

**UNIVERSIDAD TÉCNICA FEDERICO SANTA  
MARÍA**

**DEPARTAMENTO DE ELECTRÓNICA  
VALPARAISO - CHILE**



**“DISEÑO DE MOTOR DE RECOMENDACIÓN  
DE ACCIONES MARKOVIANO PARA UN  
SISTEMA DE COBRANZA DIGITAL”**

**ALEX ALONSO PEÑA PAVEZ**

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO  
CIVIL TELEMÁTICO**

**Profesor Guía: Mauricio Araya**  
**Profesor Correferente: Nicolás Jara**

**Agosto - 2020**

## **DEDICATORIA**

A mis padres, a mi hermano y a toda mi familia

Gracias por todo su apoyo incondicional.

## AGRADECIMIENTOS

Agradezco en primer lugar a mi familia, que siempre me han dado apoyo incondicional al ser uno de los primeros de la familia en ir a la Universidad y que ha significado un motivo de orgullo para ellos. Tuve la suerte de que mis abuelos por parte materna y paterna me vieran entrar, pero desafortunadamente tres de ellos partieron de esta vida antes de poder terminar la carrera, así que este trabajo está dedicado con mucho cariño a mis abuelos Benjamín, Agustín y Teresa que ya no se encuentran en este mundo y a mi abuela Susana quien tengo la suerte de que podrá ver la culminación de mi carrera. Agradezco especialmente a mis padres, Marcela y Alex, quienes me han apoyado en este tan largo camino y que me han inculcado grandes valores muy importante para mi formación valórica y personal. Gracias a su tremendo esfuerzo siempre constante que me permitió poder estudiar en la Universidad y a ellos les debo esta oportunidad que no muchas personas tienen la suerte de tenerla.

A mi hermano menor Diego, que más de alguna vez significó una fuente de inspiración para nunca rendirme en el camino y siempre he querido ser un buen ejemplo para él y ser su inspiración cuando sea su turno de aventurarse en la educación superior.

También doy gracias a todos los amigos que tuve la oportunidad de conocer en la universidad, especialmente dentro de mi generación de telemáticos los cuales han sido un gran aporte dentro de mi formación como persona. He tenido la suerte de vivir grandes experiencias gracias a ellos tanto en el ámbito académico como en nuestros tiempos libres y me han quedado grabado para siempre todo lo que he aprendido de ellos. Mis amigos y compañeros, Sergio, Luis, Jonathan, Simón, Javier, Francisco, Patricio y Pedro muchas gracias por haber sido parte de mi vida universitaria.

Doy gracias también a Antonio Martínez, quien me ha dado consejos muy útiles para adentrarse en el mundo del *data science* durante mi periodo de práctica y posterior proceso de memoria.

Finalmente, agradezco a todos los profesores que han sido parte de mi formación académica especialmente a mi profesor guía Mauricio Araya que me ha guiado durante el proceso de titulación y que ha tenido harta paciencia para enseñarme los caminos que debía tomar dentro de esta memoria.

## RESUMEN

**Resumen**— Este documento presenta un problema existente en los procesos de cobranza de deuda actuales, en donde se busca planificar una estrategia adecuada para interactuar, a través de múltiples canales o mecanismos, con deudores de un sistema. Esto se hace con el objetivo de obtener un retorno de sus deudas impagas. Para esto, se propone una solución, modelando el problema como un proceso decisional de Markov de horizonte finito, el cual se pretende resolver con algoritmos de programación dinámica. De esta forma se busca estimar a través del aprendizaje de máquinas y un enfoque estadístico la incerteza del comportamiento de los deudores, presente en un sistema. Finalmente, se busca una solución aproximada que sea capaz de aprender de un modelo a priori y lograr automatizar el proceso de forma inteligente para un sistema digital de cobranza.

**Palabras Clave**—Proceso de decisión de Markov, Programación Dinámica, Optimización, Planificación, Proceso de cobranza de deudas.

## ABSTRACT

**Abstract**— *This document presents an existing problem in the current debt collection processes, where it seeks to plan an adequate strategy to interact, through multiple channels or mechanisms, with debtors of a system. This is done with the objective of obtaining a return on their unpaid debts. For this, a solution is proposed, modeling the problem as a finite horizon Markov decisional process, which is intended to solve with dynamic programming algorithms. In this way, we seek to estimate through machine learning and a statistical approach the uncertainty of the debtors' behavior, present in a system. Finally, an approximate solution is sought that is capable of learning from an a priori model and achieving intelligent automation of the process for a digital collection system.*

**Keywords**— *Markov Decision Process, Dynamic Programming, Optimization, Planning, Debt Collection Process.*

## GLOSARIO

AI: *Artificial Intelligence*, traducción al inglés de “Inteligencia Artificial”

DP: *Dynamic Programming*, traducción al inglés de “Programación Dinámica”

GB: *Gradient Boosting*, técnica de Machine Learning que es utilizado como clasificador.

MDP: *Markov Decision Process*, traducción al inglés de “Proceso de decisión de Markov”

ML: *Machine Learning*, traducción al inglés de “Aprendizaje Automático”

RF: *Random Forest*, técnica de Machine Learning que es utilizado como clasificador.

RL: *Reinforcement Learning*, traducción al inglés de “Aprendizaje Reforzado”

SVM: *Support Vector Machines*, técnica de Machine Learning que es utilizado como clasificador.

XGB: *Extreme Gradient Boosting*, técnica de Machine Learning que es utilizado como clasificador.

# INDICE DE CONTENIDOS

|                                                          |    |
|----------------------------------------------------------|----|
| ABSTRACT .....                                           | 4  |
| INDICE DE FIGURAS .....                                  | 8  |
| INDICE DE TABLAS.....                                    | 10 |
| INTRODUCCIÓN.....                                        | 11 |
| CONTEXTO .....                                           | 12 |
| 1. DEFINICIÓN DEL PROBLEMA.....                          | 13 |
| 2. OBJETIVOS.....                                        | 14 |
| 2.1. OBJETIVOS GENERALES .....                           | 14 |
| 2.2. OBJETIVOS ESPECÍFICOS .....                         | 14 |
| 3. MARCO CONCEPTUAL .....                                | 15 |
| 3.1. REINFORCEMENT LEARNING Y PROGRAMACIÓN DINÁMICA..... | 15 |
| 3.2. PROCESO DE DECISIÓN DE MARKOV .....                 | 15 |
| 3.3. MODELOS CLASIFICADORES DE MACHINE LEARNING .....    | 16 |
| 3.4. TÉCNICAS DE MUESTREO .....                          | 16 |
| 3.4.1. UNDERSAMPLING.....                                | 16 |
| 3.4.2. OVERSAMPLING .....                                | 17 |
| 3.5. MÉTRICAS DE DESEMPEÑO .....                         | 18 |
| 3.5.1. MATRIZ DE CONFUSIÓN.....                          | 18 |
| 3.5.2. PRECISION .....                                   | 18 |
| 3.5.3. RECALL .....                                      | 19 |
| 3.5.4. F1-SCORE .....                                    | 19 |
| 3.6. MATRIZ DE TRANSICIÓN .....                          | 19 |
| 3.7. ALGORITMO DE ITERACIÓN POR VALOR.....               | 20 |
| 3.8. HERRAMIENTAS TECNOLOGICAS.....                      | 20 |
| 4. PROPUESTA DE SOLUCIÓN.....                            | 21 |
| 4.1. RESTRICCIONES DEL PROBLEMA .....                    | 21 |
| 4.2. GRAFO MDP .....                                     | 22 |

|        |                                            |    |
|--------|--------------------------------------------|----|
| 4.3.   | MICRO MATRICES DE TRANSICIÓN .....         | 23 |
| 4.4.   | SOLUCIÓN MDP: ITERACIÓN POR VALOR .....    | 24 |
| 4.5.   | DATOS REALES .....                         | 25 |
| 4.6.   | ENTRENAMIENTO Y TESTEO .....               | 27 |
| 4.7.   | MODELOS CLASIFICADORES .....               | 28 |
| 4.8.   | EXPERIMENTOS CON DATOS REALES .....        | 29 |
| 5.     | VALIDACIÓN DE LA SOLUCIÓN .....            | 35 |
| 5.1.   | SIMULACIÓN .....                           | 35 |
| 5.1.1. | DISTRIBUCIÓN DE PROBABILIDAD DE PAGO ..... | 35 |
| 5.1.2. | CREACIÓN DE DATASET FICTICIOS .....        | 36 |
| 5.1.3. | ENTRENAR MODELOS DE CLASIFICACIÓN .....    | 37 |
| 5.1.4. | RESOLUCIÓN MDP .....                       | 38 |
| 5.1.5. | RESULTADOS .....                           | 40 |
| 6.     | CONCLUSIONES.....                          | 49 |
|        | REFERENCIAS BIBLIOGRÁFICAS .....           | 53 |

## INDICE DE FIGURAS

|                                                                                                          |    |
|----------------------------------------------------------------------------------------------------------|----|
| FIGURA 1. ESQUEMA UNDERSAMPLING.....                                                                     | 16 |
| FIGURA 2. ESQUEMA OVERSAMPLING.....                                                                      | 17 |
| FIGURA 3. ESQUEMA SMOTE.....                                                                             | 17 |
| FIGURA 4. MATRIZ DE CONFUSIÓN .....                                                                      | 18 |
| FIGURA 5. PRECISIÓN. ....                                                                                | 18 |
| FIGURA 6. RECALL. ....                                                                                   | 19 |
| FIGURA 7. F1-SCORE. ....                                                                                 | 19 |
| FIGURA 8. MATRIZ DE TRANSICIÓN.....                                                                      | 19 |
| FIGURA 9. ALGORITMO ITERACIÓN POR VALOR. FUENTE: “ <i>REINFORCEMENT LEARNING, AN INTRODUCTION</i> ” .... | 20 |
| FIGURA 10. GRAFO MDP. ....                                                                               | 22 |
| FIGURA 11. EJEMPLO DE TRANSICIÓN DE UN USUARIO EN PERIODO DE 6 DÍAS.....                                 | 23 |
| FIGURA 12. ENFOQUE DE MICRO MATRICES .....                                                               | 24 |
| FIGURA 13. ALGORITMO DE ITERACIÓN POR VALOR. FUENTE: ELABORACIÓN PROPIA. ....                            | 24 |
| FIGURA 14. ESQUEMA ENTRENAMIENTO Y TESTEO. ....                                                          | 27 |
| FIGURA 15. ESQUEMA SIN PERFIL DE USUARIO. ....                                                           | 28 |
| FIGURA 16. ESQUEMA CON PERFIL DE USUARIO. ....                                                           | 28 |
| FIGURA 17. ROC AUC MODELOS CON DATASET ORIGINAL.....                                                     | 29 |
| FIGURA 18. EXPLICACIÓN DESBALANCE EN DATASET. ....                                                       | 30 |
| FIGURA 19. RELACIÓN PAGO V/S NO PAGO DE UN MES DATASET REAL. ....                                        | 30 |
| FIGURA 20. PAGADORES DIARIOS DE UN MES DE DATASET REAL.....                                              | 31 |
| FIGURA 21. MÉTRICAS DATASET ORIGINAL.....                                                                | 31 |
| FIGURA 22. MÉTRICAS UNDERSAMPLING NO USER.....                                                           | 32 |
| FIGURA 23. MÉTRICAS UNDERSAMPLING + USER .....                                                           | 32 |
| FIGURA 24. MÉTRICAS OVERSAMPLING NO USER .....                                                           | 33 |
| FIGURA 25. MÉTRICAS OVERSAMPLING + USER.....                                                             | 33 |
| FIGURA 26. ESTRUCTURA DE SIMULACIÓN. ....                                                                | 35 |
| FIGURA 27. PORCENTAJE DE PAGO, SEGÚN DISTRIBUCIONES PROBABILÍSTICAS WEIBULL Y BETA.....                  | 36 |
| FIGURA 28. EJEMPLO COMPORTAMIENTO DE PAGO DE CARTERA (5000 USUARIOS).....                                | 37 |
| FIGURA 29. SECUENCIA DE ENTRENAMIENTO DE MODELO ML.....                                                  | 37 |
| FIGURA 30. ESTRUCTURA ETAPA 4, SUB-ETAPA 1.....                                                          | 38 |
| FIGURA 31. ESTRUCTURA DE ALMACENAMIENTO DE RESULTADO.....                                                | 39 |
| FIGURA 32. ESTRUCTURAS DE VALOR, POLÍTICA Y PROBABILIDAD DE PAGO. ....                                   | 39 |
| FIGURA 33. ESTRUCTURA ETAPA 4, SUB-ETAPA 2.....                                                          | 40 |
| FIGURA 34. FORMULA DE VALOR ESPERADO. ....                                                               | 40 |
| FIGURA 35. VALORES POR DÍA DE MODELOS OVERSAMPLING. ....                                                 | 41 |
| FIGURA 36. VALORES POR DÍA DE MODELOS UNDERSAMPLING.....                                                 | 41 |
| FIGURA 37. XGB & GB SOBREMUESTREO V/S TEST .....                                                         | 42 |
| FIGURA 38. XGB & GB SUBMUESTREO V/S TEST. ....                                                           | 42 |
| FIGURA 39. CANTIDAD DE PAGOS POR PERFIL CON MODELOS UNDERSAMPLING.....                                   | 43 |
| FIGURA 40. CANTIDAD DE PAGOS POR PERFIL CON MODELOS OVERSAMPLING. ....                                   | 44 |
| FIGURA 41. ESTRATEGIA PERFIL 1 VS ESTRATEGIA MDP UNDERSAMPLING. ....                                     | 45 |
| FIGURA 42. ESTRATEGIA PERFIL 3 VS ESTRATEGIA MDP UNDERSAMPLING. ....                                     | 45 |
| FIGURA 43. ESTRATEGIA PERFIL 5 VS ESTRATEGIA MDP UNDERSAMPLING. ....                                     | 46 |
| FIGURA 44. ESTRATEGIA PERFIL 1 VS ESTRATEGIA MDP OVERSAMPLING. ....                                      | 46 |

|                                                                     |    |
|---------------------------------------------------------------------|----|
| FIGURA 45. ESTRATEGIA PERFIL 3 VS ESTRATEGIA MDP OVERSAMPLING. .... | 47 |
| FIGURA 46. ESTRATEGIA PERFIL 5 VS ESTRATEGIA MDP OVERSAMPLING. .... | 47 |
| FIGURA 47. PROPUESTA 1 ALTERNATIVA GRUPO CONTROL. ....              | 50 |
| FIGURA 48. PROPUESTA 2 ALTERNATIVA GRUPO CONTROL. ....              | 51 |

## INDICE DE TABLAS

|                                                         |    |
|---------------------------------------------------------|----|
| TABLA 1. EJEMPLO DATASET CON HISTORIAL DE ACCIONES..... | 26 |
| TABLA 2. EJEMPLO DATASET CON PERFILES DE USUARIO.....   | 26 |
| TABLA 3. RESULTADO DE MÉTRICAS DATASET REAL.....        | 34 |
| TABLA 4. MÉTRICAS DATASET SIMULADO.....                 | 38 |
| TABLA 5. TABLA DE RESULTADOS FINALES.....               | 43 |

## INTRODUCCIÓN

A lo largo de este documento se entrega una propuesta de cómo abordar el problema de recolección de deudas dentro de un sistema digital de cobranza. Esto será abordado con mayor detalle dentro del capítulo 1, en donde se explica cómo es abordado este tema en la actualidad y cuál es la tendencia que existe respecto a implementar tecnologías modernas. En el capítulo 2 se describe en detalle los objetivos de esta memoria que, en resumen, se busca entregar una solución a un problema de gestión o planificación de estrategias que intentan conseguir un retorno de deudas impagas dentro de un sistema de recolección de deudas digital y así tener un algoritmo que sea capaz de recomendar acciones. El capítulo 3 abordará en detalle los conceptos utilizados para resolver un proceso de decisión de Markov (MDP), en donde se explica para que son utilizados y también explicación de algoritmos existentes para resolverlos. También se explicarán técnicas de Machine Learning (ML), como lo son los modelos predictivos de clasificación como “*Random Forest (RF)*”, “*Support Vector Machines (SVM)*”, “*Gradient Boosting (GB)*” y “*Extreme Gradient Boosting (XGB)*”. Se expone la razón de su uso y métricas que son fundamentales para evaluar el rendimiento de las técnicas. Además, se introducen técnicas de muestreo o “sampling”, las cuales mejoran el desempeño en situaciones donde los modelos predictivos están sesgados por el desbalance presente en los datos de entrenamiento. En el capítulo 4 se aplican los conceptos del capítulo anterior definiendo restricciones para abordar el problema y simplificar el análisis de este. Estas restricciones convierten el MDP en un problema de horizonte finito lo que implica que el problema se aborda en ventanas temporales acotadas ya que para este problema en particular no se requiere tener soluciones a largo plazo. La comprobación de esta respuesta y lo efectiva que es, se analiza con mayor detalle en el capítulo 5 en donde se utiliza la simulación como herramienta de validación. Finalmente, en el capítulo 6 de conclusiones, se alude a los resultados obtenidos y también se destacan que aspectos se pueden mejorar, entregando propuestas de trabajos futuros que se podrían desarrollar.

## CONTEXTO

Este trabajo se desarrolla dentro de la empresa DIGEVO. Esta empresa se especializa en ofrecer soluciones tecnológicas para problemas existentes en distintas áreas del mundo empresarial. Mas en específico, el trabajo se desarrolla en el área de inteligencia artificial, la cual está diseñando un producto denominado “Stack de cobranza digital” que consiste en un conjunto de tecnologías que utilizan la inteligencia artificial para solucionar el problema de recolección de deuda. El documento busca entregar una propuesta para el motor cognitivo del producto que se encarga de generar la recomendación de acciones que deben ser ejecutadas y que buscan como objetivo el retorno de la deuda de un usuario que presenta morosidad en un sistema. El producto es ofrecido a distintas empresas que buscan gestionar de manera efectiva la recuperación de deudas impagas y automatizar los procesos disponibles. DIGEVO dispone de algunos datos brutos facilitados por las empresas que adquieren el producto como, por ejemplo, historiales de deudas, acciones gestionadas y datos demográficos de los cuales se utiliza una pequeña porción de ellos para poder entregar un prototipo que pueda ser utilizado dentro del motor cognitivo del producto.

## 1. DEFINICIÓN DEL PROBLEMA

El mundo de los negocios a desarrollado un interés en incorporar tecnología para aprovechar el potencial uso de los datos. Según algunos expertos, los datos se han vuelto uno de los recursos más valiosos del mundo, el petróleo de la nueva era [1]. El análisis y comprensión de estos ha tomado una gran importancia, ya que esto permite generar nuevas oportunidades de negocio y la optimización de procesos ya existentes. Actualmente muchas empresas desean innovar incorporando inteligencia artificial para optimizar y automatizar el proceso de recolección de deudas [2] que en general, son llevadas a cabo por agencias de cobranza. Este proceso consiste en el seguimiento de cuentas de clientes que registran morosidad o deudas impagas [3], por lo tanto, se busca un retorno de sus deudas a través de canales que son permitidos dentro del proceso como por ejemplo las llamadas, cartas, correos electrónicos, etc. Este tipo de proceso existe desde que se ha usado el dinero como medio de pago por productos y servicios, remontándose hasta la antigüedad. Es un tema que a lo largo de la historia se ha tratado de dar solución de distintas formas, desde los métodos más arcaicos como la extorsión y abusos de poder para cobrar una deuda, hasta métodos más civilizados como las acciones legales o las cartas a domicilio [4]. Muchas empresas no disponen de la experiencia necesaria, por lo que aparecen distintas organizaciones y agencias que se especializan en el ámbito de cobranzas [5]. Estas agencias actualmente utilizan distintas herramientas para conseguir estos objetivos y es muy necesario que optimicen la cantidad de recursos utilizados para ejercer estrategias de retornos de deudas. Un problema importante que se busca solucionar es la cantidad de acciones necesarias para que un deudor pague sus deudas, por lo que también importa la secuencia y como deben ser ejecutadas las acciones. A esto se le denomina “política” y es de suma importancia buscar la estrategia adecuada para los distintos perfiles de deudores que se puedan presentar en un sistema de deudores para lograr conseguir un buen porcentaje de retorno bajo el contexto de deudas.

## **2. OBJETIVOS**

### **2.1. OBJETIVOS GENERALES**

Diseñar un modelo de datos y programar un sistema que recomiende acciones y que posteriormente funcione como motor dentro de un sistema inteligente de gestión de cobranzas. Se busca una forma de recomendar la mejor acción que tenga una mayor probabilidad de obtener un retorno de deuda dentro de un marco temporal determinado. Esto se diferencia de los actuales sistemas de recomendación ya que estos en su esencia están ideados para realizar recomendaciones de ítems o productos como un libro o una película a través de las interacciones explícitas o implícitas que existen de parte del usuario con el producto. Finalmente se busca entregar un prototipo de modelo que involucre experimentos con distintos modelos y posteriormente evaluar las respuestas que entrega el modelo haciendo un análisis de lo obtenido.

### **2.2. OBJETIVOS ESPECÍFICOS**

- Construir un modelo basado en un Proceso de decisión de Markov de horizonte finito que represente los estados en que se encuentra un deudor durante el tiempo y las acciones que se toman en el proceso de gestión de cobranza para obtener retorno de deuda.
- Diseñar un modelo o vector de datos que sirva como representación del estado en que se encuentra un deudor y que contenga la información del pasado.
- Evaluar distintos modelos de machine learning para predecir probabilidades y comparar sus rendimientos con los datos disponibles.
- Implementar un algoritmo de programación dinámica para generar recomendación de acciones.
- Crear un simulador que genere una cartera de deudores y evaluar las recomendaciones generadas por el algoritmo de manera off-line.

### 3. MARCO CONCEPTUAL

#### 3.1. REINFORCEMENT LEARNING Y PROGRAMACIÓN DINÁMICA

La programación dinámica (DP) y el Aprendizaje Reforzado (RL) son técnicas que se utilizan para resolver problemas en donde se busca tomar una decisión en cada instante de tiempo y alcanzar un objetivo dentro de un determinado entorno. Usualmente estos problemas son formulados matemáticamente como un problema de optimización que busca la mejor “acción” o una regla que genera distintas secuencias de acciones dependiendo del entorno también denominada como “política” que se usa para maximizar una recompensa. Por lo general se considera una variable de tiempo que es discreta y se decide tomar una acción en cada paso de tiempo, lo que deriva en un problema secuencial de toma de decisiones. Estos problemas secuenciales aparecen en distintos campos como lo es la robótica y la Inteligencia artificial (AI). Ambas técnicas, RL y DP son utilizadas para resolver procesos decisionales de Markov, pero DP entrega políticas óptimas asumiendo un modelo matemático perfecto del entorno tal como lo es un MDP debido a esta razón tienen un límite ya que no en todos los casos se puede modelar perfectamente el funcionamiento de un entorno y además mientras más grande es el modelo, mayor es el gasto computacional para resolverlo con estos métodos [6]. En cambio, RL usa DP como base teórica de la resolución, pero no es necesario tener un modelo perfecto para solucionar el problema ya que lo hace mediante un agente que interactúa racionalmente con el entorno y en base a la experiencia busca maximizar la recompensa esperada a largo plazo [7]. Sin embargo, por más que RL se describe como un método que podría considerarse más actual que el enfoque clásico de DP, para este trabajo no será necesario usarlo debido a las restricciones que se utilizan para resolver el problema, que serán explicadas en la sección 4.

#### 3.2. PROCESO DE DECISIÓN DE MARKOV

Un proceso de decisión de Markov (MDP) [8] es una forma de modelar entornos estocásticos de toma de decisiones secuenciales que se compone por:

- Un conjunto de posibles estados  $S$ .
- Un conjunto de posibles acciones  $A$ .
- Una matriz de transición  $T(s, a, s')$ , donde:
  - $s$ : estado, donde  $s \in S$ .
  - $a$ : acción, donde  $a \in A$ .
  - $s'$ : estado siguiente, donde  $s' \in S$ .
- Una función de recompensa  $R(s, a)$ , con  $s \in S, a \in A$ .
- Una política  $\pi(s)$  que corresponde a la solución del MDP.

Un MDP cumple con la propiedad de Markov que dice: “*El futuro es independiente del pasado, dado el presente*” [8]. Esto quiere decir que el estado actual ( $s_t$ ) contiene toda la información relevante de los estados anteriores ( $s_0, s_1, s_2, s_3, \dots, s_{t-1}$ ), por lo tanto, no sirve tener más información de los estados pasados.

### 3.3. MODELOS CLASIFICADORES DE MACHINE LEARNING

Existen distintos tipos de enfoques dentro de Machine learning, pero el enfoque a utilizar será el uso de modelos de aprendizaje supervisado en los que el set de datos en que los resultados de salida son conocidos. Los modelos aprenden del resultado y se trabaja con los parámetros internos y el modelo podrá hacer predicciones adecuadas ante datos no procesados previamente. Dentro de este enfoque existen 2 subcategorías que son la clasificación y la regresión, pero para este problema en particular solo se usará la subcategoría de clasificación ya que se busca predecir una clase categórica binaria. Esto se aborda con más detalle en las secciones 4 y 5. Dentro de los modelos predictivos existentes se decide utilizar los siguientes 4 modelos acompañado de técnicas de muestreo que se presentan en la sección 3.4. Los modelos son:

- Random Forest (RF)
- Support Vector Machines (SVM)
- Gradient Boosting (GB)
- Extreme Gradient Boosting (XGB)

### 3.4. TÉCNICAS DE MUESTREO

Estas técnicas ayudan a equilibrar el espacio muestral cuando se presenta un set de datos desequilibrados, con el fin de eliminar parte del sesgo presente en la muestra. Con esto se busca hacer un entrenamiento de clases con un espacio muestral equilibrado.

#### 3.4.1. UNDERSAMPLING

Es una técnica que consiste en descartar muestras de la clase mayoritaria. La forma más común es eliminar de forma aleatoria muestras de la clase mayoritaria hasta tener un balance, sin embargo, se corre el riesgo de perder información importante y se modifica la distribución de datos original [9].

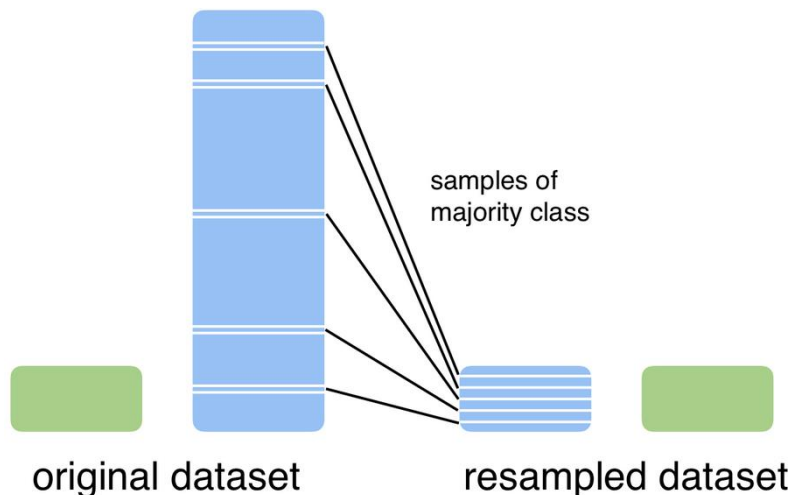


Figura 1. Esquema Undersampling.

### 3.4.2. OVERSAMPLING

Esta técnica al contrario de la técnica anterior busca el balance de datos a través de la creación de nuevas muestras de la clase minoritaria. Una alternativa es el duplicado de muestras minoritaria de manera aleatoria y también está la alternativa SMOTE (Synthetic Minority Oversampling Technique) que genera muestras sintéticas hasta alcanzar un balance. La dificultad de este método es que al generar más muestras cabe la posibilidad de generar ruido especialmente el método SMOTE y esto puede afectar al rendimiento de los clasificadores [9].

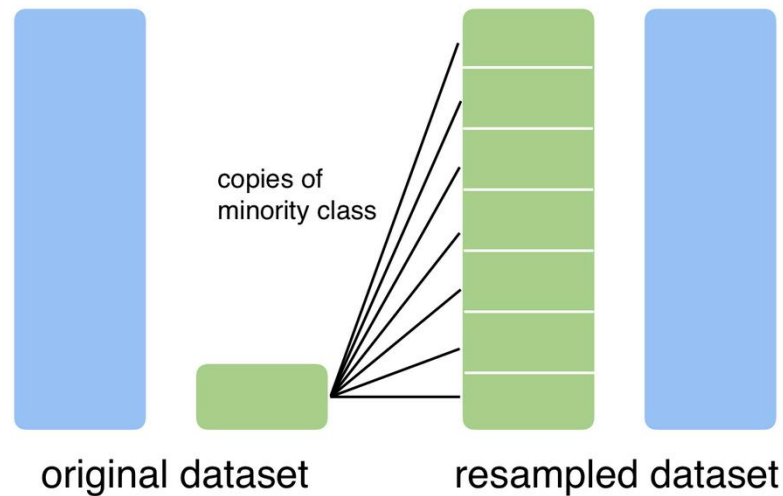


Figura 2. Esquema Oversampling.

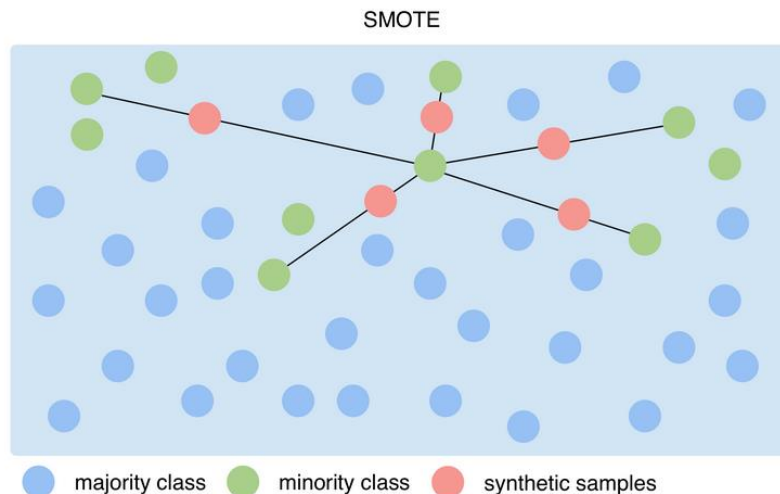


Figura 3. Esquema SMOTE

### 3.5. MÉTRICAS DE DESEMPEÑO

#### 3.5.1. MATRIZ DE CONFUSIÓN

Esta matriz es una herramienta que ayuda a visualizar el desempeño de un modelo. De donde se puede desprender mejores métricas que el tradicional “*accuracy*” [10]. La idea general de la matriz es contar el número de instancias de la clase de interés que son correctamente clasificadas y las veces en que falla. Esto se traduce de manera gráfica en la figura 4.

|          |   | predicción |    |
|----------|---|------------|----|
|          |   | 0          | 1  |
| realidad | 0 | TN         | FP |
|          | 1 | FN         | TP |

Figura 4. Matriz de confusión

- TP es la cantidad de *positivos* que fueron clasificados **correctamente** como positivos por el modelo.
- TN es la cantidad de *negativos* que fueron clasificados **correctamente** como negativos por el modelo.
- FN es la cantidad de *positivos* que fueron clasificados **incorrectamente** como negativos.
- FP es la cantidad de *negativos* que fueron clasificados **incorrectamente** como positivos.

#### 3.5.2. PRECISION

A través de la matriz de confusión se puede extraer información útil respecto a un modelo. Una métrica que se puede extraer es la precisión que es la exactitud de las predicciones positivas [10], lo que implica que es una métrica orientado a la calidad de las predicciones y se mide con la formula expresada en la figura 5.

$$precision = \frac{TP}{TP + FP}$$

Figura 5. Precisión.

### 3.5.3. RECALL

También denominada “*sensitivity*”, es una métrica complementaria a la precisión. Esta métrica entrega información respecto al ratio de instancias positivas que son correctamente detectadas por el modelo clasificador [10], lo que de cierta forma está orientado a la cantidad más que la calidad de las predicciones. Su fórmula esta descrita en la figura 6.

$$recall = \frac{TP}{TP + FN}$$

Figura 6. Recall.

### 3.5.4. F1-SCORE

Esta métrica agrupa la información contenida en las 2 métricas anteriores en un solo valor y es perfecta para comparar el desempeño entre modelos clasificadores. F1-score es la media armónica de la precisión y el recall, que les da un mayor peso a los valores pequeños y como consecuencia, la métrica será grande solo si ambas métricas tienen valores altos [10]. La forma de calcularlo se describe en la figura 7.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Figura 7. F1-score.

## 3.6. MATRIZ DE TRANSICIÓN

La matriz o función de transición representa las probabilidades que existen al pasar de un estado  $s_t$  a un estado  $s_{t+1}$ . En alguna literatura se denomina también como “modelo” o “modelo de comportamiento”.

$$\begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \\ P_{n1} & & P_{nn} \end{bmatrix}$$

Figura 8. Matriz de Transición

Esta matriz es de vital importancia para el algoritmo de DP que se va a utilizar para resolver el MDP propuesto en las secciones posteriores, ya que esta matriz es el corazón del algoritmo de iteración por valor, encargado de realizar recomendación de acciones y depende de esta matriz la calidad de las recomendaciones.

### 3.7. ALGORITMO DE ITERACIÓN POR VALOR

Este algoritmo es uno de los algoritmos para calcular una estrategia óptima de un MDP. El algoritmo busca resolver el MDP maximizando la ecuación de optimización Bellman para entregar como resultado final una política denominada  $\pi$  que es la estrategia que elige la acción que maximiza las recompensas del MDP. El pseudocódigo genérico de este algoritmo para casos de horizonte infinito se muestra en la figura 9. Para el caso de horizonte finito, se debe cambiar el factor  $\gamma$  por 1 y limitar el ciclo infinito sin usar un criterio de detención [11].

```
Initialize  $V(s)$ , for all  $s \in \mathcal{S}^+$ , arbitrarily except that  $V(\text{terminal}) = 0$ 

Loop:
|  $\Delta \leftarrow 0$ 
| Loop for each  $s \in \mathcal{S}$ :
|    $v \leftarrow V(s)$ 
|    $V(s) \leftarrow \max_a \sum_{s',r} p(s', r | s, a) [r + \gamma V(s')]$ 
|    $\Delta \leftarrow \max(\Delta, |v - V(s)|)$ 
until  $\Delta < \theta$ 

Output a deterministic policy,  $\pi \approx \pi_*$ , such that
 $\pi(s) = \operatorname{argmax}_a \sum_{s',r} p(s', r | s, a) [r + \gamma V(s')]$ 
```

**Figura 9. Algoritmo Iteración por valor. Fuente: “Reinforcement learning, an introduction”**

### 3.8. HERRAMIENTAS TECNOLÓGICAS

Para cubrir y hacer experimentos con toda la teoría explicada en la sección 3 (3.1 a 3.7) existen herramientas que ayudarán a facilitar el estudio de esto, dentro de ellas se utiliza el lenguaje de programación Python que proporciona las bibliotecas necesarias para aplicar toda la teoría. Dentro de estas se encuentra:

- Numpy
- Pandas
- Matplotlib
- Scikit-Learn

Las 2 primeras ayudan a organizar y almacenar información de manera estructurada y además se pueden realizar operaciones matriciales de manera más sencilla. La tercera herramienta ayuda en temas de visualización de resultados en forma gráfica. Finalmente, la última entrega soporte para realizar muestreos, generar modelos de clasificación para ser entrenados y es capaz de entregar información respecto a las métricas asociadas al modelo.

## 4. PROPUESTA DE SOLUCIÓN

Dentro de la propuesta se entrega una estructura de flujo de trabajo que será abordado con más detalles a lo largo de este capítulo. Finalmente se busca generar una recomendación o la mejor política que se pueda hacer en un instante de tiempo.

Este problema se formula como un proceso de decisión de Márkov (MDP) con horizonte finito y con decisiones realizadas en tiempo discreto. Esto quiere decir que las decisiones que entrega el modelo serán realizadas dentro de un marco temporal finito, para este caso, un mes y las decisiones serán tomadas en puntos discretos a través del tiempo como los son los días dentro de ese mes. Se plantea un modelo de datos para representar los estados (S) tal como se define dentro de un MDP y como salida se obtiene un dataset con el modelo propuesto. Posteriormente, se entrega el dataset a un módulo de Machine Learning clasificador para determinar la probabilidad de transición y crear una matriz de transición de estados. Finalmente, el MDP será resuelto con un algoritmo de iteración por valor. Para validar la respuesta, se hará una simulación previa, en donde se pretende crear desde cero un dataset ficticio con datos de usuario y acciones simuladas, que siga una distribución de probabilidad de pago inventada. La idea de la etapa de simulación es que el algoritmo sea capaz de encontrar las mejores estrategias que se hayan asignado durante la simulación o que encuentre una alternativa incluso mejor que la simulada. Al tener un simulador se pueden realizar pruebas off-line para compensar escasez de datos reales y ahorrar la opción de una prueba on-line que puede ser riesgosa para un prototipo. Esto se comprueba comparando los valores esperados que ofrece el simulador y se comparan con la solución del MDP que entrega el algoritmo.

### 4.1. RESTRICCIONES DEL PROBLEMA

Para efectos de simplificación del problema se establecen algunos supuestos y restricciones que facilitan el diseño de la solución y son adecuados para un prototipo inicial. Las restricciones son:

- Se considera que un mes contiene 30 días por lo tanto se establece un horizonte finito que será considerado como la cantidad máxima de días dentro del mes.
- Se modela el estado como lo que ocurre dentro de un día, por lo que un usuario como máximo puede pasar por 30 estados a lo largo del mes.
- Respecto a las acciones, se establece la cardinalidad de las acciones como 3, debido a la información que fue facilitada para este trabajo y solo se puede ejecutar una de estas por día.
- Al existir la primera restricción, se desprende que existe un avance en el tiempo y por lo tanto no existe forma de volver a un estado anterior. Esto afecta directamente a la matriz de transición de estado ya que contendrá una gran cantidad de valores nulos.

## 4.2. GRAFO MDP

A partir de las restricciones del problema, se plantea el diseño de un grafo para representar la probabilidad condicional de transición entre estados de pago o estados de no pago. Esta probabilidad ignora las acciones que se ejecutaron antes de obtener un pago y la probabilidad queda definida por la acción anterior debido a la propiedad de Markov.

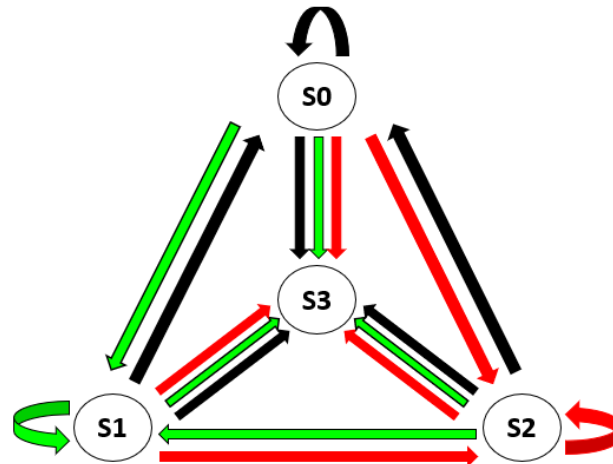


Figura 10. Grafo MDP.

Este grafo, representa la cantidad de estados presentes dentro de un día y las posibles acciones que se pueden tomar desde un determinado estado, en donde se tienen los siguientes elementos:

- Conjunto de acciones  $a_i$ :
  - $a_0$  : Hacer nada o hacer ninguna acción (NO ACC).
  - $a_1$  : Enviar un mensaje de texto (SMS).
  - $a_2$  : Hacer un llamado (CALL).
- Conjunto de estados  $s_i$ :
  - $s_0$  : Estado de “NO PAGO” dado que la última acción fue  $a_0$ .
  - $s_1$  : Estado de “NO PAGO” dado que la última acción fue  $a_1$ .
  - $s_2$  : Estado de “NO PAGO” dado que la última acción fue  $a_2$ .
  - $s_3$  : Estado de “PAGO” dado cualquier acción (estado terminal).
- Los colores de las flechas del grafo están asociados a cada acción disponible:
  - $a_0$  : Flechas de color Negro.
  - $a_1$  : Flechas de color Verde.
  - $a_2$  : Flechas de color Rojo.

En la figura 11 se puede ver un ejemplo de cómo implementar el grafo del MDP por día en una secuencia simple de acciones de un deudor en particular.

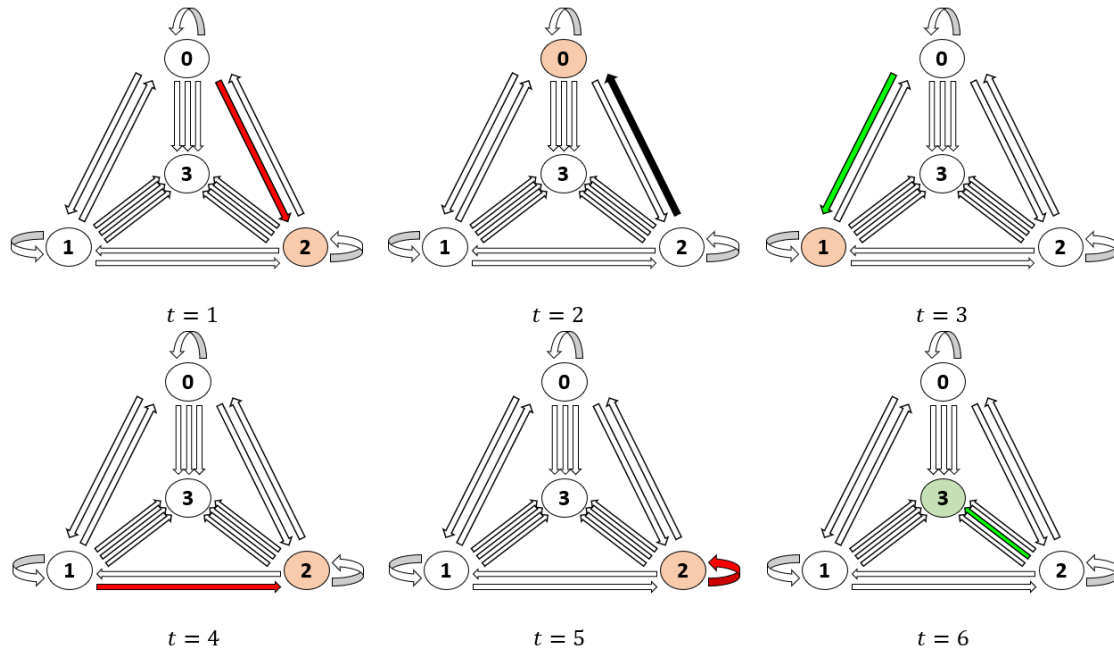


Figura 11. Ejemplo de transición de un usuario en periodo de 6 días.

#### 4.3. MICRO MATRICES DE TRANSICIÓN

Al resolver el problema se busca obtener una función o matriz de transición de estados, para posteriormente utilizarlo en el algoritmo con el que se pretende resolver el problema. Sin embargo, esta función se vuelve más compleja de calcular, a medida que aparecen más estados. Pero, dada las condiciones de este problema en particular y en cómo se ha optado por modelar el MDP, la función de transición de estados tendrá un enfoque de “micro matrices” de transición de estados, debido a que no es necesario calcular todas las probabilidades de transición de todos los estados como consecuencia de las restricciones del problema (ej. calcular transición de pagar con SMS en el día 30 dado que se está en el día 1, en este caso necesito las transiciones de lo que va a ocurrir al día siguiente, osea el día 2). Esto ayuda a simplificar el problema ya que calcular una matriz muy grande puede ser costoso en cuanto a tiempo, memoria y costo computacional al tener más estados, acciones y perfiles de usuario. En vez de eso, se calculan las transiciones necesarias para un día determinado, almacenando las probabilidades en matrices más pequeñas. Estas probabilidades se almacenan por acción, día y perfil de usuario, reduciendo la cantidad de valores nulos que puedan llegar a existir en una matriz voluminosa al guardar transiciones inexistentes.

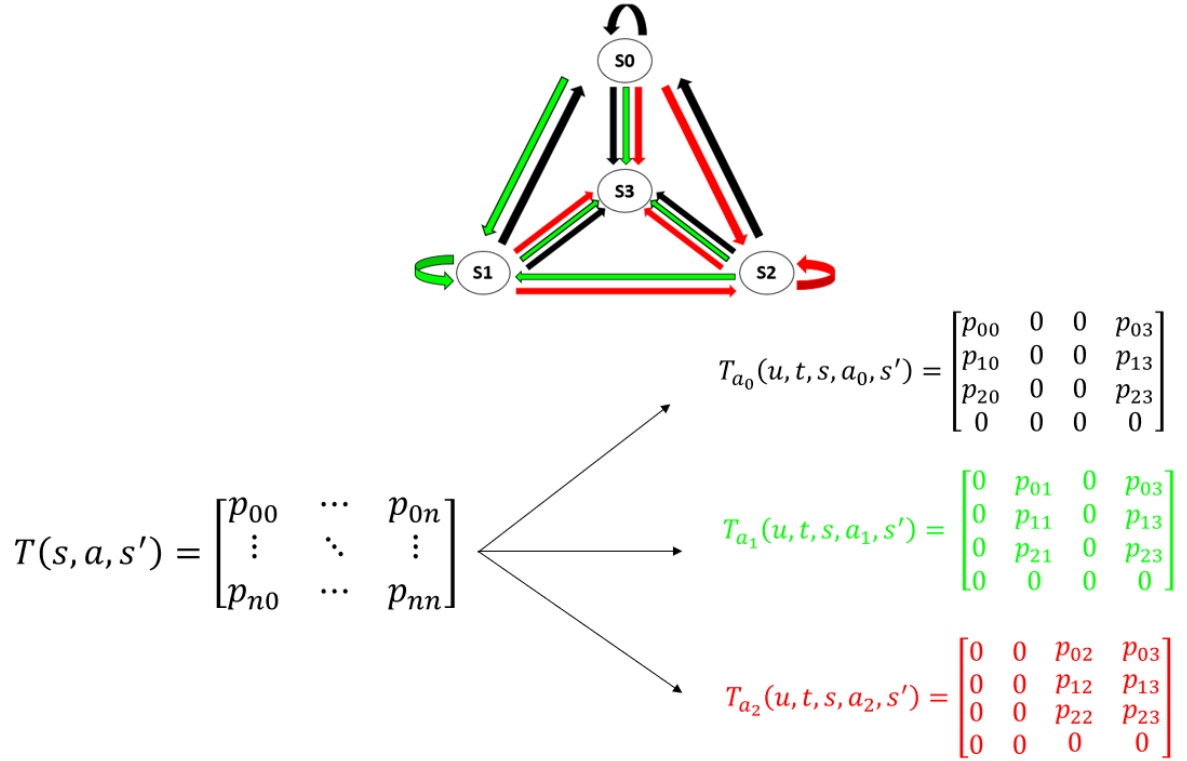


Figura 12. Enfoque de Micro Matrices

#### 4.4. SOLUCIÓN MDP: ITERACIÓN POR VALOR

Este MDP se busca resolver con el algoritmo de iteración por valores en donde se pretende buscar la acción que entregue la mayor recompensa en un determinado estado. El algoritmo usa la ecuación de Bellman para optimización y se fija un horizonte finito debido a las restricciones que se imponen inicialmente al modelar el problema. El pseudocódigo se formula como se muestra a continuación:

```

inicialización  $V(s)$  arbitrariamente;
while  $k \leq 30$  do
    for  $s \in S$  do
        for  $a \in A$  do
             $Q(s, a) = R(s, a) + \gamma \times \sum_{s' \in S} T(s, a, s') \times V(s')$ 
        end
         $V(s) = \max_a Q(s, a)$ 
    end
end

```

Figura 13. Algoritmo de iteración por valor. Fuente: Elaboración propia.

En donde:

- $k$ : cantidad de iteraciones del algoritmo, para este caso  $k = 30$  correspondiente a los días del mes.
- $s$ : Estado individual.
- $S$ : conjunto de estados
- $a$ : acción individual
- $A$ : Conjunto de acciones.
- $V(s)$ : Función de valor, resultante al final de cada iteración. Al obtener  $V(s)$  por cada  $k$ , se obtiene el máximo valor por día que entrega un estado determinado.
- $Q(s, a)$ : Función de calidad (*quality*) que determina recompensa dado una acción y estado determinado.

#### 4.5. DATOS REALES

Se tienen los datos de gestión de cobranza de una empresa X, correspondientes a los meses de marzo y abril del año 2019. Una vez procesado los datos se obtiene una cantidad total de 11.597 cuentas con 113.603 registros con información del historial de acciones y además con información para identificar la cuenta de usuario, la última acción que se ejecutó en el registro e información demográfica complementada con antecedentes crediticios.

El estado se representa como un vector denominado “*STATE*” que resume toda la información de gestiones que se han realizado a un con lo que ha ocurrido hasta el momento y que se necesita para determinar cuál es la decisión que se debe tomar para cambiar de estado. El vector de estado contiene información de historial con la secuencia de acciones ejecutadas hasta cierto instante de tiempo en el mes. Las características del son las siguientes:

- $\delta_t$  : Cantidad de días que han transcurrido en el proceso de cobranza, hasta que el deudor registre un pago.
- $\delta_{ai}$  : Cantidad de días que han pasado desde que se ejecutó la acción  $i$ , donde:
  - $a_1$ : SMS
  - $a_2$ : CALL
  - $a_3$ : NO\_ ACCION
- $n_{ai}$  : Cantidad de acciones que  $a_i$  que han sido ejecutadas en un usuario hasta un determinado día del mes.
- pago: Variable binaria que indica el pago o no pago del deudor.

Estos vectores son agrupados y reordenados en un nuevo dataset con orden cronológico en donde se aprecia el historial de acciones gestionadas a un deudor.

**Tabla 1. Ejemplo dataset con historial de acciones.**

| cta | $\delta_t$ | $\delta_{a_1}$ | $\delta_{a_2}$ | $\delta_{a_3}$ | $n_{a_1}$ | $n_{a_2}$ | $n_{a_3}$ | pago |
|-----|------------|----------------|----------------|----------------|-----------|-----------|-----------|------|
| a   | 0          | 0              | 0              | 0              | 0         | 0         | 0         | 0    |
| a   | 1          | 0              | 1              | 1              | 1         | 0         | 0         | 0    |
| a   | 2          | 0              | 2              | 2              | 2         | 0         | 0         | 0    |
| a   | 3          | 0              | 3              | 3              | 3         | 0         | 0         | 0    |
| a   | 4          | 1              | 0              | 4              | 3         | 1         | 0         | 0    |
| a   | 5          | 2              | 1              | 0              | 3         | 1         | 1         | 0    |
| a   | 6          | 3              | 2              | 0              | 3         | 1         | 2         | 0    |
| a   | 7          | 0              | 2              | 1              | 4         | 1         | 2         | 1    |
| b   | 0          | 0              | 0              | 0              | 0         | 0         | 0         | 0    |
| b   | 1          | 1              | 0              | 1              | 0         | 1         | 0         | 0    |
| b   | 2          | 2              | 0              | 2              | 0         | 2         | 0         | 1    |
| c   | 0          | 0              | 0              | 0              | 0         | 0         | 0         | 0    |
| c   | 1          | 0              | 1              | 1              | 1         | 0         | 0         | 0    |

Dentro de los datos disponibles se tiene también información adicional respecto a perfiles de usuario que pueden influir en los experimentos que se pretende realizar en las siguientes secciones. Se agrupa entonces la información dentro de un dataset con información de cada usuario para posteriormente extraer un vector de perfil de usuario denominado “*USER*” con la siguiente información:

- Sexo: Genero del cliente, donde 0 indica si es mujer o 1 en caso de ser hombre.
- Edad: dentro del rango de valores entre 19 y 86 años.
- Clasificación de riesgo: Indicador que se entrega según el grado de deuda
- Cupo total: Crédito total otorgado al cliente con valores entre \$0 y \$5M.
- Cupo disponible: Crédito con valores de \$-2M y \$2M. El cupo negativo indica un sobregiro en la cuenta.
- Cupo disp. relativo: Ratio entre cupo disponible y total con valores entre 0 y 1.
- Sobre deuda: 0 en caso de poseer un cupo disponible mayor a 0 y 1 en caso contrario

**Tabla 2. Ejemplo dataset con perfiles de usuario.**

| cta | sexo | edad | clasif.riesgo | cupo    | disp.   | relativo | sobre deuda |
|-----|------|------|---------------|---------|---------|----------|-------------|
| A   | 0    | 20   | 0             | 100.000 | 80.000  | 0.8      | 0           |
| B   | 0    | 30   | 11            | 50.000  | 30.000  | 0.6      | 0           |
| C   | 0    | 19   | 2             | 200.000 | -10.000 | 0        | 1           |
| D   | 1    | 25   | 3             | 500.000 | -20.000 | 0        | 1           |
| E   | 1    | 54   | 7             | 300.000 | 45.000  | 0.15     | 0           |
| F   | 0    | 72   | 8             | 500.000 | -50.000 | 0        | 1           |
| G   | 1    | 28   | 1             | 200.000 | 50.000  | 0.25     | 0           |

Cabe destacar que estos datos no poseen una estrategia optima ya que son datos de 2 meses en donde no se aplicó ninguna estrategia, por lo que estos datos serán utilizados solamente para experimentar con los modelos predictivos de clasificación y observar rendimientos. La sección de simulación (sección 5) se realizará un experimento de cómo debería funcionar de manera completa (resolución MDP).

#### 4.6. ENTRENAMIENTO Y TESTEO

El dataset se opta por dividirlo por cuentas en una proporción de 80:20. Al dividirlo por cuentas se asegura que la división no divida los historiales y que cada set (test y training) tengan los historiales completos de un deudor. En el siguiente esquema se representa la idea, en donde los registros que comparten color pertenecen al historial de un solo deudor.

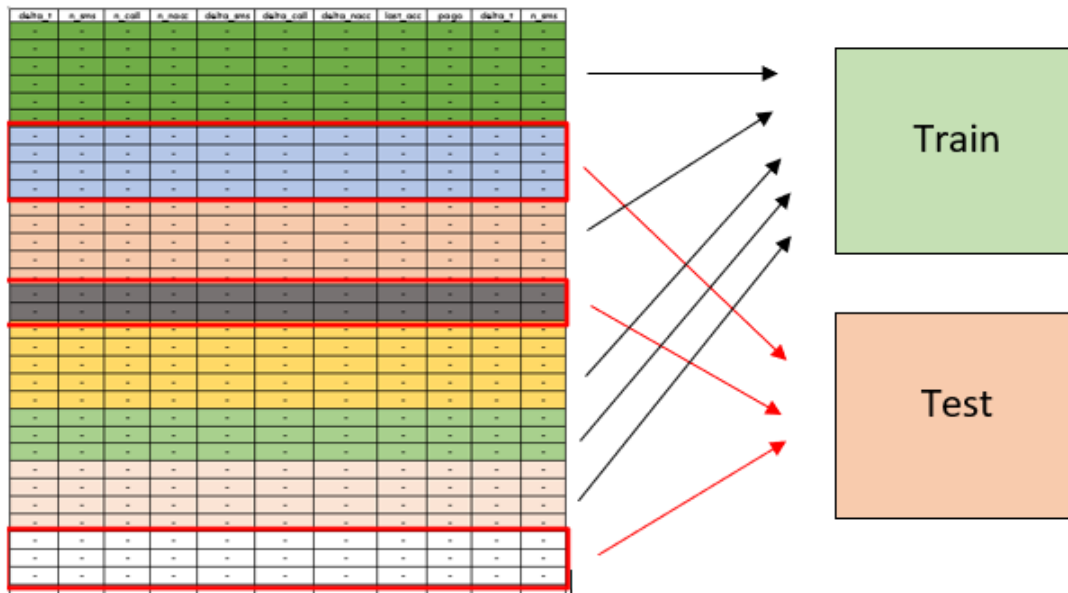


**Figura 14. Esquema entrenamiento y Testeo.**

Finalmente, los registros quedan distribuidos de la siguiente forma:

- Entrenamiento: 89.868 registros
- Testing: 23.735 registros

#### 4.7. MODELOS CLASIFICADORES

En esta sección se explican los modelos utilizados para determinar las probabilidades de transición entre estados, como fueron entrenados y finalmente los resultados obtenidos de cada modelo. Para estos experimentos se utiliza el vector *STATE* sin agregar perfiles de usuarios y posteriormente se incluye la información del perfil de usuario del vector *USER* para comprobar si existe alguna mejora respecto a las métricas de desempeño.

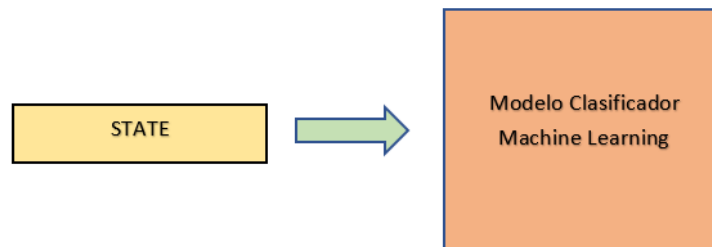


Figura 15. Esquema sin perfil de usuario.

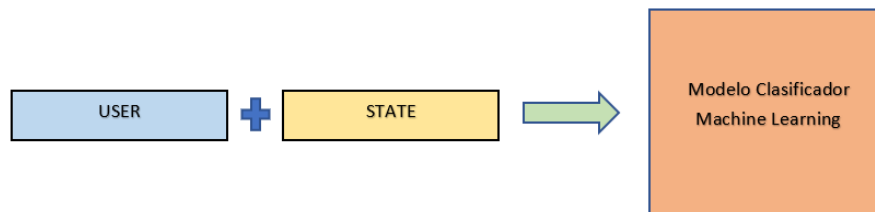


Figura 16. Esquema con perfil de usuario.

#### 4.8. EXPERIMENTOS CON DATOS REALES

En esta sección se realizarán experimento para evaluar el rendimiento de los modelos escogidos como candidatos a estimador de transición de estados. Se busca encontrar que modelos y que combinación de técnicas de muestreo, tienen mejor rendimiento y dependiendo que tan buenas sean sus métricas y que modelos serán escogidos como el mejor estimador de función de transiciones. En la figura 17 se presenta como métrica de comparación el ROC AUC de todos los modelos entrenados con el dataset original siendo XGB+USER el más alto y SVM en sus 2 versiones, tiene un desempeño muy pequeño bajo este criterio.

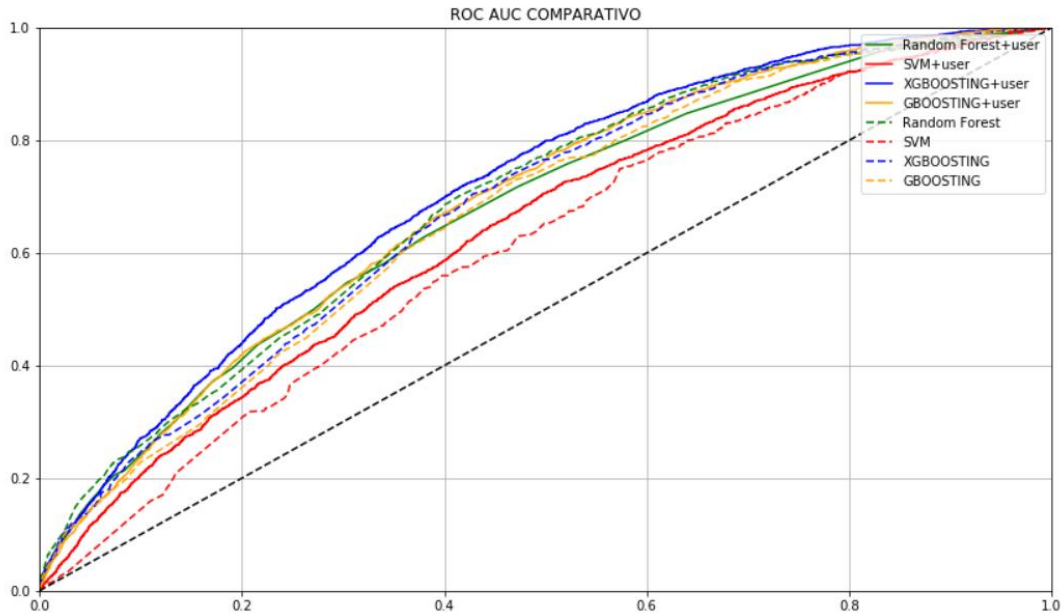


Figura 17. ROC AUC modelos con dataset original.

Sin embargo, se tiene la presencia de un dataset desbalanceado debido al diseño del modelo que se ha propuesto como se puede observar en las figuras 18, donde se destaca en color rojo el único registro que presenta un pago de la deuda y en la figura 19, donde se puede apreciar la relación entre registros con pago y no pago es muy desbalanceado siendo los registros con no pago la clase que predomina. Esto afecta especialmente la métrica de *accuracy* ya que va a entregar un buen resultado, pero es falso ya que en realidad el modelo está haciendo una buena predicción de la clase que predomina, siendo que, para efectos de este trabajo, interesa que el modelo sea capaz de predecir bien la clase minoritaria, es decir los registros con pago.

| cta | id_state | delta_t | n_sms | n_call | n_no_acc | d_sms | d_call | d_no_acc | last_acc | pago |
|-----|----------|---------|-------|--------|----------|-------|--------|----------|----------|------|
| 4   | 0        | 0       | 0.0   | 0.0    | 1.0      | 0     | 0      | 0        | SMS      | 0.0  |
| 4   | 4        | 1       | 0.0   | 0.0    | 2.0      | 1     | 1      | 0        | NO_ACC   | 0.0  |
| 4   | 12       | 2       | 0.0   | 0.0    | 3.0      | 2     | 2      | 0        | NO_ACC   | 0.0  |
| 4   | 22       | 3       | 0.0   | 0.0    | 4.0      | 3     | 3      | 0        | NO_ACC   | 1.0  |
| 4   | 0        | 0       | 0.0   | 0.0    | 1.0      | 0     | 0      | 0        | SMS      | 0.0  |
| 4   | 4        | 1       | 0.0   | 0.0    | 2.0      | 1     | 1      | 0        | NO_ACC   | 0.0  |
| 4   | 12       | 2       | 0.0   | 0.0    | 3.0      | 2     | 2      | 0        | NO_ACC   | 0.0  |
| 4   | 21       | 3       | 0.0   | 0.0    | 4.0      | 3     | 3      | 0        | NO_ACC   | 0.0  |
| 4   | 33       | 4       | 0.0   | 0.0    | 5.0      | 4     | 4      | 0        | NO_ACC   | 0.0  |
| 4   | 47       | 5       | 0.0   | 1.0    | 5.0      | 5     | 0      | 1        | CALL     | 0.0  |
| 4   | 64       | 6       | 0.0   | 1.0    | 6.0      | 6     | 1      | 0        | NO_ACC   | 0.0  |
| 4   | 82       | 7       | 0.0   | 1.0    | 7.0      | 7     | 2      | 0        | NO_ACC   | 0.0  |
| 4   | 111      | 8       | 0.0   | 2.0    | 7.0      | 8     | 0      | 1        | CALL     | 0.0  |
| 4   | 137      | 9       | 0.0   | 2.0    | 8.0      | 9     | 1      | 0        | NO_ACC   | 0.0  |
| 4   | 175      | 10      | 0.0   | 3.0    | 8.0      | 10    | 0      | 1        | CALL     | 0.0  |
| 4   | 212      | 11      | 0.0   | 3.0    | 9.0      | 11    | 1      | 0        | NO_ACC   | 1.0  |
| 4   | 0        | 0       | 0.0   | 0.0    | 1.0      | 0     | 0      | 0        | SMS      | 0.0  |
| 4   | 4        | 1       | 0.0   | 0.0    | 2.0      | 1     | 1      | 0        | NO_ACC   | 0.0  |
| 4   | 12       | 2       | 0.0   | 0.0    | 3.0      | 2     | 2      | 0        | NO_ACC   | 0.0  |
| 4   | 22       | 3       | 0.0   | 0.0    | 4.0      | 3     | 3      | 0        | NO_ACC   | 1.0  |

Figura 18. Explicación desbalance en dataset.

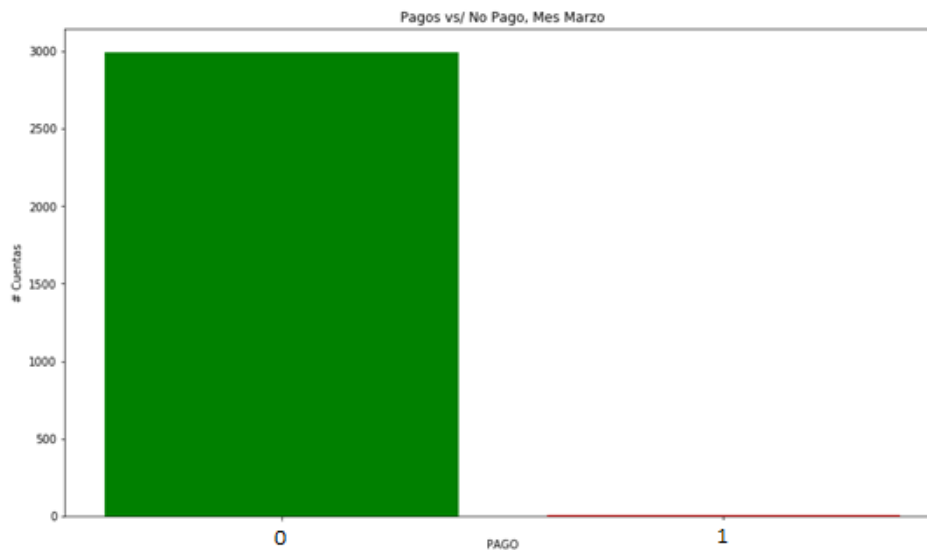


Figura 19. Relación pago v/s no pago de un mes dataset real.

Esto no quiere decir que dentro del mes hay una tasa de pago baja, ya que por día la tasa de pagos es baja, pero en el total del mes se tiene una tasa más alta de pago como se puede observar en la figura 20.

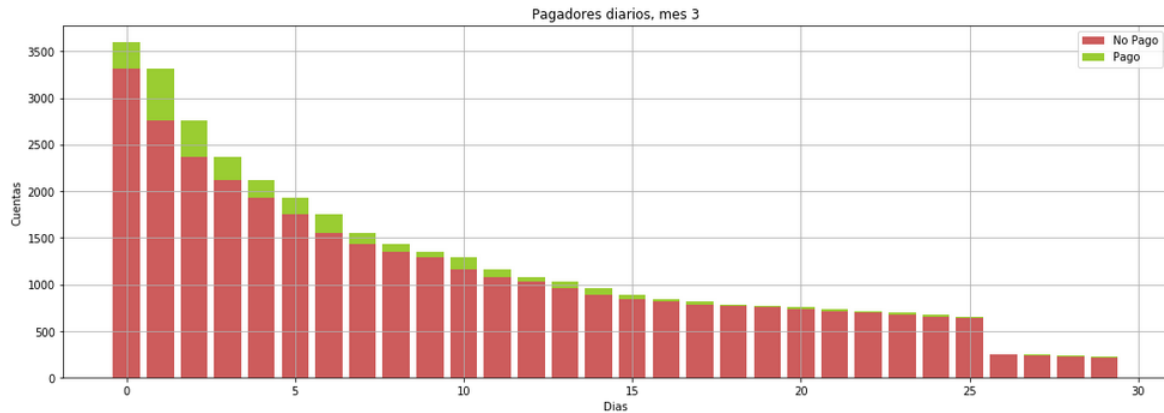


**Figura 20. Pagadores diarios de un mes de dataset real.**

Debido a este desbalance, el ROC AUC es una métrica insuficiente y no es muy claro de interpretar. Para este caso es mejor usar las métricas de precisión, recall y F1-score como se presenta en la figura 21 que indica que los modelos SVM en sus 2 versiones entregan un buen rendimiento respecto a los otros modelos.

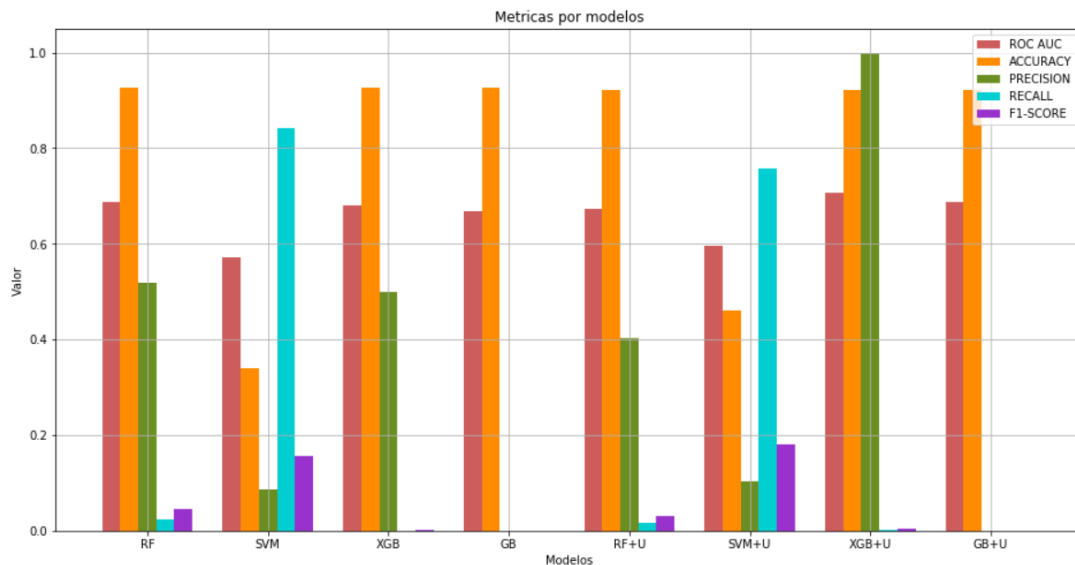


**Figura 21. Métricas dataset original.**

Al realizar los mismos experimentos con la técnica de submuestreo entregan un resultado muy distinto al experimento anterior. Ya que los modelos RF, GB y XGB tienen mejor F1-score que SVM en sus versiones sin vector USER y con vector USER siendo XGB y RF los métodos que lideran en este experimento, como se puede observar en las figuras 22 y 23 respectivamente.

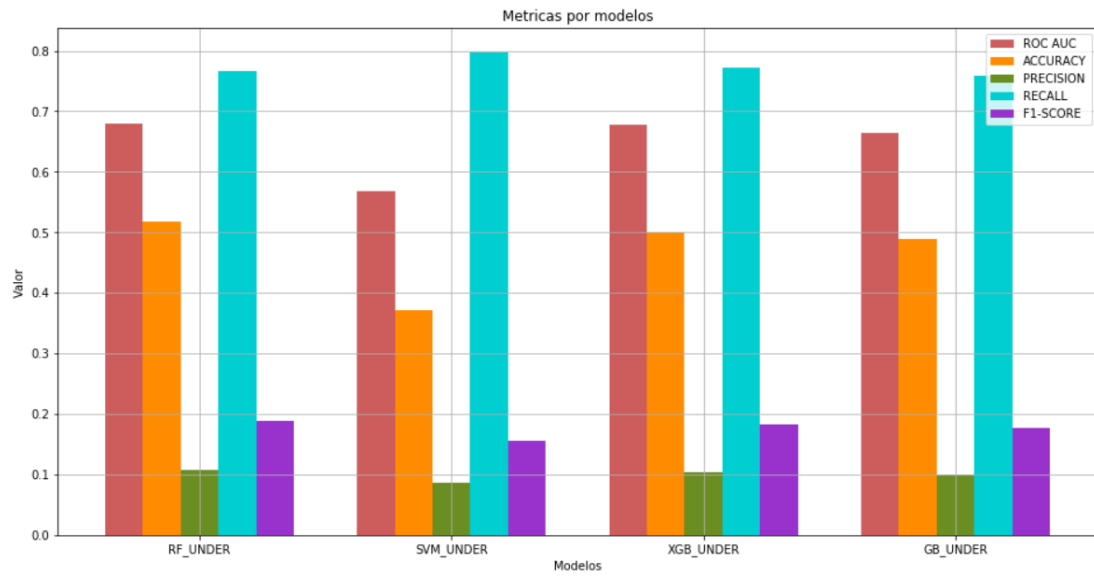


Figura 22. Métricas undersampling NO USER

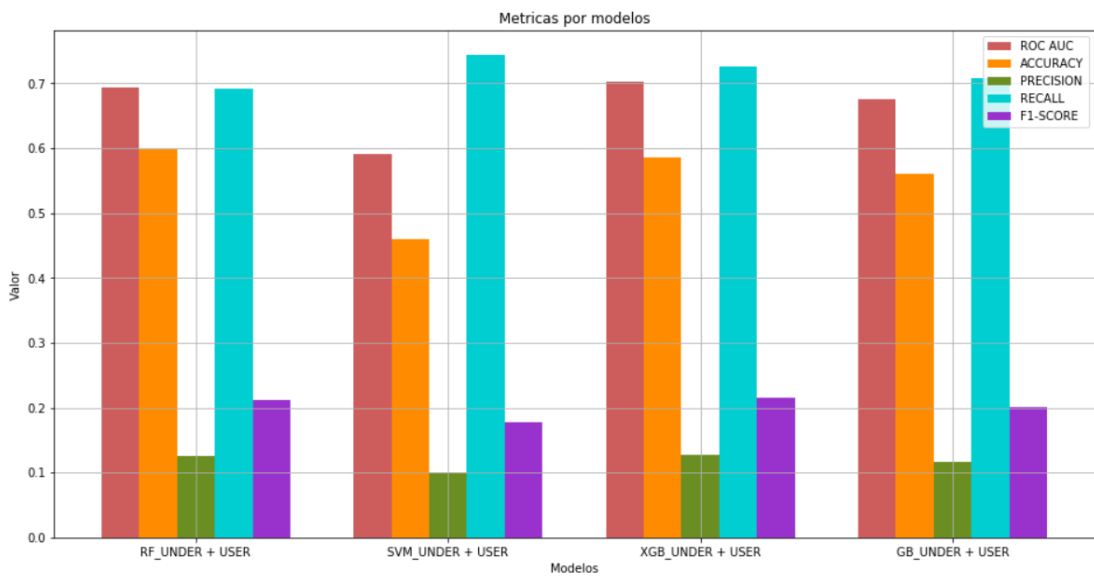


Figura 23. Métricas undersampling + USER

Finalmente, al realizar el experimento con sobremuestreo también se tienen mejores resultados que el primer experimento sin técnicas de muestreo y nuevamente RF y XGB lidera como el mejor modelo como se puede observar en las figuras 24 y 25 en las versiones sin perfil de usuario y con perfil de usuario respectivamente.

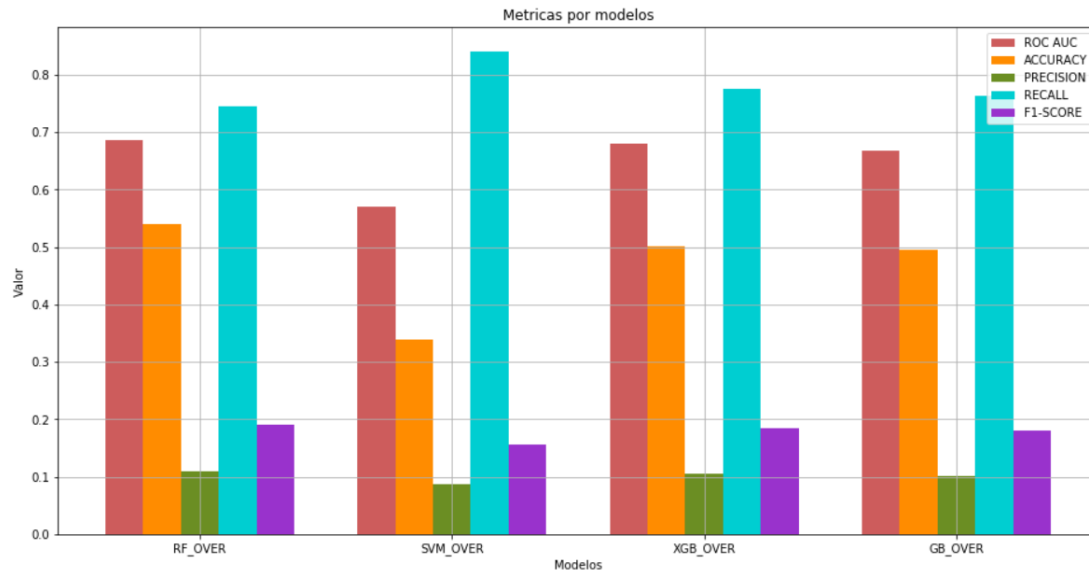


**Figura 24. Métricas oversampling NO USER**

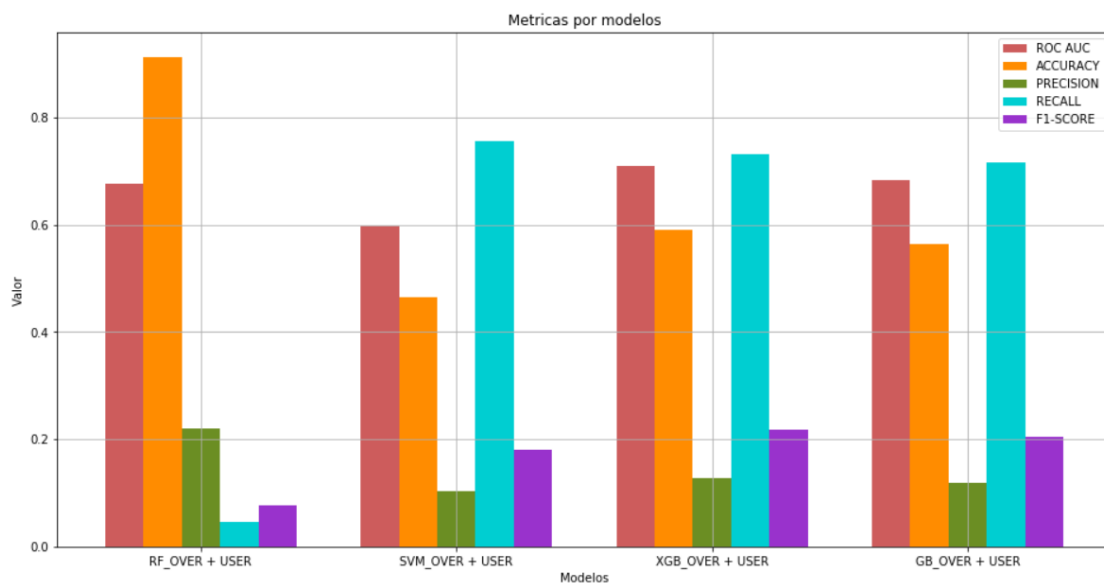


**Figura 25. Métricas oversampling + USER**

Finalmente, para resumir todos los experimentos, las métricas son agrupadas en la tabla 3 en donde se puede tener el detalle numérico de los resultados. Una tendencia que se observa claramente es que los resultados con técnicas de muestreo entregan resultados mucho mejores que la versión sin estas. Las técnicas de muestreo ayudan a los modelos RF, GB y XGB a entregar una mejor respuesta por lo que SVM es un mejor algoritmo

para trabajar los datos sin técnicas de muestreo. Otro detalle que se puede notar es que las versiones con perfil de usuario tienen un mejor rendimiento que la versión sin perfil y esto indica que a mayor cantidad de datos se disponga, se puede tener un modelo estimador mucho más preciso. A pesar de que XGB con sobremuestreo es el modelo que lidera en cuanto a la métrica de F1-score, sería importante tener un conjunto de modelos para ser utilizados como estimador de transición de estados, ya que dependiendo de la cantidad y calidad de datos con los que se entrenan los modelos, estos valores pueden cambiar, incluso el grado de desbalance de datos puede ser decisivo para evaluar el rendimiento de estos modelos.

**Tabla 3. Resultado de métricas dataset real.**

|              | <i>Original</i> |             |            |           | <i>Undersampling</i> |             |            |           | <i>Oversampling</i> |             |            |           |
|--------------|-----------------|-------------|------------|-----------|----------------------|-------------|------------|-----------|---------------------|-------------|------------|-----------|
| <i>Model</i> | <i>ROCAUC</i>   | <i>PREC</i> | <i>REC</i> | <i>F1</i> | <i>ROCAUC</i>        | <i>PREC</i> | <i>REC</i> | <i>F1</i> | <i>ROCAUC</i>       | <i>PREC</i> | <i>REC</i> | <i>F1</i> |
| RF           | 0,689           | 0,519       | 0,024      | 0,046     | 0,680                | 0,107       | 0,767      | 0,188     | 0,685               | 0,109       | 0,745      | 0,191     |
| SVM          | 0,571           | 0,086       | 0,841      | 0,156     | 0,568                | 0,086       | 0,798      | 0,156     | 0,571               | 0,086       | 0,841      | 0,156     |
| GB           | 0,669           | 0,000       | 0,000      | 0,000     | 0,664                | 0,101       | 0,759      | 0,178     | 0,668               | 0,102       | 0,763      | 0,180     |
| XGB          | 0,680           | 0,500       | 0,001      | 0,001     | 0,678                | 0,104       | 0,773      | 0,183     | 0,681               | 0,105       | 0,776      | 0,185     |
| RF+USER      | 0,674           | 0,403       | 0,016      | 0,031     | 0,693                | 0,126       | 0,692      | 0,213     | 0,677               | 0,219       | 0,046      | 0,076     |
| SVM+USER     | 0,596           | 0,102       | 0,758      | 0,180     | 0,590                | 0,101       | 0,744      | 0,178     | 0,597               | 0,103       | 0,755      | 0,181     |
| GB+USER      | 0,687           | 0,000       | 0,000      | 0,000     | 0,675                | 0,118       | 0,708      | 0,202     | 0,683               | 0,119       | 0,716      | 0,205     |
| XGB+USER     | 0,707           | 1,000       | 0,002      | 0,004     | 0,702                | 0,127       | 0,726      | 0,216     | 0,709               | 0,128       | 0,730      | 0,218     |

## 5. VALIDACIÓN DE LA SOLUCIÓN

### 5.1. SIMULACIÓN

Tal como se explica al inicio de la propuesta, se crea un simulador que sea capaz de construir datos de comportamiento similares a los datos reales, con la intención de compensar la escasez de datos reales y así poder realizar pruebas off-line de la eficiencia del algoritmo que se encargará de realizar recomendación de acciones. Esto además se hace para poner a prueba la metodología que se usa para entrenar los modelos y ver que tan efectivo es usar las métricas de precisión, recall y F1 score como criterio de selección para el estimador de transiciones. Por último, esto se usa como medida de control previa para entender cómo funciona esta metodología y no afectar directamente los datos reales ya que una simulación puede distar mucho de lo que ocurre en la realidad. En esta sección se realiza un trabajo por etapas representado por la estructura presente en la figura 26.



Figura 26. Estructura de simulación.

#### 5.1.1. DISTRIBUCIÓN DE PROBABILIDAD DE PAGO



En esta etapa se tiene como objetivo, generar una distribución probabilística de pago a la cual se le entrega parámetros de entrada para modelar la cartera de usuarios. Estos parámetros son la cantidad de usuarios del mes, fracciones de usuarios respecto al total que van a pagar dentro del mes dado una acción y finalmente parámetros de forma dependiendo de la distribución de probabilidad que se escoja. Dentro del diseño del simulador, se decide usar las distribuciones Weibull y Beta ya que son distribuciones con comportamiento similar respecto al comportamiento de pago de los datos reales. Hay que comprender que no existe una probabilidad que modele de manera exacta el comportamiento real, aunque para una simulación de validación es suficiente. Estas distribuciones solo entregan la probabilidad de pago dada una acción determinada, ignorando las características contenidas en el perfil de usuario.

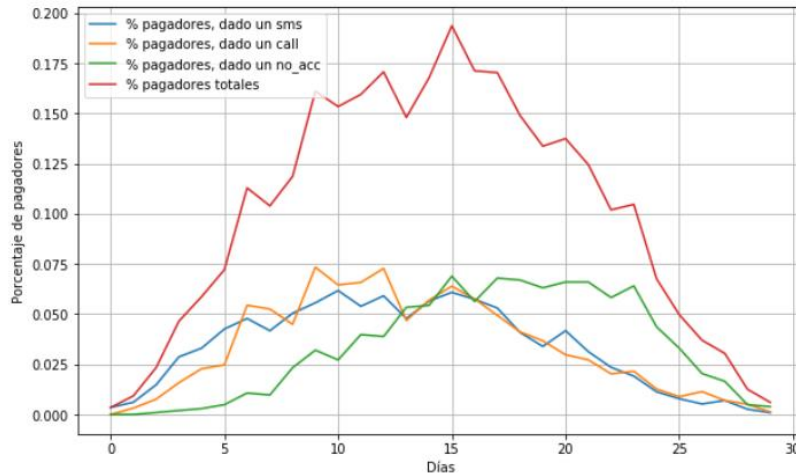


Figura 27. Porcentaje de pago, según distribuciones probabilísticas Weibull y Beta.

### 5.1.2. CREACIÓN DE DATASET FICTICIOS



Una vez creadas estas distribuciones, se entrega como entrada a un módulo creador de Dataset. En esta sección se crean “estrategia” que son secuencia de acciones que son ejecutadas a los deudores en cada día. Estas estrategias son asignadas según el perfil de usuario de la cuenta, dentro de los cuales, para efectos del diseño del simulador, se implementan 3 tipos de estrategias que son:

- Estrategias según género y edad.
- Estrategias según deuda y sobregiro.
- Estrategias según cupo y deuda.

Para realizar los experimentos de la simulación en esta sección se decide de manera arbitraria asignar estrategias según sexo y edad. Cabe reiterar que estas estrategias son creadas de forma arbitraria solo para simulación y no tienen ninguna relación con los datos reales, ya que las estrategias requieren un análisis más profundo al disponer de más datos. La intención es que el diseño del simulador sea modularizado para poder implementar y agregar otras estrategias que tengan más relación con los datos reales. Dentro de las estrategias por género y edad se asignan las siguientes secuencias de acciones:

- $\{a_2, a_0, a_1, a_2, a_2, a_1\}$ , si es género femenino y edad menor o igual a 30 años.
- $\{a_1, a_2, a_0, a_1, a_2, a_0\}$ , si es género femenino y tener entre 30 y 65 años.
- $\{a_0, a_1, a_2, a_0, a_1, a_1\}$ , si es género femenino y edad mayor o igual a 65 años.
- $\{a_2, a_2, a_0, a_1, a_0, a_1\}$ , si es género masculino y edad menor o igual a 30 años.
- $\{a_2, a_1, a_0, a_1, a_2, a_0\}$ , si es género masculino y tener entre 30 y 65 años.
- $\{a_1, a_0, a_2, a_0, a_1, a_2\}$ , si es género masculino y edad mayor o igual a 65 años.

De esta cantidad de estrategias, se desprende que existe la misma cantidad de perfiles de usuarios. Estas acciones se ejecutarán por día, por lo que se tiene la secuencia de acciones para 6 días. Una vez pasados los días se repite la secuencia entrando en un ciclo que acaba cuando el usuario pague dependiendo de la distribución de pago. Una vez terminada la asignación de estrategias se obtiene un dataset con el historial de acciones de todos los usuarios más los perfiles.

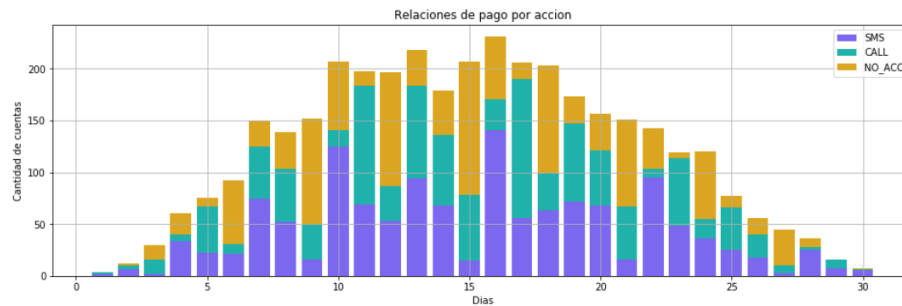


Figura 28. Ejemplo comportamiento de pago de cartera (5000 usuarios).

### 5.1.3. ENTRENAR MODELOS DE CLASIFICACIÓN



En esta etapa, se hace la respectiva división en datasets de entrenamiento y testeo. El dataset de la etapa anterior, es una cartera de 5.000 usuarios con estrategias según edad y sexo con una cantidad de 95.563 registros en total y al hacer la división en razón 80:20, quedan 2 datasets de 76.570 registros con 4.000 cuentas y 18.993 registros con 1.000 cuentas para entrenamiento y testeo respectivamente. Luego se procede a hacer la secuencia como se presenta en la figura 29.

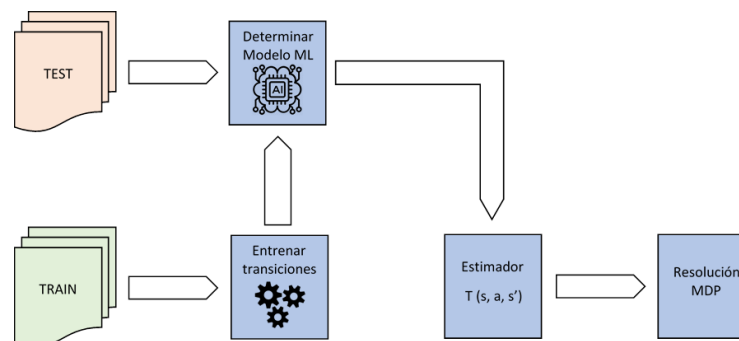


Figura 29. Secuencia de entrenamiento de modelo ML.

Lo primero es entrenar el modelo con las transiciones generando todos los modelos disponibles y en combinación con todas las estrategias de muestreo previamente mencionadas. Posteriormente se evalúa que modelo es mejor usando el dataset de testeo en función de las métricas obtenidas. Dado el resultado anterior con los datos reales, para obtener un mejor resultado se omitirán los modelos sin perfil de usuario

ya que los modelos con perfil ofrecieron mejores resultados haciendo que XGB, GB y RF sean modelos más competitivos en comparación con SVM. Finalmente, el mejor modelo es asignado como estimador de función de transición de estados para la resolución del MDP.

**Tabla 4. Métricas dataset simulado.**

| Modelos    | Undersampling |       |        |        |        | Oversampling |        |        |        |        |
|------------|---------------|-------|--------|--------|--------|--------------|--------|--------|--------|--------|
|            | ROC AUC       | Acc.  | Prec.  | Rec    | F1     | ROC AUC      | Acc.   | Prec.  | Rec    | F1     |
| RF + USER  | 0.664         | 0.545 | 0.0572 | 0.6971 | 0.1058 | 0.6328       | 0.9602 | 0.0769 | 0.0027 | 0.0052 |
| SVM + USER | 0.603         | 0.600 | 0.0574 | 0.6070 | 0.1049 | 0.5992       | 0.6159 | 0.0574 | 0.5811 | 0.1045 |
| XGB + USER | 0.683         | 0.515 | 0.0605 | 0.7967 | 0.1125 | 0.6970       | 0.5409 | 0.0626 | 0.7803 | 0.1159 |
| GB + USER  | 0.686         | 0.532 | 0.0608 | 0.7694 | 0.1128 | 0.6908       | 0.5378 | 0.0615 | 0.7708 | 0.1140 |

Al igual que los datos reales, el modelo que entrega mejores resultados en cuanto a sus métricas de precisión y F1-score, es XGB+USER, aunque que estos resultados pueden variar según el dataset y la cantidad de datos, tal como se menciona en los resultados con los datos reales. De todas formas, se realizarán pruebas con los demás modelos para hacer observaciones de cuanto varían los resultados en función del estimador que se esté utilizando para determinar las probabilidades de transición.

#### 5.1.4. RESOLUCIÓN MDP



Finalmente, en esta etapa con el estimador previamente entrenado, se utiliza el dataset de testing para comprobar si las estrategias inventadas de forma arbitraria son las que aseguran una mayor recompensa o valor esperado. Esto también ayuda a ver lo eficiente el estimador escogido cuando existan casos inexistentes en el dataset de entrenamiento pero que si existen en el dataset de testing.

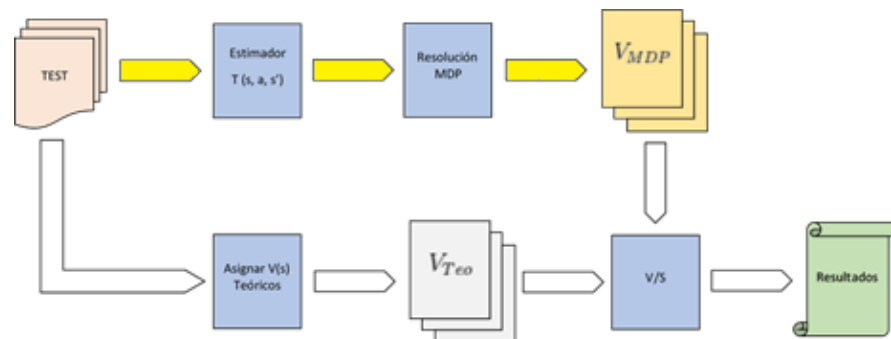


**Figura 30. Estructura etapa 4, sub-etapa 1.**

Lo primero es entregar el dataset al estimador, el cual como salida entrega la matriz de transición que es un conjunto de micro matrices con las probabilidades de ir de un estado a otro dado una determinada acción de todos los días y de todos los perfiles existentes tal como se muestra en la figura 30. Estas matrices son entregadas al

módulo de resolución en donde es ejecutado el algoritmo de iteración por valor y, además, se debe asignar los costos de realizar una acción y la recompensa al obtener un pago. Para este caso se asignan los siguientes valores:

- $C(a_0) = -1$  (costo de “hacer nada”)
- $C(a_1) = -2$  (costo de “SMS”)
- $C(a_2) = -3$  (costo de “CALL”)
- $R(pago, a_i) = 40$  (Recompensa de tener un “PAGO” dada cualquier acción)

Como resultado se entrega los valores óptimos esperados de cada estado almacenados en una estructura tipo cubo con dimensiones  $S \times U \times T$  en donde:

- $S$ : Cantidad de estados.
- $U$ : Cantidad de perfiles.
- $T$ : Cantidad de días.

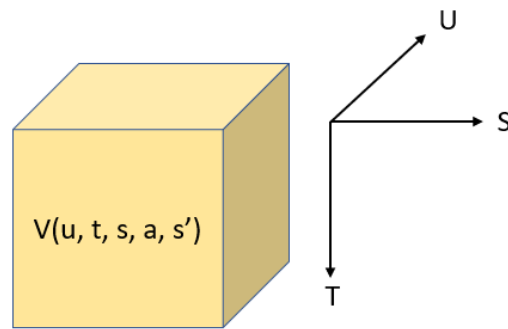


Figura 31. Estructura de almacenamiento de resultado.

No basta solo tener los valores esperados, también se necesita saber lo que hay detrás de este valor y esto es la “política” que se interpreta como la “acción” que garantiza la mayor recompensa si el usuario paga su deuda. Por esta razón, en paralelo se guarda en 2 estructuras la política y la probabilidad de pago asociada respectivamente, con la misma dimensionalidad. Esto implica que al escoger un elemento de  $V$ , en la misma coordenada dentro de la estructura  $\pi$  se obtiene la mejor acción o acción optima y en la estructura  $P$ , la probabilidad de pago asociada.

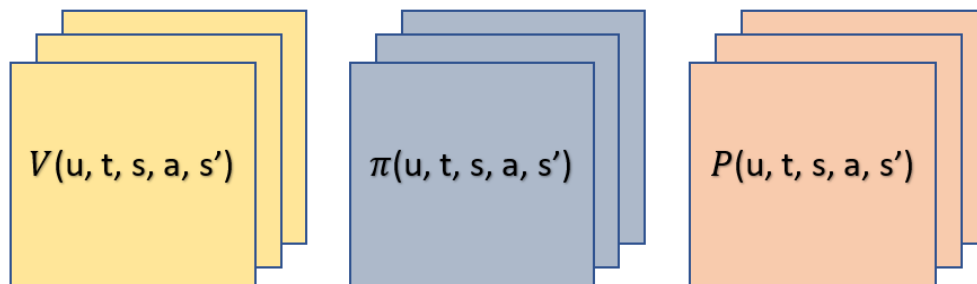


Figura 32. Estructuras de Valor, Política y probabilidad de pago.

Finalmente, estas estructuras se utilizan en una nueva simulación en donde se construye un nuevo dataset con los mismos perfiles del dataset de testing, pero esta vez con las acciones resultantes de la resolución del MDP con el fin de hacer una comparación con el dataset de testeo y observar si las estrategias inventadas son óptimas, en tal caso las políticas resultantes deberían ser las mismas que las del dataset de testeo y en caso contrario se entrega una alternativa mejor que logre maximizar la recompensa.

En segundo lugar, se asignan los mismos valores de costos y recompensas usados en el proceso de resolución de MDP al dataset de Testing con el propósito de tener un parámetro de comparación con el resultado del algoritmo. Aquí se busca calcular el valor esperado de ambos datasets y si en realidad el algoritmo cumple su función de optimizar el valor esperado como se muestra en el esquema de la figura 33.

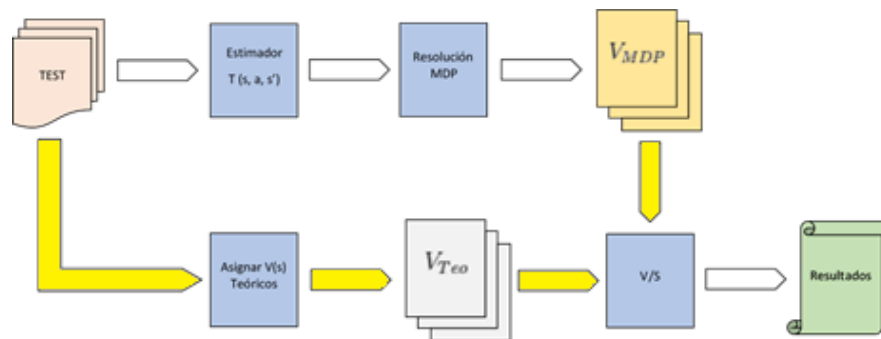


Figura 33. Estructura etapa 4, sub-etapa 2.

### 5.1.5. RESULTADOS

Primero se calcula el valor total esperado de los datasets. La idea es que el valor esperado del MDP sea mayor o igual que el valor esperado del dataset de testeo, si es igual, quiere decir que en el dataset de testeo se aplicó la estrategia óptima y por lo tanto esta optimizado sin siquiera pasar por el algoritmo y el caso contrario indicará que el algoritmo está encontrando una mejor estrategia que entrega una mayor recompensa. El valor esperado del dataset, se calcula mediante la siguiente formula.

$$V_{MDP} = \frac{\sum R_u(t, s, a, s')}{\#U}$$

Figura 34. Formula de valor esperado.

Además, se realiza un experimento para comprobar que la métrica de F1-score, es una métrica adecuada para escoger el estimador de transiciones. Para esto, se utilizan los 8 modelos con sobremuestreo y submuestreo, obteniendo los resultados de las figuras 35 y 36 respectivamente.

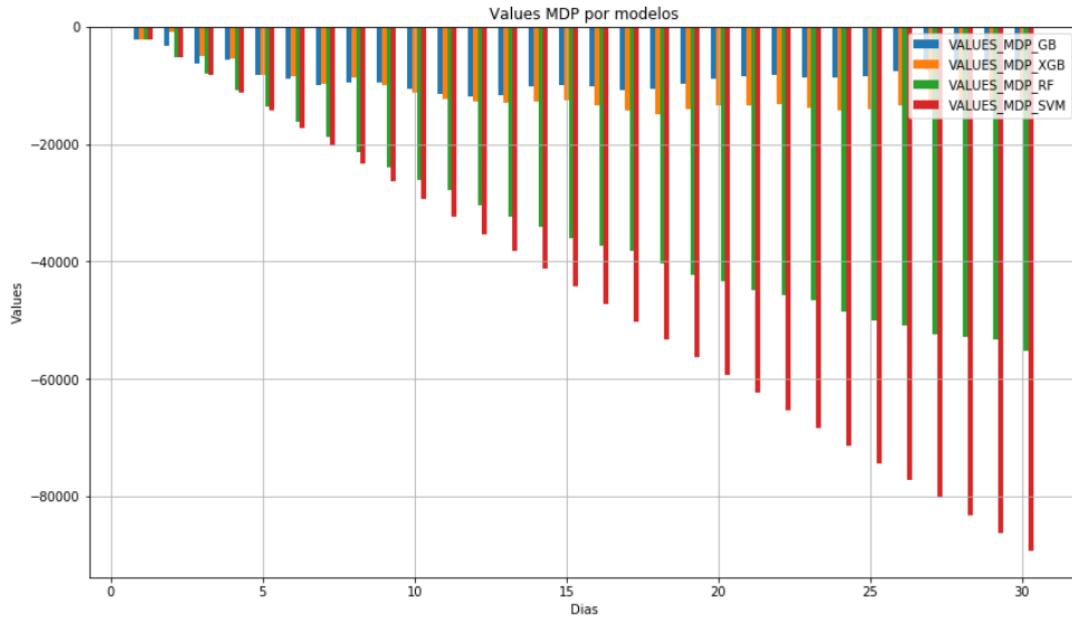


**Figura 35. Valores por día de modelos oversampling.**

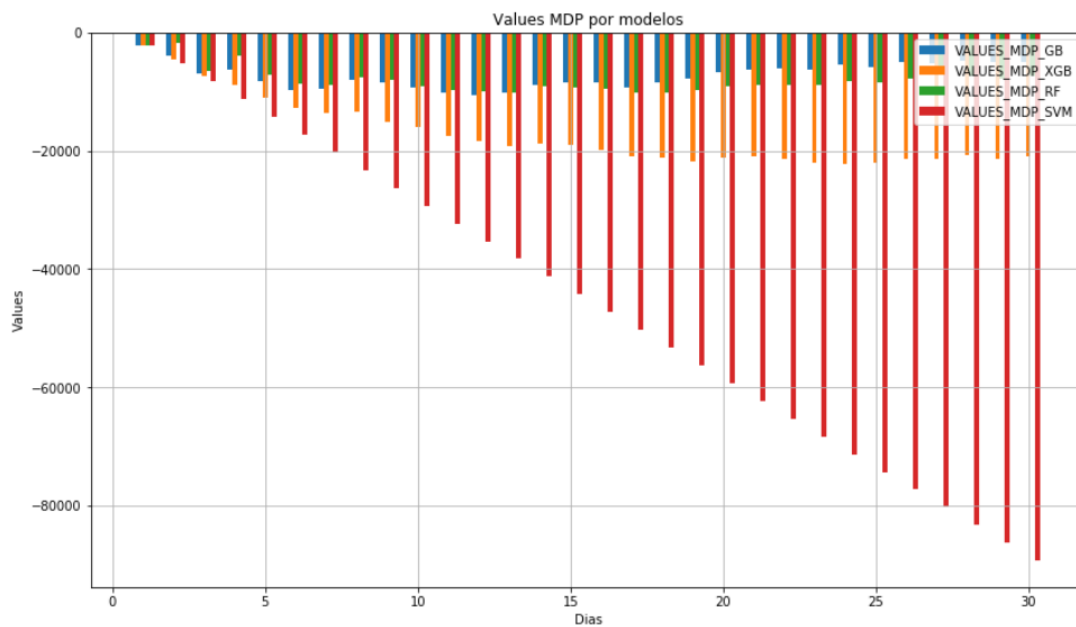


**Figura 36. Valores por día de modelos Undersampling.**

Se puede ver que los modelos SVM y RF entregan pésimos resultados en comparación de GB y XGB. Los valores por día esperados son mucho mayores y tiene directa relación con que estos 2 modelos tienen un mejor F1-score que los modelos SVM y RF (Tabla 4) en ambas versiones de muestreo. Finalmente se toman estos 2 modelos con ambas versiones, para comparar con el dataset de testeo se puede observar que GB entrega mejores resultados en ambas versiones, logrando una maximización de la recompensa mucho mejor que el testeo tal como se puede apreciar en las figuras 37 y 38.

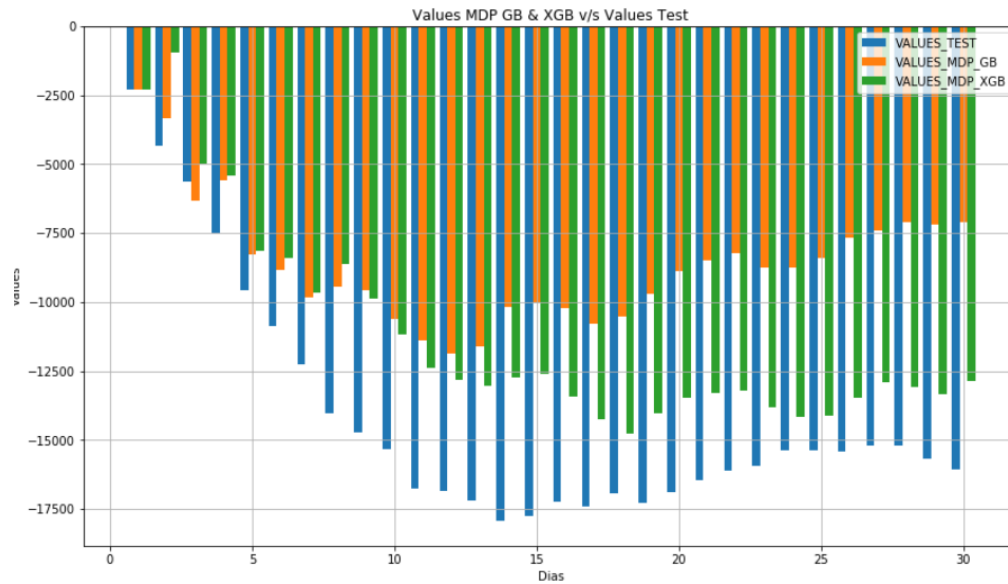


Figura 37. XGB & GB sobremuestreo v/s Test

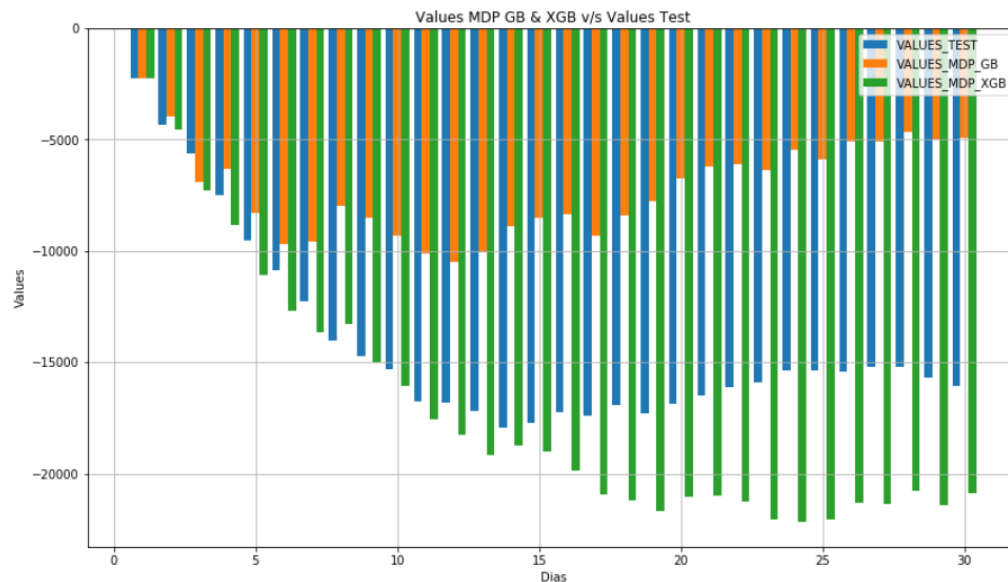


Figura 38. XGB & GB submuestreo v/s Test.

En la tabla 5 se puede observar de manera numérica los resultados finales de todos los modelos y se observa que XGB entrega un mejor rendimiento en cuanto a la métrica F1- score y precisión, sin embargo, al simular nuevamente, GB entrega un mejor resultado en cuanto al valor esperado, lo que implica que al usar un modelo para estimar transiciones, no existe un modelo único que entregue la mejor respuesta pero si hay una forma de acercarse al valor óptimo del dataset de testeo usando la métrica de F1 score como criterio de selección del modelo. Cabe destacar que estos resultados pueden variar dependiendo del dataset, las técnicas de muestreo y el entrenamiento del modelo.

**Tabla 5. Tabla de resultados finales.**

| Modelos | Undersampling |        |        |       |           | Oversampling |        |        |       |           |
|---------|---------------|--------|--------|-------|-----------|--------------|--------|--------|-------|-----------|
|         | Prec.         | Rec    | F1     | %pago | V         | Prec         | Rec    | F1     | %pago | V         |
| TEST    | -             | -      | -      | 0,733 | - 425,545 | -            | -      | -      | 0,733 | - 425,545 |
| RF      | 0,0572        | 0,6971 | 0,1058 | 0,909 | - 240,248 | 0,0769       | 0,0027 | 0,0052 | 0,364 | - 1003,92 |
| SVM     | 0,0574        | 0,6070 | 0,1049 | 0,0   | - 1373,34 | 0,0574       | 0,5811 | 0,1045 | 0,0   | - 1373,34 |
| XGB     | 0,0605        | 0,7967 | 0,1125 | 0,745 | - 516,511 | 0,0626       | 0,7803 | 0,1159 | 0,856 | - 314,467 |
| GB      | 0,0608        | 0,7694 | 0,1128 | 0,94  | - 216,248 | 0,0615       | 0,7708 | 0,1140 | 0,898 | - 287,08  |

Desde el punto de vista de la optimización de estrategias, se puede observar que el algoritmo fue capaz de crear una estrategia optimizada distinta a la estrategia que se inventó arbitrariamente dado el perfil del usuario. En la figura 39, se puede apreciar que la cantidad de pagos registrados por cada modelo con submuestreo en comparación del dataset de testing siguen un patrón de comportamiento de pago similar al original, pero registrando mayores cantidades de pagos, a excepción de XGB que solo registra mejora en los perfiles 1, 3 y 6.

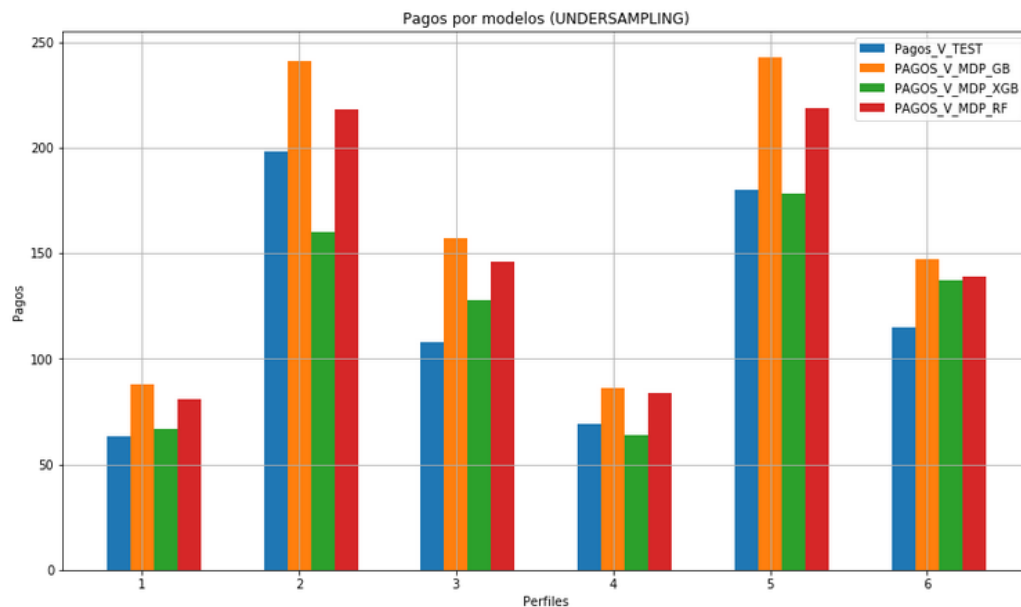


**Figura 39. Cantidad de pagos por perfil con modelos Undersampling.**

Similar al caso anterior, en la figura 40, se puede realizar este mismo análisis con los modelos de sobremuestreo y también se puede observar que el comportamiento de pago sigue un patrón de pago de los modelos que es similar al de testing, pero esta vez el modelo RF entrega un resultado peor que el testing y esta vez XGB entrega un mejor resultado. Aun así, en ambos casos GB entrega un mejor resultado de pago siguiendo un patrón similar al testing. Todo esto se condice con los resultados de la tabla 5.

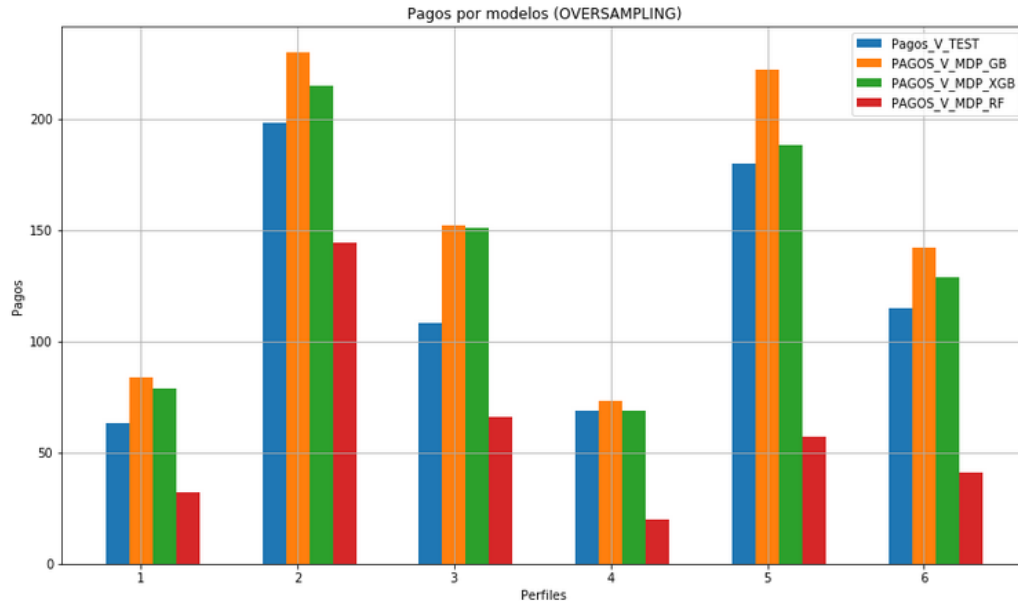


**Figura 40. Cantidad de pagos por perfil con modelos Oversampling.**

Esto implica que, en ambos casos, el algoritmo fue capaz de encontrar una estrategia para cada perfil que garantiza una mejor recompensa, lo que se puede apreciar de mejor forma en las figuras 41, 42 y 43, en donde se hace una comparación de la estrategia arbitraria, respecto a la estrategia optimizada que entrega el algoritmo, en donde la estrategia arbitraria es la que se definió en la sección 5.1.2. En el caso de submuestreo se puede ver que el algoritmo genera una estrategia que es igual usando los tres modelos, lo que implica que el factor que define un mejor resultado es la probabilidad de pago que predice el modelo clasificador. Sin embargo, el algoritmo para este caso logra determinar una estrategia cíclica al igual que la estrategia arbitraria, pero distinta en cuanto a la secuencia.

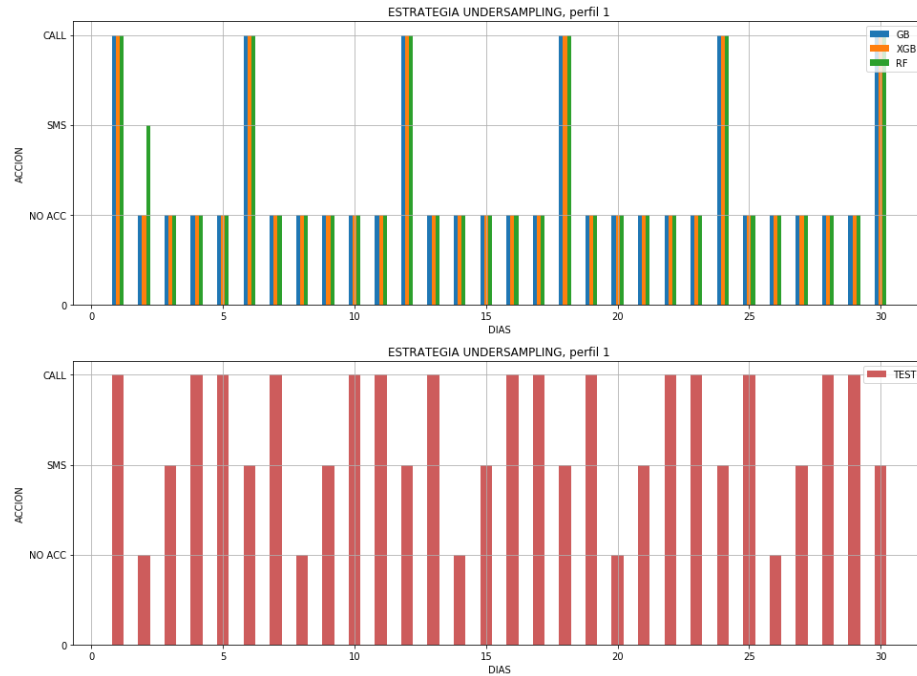


Figura 41. Estrategia perfil 1 vs estrategia MDP undersampling.

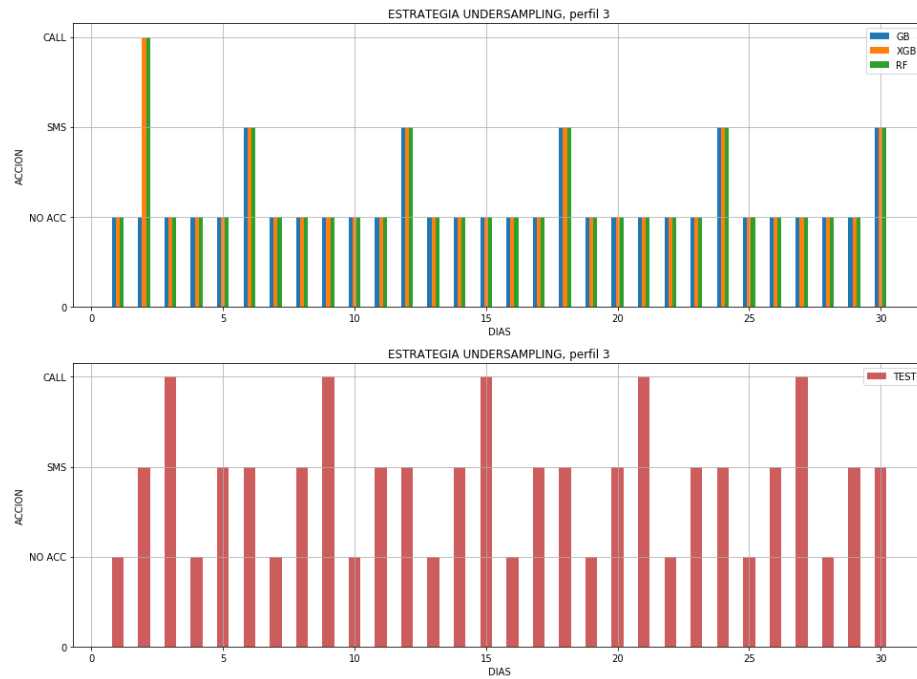


Figura 42. Estrategia perfil 3 vs estrategia MDP undersampling.

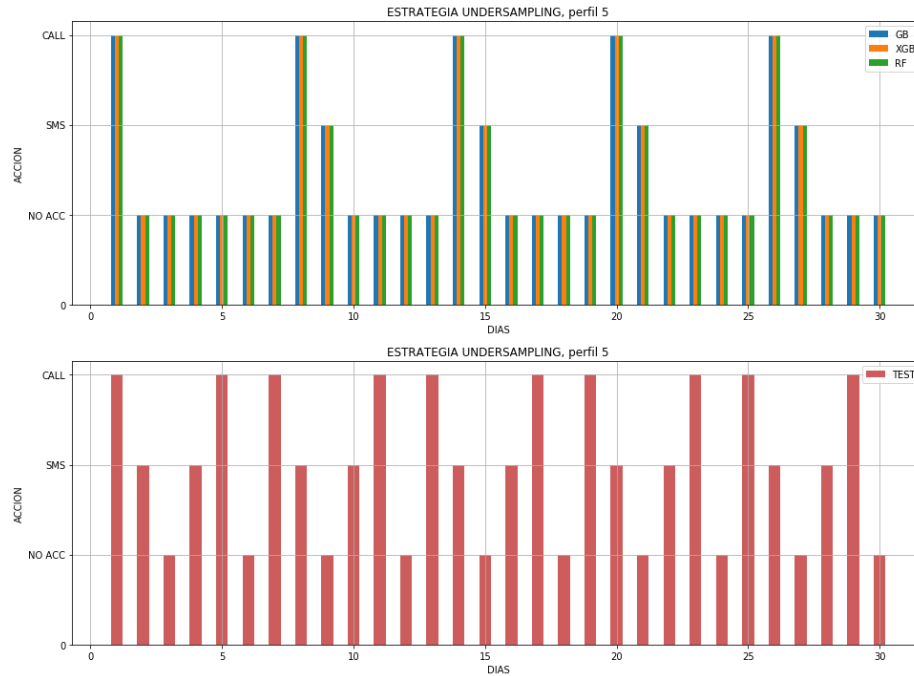


Figura 43. Estrategia perfil 5 vs estrategia MDP undersampling.

Al realizar el mismo análisis anterior, se puede ver que para el caso de sobremuestreo, las estrategias generadas por el algoritmo difieren bastante respecto al caso anterior y los modelos XGB y GB entregan la misma recomendación, por lo que nuevamente al ver la tabla 5, explica los valores tan distintos que se obtienen al usar RF como modelo predictor dentro del algoritmo e incluso la recomendación es distinta de los otros modelos y por consecuencia una mala recomendación.

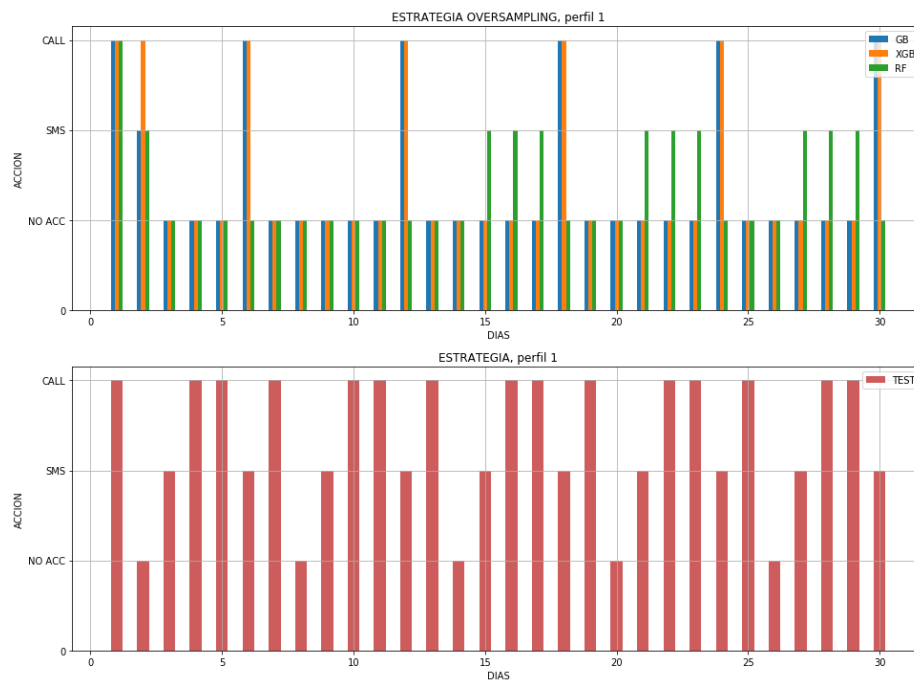


Figura 44. Estrategia perfil 1 vs estrategia MDP oversampling.

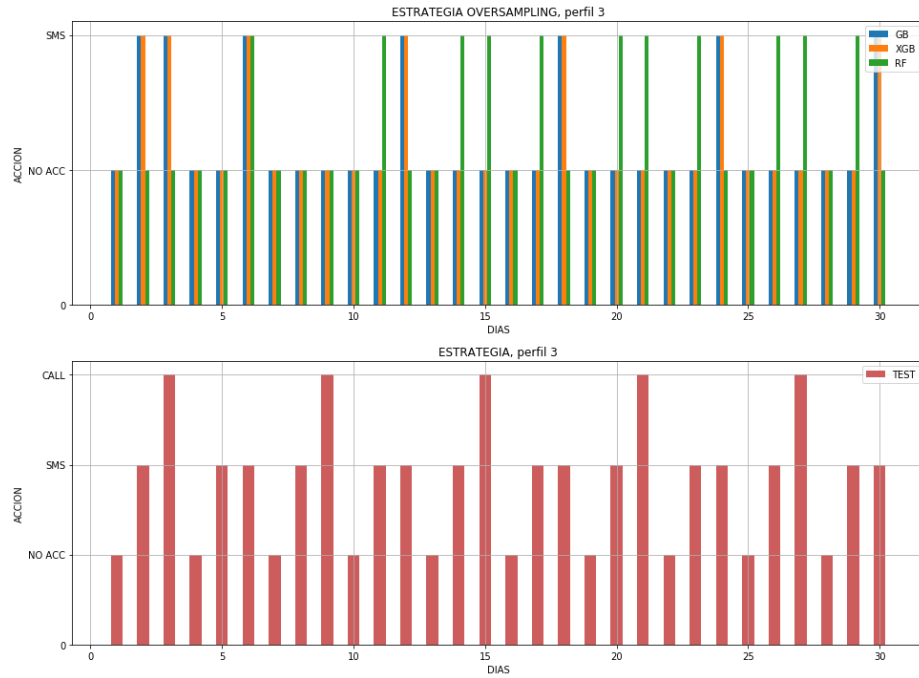


Figura 45. Estrategia perfil 3 vs estrategia MDP oversampling.

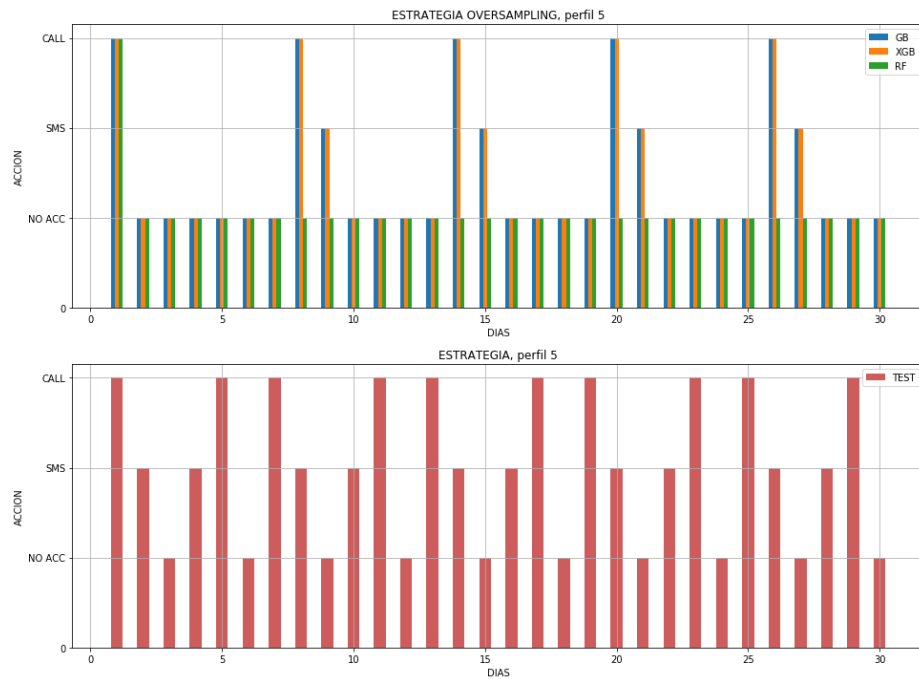


Figura 46. Estrategia perfil 5 vs estrategia MDP oversampling.

De esto se concluye que la simulación en este caso particular, se pudo obtener una estrategia óptima a pesar de la arbitrariedad de la estrategia y este algoritmo nos demuestra que existen otros caminos que pueden entregar un mejor resultado al optimizar el valor de la recompensa que realiza el algoritmo y se demuestra que este es capaz de entregar una recomendación de acción, sin embargo la tendencia del modelo fue recomendar acciones cíclicas, tal como se definió desde un principio en la simulación, logrando una tendencia de pago similar para ambos casos de muestreo pero con recomendaciones distintas. Además, comprueba que dependiendo del tipo de muestreo que se realiza en el entrenamiento, el modelo clasificador influye bastante en el resultado final que entrega el algoritmo.

## 6. CONCLUSIONES

En base a los resultados obtenidos, se puede decir que este problema en particular puede ser modelado como un MDP y las restricciones para convertirlo en un problema de horizonte finito, son características suficientes y realistas bajo este contexto en particular y que es posible entregar una solución adecuada al problema. Este modelo es capaz de entregar una forma de recomendar acciones o estrategias optimas respecto a la función de transición, esto quiere decir que el modelo puede identificar una política que permite un máximo retorno de la recompensa. Se puede concluir también, que el algoritmo de iteración por valor cumple con el objetivo de recomendar una acción gracias a la política que recomienda en cada iteración, sin embargo, la matriz de transición que se encuentra dentro del algoritmo es una pieza fundamental para realizar aquello ya que los resultados obtenidos varían en gran parte por los modelos clasificadores utilizados para crear esta matriz de transición. Respecto a los modelos utilizados, las métricas de precisión, recall y F1-score fueron importantes para evaluar el desempeño de estos modelos, dado que se presenta un desbalance de datos que existe desde un principio en el problema y que es directa consecuencia de cómo se decide modelar los datos. Las técnicas de muestreo fueron cruciales para generar modelos que tuvieran mejor desempeño y obtuvieran mayor poder de predicción en un ambiente de datos desbalanceados, no obstante se debe tener bastante cuidado con ambas técnicas de muestreo cuando se lleve este trabajo a un entorno real que dista mucho de la simulación planteada ya que tal como se menciona en la sección 3.4.1 y 3.4.2 estas técnicas pueden generar pérdida de datos en el caso de submuestreo o generación de ruido para el caso del sobremuestreo. En ambos casos se puede correr el riesgo de tener un modelo subajustado o sobreajustado, si se lleva la situación a un caso extremo por lo que en la situación real hay que tener precaución de aquello. Aun así, los modelos que fueron utilizados entregan un resultado viable para ser aplicado en la vida real, pero se debe tener en cuenta que no existe el modelo perfecto para poder crear la matriz de transición. Tal como se pudo apreciar a lo largo del documento, los datos reales entregan mejores desempeños en el modelo XGB e incluso sin información del usuario, se obtienen mejores resultados con el modelo SVM y en el caso simulado el modelo GB es el que predomina. Esto conlleva a que es una mala idea quedarse con un solo modelo para construir la matriz de transición y en vez de eso es recomendable utilizar un conjunto de modelos, incluso más de los que se proponen en este trabajo. Posteriormente, se debe hacer una simulación previa antes de aplicar recomendaciones con datos reales. Es importante mencionar que existen una combinación de factores que pueden variar mucho el resultado, como lo pueden ser la calidad de los datos, la cantidad disponible de estos, el método de entrenamiento, las técnicas de muestreo, la configuración de los hiperparámetros de los modelos de ML utilizados, entre otros. Hay que destacar que el modelo de datos propuesto sirve para resolver este problema en particular y entrega una solución factible para una etapa inicial dentro del motor cognitivo del stack de cobranza digital. En relación con la matriz de transición, el enfoque que se utilizó de usar micro matrices en vez de una matriz enorme es una solución bastante viable debido a que se busca generar en el producto final una recomendación que genere un retorno en el corto plazo. En consecuencia, no se requiere calcular una matriz con todos los datos de usuarios y estados posibles ya que eso implica estar buscando soluciones de largo plazo, además es algo costoso de implementar y escapa del alcance visto dentro de esta memoria. Cabe mencionar que la intención final de este trabajo es entregar un prototipo de motor que cumpla con la función de recomendar acciones condicionalmente óptimas y poder realizar trabajos futuros sobre la base del prototipo.

Como propuesta final de aplicación de este trabajo se sugiere, para aplicar en la vida real, extraer un pequeño grupo de control dentro de los datos reales en la empresa. La idea de este grupo es que sea una pequeña fracción de los datos reales y que estén presentes los distintos perfiles de usuario o tipos de clientes y también las distintas recomendaciones de acciones que se puedan realizar. Posteriormente usar el algoritmo propuesto previamente entrenado con datos reales e ir comparando si el motor está haciendo mejores recomendaciones que las estrategias actualmente creadas y finalmente retroalimentar el motor con lo que se va realizando en el momento. Si el motor cumple el objetivo de tener mejores retornos, se puede pensar en ir aumentando el pequeño grupo de control de manera gradual hasta tener un sistema autónomo que pueda realizar recomendaciones. Puede ser un solo pequeño grupo de control o varios grupos pequeños para poder probar más de un solo modelo y comparar resultados entre ellos como se grafica en las figuras 47 y 48 respectivamente.

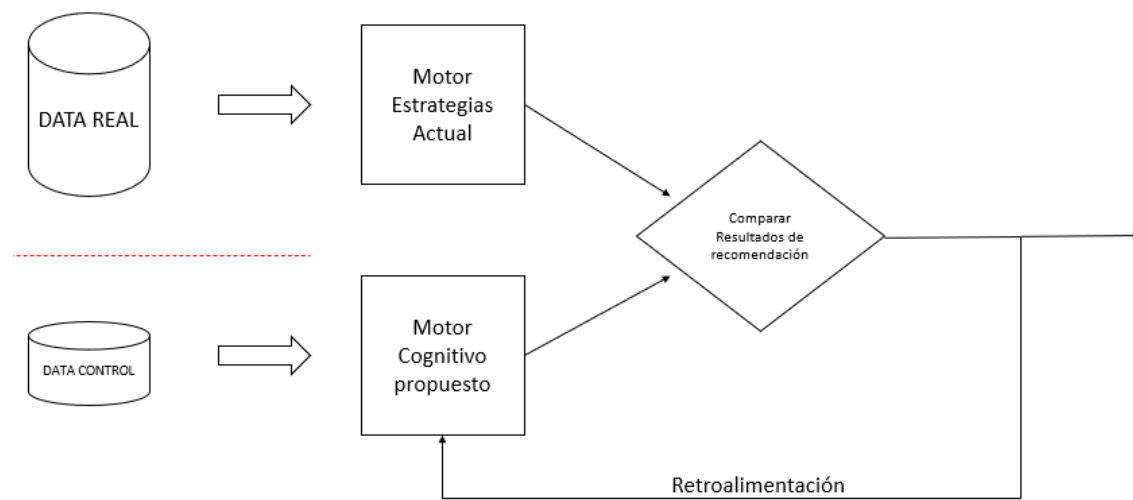


**Figura 47. Propuesta 1 alternativa grupo control.**

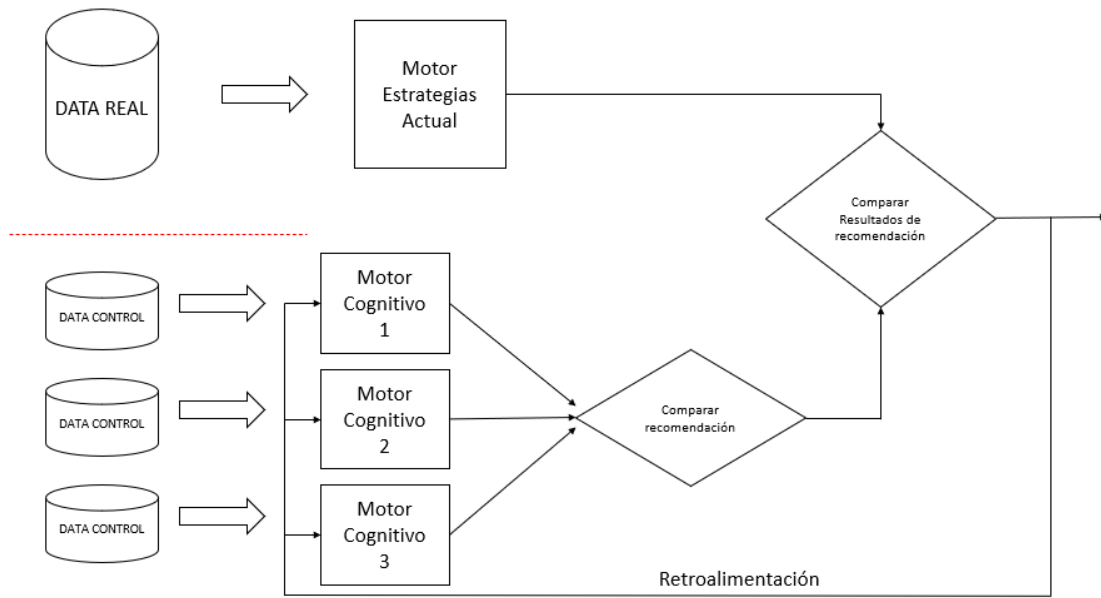


**Figura 48. Propuesta 2 alternativa grupo control.**

La idea es retroalimentar el motor con los resultados obtenidos para mejorar la recomendación y tener como dato de entrenamiento más casos de recomendaciones exitosas. Esto también sirve para tener más casos de posibles estados que puedan aparecer y ver como el motor es capaz de recomendar en base a casos que jamás ha visto. Si el motor recomienda bien, se puede ir aumentando la cantidad de datos con los que va a operar el motor hasta lograr un sistema de recomendación totalmente autónomo.

Dentro de posibles mejoras que se pueden realizar esta la modificación de los vectores de usuario y la información que deben contener. Esto depende principalmente de la disponibilidad de los datos a la hora de implementar el motor, pero sería interesante agregar otros vectores de información, adicionales al que se implementó en la memoria, por ejemplo, un vector de tiempo que contenga información de cuando se debería hacer una acción, si es lo mismo hacer una llamada de día que de noche, mejor respuesta de parte del deudor dependiendo de la hora, enviar un SMS en un fin de semana que en día de semana, entre otros. También puede haber información respecto al mes en curso con información característica del mes y comprobar si esta información influye en el comportamiento, se podría analizar cosas como si el deudor se comporta de igual manera en abril que en diciembre considerando las festividades y el nivel de deudas que aparecen durante este último mes. Otra información que puede ser relevante agregar, sería un vector con información geográfica que también puede influir en el comportamiento de pago de un usuario; tal vez deudores que viven lejos del lugar donde deben pagar, son peores pagadores que alguien que vive cerca. Todas estas preguntas van de la mano con la cantidad de datos que se tengan a disposición y eso depende de la cantidad de datos que una empresa está autorizada legalmente a facilitar para efectos del producto que se pretende vender. También se pueden hacer mejoras respecto a optimización del código. Si bien el enfoque de micro matrices ayuda un poco en esta dirección, al aumentar la cantidad de acciones, estados y perfiles de usuario, se va a requerir mayores cantidades de memoria y capacidad de cómputo para operar con mayor volumen de datos. Se espera que este motor no demore demasiado en realizar una

recomendación cuando se tienen 100.000 usuario y más de 1.000.000 de registros en los datos. Una alternativa de mejora es el uso de matrices en formato comprimido cuando se está en presencia de matrices *sparse* que son matrices con mucha presencia de valores nulos o ceros que no aportan nada y por consecuencia son datos innecesarios que ocupa espacio en la memoria. Por último, para hacer un uso real de esto, es importante que sea un modelo ML de aprendizaje Online en donde el sistema podrá aprender de manera gradual desde un flujo de datos entrante. Esto se debe a que el enfoque entregado en este trabajo es un aprendizaje por lotes o *batch* lo que implica que cada vez que se obtengan nuevos datos para retroalimentar el modelo, hay que entrenarlo nuevamente incluso con los datos antiguos con los que fue previamente entrenado. En consecuencia, el sistema debe detenerse quedando fuera de operación mientras se reentrena con los datos nuevos y antiguos. Si bien esta solución es simple, puede ser un factor crítico debido al tiempo que se pueda demorar en entrenar el modelo y además se requiere una gran capacidad de memoria para almacenar cantidades grandes de datos. Es por esto, que un aprendizaje online puede asegurar una recomendación más rápida al ir entrenando en tiempo real y cuida la cantidad de recursos que se dispone cuando son limitados. Sin embargo, se debe tener precaución con la tasa de aprendizaje ya que, si es muy alta, el modelo tiende a adaptarse a los datos entrantes, pero olvidará información pasada que puede ser relevante. Es importante encontrar un equilibrio en la tasa de aprendizaje que sea adecuado y que permita que el sistema pueda aprender dentro de un marco temporal para tener información más completa a la hora de realizar una recomendación.

Finalmente, queda decir, que este trabajo cumple con los objetivos establecidos al inicio y logra uno de los objetivos más importantes que es generar una recomendación inteligente a través de la solución del MDP diseñado para este problema particular. Si bien son muchas las mejoras que se pueden realizar en este trabajo, como prototipo inicial de motor de recomendaciones cumple los objetivos de realizar recomendaciones a corto plazo.

## REFERENCIAS BIBLIOGRÁFICAS

|           |                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [1]       | M. Leon, «trendTic,» 10 3 2020. [En línea]. Available: <a href="https://www.trendtic.cl/2020/03/datos-digitales-%EF%BB%BFel-nuevo-petroleo-del-siglo-xxi/">https://www.trendtic.cl/2020/03/datos-digitales-%EF%BB%BFel-nuevo-petroleo-del-siglo-xxi/</a> .                                                                                                                                      |
| [2]       | M. Salazar, «Innovacin,» 6 11 ´2019. [En línea]. Available: <a href="http://www.innovacion.cl/2019/11/empresa-utiliza-inteligencia-artificial-para-cobrar-deudas-via-redes-sociales/">http://www.innovacion.cl/2019/11/empresa-utiliza-inteligencia-artificial-para-cobrar-deudas-via-redes-sociales/</a> .                                                                                     |
| [3]       | J. Kagan, 25 2 2020. [En línea]. Available: <a href="https://www.investopedia.com/terms/d/debt-collector.asp">https://www.investopedia.com/terms/d/debt-collector.asp</a> .                                                                                                                                                                                                                     |
| [4]       | M. H. Law, «Mchugheslaw,» 5 11 2015. [En línea]. Available: <a href="https://www.mchugheslaw.com/the-history-of-debt-collection/">https://www.mchugheslaw.com/the-history-of-debt-collection/</a> .                                                                                                                                                                                             |
| [5]       | A. Fontinelle, «Investopedia,» 26 5 2020. [En línea]. Available: <a href="https://www.investopedia.com/articles/personal-finance/121514/how-debt-collection-agency-business-works.asp">https://www.investopedia.com/articles/personal-finance/121514/how-debt-collection-agency-business-works.asp</a> .                                                                                        |
| [6]       | R. S. S. a. A. G. Barto, «Dynamic Programming,» de Reinforcement Learning: An introduction, Cambridge, Massachusetts & London, England, The MIT Press, 2018, p. 73.                                                                                                                                                                                                                             |
| [7]       | R. S. S. & A. G. Barto, «Reinforcement Learning,» de Reinforcement Learning, An introduction, Cambridge, Massachusetts & London, England, The Mit Press, 2018, p. 1.                                                                                                                                                                                                                            |
| [8]       | M. Silva, «Medium,» 30 5 2019. [En línea]. Available: <a href="https://medium.com/aprendizaje-por-refuerzo-introducci%C3%B3n-al-mundo-del/aprendizaje-por-refuerzo-procesos-de-decisi%C3%B3n-de-markov-parte-1-8a0aed1e6c59">https://medium.com/aprendizaje-por-refuerzo-introducci%C3%B3n-al-mundo-del/aprendizaje-por-refuerzo-procesos-de-decisi%C3%B3n-de-markov-parte-1-8a0aed1e6c59</a> . |
| [9]       | J. K. Hoyos, «Trabajo de grado, UTP,» 2019. [En línea]. Available: <a href="http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/10868/T005.746%20H868.pdf?sequence=1&amp;isAllowed=y">http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/10868/T005.746%20H868.pdf?sequence=1&amp;isAllowed=y</a> .                                                                            |
| [10]<br>] | A. Géron, «Confusion Matrix,» de hands-On Machine learning with Scikit-Learn & TensorFlow, O'reilly, 2017, pp. 86-89.                                                                                                                                                                                                                                                                           |
| [11]<br>] | R. S. S. & A. G. Barto, «Finite markov decision procees,» de Reinforcement Learning: An Introduction, Cambridge, Massachusetts & London, England, The MIT Press, 2018, pp. 47-49.                                                                                                                                                                                                               |