las técnicas de modelamiento que traen beneficios respecto a *Logistic Regression* son las de ensamblaje, específicamente *Random Forest*, *Extra Tree* y *Bagging*, ya que el KS y el *AUC Score* siempre son mejores en los seis datasets utilizados y además, no presentan sobreajuste. Se tendrán en cuenta estos resultados para la siguiente etapa.

4.3. Etapa 2: Selección de Variables + Modelamiento

Esta etapa tiene el propósito de implementar nuevas técnicas de selección de variables, por lo que se reduce la cantidad de técnicas de modelamiento para utilizar las métricas, con el objetivo de enfocar los esfuerzos de la investigación en la selección y no en el modelamiento. Las técnicas de modelamiento utilizadas en esta etapa son las siguientes:

- *Logistic Regression*
- *Random Forest*
- *Extra Tree*
- *Gradient Boosting*

Respecto a *Bagging*, a pesar de tener buenos resultados en la etapa anterior, no tiene todas las funciones necesarias en la biblioteca de *Python* para obtener los mismos tipos de resultados de las técnicas seleccionadas.

Dado el mismo argumento de reducir las técnicas, también se reducen los *datasets* a utilizar, seleccionando los siguientes entre los seis iniciales (siguiendo la misma referencia numérica) en su primera versión:

- **Dataset 1**: 118.307 registros, donde la primera versión cuenta con 233 variables.
- **Dataset 2**: 54.773 registros, donde la primera versión cuenta con 206 variables.
- **Dataset 6**: 65.539 registros, donde la primera versión cuenta con 117 variables.

Las técnicas de selección de variables implementadas para la investigación son las siguientes:

- *Principal Component Analysis* **(PCA)**: se basa en la agrupación de variables que están correlacionadas en un mismo componente; luego, para cada componente que se genera, se selecciona la variable más predictiva con el fin que el *output* sean variables que no estén correlacionadas entre si y sean las mejores de cada componente.
- *Ranking* de Importancia "Técnica" + Convergencia KS **(CP KS 1)**: esta técnica de selección se basa en dos etapas, en la primera se calcula la importancia de todas las variables con una técnica de ensamblaje, y se genera un *ranking* de las variables dada su importancia. La segunda etapa se toma este *ranking* y se va ingresando una sola variable por iteración y en ese mismo paso se calcula el KS del modelo. Desde la

segunda iteración en adelante se compara el KS del modelo actual con n variables y el KS del modelo anterior de $n - 1$ variables, en el caso que el KS no haya variado significativamente, el algoritmo se detiene y se retorna las n variables de la iteración.

- *Ranking* KS + Convergencia KS **(CP KS 2)**: es similar a la técnica CP KS 1, pero el *ranking* de importancia se reemplaza por un *ranking* del KS de cada variable en orden descendente.

- *Ranking* IV + Convergencia KS **(CP KS 3)**: es similar a las técnicas CP KS 1 y 2, pero el *ranking* de importancia/KS se reemplaza por un *ranking* del IV de cada variable en orden descendente.

- Correlación con *ranking* de Importancia "Técnica"**(Correlación 1)**: esta técnica calcula la matriz de correlación entre todas las variables con los coeficientes de correlación de Pearson, Spearman o Tau-b de Kendall; luego, se genera un *ranking* de importancia para cada variable. La metodología de selección de variables consiste en tomar la mejor candidata en base a su importancia y filtrar del *set* de datos las que posean una correlación mayor o igual a 0,5. Luego de forma iterativa se realiza el mismo procedimiento con la siguiente mejor candidata del conjunto de variables actualizado, luego de la aplicación del filtro.

- Correlación con *Ranking* IV **(Correlación 2)**: es similar a la técnica de correlación anterior, pero el *ranking* con los IV de cada variable.

- Correlación con *Ranking* KS **(Correlación 3)**: Es similar a la técnica de correlación anterior, pero el *ranking* con los KS de cada variable.

- Correlación con Resultados anteriores de los 3 métodos de correlación **(Correlación 4)**: esta técnica es una combinación de los tres métodos de correlación anteriores, es decir, el *input* de variables es el *output* de las tres técnicas anteriores. En base a estas variables se genera un *ranking* por importancia utilizando una técnica de ensamblaje, luego se calcula la matriz de correlación entre las variables y se repite el procedimiento de filtrado que las tres técnicas anteriores.

Cabe destacar que las técnicas de **(PCA)** y Correlación con *ranking* IV **(Correlación 2)** ya están presentes en la metodología actual, pero son implementadas dentro de un software estadístico, por lo que para fines de la investigación no se considera un punto de referencia, si no como una técnica de selección más implementada junto al resto.

Antes de comenzar el análisis se destaca que algunos de estos métodos de selección implican utilizar una técnica de modelamiento dentro del algoritmo; por ejemplo, para obtener la importancia por una "Técnica" (en las técnicas CP KS 1, Correlación 1 y Correlación 4), se requiere el uso de una técnica de modelamiento para su ejecución. Por lo tanto, para evaluar el desempeño del método con un modelo, se utiliza la misma técnica usada en el algoritmo con el fin de no perjudicarlo. Con esto, estas tres técnicas de selección aplican un modelamiento dentro de su algoritmo.

Los resultados de las 8 técnicas de selección de variables en los 3 *datasets* se presentan en los cuadros 4.8 y 4.9.

Cuadro 4.8: Resultados de las técnicas de selección de variables en el *Dataset* 1 (parte 1)

	Random Forest			***Extra Tree***		
	N Variables	**Train KS**	**Test KS**	**N Variables**	**Train KS**	**Test KS**
Etapa 0	5	0,597	0,594	5	0,597	0,594
Etapa 1	5	**0,6097**	**0,6056**	5	**0,6097**	**0,6056**
PCA	7	0,7134	0,697	7	0,7134	0,6965
CP KS 1	6	0,6149	0,611	6	0,6191	0,6148
CP KS 2	6	0,6147	0,6088	6	0,6147	0,6088
CP KS 3	6	0,6119	0,6064	6	0,6119	0,6064
Correlación 1	10	0,7545	0,7371	10	0,7101	0,6895
Correlación 2	10	0,7403	0,7234	10	0,7405	0,7229
Correlación 3	10	0,7045	0,6877	10	0,7046	0,6872
Correlación 4	10	0,7644	0,7474	10	0,7607	0,7386

Cuadro 4.9: Resultados de las técnicas de selección de variables en el *Dataset* 1 (parte 2)

	Gradient Boosting			***Logistic Regression***		
	N Variables	**Train KS**	**Test KS**	**N Variables**	**Train KS**	**Test KS**
Etapa 0	5	0,597	0,594	5	0,597	0,594
Etapa 1	5	**0,6026**	**0,5995**	-	-	-
PCA	7	0,6265	0,6246	7	0,6178	0,6136
CP KS 1	6	0,6258	0,6248	6	0,6119	0,6069
CP KS 2	6	0,6147	0,6088	6	0,6129	0,6072
CP KS 3	6	0,6119	0,6064	6	0,6106	0,6054
Correlación 1	10	0,6066	0,5995	10	0,5972	0,5929
Correlación 2	10	0,6092	0,6047	10	0,5983	0,5943
Correlación 3	10	0,6064	0,6057	10	0,5972	0,5932
Correlación 4	10	0,612	0,6107	10	0,5983	0,5941

Como se puede ver, para el *Dataset* 1 la mayoría de las técnicas entrega una mejora en el desempeño del KS, pero con mayor cantidad de variables respecto a la etapa 1. Cabe destacar que las cuatro técnicas de selección por correlación devuelven 10 variables (distintas entre si), las cuales no generan mucha ganancia para las técnicas de *Logistic Regression* y *Gradient Boosting*, pero sí para *Random Forest* y *Extra Tree*, demostrando que al complejizar el modelo con más variables, ambas técnicas logran discriminar mucho mejor a un cliente ”bueno” o ”malo”.

Otro detalle es que las cinco variables extras seleccionadas por las técnicas de correlación no generan una ganancia significativa respecto a las otras cinco variables del entregadas en la metodología actual con *Logistic Regression* y *Gradient Boosting*.

Para el *Dataset* 2 (cuadro 4.10) se puede observar que las técnicas de correlación seleccionan entre 40 y 42 variables, lo cual genera un problema en no lograr reducir la complejidad del modelo significativamente; dado esto, *Random Forest* y *Extra Tree* obtienen buenos resultados por esta complejidad, pero el modelo no es interpretable por analistas y el negocio asociado dada la cantidad excesiva de variables.

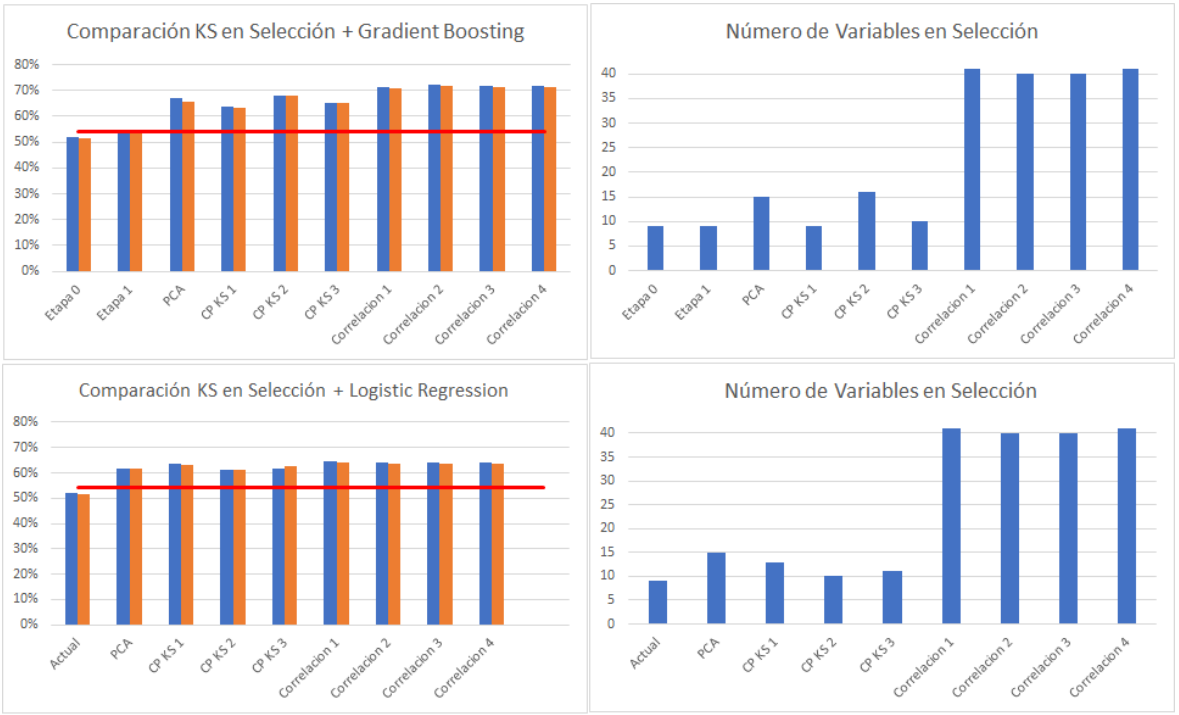
Cuadro 4.10: Resultados de las técnicas de selección de variables en el *Dataset* 2 (parte 1)

	Random Forest			***Extra Tree***		
	N Variables	**Train KS**	**Test KS**	**N Variables**	**Train KS**	**Test KS**
Etapa 0	9	0,5189	0,5165	9	0,5189	0,5165
Etapa 1	9	**0,6628**	**0,6521**	9	**0,6632**	**0,6527**
PCA	15	0,963	0,9545	15	0,9638	0,9573
CP KS 1	20	0,9458	0,9381	24	0,961	0,9578
CP KS 2	19	0,9101	0,9058	19	0,9108	0,9075
CP KS 3	20	0,9212	0,9194	20	0,9222	0,9216
Correlación 1	42	0,9986	0,9915	42	0,9998	0,9964
Correlación 2	40	0,9984	0,99	40	0,9989	0,995
Correlación 3	40	0,9988	0,9885	40	0,9991	0,9952
Correlación 4	42	0,9991	0,9912	42	0,9998	0,997

Cuadro 4.11: Resultados de las técnicas de selección de variables en el *Dataset* 2 (parte 2)

	Gradient Boosting			***Logistic Regression***		
	N Variables	**Train KS**	**Test KS**	**N Variables**	**Train KS**	**Test KS**
Etapa 0	9	0,5189	0,5165	9	0,5189	0,5165
Etapa 1	9	**0,5493**	**0,5412**	-	-	-
PCA	15	0,6713	0,6583	15	0,6184	0,617
CP KS 1	9	0,6392	0,633	13	0,6344	0,6336
CP KS 2	16	0,6823	0,6796	10	0,6114	0,6132
CP KS 3	10	0,6499	0,6502	11	0,6185	0,6251
Correlación 1	41	0,7149	0,7091	41	0,6437	0,6393
Correlación 2	40	0,7216	0,7162	40	0,6416	0,6376
Correlación 3	40	0,7194	0,7139	40	0,6421	0,6359
Correlación 4	41	0,7198	0,7125	41	0,6419	0,6375

Figura 4.13: Comparación del KS en el *Dataset* 2 y número de variables post-selección para *Gradient Boosting* y *Logistic Regression*



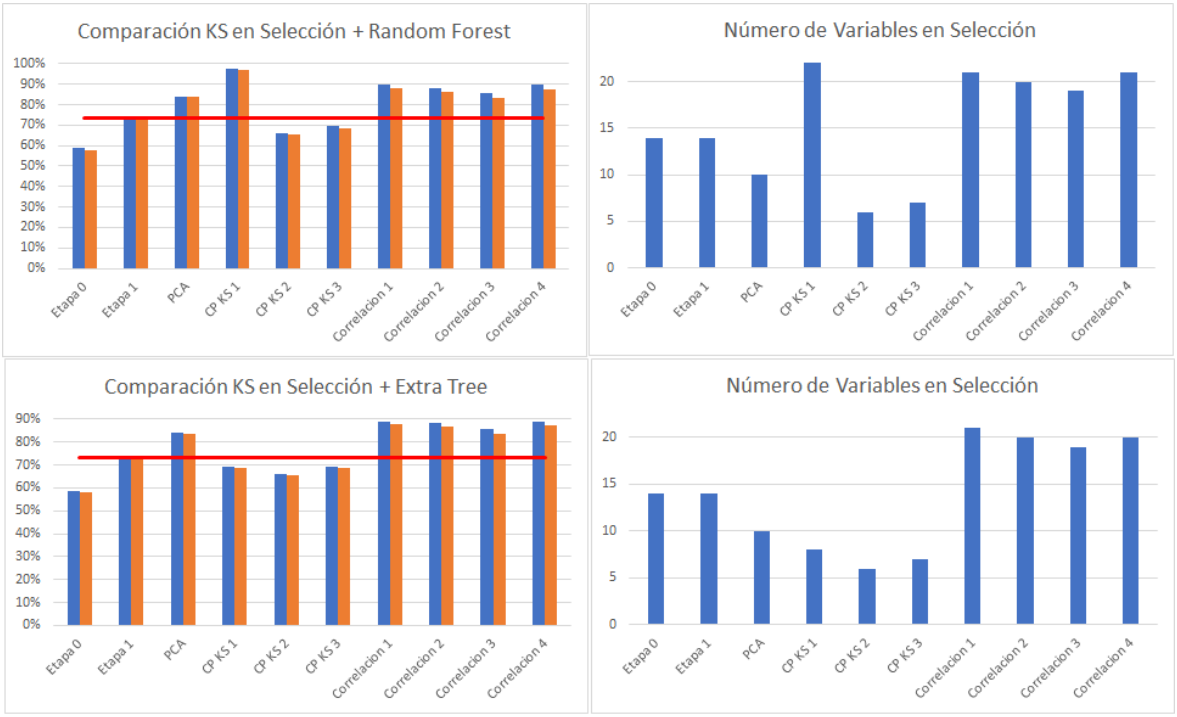
Cuadro 4.12: Resultados de las técnicas de selección de variables en el *Dataset* 6 (parte 1)

	Random Forest			***Extra Tree***		
	N Variables	**Train KS**	**Test KS**	**N Variables**	**Train KS**	**Test KS**
Etapa 0	14	0,5882	0,5787	14	0,5882	0,5787
Etapa 1	14	**0,7378**	**0,7334**	14	**0,7382**	**0,7331**
PCA	10	0,8393	0,8368	10	0,8395	0,8355
CP KS 1	22	0,9754	0,9708	8	0,6922	0,6893
CP KS 2	6	0,6585	0,6554	6	0,6586	0,6555
CP KS 3	7	0,6942	0,6872	7	0,6942	0,6873
Correlación 1	21	0,8956	0,8826	21	0,8908	0,8764
Correlación 2	20	0,882	0,8633	20	0,8825	0,8657
Correlación 3	19	0,8544	0,8359	19	0,8548	0,8351
Correlación 4	21	0,8958	0,8774	20	0,8869	0,871

Cuadro 4.13: Resultados de las técnicas de selección de variables en el *Dataset* 6 (parte 2)

	Gradient Boosting			***Logistic Regression***		
	N Variables	**Train KS**	**Test KS**	**N Variables**	**Train KS**	**Test KS**
Etapa 0	14	0,5882	0,5787	14	0,5882	0,5787
Etapa 1	14	**0,6175**	**0,6127**	-	-	-
PCA	10	0,701	0,689	10	0,6722	0,6651
CP KS 1	12	0,7218	0,7296	12	0,6848	0,6913
CP KS 2	6	0,6553	0,6571	6	0,6426	0,6403
CP KS 3	6	0,6743	0,6775	6	0,6431	0,6426
Correlación 1	21	0,5628	0,5505	21	0,4992	0,4846
Correlación 2	20	0,5613	0,5318	20	0,4966	0,483
Correlación 3	19	0,5596	0,5355	19	0,5005	0,4905
Correlación 4	19	0,5596	0,5436	19	0,4864	0,4707

Figura 4.14: Comparación del KS en el *Dataset* 6 y número de variables post-selección para *Random Forest* y *Extra Tree*



Comparando las ocho técnicas se puede observar que PCA selecciona 15 variables (figura 4.13), las cuales tienen un buen desempeño en KS, y supera con significancia a los modelos entrenados con 9 variables en la etapa 0 y 1.

Para el *Dataset* 6, se puede observar que las ocho técnicas entregan buenos resultados para *Random Forest* y *Extra Tree*, pero para las otras dos técnicas los métodos de selección de correlación no aportan variables que logren superar al modelo entrenado con la metodología actual (etapa 0).

Se puede destacar que hay técnicas de selección que obtuvieron menos variables que la Etapa 1 y el desempeño en KS logra subir significativamente, por lo que se encontraron entre 6 y 12 variables que impactan mucho más en el modelo.

Los resultados de esta etapa son bien diversos. Esto puede justificarse debido a que las variables que quedan después de una técnica de selección influyen mucho en los resultados finales, por lo que en base a los experimentos no se puede declarar que una de las ocho técnicas de selección es la mejor, aunque las cuatro técnicas de correlación tienden a ser superadas por las otras cuatro al tener en cuenta el *trade off* entre **ganar desempeño vs seleccionar pocas variables**.

Cabe destacar que las tres técnicas de CP KS son las que tienden a destacarse entre las ocho técnicas de selección; además, para el área de desarrollo de modelos de la Institución Bancaria es una innovación importante, ya que estas técnicas de selección contienen en cierto grado, el beneficio de las técnicas de ensamblaje (*Random Forest*, *Extra Tree* y *Gradient Boosting*), como por ejemplo, calcular una importancia para cada variable luego de utilizarlas dentro de un modelo.

4.4. Etapa 3: Transformación de las Variables + Selección de Variables + Modelamiento

Para comparar transformaciones de variables, se utilizan las variables de la siguiente manera, dependiendo si son variables continuas o categóricas:

- **Transformación WoE**: variables continuas categóricas con transformación WoE, tal como se ha utilizado durante todas las etapas anteriores.
- **Sin Transformación**: variables continuas en su forma natural y variables categóricas con transformación WoE.
- **Transformación LogN**: variables continuas con transformación, aplicando logaritmo natural y variables categóricas con transformación WoE.
- **Transformación Log**: variables continuas con transformación, aplicando logaritmo en base 10 y variables categóricas con transformación WoE.
- **Transformación Sqrt**: variables continuas con transformación, aplicando raíz cuadrada y variables categóricas con transformación WoE.

Cabe destacar que al utilizar las tres últimas transformaciones mencionadas, se tomó la decisión de dejar los datos que se indefinen al aplicar la transformación en su forma natural para lograr realizar los experimentos.

Dado que en la etapa 1 se utilizaron seis *datasets*, nueve técnicas de modelamiento y transformación WoE, se compara las otras tres transformaciones restantes y las variables sin transformación con el fin de observar el impacto en el desempeño de los modelos. Los resultados se encuentran en los cuadros 4.14 a 4.19.

Cuadro 4.14: Resultados de la transformación de variables en el *Dataset* 1 respecto de la Etapa 1

Techniques/Transform	Etapa 1 (WoE)		Sin Transformación		Transformación LogN		Transformación Log		Transformación Sqrt	
	Train KS	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**
Logistic Regression	**0,597**	**0,594**	0,5944	0,5926	0,601	0,6002	0,6034	0,6007	0,6008	0,6003
Random Forest	0,6097	0,6056	0,8286	0,8119	0,827	0,8105	0,8286	0,8116	0,8286	0,8115
Extra Tree	0,6097	0,6056	0,8292	0,8154	0,8276	0,8149	0,8292	0,8163	0,8292	0,8164
Bagging	0,6097	0,6056	0,8286	0,8124	0,827	0,8123	0,8286	0,8126	0,8286	0,8118
Gradient Boosting	0,6026	0,5995	0,62	0,6147	0,6234	0,6153	0,62	0,6147	0,62	0,6147
AdaBoost	0,5978	0,5947	0,6086	0,6045	0,611	0,6062	0,6086	0,6045	0,6086	0,6045
Decision Tree	0,6097	0,6056	0,8292	0,8097	0,8276	0,8099	0,8292	0,8104	0,8292	0,8095
Bernoulli Naive Bayes	0,557	0,5513	0,59	0,5854	0,5893	0,5846	0,59	0,5854	0,59	0,5854
Gaussian Naive Bayes	0,5872	0,5824	0,5741	0,5749	0,5911	0,5879	0,5977	0,5995	0,5832	0,5848

Cuadro 4.15: Resultados de la transformación de variables en el *Dataset* 2 respecto de la Etapa 1

Techniques/Transform	Etapa 1 (WoE)		Sin Transformación		Transformación LogN		Transformación Log		Transformación Sqrt	
	Train KS	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**
Logistic Regression	**0,5189**	**0,5165**	**0,2381**	**0,2292**	0,4927	0,4872	0,493	0,4908	0,5154	0,5028
Random Forest	0,6627	0,6517	1	0,9945	1	0,9918	1	0,9944	1	0,9944
Extra Tree	0,6632	0,6527	1	0,9968	1	0,997	1	0,9968	1	0,9968
Bagging	0,6627	0,6519	1	0,9913	1	0,9914	1	0,9914	1	0,9914
Gradient Boosting	0,5493	0,5412	0,7039	0,6907	0,7052	0,6873	0,7039	0,6907	0,7039	0,6907
AdaBoost	0,5174	0,5134	0,671	0,6671	0,6675	0,6594	0,671	0,6671	0,671	0,6671
Decision Tree	0,6632	0,6526	1	0,9826	1	0,9795	1	0,9834	1	0,9845
Bernoulli Naive Bayes	0,4869	0,4759	0,2578	0,2423	0,2616	0,2539	0,2578	0,2423	0,2578	0,2423
Gaussian Naive Bayes	0,4963	0,4813	0,4491	0,4413	0,4814	0,4644	0,4887	0,4788	0,4926	0,4955

Cuadro 4.16: Resultados de la transformación de variables en el *Dataset* 3 respecto de la Etapa 1

Techniques/Transform	Etapa 1 (WoE)		Sin Transformación		Transformación LogN		Transformación Log		Transformación Sqrt	
	Train KS	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**
Logistic Regression	**0,7165**	**0,6979**	**0,205**	**0,1748**	0,7012	0,6824	0,6867	0,6702	0,6754	0,6568
Random Forest	0,8553	0,8189	0,9956	0,9656	0,9956	0,9674	0,9955	0,9681	0,9955	0,9678
Extra Tree	0,8586	0,8195	0,9959	0,9703	0,9959	0,9672	0,9959	0,9682	0,9959	0,969
Bagging	0,8546	0,8187	0,9955	0,9647	0,9956	0,9682	0,9955	0,9692	0,9955	0,9675
Gradient Boosting	0,7295	0,7102	0,7531	0,7323	0,7567	0,7343	0,7531	0,7323	0,7531	0,7323
AdaBoost	0,7082	0,6877	0,7434	0,7243	0,733	0,7155	0,7434	0,7243	0,7434	0,7243
Decision Tree	0,8586	0,8217	0,9959	0,9539	0,9959	0,9591	0,9959	0,9597	0,9959	0,9588
Bernoulli Naive Bayes	0,6627	0,6494	0,3158	0,319	0,6768	0,6682	0,3158	0,319	0,3158	0,319
Gaussian Naive Bayes	0,6854	0,6721	0,1956	0,1659	0,6373	0,6155	0,6256	0,6001	0,6051	0,5695

Cuadro 4.17: Resultados de la transformación de variables en el *Dataset* 4 respecto de la Etapa 1

Techniques/Transform	Etapa 1 (WoE)		Sin Transformación		Transformación LogN		Transformación Log		Transformación Sqrt	
	Train KS	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**
Logistic Regression	**0,3396**	**0,3568**	**0,2567**	**0,2696**	0,2955	0,3074	0,2782	0,3023	0,3013	0,3134
Random Forest	0,6794	0,6476	0,9572	0,9247	0,9559	0,9208	0,9573	0,9251	0,9573	0,9247
Extra Tree	0,6798	0,6543	0,9575	0,9251	0,956	0,9209	0,9575	0,9247	0,9575	0,9259
Bagging	0,6793	0,6471	0,957	0,9216	0,9559	0,9202	0,957	0,9203	0,957	0,9211
Gradient Boosting	0,4011	0,4033	0,4722	0,4577	0,4694	0,4552	0,4722	0,4577	0,4722	0,4577
AdaBoost	0,3453	0,3628	0,3832	0,3804	0,3861	0,3822	0,3832	0,3804	0,3832	0,3804
Decision Tree	0,6798	0,6449	0,9575	0,8906	0,956	0,8899	0,9575	0,8931	0,9575	0,8921
Bernoulli Naive Bayes	0,3111	0,3276	0,3014	0,3146	0,2778	0,2941	0,3014	0,3146	0,3014	0,3146
Gaussian Naive Bayes	0,3265	0,3389	0,2329	0,2206	0,2855	0,293	0,2859	0,2998	0,2858	0,3047

Cuadro 4.18: Resultados de la transformación de variables en el *Dataset* 5 respecto de la Etapa 1

Techniques/Transform	Etapa 1 (WoE)		Sin Transformación		Transformación LogN		Transformación Log		Transformación Sqrt	
	Train KS	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**
Logistic Regression	**0,3458**	**0,3575**	**0,2805**	**0,2882**	0,2994	0,3228	0,2984	0,3246	0,3038	0,3137
Random Forest	0,4998	0,5039	0,9402	0,921	0,9372	0,9175	0,9402	0,9207	0,9402	0,921
Extra Tree	0,4998	0,503	0,9402	0,9212	0,9372	0,9183	0,9402	0,9195	0,9402	0,9214
Bagging	0,4998	0,503	0,9402	0,9195	0,9372	0,9169	0,9402	0,9197	0,9402	0,9195
Gradient Boosting	0,4291	0,4365	0,5203	0,5156	0,5086	0,5087	0,5203	0,5156	0,5203	0,5156
AdaBoost	0,3558	0,3684	0,4204	0,4246	0,4226	0,4289	0,4204	0,4246	0,4204	0,4246
Decision Tree	0,4998	0,503	0,9402	0,9194	0,9372	0,9169	0,9402	0,9181	0,9402	0,919
Bernoulli Naive Bayes	0,3378	0,3533	0,3139	0,3301	0,3296	0,3427	0,3139	0,3298	0,3139	0,3301
Gaussian Naive Bayes	0,3458	0,3561	0,1807	0,2138	0,3155	0,3245	0,3062	0,3176	0,298	0,303

Cuadro 4.19: Resultados de la transformación de variables en el *Dataset* 6 respecto de la Etapa 1

Techniques/Transform	Etapa 1 (WoE)		Sin Transformación		Transformación LogN		Transformación Log		Transformación Sqrt	
	Train KS	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**	**Train KS**	**Test KS**
Logistic Regression	**0,5071**	**0,4973**	**0,0911**	**0,08**	0,3889	0,3936	0,3698	0,3815	0,3593	0,3762
Random Forest	0,6344	0,6172	1	0,9932	1	0,993	1	0,9933	1	0,9931
Extra Tree	0,6347	0,6215	1	0,9972	1	0,9969	1	0,9972	1	0,9962
Bagging	0,6342	0,6172	1	0,9932	1	0,993	1	0,9933	1	0,9932
Gradient Boosting	0,5412	0,5279	0,7042	0,6728	0,7064	0,6833	0,7042	0,6728	0,7042	0,6728
AdaBoost	0,5014	0,4913	0,6496	0,6239	0,6354	0,6194	0,6496	0,6239	0,6496	0,6239
Decision Tree	0,6347	0,6217	1	0,9889	1	0,9938	1	0,9888	1	0,9888
Bernoulli Naive Bayes	0,4926	0,4882	0,4258	0,4068	0,4304	0,4125	0,4258	0,4068	0,4258	0,4068
Gaussian Naive Bayes	0,4908	0,4867	0,1593	0,1368	0,3765	0,3672	0,2266	0,2151	0,2531	0,2381

Como puede observarse en los cuadros 4.14 al 4.19, al no realizar transformación en las variables continuas, todas las técnicas asociadas a árboles (*Random Forest*, *Extra Tree*, *Bagging*, *Gradient Boosting*, *AdaBoost* y *Decision Tree*) se potencian y aumentan su KS en los dos *set* de datos (entrenamiento y prueba); esto también se ve reflejado al aplicar las transformaciones de *LogN*, *Log* y *Sqrt*. Este mejoramiento de los indicadores se debe a que las variables continuas, al no estar categorizadas con la transformación WoE, mantienen su poder predictivo debido a que los valores no fueron agrupados, por lo tanto los árboles de decisión pueden desarrollar una mayor complejidad al momento de predecir, haciendo que su poder de discriminación aumente, aunque siendo más propensos a sobreajuste al

utilizar datos fuera del conjunto de entrenamiento y prueba; es decir, si se llega a utilizar el modelo con todos los datos de la población, es altamente probable que los modelos que utilizaron árboles contengan sobreajuste o no sirva para explicar el problema. Esto último se comprueba al ver que las métricas de KS estan muy cercanas al 100 % para las técnicas de *Random Forest*, *Extra Tree*, *Bagging* y *Decision Tree*.

Por otro lado, se puede ver en los cuadros 4.15 al 4.19 que *Logistic Regression* pierde su poder predictivo considerablemente si no se utiliza la transformación WoE. Esto es de esperarse, debido a que esta transformación busca potenciar los supuestos de la regresión para que se obtenga un buen modelo tales como variables con monotonía (creciente o decreciente), agrupación de *outliers* y no de variables correlacionadas, entre otras. Por último, para las técnicas de *Naive Bayes*, el no utilizar transformación WoE también es perjudicial por motivos similares a *Logistic Regression*.

Figura 4.15: Desempeño del KS en el *Dataset* 1 al no realizar transformación en las variables continuas

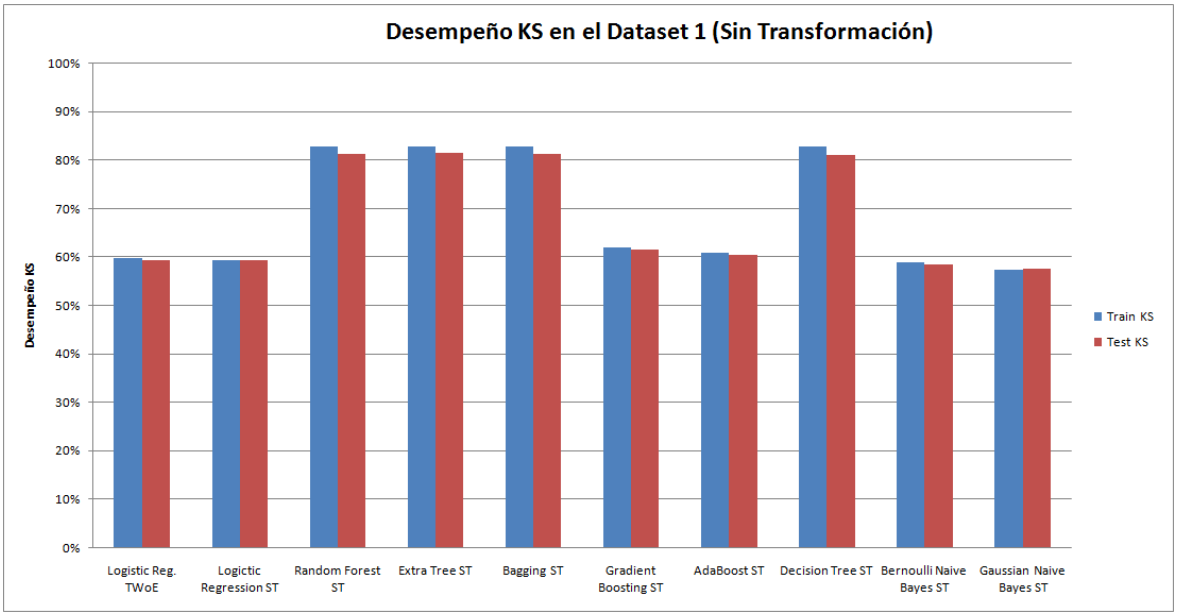


Figura 4.16: Desempeño del KS en el *Dataset* 1 al realizar transformación *LogN* en las variables continuas

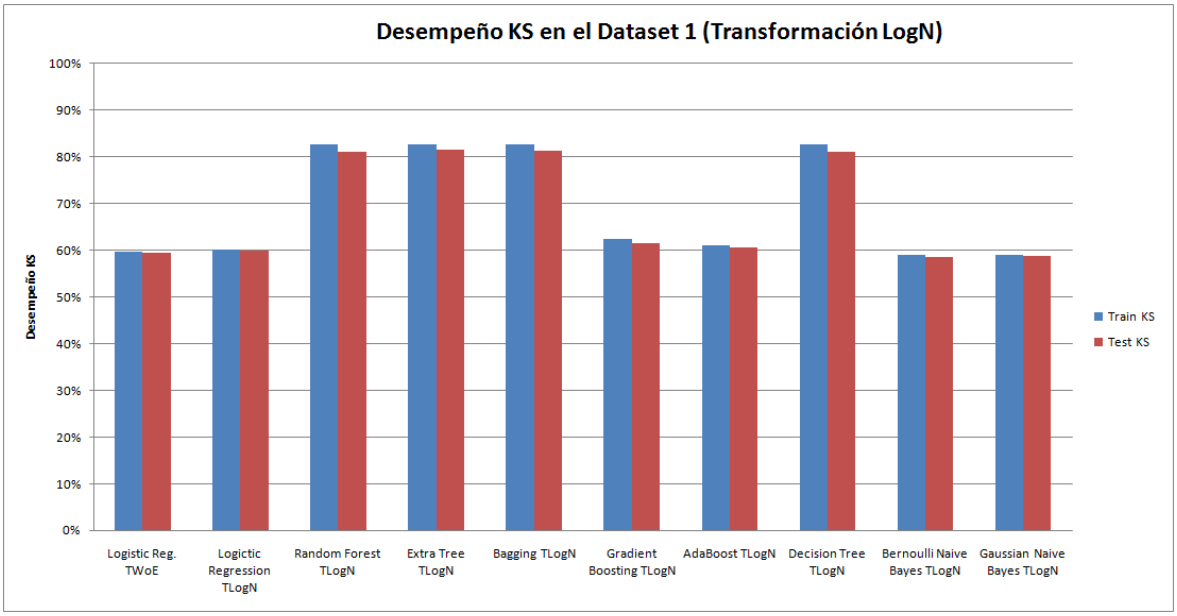


Figura 4.17: Desempeño del KS en el *Dataset* 1 al realizar transformación *Log* en las variables continuas

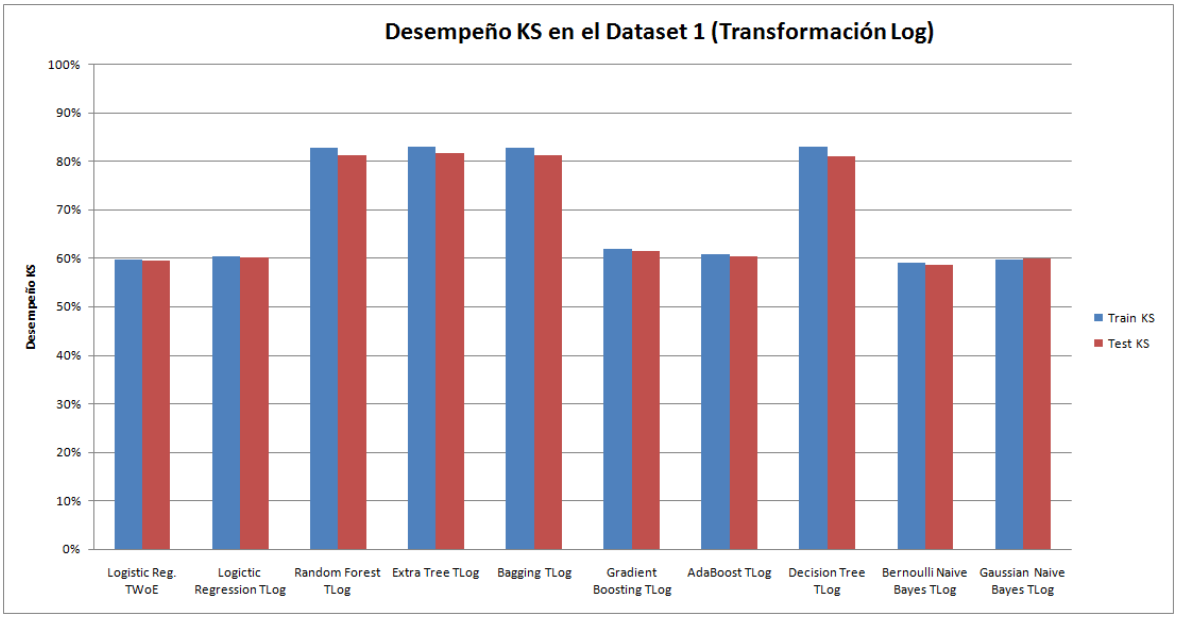


Figura 4.18: Desempeño del KS en el *Dataset* 1 al realizar transformación *Sqrt* en las variables continuas

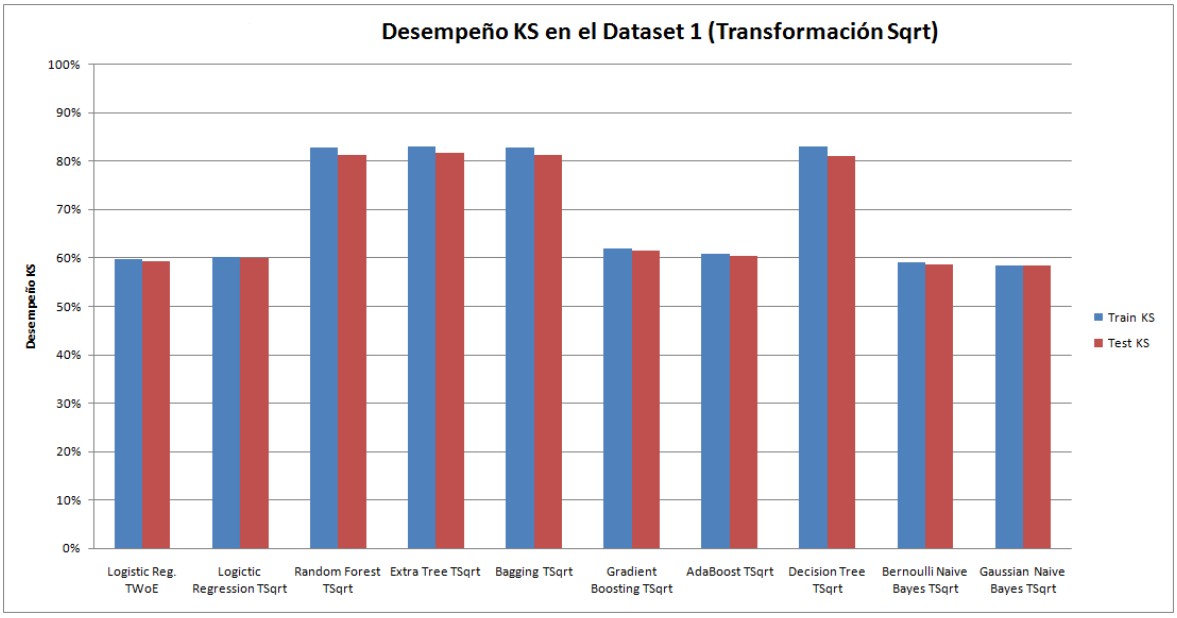


Figura 4.19: Desempeño del KS en el *Dataset* 4 sin transformación en las variables continuas

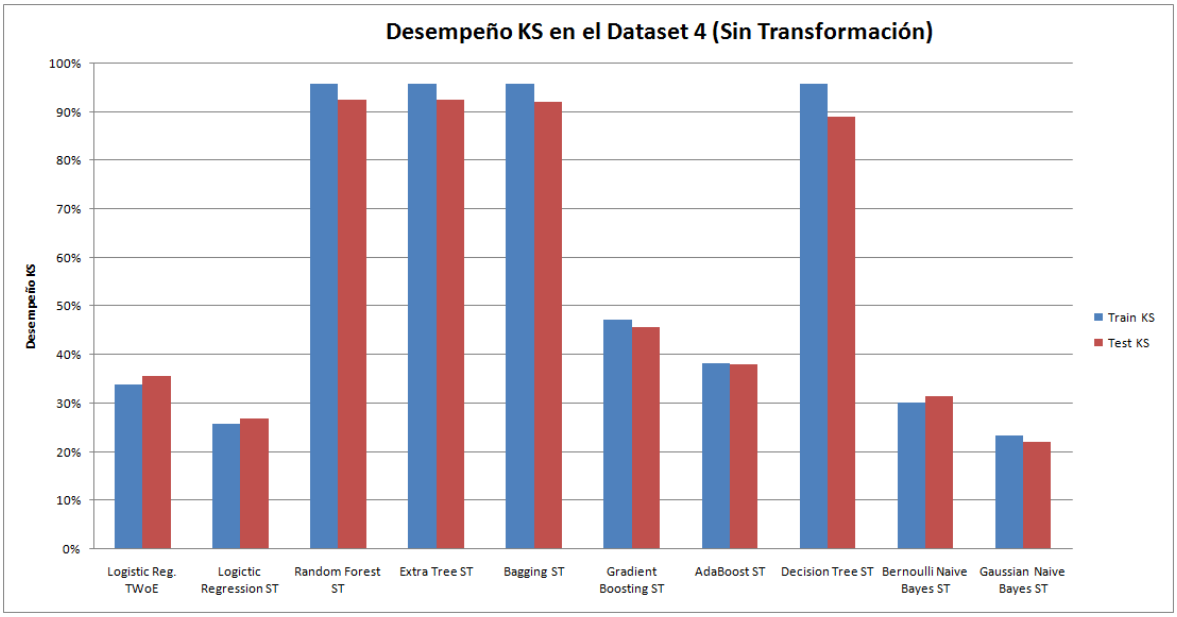


Figura 4.20: Desempeño del KS en el *Dataset* 4 al realizar transformación *LogN* en las variables continuas

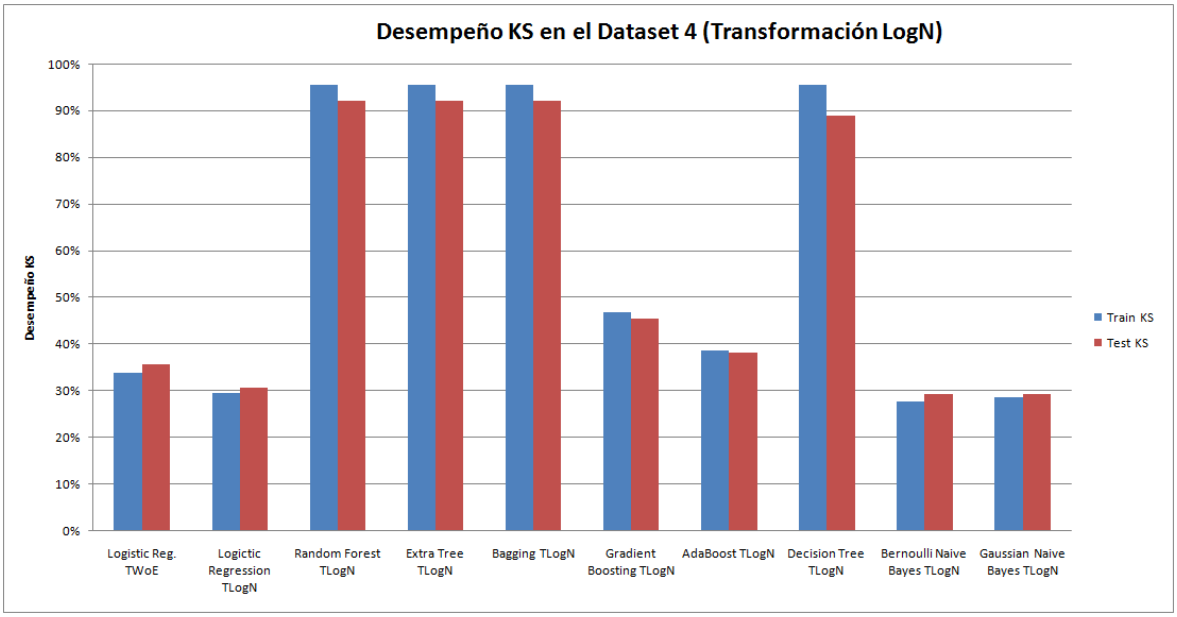


Figura 4.21: Desempeño del KS en el *Dataset* 4 al realizar transformación *Log* en las variables continuas

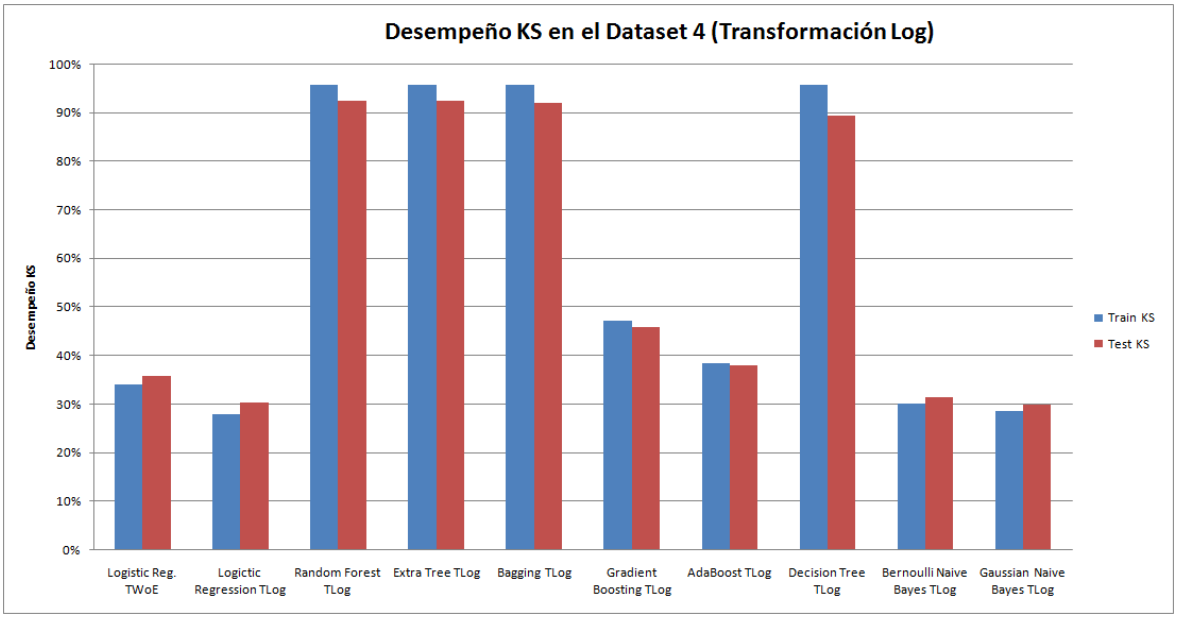


Figura 4.22: Desempeño del KS en el *Dataset* 4 al realizar transformación *Sqrt* en las variables continuas

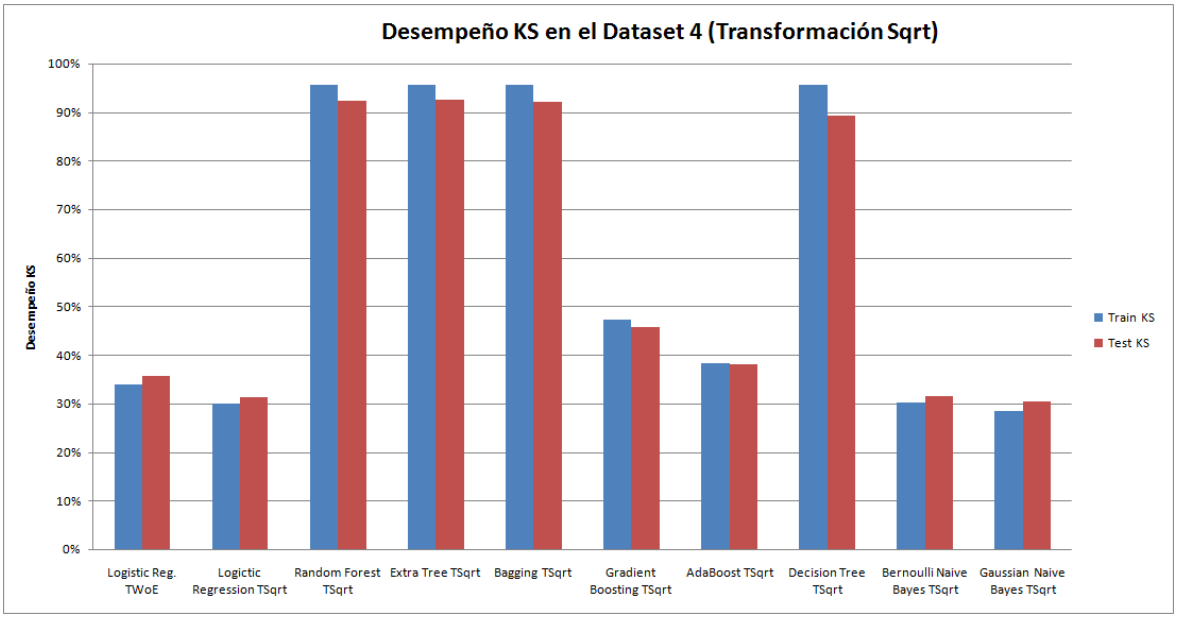


Figura 4.23: Desempeño del KS en el *Dataset* 6 al sin transformación en las variables continuas

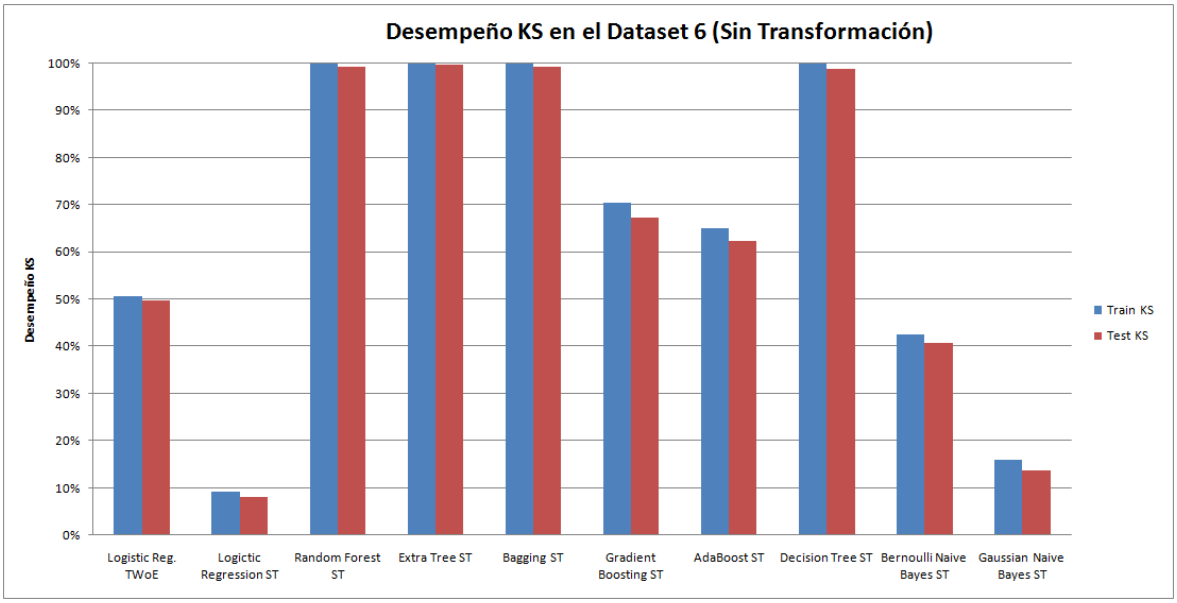


Figura 4.24: Desempeño del KS en el *Dataset* 6 al realizar transformación *LogN* en las variables continuas

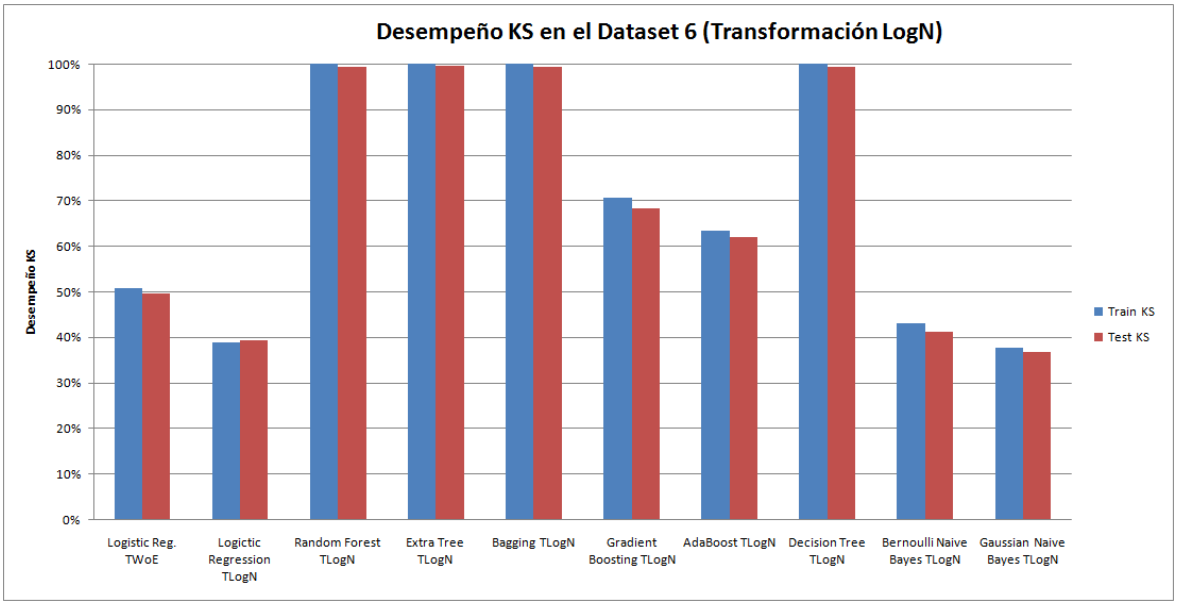


Figura 4.25: Desempeño del KS en el *Dataset* 6 al realizar transformación *Log* en las variables continuas

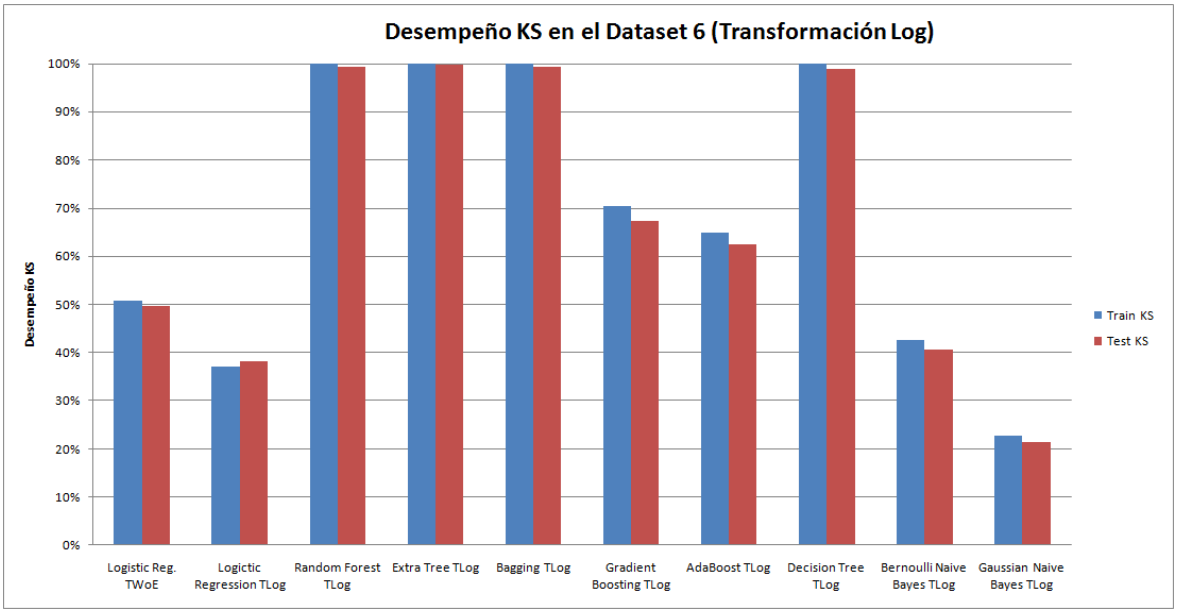
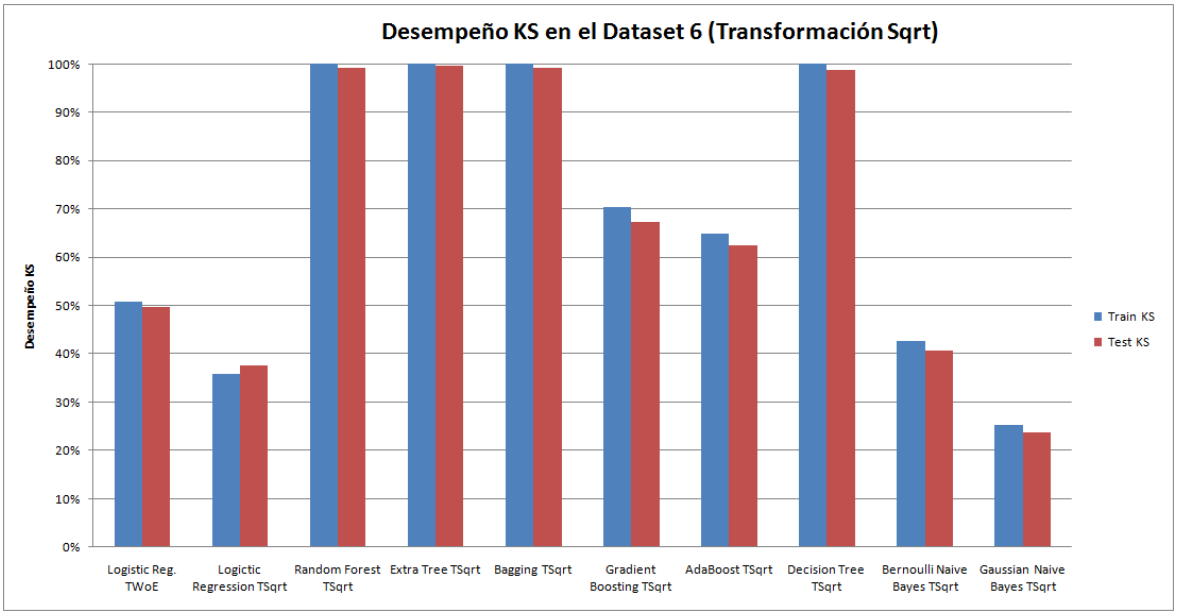


Figura 4.26: Desempeño del KS en el *Dataset* 6 al realizar transformación *Sqrt* en las variables continuas



Respecto al beneficio de no utilizar la transformación WoE, se puede observar en las figuras 4.15 a 4.18 que la ganancia en poder de discriminación (KS) entre clientes ”buenos” y ”malos” es considerable para las variables del *Dataset* 1 para *Random Forest*, *Extra Tree*, *Bagging* y *Decision Tree*, dados los argumentos anteriores.

Una situación particular que se dio durante la etapa 2, fue que el *Dataset* 1 tiene variables más estables, provocando que la ganancia o pérdida de KS no fuera tan significativa respecto al resto de los *dataset*. Esto se vuelve a reflejar nuevamente para esta etapa, provocando en los otros *datasets* una diferencia en el resultado para *Logistic Regression* cuando no se utiliza la transformación WoE; lo cual es notorio al ver en las figuras 4.19 a 4.26, donde el KS, tanto en los datos de entrenamiento como de prueba, baja significativamente para los *Datasets* 4 y 6.

Luego de todos los experimentos realizados, se comprueba empíricamente que las tres etapas realizadas arrojan buenos resultados para las técnicas de ensamblaje y en las técnicas de selección de variables que involucran algoritmos de ensamblaje para el calculo de la importancia de las variables.

El equipo de analistas de desarrollo de modelos de la Institución Bancaria encuentra valor en los descubrimientos encontrados y esperan poder llevar a la práctica todo lo positivo para que su metodología sea actualizada. Se aspira que el trabajo técnico implementado (los *scripts* en *Python*) se adapten para que estén dentro de los procesos automatizados y se construyan modelos de *credit scoring* con mejor poder de predicción.

Conclusiones

Luego de esta investigación se tiene una mirada más clara del proceso de desarrollo de un modelo de *scoring* y los potenciales beneficios de las nuevas técnicas de modelamiento que existen en herramientas de programación como Python. Cabe destacar que estas conclusiones aplican para el contexto de la investigación realizada, y podrían variar para otros *datasets*, aunque se recomienda tomarlas como referencia para trabajos similares.

Los primeros experimentos usando técnicas de modelamiento para comparar los beneficios versus la regresión logística comprueban que las técnicas de ensamblaje, específicamente, ***Random Forest***, ***Extra Tree***, ***Bagging*** y ***Gradient Boosting*** logran generar un modelo con mejor poder de discriminación capaz de acertar con mayor precisión qué cliente sera bueno. Además, si el contexto del problema a resolver es más complejo, es decir, presenta una cantidad de variables explicativas mayor, estas técnicas aumentan su diferencia de desempeño contra la regresión logística. Esto se debe principalmente porque estos algoritmos utilizan árboles de decisión para aprender los patrones del problema, los que se adaptan de mejor manera a clasificar en niveles altos de complejidad; pero por esta misma capacidad de adaptación, son más propensos a sobreajustarse a los datos de entrenamiento en el caso que no se controle la expansión de los árboles (parámetros de profundidad y división).

Los experimentos en la selección de variables con nuevas técnicas demuestran que una buena heurística a seguir, es probar muchas técnicas de selección (como las de correlación y convergencia en KS) en paralelo y en base a los resultados, **escoger la que mejor se adapte al problema**. Debido a que no existe la mejor técnica de selección para todos los problemas, conviene aplicar por lo menos dos, para realizar un análisis con las variables resultantes que devuelve cada técnica y su desempeño en el modelo. Además, existe un *trade off* de a mayor

cantidad de variables, mejor es el desempeño del modelo. La selección de variables busca reducir justamente esta cantidad de variables al mínimo posible sin perder tanto desempeño, con el fin de facilitar la salida del modelo en la explicación de por qué el cliente es ”bueno” o ”malo”.

Los últimos experimentos de la investigación en la transformación de las variables, demuestran que la **transformación WoE es perjudicial** para las técnicas de ensamblaje y los arboles de decisión, debido a que la categorización realizada con WoE acota en gran parte el poder predictivo de cada variable, por lo que el modelo va a perder mucho desempeño debido a esta pérdida de poder predictivo en sus variables. Esto sólo es válido para las técnicas basadas en árboles de decisión utilizadas en los experimentos.

Esta investigación espera que beneficie directamente al área de desarrollo de modelos de la Institución Bancaria, pero para lograr esto se debe comenzar a utilizar las herramientas *open source* como Python dentro de su metodología por el gran potencial que entrega junto a la biblioteca de *machine learning* **Scikit-learn**. Al realizarse estos cambios, la calidad de los modelos desarrollados aumentará significativamente y los analistas tendrán un abanico de posibilidades para desarrollar un modelo. Por otro lado, los tiempos de ejecución de procesos automatizados con herramientas de programación reducen significativamente los tiempos de espera para cada etapa del desarrollo, ahorrando horas en la ejecución de los proyectos.

Una de las medidas sobre la que se debería hacer hincapié, es replicar y mejorar todo el proceso de desarrollo de modelos fuera del Software IBM SPSS Modeler, con el fin de aprovechar las herramientas *open source* que existen hoy para construir modelos predictivos, siempre y cuando estos procesos sean automatizados y no se requiera programar todo desde cero para cada proyecto.

En relación al enfoque CRISP-DM en reversa, resulta una buena forma de abordar la investigación si el enfoque es siempre experimentar con nuevas técnicas de modelamiento, con el fin de comparar los resultados de las métricas en base a los modelos existentes ya entrenados con parte del *set* de datos y testeados con el resto del mismo. En el caso que la investigación buscara otro enfoque, se recomienda utilizar CRISP-DM para abordar un proyecto completo desde lo que quiere el negocio, hasta tener resultados de un modelo.

Respecto a la herramienta de programación usada, Python como herramienta para la minería de datos y máquinas de aprendizaje, es muy recomendada ya que existen bibliotecas bien avanzadas para el tratamiento de datos como **pandas** y para el uso de técnicas de modelamiento como **Scikit-learn**, además cuenta con la biblioteca de visualización **matplotlib** para ver de mejor forma los resultados. Por último, se destaca que la comunidad de desarrolladores en Python es amplia, por lo que existe buena documentación y soporte respecto a las dudas del uso de estas bibliotecas en distintos sitios web.

En cuanto a la relación de esta memoria con lo aprendido en la carrera de Ingeniería Civil en Informática, se destaca el pensamiento lógico para resolver problemas, la aplicación de métodos cuantitativos desde el punto de vista estadístico y el uso de herramientas de inteligencia artificial para resolver problemas reales. Por otro lado, para esta investigación, fue útil lo aprendido en las siguientes asignaturas:

- **Estadística Computacional**: el uso de la estadística es crucial para empezar a entender los datos y sus cualidades desde el lado cuantitativo.
- **Computación Científica**: lo aprendido en este curso por el lado aplicativo fue importante para el manejo de datos con Python.
- **Inteligencia Artificial**: las nociones de heurísticas y forma de resolver problemas mediante algoritmos de optimización motivo en querer especializarse en esta área.
- **Patrones de Reconocimiento en Minería de Datos**: lo enseñado en este curso de forma introductoria a esta área motivó a querer desarrollar este tema de memoria.
- **Máquinas de Aprendizaje**: este electivo de especialidad fue crucial para aplicar las técnicas de modelamiento desde las habilidades técnicas gracias a las tareas y presentaciones realizadas en el curso usando **Scikit-learn**.

Por otro lado, las asignaturas de Programación y Estructura de Datos fueron cruciales para implementar la solución tecnológica de forma correcta y eficiente, ya que no sólo basta con que la solución de automatización funcione, si no que ésta haya sido desarrollada en forma óptima para agregar valor al negocio desde el punto de vista de tener algoritmos de

tiempos de ejecución cortos y que no fallan al usarse por profesionales que necesitan estas automatizaciones.

Finalmente, se recomienda como extensiones a este estudio:

- **Optimización de hiper-parámetros en técnicas de modelamiento**: la investigación se vio complicada varias veces por generar modelos con sobreajuste debido a no usar los mejores parámetros para restringir los árboles de decisión que utilizaban gran parte de las técnicas de modelamiento. Por otro lado, la búsqueda de hiper-parámetros es costosa y compleja, ya que requiere distintas pruebas donde se utilice la misma técnica de modelamiento varias veces, pero modificando los parámetros de entrada hasta encontrar el modelo óptimo.

- **Uso de más técnicas de modelamiento en bibliotecas distintas**: la investigación se acotó al uso de técnicas dentro de Scikit-learn, pero existen mas técnicas interesantes de comparar con la regresión logística como Redes Neuronales, que se encuentran en **TensorFlow** [11].

- **Desarrollo de modelos de *credit scoring* con *cluster computing***: un gran problema que cada vez es más notorio para las instituciones financieras, es la capacidad de procesar grandes volúmenes de datos en un tiempo prudente, para esto se puede utilizar bibliotecas de *cluster computing* para *machine learning*, como por ejemplo **MLib en Apache Spark**, con el fin de desarrollar modelos de *credit scoring* en tiempo real a gran escala utilizando tecnologías de *Big Data* con *Hadoop*.

Bibliografía

[1] Raymond Anderson. *The credit scoring toolkit: theory and practice for retail credit risk management and decision automation*. Oxford University Press, 2007.

[2] Ana Isabel Rojão Lourenço Azevedo and Manuel Filipe Santos. Kdd, semma and crisp-dm: a parallel overview. *IADS-DM*, 2008.

[3] Jason Brownlee. A tour of machine learning algorithms. *Machine Learning Mastery*, 2013.

[4] Goedele Dierckx. Logistic regression for credit scoring. *Wiley StatsRef: Statistics Reference Online*, 2004.

[5] David W Hosmer and Stanley Lemeshow. *Special topics*. Wiley Online Library, 2000.

[6] Stefan Lessmanna, H Seowb, Bart Baesenscd, and Lyn C Thomasd. Benchmarking state-of-the-art classification algorithms for credit scoring: A ten-year update. In *Credit Research Centre, Conference Archive*, 2013.

[7] J.H. Orallo, M.J.R. Quintana, and C.F. Ramírez. *Introducción a la minería de datos*. Fuera de colección. Editorial Alhambra S. A. (SP), 2004.

[8] SBIF. Compendio de normas contables para bancos, capitulo b-1 al b-7. *Superintendencia de Bancos e Instituciones Financieras de Chile*, 2014.

[9] Scikit-learn.org. API Machine Learning en Python, parámetros técnicas en scikit-learn. `http://scikit-learn.org/stable/modules/classes.html`. Visitado: 2017-10-22.

[10] Naeem Siddiqi. *Credit risk scorecards: developing and implementing intelligent credit scoring*, volume 3. John Wiley & Sons, 2012.

[11] Tensorflow.org. API Neural Network en Python, tensorflow. `https://www.tensorflow.org/api_guides/python/nn`. Visitado: 2017-11-05.