ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

DEPARTAMENTO DE POSTGRADOS

PROYECTO DE TITULACION

PREVIO A LA OBTENCION DEL TITULO DE:

MAGISTER EN CIENCIA DE DATOS

TEMA:

“CLASIFICADOR DE EVENTOS PARA TRANSACCIONES DE PAGOS ENTRANTES UTILIZANDO MODELOS DE MACHINE LEARNING EN UNA EMPRESA ENRUTADORA DE PAGOS CON TARJETAS”

Presentado por:

CARLOS FRANCISCO OÑATE BRAVO

Guayaquil – Ecuador

Año: 2023

DEDICATORIA

Como todo en la vida, va dedicado primero a Dios.

A mi esposa, Silvia, por su apoyo, paciencia y trabajo duro en todos los ámbitos que estuve ausente durante mi tiempo de estudio.

A mis hijos: Francisco y Carlos, quienes han sabido comprender el esfuerzo y sacrificio de su papá y mamá, a pesar de su corta edad.

A mis dos ángeles guardianes, mis dos madres: Lupe y Meche, por estar siempre en todo momento. Muy importantes en este logro académico como en otros ya conseguidos.

Carlos Oñate Bravo.

AGRADECIMIENTOS

A Dios, por todas la bendiciones que recibo día a día.

A mi esposa e hijos y mis madres.

A los profesores que supieron transmitir sus conocimientos, algunos desde la ingeniería y ahora en la maestría. Es inspirador ver su dedicación y esfuerzo en la enseñanza.

A mi tutor, Phd. Sergio, por su guía y enseñanzas en este proyecto.

Carlos Oñate Bravo.

DECLARACIÓN EXPRESA

La responsabilidad por los hechos y doctrinas expuestas en este Proyecto de Titulación me corresponde exclusivamente y ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría. El patrimonio intelectual del mismo corresponde exclusivamente a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

_____

Carlos Francisco Oñate Bravo

Autor

COMITÉ EVALUADOR

Firmado electrónicamente por:
SERGIO ALEX BAUZ OLVERA

Ph.D. Sergio Bauz O.

PROFESOR TUTOR

Firmado electrónicamente por:
ALLAN ROBERTO AVENDANO SUDARIO

MSc. Allan Avendaño S.

PROFESOR EVALUADOR

RESUMEN

En la actualidad el uso tarjetas para realizar pagos ha evolucionado de manera que se ha vuelto común realizar este tipo de transacciones en el día a día. La visión de un grupo de inversionistas llevó a la creación de una empresa que se encargue de hacer el nexo entre el comercio que cobra y los autorizadores, quienes aprueban o declinan el pago. Esta tarea es posible mediante el uso de redes de comunicación para el enrutamiento de las transacciones, donde intervienen distintos factores como: tecnologías, servicios, empresas y datos. La empresa es la encargada de hacer que esta comunicación sea lo más transparente posible para el comercio que realiza el cobro de un producto o servicio.

Este proyecto de tesis propone la creación de una herramienta que aplique un algoritmo de *Machine Learning (ML)* con aprendizaje supervisado, que clasifique los eventos o errores que se presentan en las transacciones que pasan por la empresa; se evaluaron distintos modelos de los algoritmos; el primero es Clasificador de Arboles de Decisión (DT), continuaremos con Redes Neuronales (NN) y, finalmente, Máquina de Vectores de Soporte (SVM); de la comparación entre los resultados que estos arrojaron, se seleccionó el algoritmo que obtuvo el mejor resultado en la precisión de la clasificación de eventos.

Esta clasificación de eventos, que permite identificar la severidad y grupo de personas que deben atenderlos, es visualizada en tiempo casi real; ayuda en la reducción de tiempo en la identificación de errores, uso eficiente de los recursos, reducción de pérdidas económicas, reducción de posibilidad de sanciones por parte entes reguladores ante la falta de servicios, aumento de satisfacción del comercio que realiza el cobro con tarjeta, entre otros.

ABSTRACT

In current times, the use of cards to make payments has evolved in such a way that it has become common to carry out this type of transactions in everyday life. The vision of investors led the creation of a company that oversees making the link between the merchant that collects and the authorizers, who approve or decline the payments. The task is possible by communication networks for routing transactions, where different factors intervene such as: technologies, services, companies, and data. The company is in charge of making this communication as transparent as possible for the business that charges for a product or service.

This project proposes the creation of a tool that applies a Machine Learning algorithm with supervised learning, which classifies the events or errors that occur in the transactions that go through the company; different models of the Support Vector Machine, Decision Tree Classifiers and Neural Network algorithms were evaluated; from the comparison between the results that these produced, the algorithm that obtained the best result in the precision of event classification was selected.

This classification of events, which makes it possible to identify the severity and group of people who must attend to them, is displayed in almost real time; help in reducing time in identifying errors, efficient use of resources, reduction of economic losses, reduction of the possibility of sanctions by regulatory entities due to the lack of services, increase in satisfaction of the merchant that collects by card, among others.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICES DE TABLAS

CAPÍTULO 1

1. Análisis de la problemática

1.1. Descripción del Problema

En la actualidad los servicios financieros se han ampliado en una gran variedad, uno de estos servicios corresponde a los medios de pago electrónicos, dentro de los cuales se encuentran las tarjetas de crédito y débito, que son parte fundamental en los insumos del problema que objetivo de esta investigación.

Ecuador tiene un aproximado de 17.64 millones de habitantes entre los cuales se manejan alrededor de 5 millones de tarjetas, entre crédito y débito hasta agosto de 2022, esto genera un promedio de 14 millones de transacciones mensuales.

Las transacciones de pago con tarjetas de crédito o débito se inician en los puntos de cobros, físicos o virtuales, de los comercios en todo el territorio ecuatoriano. Estas transacciones deben ser aprobadas por el banco emisor de la tarjeta, para llegar al banco que aprueba o declina la transacción se ejecutan validaciones y análisis de algunos datos, entre los cuales se encuentran: la marca de tarjeta (Diners, Visa, Mastercard, American Express), el BIN (primeros 6 dígitos del número de la tarjeta), banco seleccionado por el comercio para procesamiento de la aprobación, tipo de pago (físico o virtual), entre otros.

La empresa enrutadora de pagos, que de ahora en adelante se la conocerá como “LA EMPRESA”, recibe cada segundo decenas de transacciones de pago desde los terminales. Estas transacciones entran por distintos canales (cajas, POS, botones de pago) y se enrutan a través de: el enrutador de transacciones, diferentes switches transaccionales y core transaccional; llega al

autorizador quien genera la respuesta en forma de aprobación o declinación; finalmente, esta respuesta es la que se devuelve hacia el comercio recorriendo el camino de regreso hasta llegar al terminal que inició la transacción.

Durante la petición de una transacción pueden ocurrir muchos errores en el camino, por diferentes motivos como: falla en el enrutador de entrada, servicios deshabilitados en los switches o core transaccional, el autorizador no está atendiendo, entre muchos otros. Estos errores se dan en todo momento, teniendo varios tipos de afectaciones como: estadísticas de transacciones, donde se busca tener el mayor cantidad de transacciones completas, donde el flujo de ida y regreso tiene que completarse; satisfacción del cliente de LA EMPRESA; satisfacción del tarjetahabiente y afectación económica, ya que cada transacción completa representa un ingreso monetario.

El análisis de un evento, cuando sucede, conlleva a utilizar recursos de varias áreas dentro LA EMPRESA, estos son: analistas de terminales, analistas de core, desarrolladores, supervisores de producción, oficiales de networking y administradores de sistemas. Esta metodología de trabajo incrementa el tiempo de identificación del problema, cada recurso de las distintas áreas tiene acceso a los datos propios de su campo laboral lo cual ayuda a la identificación del problema del evento.

1.2. Justificación

La implementación de este proyecto pretende ayudar en la identificación de los eventos mediante la implementación de un panel para visualizar la clasificación de las transacciones entrantes con códigos de errores, basada en la información que se genera a partir de los datos que

ingresan a LA EMPRESA; esto permitirá disminuir los tiempos de atención de los eventos, uso de recursos específicos por evento, mejorar la experiencia del cliente y tarjetahabiente.

Se aplicará un modelo de Machine Learning donde cada evento clasificado indicará el origen del problema; esto permitirá asignar el o los recursos específicos para dar solución al evento, ayudando a disminuir del tiempo de identificación del problema; esto se traduce en aumento de disponibilidad del servicio de transaccionalidad, reducción de pérdidas económicas, uso eficiente del tiempo de los recursos para atención de eventos.

1.3. Solución propuesta

Se propone implementar una herramienta que logre la mejor clasificación de eventos mediante la selección del modelo de *ML* más preciso que surja comparando entre: NN, SVM y DT; aquel que tenga los mejores resultados de Error Medio Cuadrático (MSE), matriz de confusión y precisión en la clasificación de los eventos será el seleccionado para la implementación.

Esta implementación ayudará a la visualización de la clasificación de los eventos y su respectivo origen del problema mediante una aplicación web interactiva, que permita tomar decisiones en base a los datos.

1.4. Objetivos

1.4.1. Objetivo general

Establecer un modelo para el algoritmo seleccionado entre SVM, NN y DT mediante la comparación de precisión entre ellos, para clasificar los eventos o errores que ocurren en el durante el proceso de la transacción para disminuir el tiempo en la identificación de los problemas.

1.4.2. Objetivos específicos

- Estudiar la base de datos de enrutamientos de transacciones de pagos para identificar las características y factores relevantes asociadas a dichos eventos.
- Analizar de manera multivariante los eventos del enrutador, los switches y core transaccional para la caracterización de dichos eventos basados en hardware, el software y la gestión.
- Aplicar los modelos de Machine Learning para la clasificación de los eventos mediante aprendizaje supervisado y proponer una estrategia de actualización del modelo ante nuevos eventos.
- Desarrollar una herramienta de visualización web interactiva de informes para ayudar a tomar las decisiones gerenciales apoyándose en datos. La herramienta seleccionada estará dada por la decisión indicada por las gerencias.

1.5. Metodología

Se implementará un análisis segmentado por cada uno de los 3 componentes principales dentro del flujo de una transacción de pago, los cuales son: enrutador de transacciones, switches transaccionales y core transaccional, como se aprecia en la **Figura 1**.

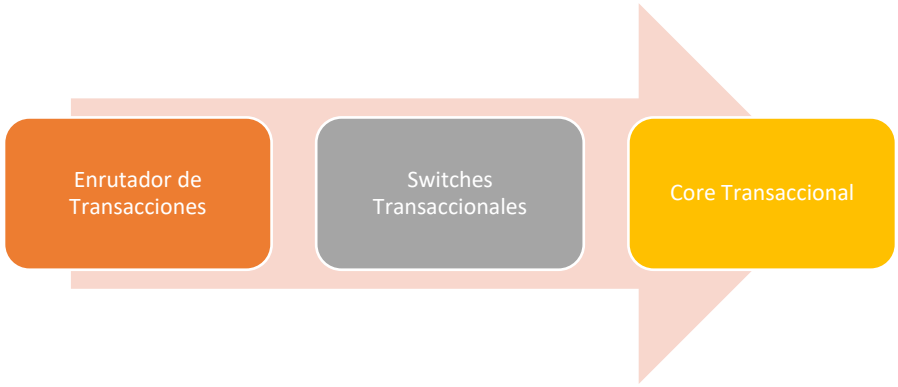


Figura 1. Componentes dentro del proceso de enrutamiento transaccional de Datafast

Dentro del análisis se busca seleccionar el mejor modelo entre NN, SVM y DT. El proceso para la implementación a ejecutar se basará en el modelo de *pipeline* de ciencia de datos, detallado en la **Figura 2**.

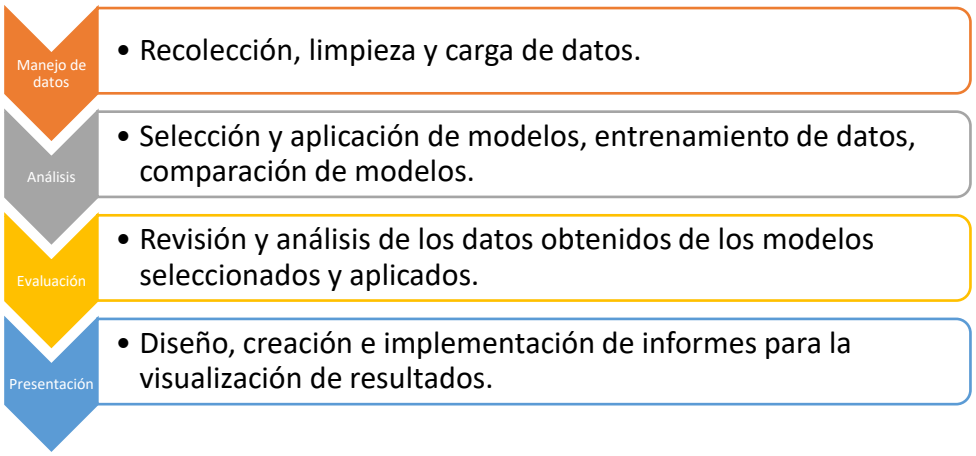


Figura 2. Proceso de implementación

1.6. Resultados Esperados

Una vez implementada la solución utilizando el modelo de *Machine Learning* seleccionado, se espera tener:

- Toma de decisión basada en datos.
- Reducción de tiempo en la identificación de los eventos.
- Reducción en los recursos de las distintas áreas requeridos para análisis de los eventos.
- Reducción de pérdidas económicas.
- Aumento del tiempo actividad de servicios de LA EMPRESA.

1.7. Conjunto de datos (*dataset*)

Los datos existentes en los distintos componentes se encuentran en Bases de Datos SQL (BD), lo que permite tener información en modelos relacionales. Cada BD contiene datos específicos de la transacción correspondiente al componente.

Del componente Enrutador de Transacciones se contará con información de la tabla principal, donde se registran las transacciones que ingresan y son enrutadas a la siguiente fase del proceso; del componente Switch Transaccional se tomarán los datos de las tabla que relaciona las transacciones con los distintos componentes internos del switch; del componente Core Transaccional se tomarán los datos de la tabla principal de transacciones. Los dos últimos componentes comparten datos similares a tratar la transacción a nivel de aplicación.

Los datos de cada componente serán pasados a una nueva Base de Datos centralizada donde reside el modelo para el clasificador de ML. Estos serán llenados en la fase Manejo de datos. Estos nuevos modelos serán llenados con procesos ETL con una frecuencia a determinar en base a la necesidad del negocio.

1.7.1. Detalle de datos del Enrutador de transacciones

Datos correspondientes a las transacciones que ingresan en el componente Enrutador de transacciones, es la primera capa del enrutamiento dentro de LA EMPRESA. Dentro de estos datos se tienen valores de fecha, hora, estado, duración, cajas enrutadora, comercio, terminal, entre otros; para mayor detalle se puede tener más detalles en el **Anexo 1**.

1.7.2. Detalle de datos del Switches Transaccionales

Datos correspondientes a las transacciones que ingresan en el componente Switch Transaccional, es la segunda capa del enrutamiento dentro de LA EMPRESA. Se tienen datos como: fecha, hora, duración, comercio, terminal, tipo de switch, respuesta del switch, respuesta de core, entre otros. La lista y descripción de cada campo se encuentran en el **Anexo 2**.

1.7.3. Detalle de datos del Core Transaccional

Datos correspondientes a las transacciones que ingresan en el componente Core Transaccional, es la tercera capa del enrutamiento dentro de LA EMPRESA. Se cuenta con datos como: fecha, hora, duración, comercio, terminal, respuesta de core, respuesta de autorizador, emisor, pinblock, entre otros. El total de campos y sus detalles se encuentran en el **Anexo 3**.

CAPÍTULO 2

2. Marco Teórico y Estado Del Arte

El ser humano se enfrenta a tomas de decisiones en todo momento, para para esta tarea nos basamos en la experiencia y conocimientos que hemos adquirido a lo largo de nuestras vidas, toda existencia a nuestro alrededor genera datos cada segundo; es por eso por lo que, la clasificación o categorización de los datos generados en los escenarios o eventos nos ofrece una ayuda y orden al momento de tomar una decisión.

Dentro de la investigación relacionada a transacciones con tarjetas de pago, de débito y de crédito, se han encontrado artículos que abordan ciertas problemáticas relacionadas a este trabajo, como el caso de Análisis de Detección de Fraude en Tarjetas de Crédito usando Técnicas de Machine Learning (C.H., POKALA, BOLISETTI, & BALASUBRAMANI, 2022), donde se utiliza una técnica de redes neuronales profunda para clasificar un conjunto de datos de tarjetas de crédito, también utiliza Máquina de Vectores de Soporte (SVM) y Naive Bayes; el objetivo del proyecto es la detección de estafas con tarjetas de crédito ya que van en aumento por la creciente transaccionalidad en línea.

De la misma manera, en la investigación Un enfoque novedoso para la detección de fraudes con tarjetas de crédito utilizando árboles de decisión y algoritmos de bosques aleatorios (DILEEP, NAVANEETH, & ABHISHEK, 2021) se utiliza dos métodos: el primero construye un árbol de decisión contra las actividades realizadas por el usuario con estafas de las cuales se sospechará; el segundo construye un bosque aleatorio basado en la actividad del usuario con el cual se intenta identificar al sospechoso. Toda esta investigación nace del aumento de nuevos sistemas de pago y de la deficiencia de los datos en las tarjetas en mora por temas de privacidad.

Las empresas, al igual que el ser humano, debe basar su crecimiento en la toma de decisiones, pero es indispensable basar esta acción en los datos que se poseen. Para de la clasificación de los datos se distinguen dos métodos: el aprendizaje supervisado y el aprendizaje no supervisado.

2.1. Aprendizaje no supervisado

En los métodos no supervisados, ninguna variable objetivo se identifica como tal. En cambio, el algoritmo de minería de datos busca patrones y estructuras entre todas las variables (LAROSE & LAROSE, 2014).

El método no supervisado es aquel donde los datos se agrupan sin una clasificación previa en la cual basarse, este nos permitirá identificar características que son comunes entre los registros agrupados (LAROSE & LAROSE, 2014).

2.2. Aprendizaje supervisado

El aprendizaje supervisado, o aproximación de funciones, consiste simplemente en ajustar datos a una función de cualquier variedad (KIRK, 2017).

la mayoría de los métodos de minería de datos son métodos supervisados, lo que significa que (1) hay una variable objetivo-preespecificada en particular, y (2) el algoritmo recibe muchos ejemplos en los que se proporciona el valor de la variable objetivo, de modo que el algoritmo pueda aprender qué valores de la variable objetivo están asociados con qué valores de las variables predictoras (LAROSE & LAROSE, 2014).

El método supervisado es aquel donde los datos tienen una clasificación o valoración previa, en la cual se basan los algoritmos para aprender a predecir o clasificar datos nuevos.

2.3. Análisis exploratorio de datos (EDA)

Dentro del modelamiento de Machine Learning es importante tener la etapa de donde se explora y analiza los datos a utilizar, de manera que podamos contar con un resumen de la información para un mejor entendimiento. El análisis exploratorio de datos nos ayuda a identificar errores en las características, tipos de datos, datos atípicos, datos faltantes, relación entre variables donde podemos obtener un análisis descriptivo. El análisis exploratorio de datos se incluirá la técnica estadística Análisis de Componentes Principales.

2.3.1. PCA

El análisis de componentes principales (PCA) Es la técnica de reducción de dimensiones con muchas ventajas y aplicaciones, nos permite reducir las dimensiones de un *dataset* en un subespacio dimensional menor. Tienes las siguientes utilidades (DANGETI, 2017):

- Mitigar el curso de la dimensionalidad.
- Comprimir los datos mientras se minimiza la perdida de información.
- Se los utiliza más en la etapa de aprendizaje supervisado.
- Comprender estructura de datos con cientos de dimensiones puede ser difícil, al reducir las dimensiones a 2 o 3, las observaciones pueden ser visualizados fácilmente.

2.3.2. Análisis de Factor

El análisis de factor (FA) utiliza variables estandarizadas para reducir conjuntos de datos mediante el Análisis de Componentes Principales. Está basada en una descomposición ortogonal de la matriz de entrada, esta genera una nueva matriz con un grupo de componentes o factores ortogonales que maximizan la cantidad de variación en las variables de la matriz de entrada (BOSLAUGH, 2012).

Aunque PCA es usado para producir variables que son ortogonales en un modelo linear y procesar un gran número de variables en una menor cantidad, un conjunto de datos más manejable, FA nos ayuda a identificar variables latentes que son representados por variables de entrada altamente correlacionados.

2.3.3. Análisis de Cluster

Es un conjunto de técnicas que permiten a los casos ser agrupados basados en los valores para una o más variables. Una técnica relacionada es el Análisis de Función Discriminante (DFA), puede ser usada para desarrolla reglas para asignar casos a los grupos. DFA es mejor para predecir la pertenencia al grupo que hacer un solo un análisis cluster (BOSLAUGH, 2012).

El análisis de cluster es útil en dos escenarios. Primero, si se sabe la cantidad de grupos, se puede pasarlo en algoritmo; segundo, es cuando no se sabe cuántos grupos existen, el algoritmo puede estimarlos.

2.4. Modelo de Clasificación de anomalías

Es importante conocer casos en los que se han utilizado modelos de clasificación de anomalías de eventos, con una naturaleza similar a la de este proyecto, de manera que podamos contar con un criterio previamente obtenido, en la investigación Simulación y modelado para la detección de anomalías en la red IoT mediante el aprendizaje automático (MUKHERJEE, SAHU, & SAHANA, 2021), cuyo objetivo principal es seleccionar un modelo de aprendizaje supervisado para predecir anomalías en los datos históricos para luego incorporarlos al mundo real y proteger de futuros ataques o anomalías; se utilizaron los algoritmos de clasificación Regresión Logística (LR), Naive Bayes (NB), Arboles de Decisión (DT), Bosques Aleatorios (RF) y Redes Neuronales Artificiales (ANN). Finalmente, en la conclusión se tiene que

Regresión Logística, Arboles de Decisión, Bosques Aleatorios y Redes Neuronales son mejores que Naive Bayes en el caso donde se modelan todas las características, mientras que Aboles de Decisión y Bosques Aleatorios son mejores cuando se excluyen las características con valores binarios.

2.4.1. Clasificación por Árboles de decisión

Los árboles en la ciencia de la computación han sido utilizados para representar diferentes tipos de situaciones y estructuras, siendo una de las más usadas las representaciones jerárquicas; permitiendo ejecuciones y recorridos recursivos. Las estructuras de los árboles consisten en un nodo raíz, ramas, nodos hijos y nodos hojas (BONACCORSO, 2018).

Un árbol de decisión es un conjunto de nodos para tomar decisiones, están relacionados por ramas, este crece hacia abajo y parte de un nodo raíz, terminan en nodos llamados hoja. El nodo raíz, se lo coloca en la parte arriba de todos los nodos para una mejor guía gráfica; las características o atributos van en los, nodos donde son probados, los resultados que se puedan obtener se transforman en ramas. Como paso final, las ramas llevan a otros nodos que generan nuevas decisiones o a una rama. (LAROSE & LAROSE, 2014).

Para lograr que un árbol de decisión logre la clasificación se debe cumplir con 3 requerimientos principales:

- Contar con un *dataset* de entrenamiento para basar el aprendizaje supervisado en una variable objetivo.
- El *dataset* de entrenamiento debe ser variado sin sesgo.
- La variable objetivo debe ser discreta.

Para tener una mejor clasificación se utilizan los siguientes criterios en la selección en los nodos (AMAT RODRIGO, 2017):

- Impureza Gini: Mide la tasa de impureza de la distribución de clase o de la tasa de la mezcla de clases. Un indicador de impureza Gini más bajo indica más pureza. Si se tiene K clases, donde los datos de la clase k toma hasta una fracción ($0 \leq f_k \leq 1$) de la totalidad del *dataset* (LIU Y. H., 2020).

$$Impureza\ Gini = 1 - \sum_{k=1}^{K} fk^2$$

- Ganancia de la información: Mide la mejora de la pureza luego de la división, o, en otras palabras, la reducción de la incertidumbre debido a una división. Una mayor ganancia de información implica una mejore división. Obtenemos la ganancia de información de una división comparando la entropía antes y después de la división. La entropía es una medida probabilística de la incertidumbre. Menor entropía implica un *dataset* más puro con menor ambigüedad (Yuxi Hayden Liu, 2020).

$$Entropía = -\sum_{k=1}^{K} f_k * log_2 f_k$$

2.4.1.1. Algoritmo C4.5

Ross Quinlan desarrolló el algoritmo C4.5, en cual es una mejora de ID3, desarrollado con anterioridad por el mismo Quinlan. Este árbol de decisión permite manejar atributos discretos y continuos, umbrales para división atributos continuos, manejar datos que tengan valores faltantes en las características o atributos, uso de costos diferentes para los atributos y

finalmente tiene poda de árboles, esto permite que se eliminen las ramificaciones que no aportan a la clasificación, convirtiéndolos en nodo hoja (ROSS QUINLAN, 1993).

2.4.1.2. Algoritmo C5.0.

La versión C4.5 fue mejorada con fines comerciales, esto dio paso a la versión C5.0 fue desarrollado, una vez más Quinlan fue el creador; entre las características a resaltar están: mayor rapidez que la versión anterior, uso de memoria más eficiente, soporte para boosting y ponderación de variedad de casos y distintos tipos de errores que se puedan dar en la clasificación. Permite solo valores categóricos en la variable objetivo (LANTZ, 2015).

2.4.2. Clasificación por Maquina de Soportada por Vectores (SVM)

La Máquina Soportada por Vectores o SVM, por sus siglas en inglés, es utilizado en método de aprendizaje supervisado, desarrollado por Vladimir Vapnik, puede ser utilizado para clasificación o regresión.

SVM es uno de los algoritmos más populares cuando se trata de espacios de alta dimensión. El objetivo del algoritmo es encontrar un límite de decisión para separar datos de diferentes clases (LIU Y. H., 2020).

El funcionamiento consiste en separar las clases a través de hiperplanos, tratando de obtener la máxima distancia de separación entre los hiperplanos asignado para separar cada clase. El vector que se forma por los puntos que estén más cerca del hiperplano es conocido como Vector de Soporte (NAZARATHY & KLOK, 2021).

Existen dos tipos de SVM, estos son (SAKARKAR, PATIL, & DUTTA, 2021):

- **Lineal:** SVM lineal se usa para datos separables linealmente, lo que significa que, si un conjunto de datos se puede clasificar en dos clases usando una sola línea recta, dichos

datos se denominan datos separables linealmente, y el clasificador se usa como clasificador SVM lineal.

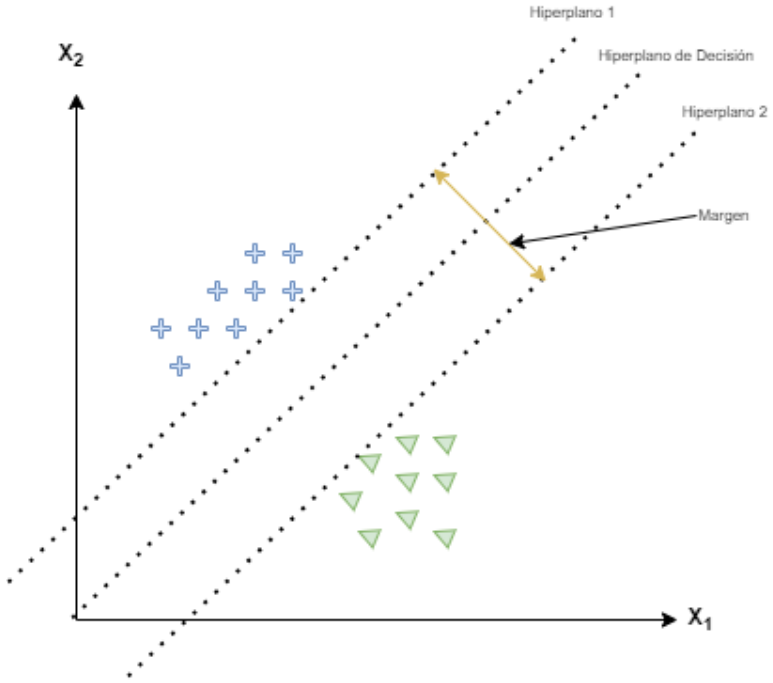


Figura 3. Ejemplo de un SVM lineal.

- **No lineal:** Se usa para datos separados no linealmente, lo que significa que, si un conjunto de datos no se puede clasificar usando una línea recta, dichos datos se denominan datos no lineales y el clasificador utilizado se denomina clasificador SVM no lineal.

Figura 4. Ejemplo de un SVM no lineal

2.4.3. Clasificación por Redes Neuronales

Nuestro cerebro es excelente en la distinción de patrones, este cuenta con alrededor de 100.000 millones de neuronas, las cuales se comunican entre sí, permitiéndole controlar la mayor parte de cuerpo humano. Los científicos han tratado de descifrar el funcionamiento complejo del cerebro y los matemáticos han tratado, desde hace mucho tiempo, lograr la representación matemática de este funcionamiento.

La historia de las Redes Neuronales (NN) o Redes Neuronales Artificiales (ANN) tiene casi 80 años, cuando allá por 1943 Warren McCulloch, un neurofisiólogo, y Walter Pitts, un matemático, lanzaron una teoría acerca de la forma de trabajar de las neuronas, modelaron una NN simple con circuitos electrónicos.

En 1957 Frank Rosenblatt desarrolló el perceptrón, conocido hoy en día como neurona, esta le permitía el reconocimiento de patrones nuevos después de haber aprendido otros similares, este tenía sus limitaciones ya que no era capaz de clasificar clases no separables linealmente.

Una red neuronal consiste básicamente en 3 capas; la primera es la entrada de datos, la segunda es la oculta o interna donde se aplican funciones de activación y la tercera es la de salida de los resultados. Las redes neuronales se distinguen por la cantidad de capas y cuantas neuronas tiene en cada capa. En la siguiente imagen se aprecia un ejemplo de un Red Neuronal de 3 capas, en la primera capa se tiene cuatro neuronas, la segunda capa cuenta con dos neuronas y la última capa tiene una neurona, que porta con el resultado (SRINIVAS, SUCHARITHA, & MATTA, 2021).

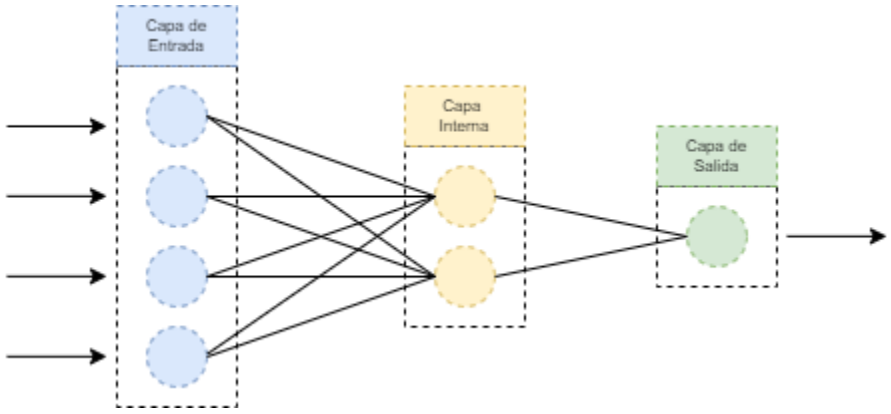


Figura 5. Modelo de Red Neuronal de 3 capas.

Las redes neuronales de una capa se van alimentando con la salida que produce cada neurona de una capa previa. La información recibida en cada neurona se la procesa con una función llamada: función de activación.

Las funciones de activación son aquellas que permiten distinguir patrones y suavizar los ruidos que exista en la entrada. Existen 3 tipos de funciones de activación, los cuales son:

- **Binarias:** Esta función depende de un umbral que indica a la neurona si se activa o no.

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

- **Lineales:** Basada en ecuaciones lineales.

$$f(x) = x + b$$

- **No lineales:** Son funciones que toma entradas de valor real y producen las salidas basadas en una función lineal de regresión. Ayuda a no tener un mapeo complejo entre la entrada y salida (LIU, y otros, 2021). En la tabla a continuación se lista las funciones de activación más conocidas.

Tabla 1. Funciones de Activación no lineales.

Nombre	**Descripción**	**Función**
Sigmoide / Lógica	Toma valores reales y produce una salida entre 0 y 1.	$f(x) = \frac{1}{1 + e^{-x}}$
TANH	Toma valores reales y produce una salida entre -1 y 1.	$f(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$
ReLU	El nombre viene de Unidad Lineal Rectificada, por sus siglas en inglés. No activa todas las neuronas al mismo tiempo. Se activa cuando la transformación lineal es mayor que cero.	$f(x) = \max(0, x)$
Leaky ReLU	Es la versión mejorada de la función ReLU para resolver el problema conocido como Dying ReLU.	$f(x) = \max(0{,}1x, x)$
ReLU	Resuelve el problema cuando la gradiente se vuelve	$f(x) = \max(ax, x)$

paramétrica	cero en la parte izquierda del eje x. “a” es el valor de pendiente para valores negativos.	
Softmax	Corrige el comportamiento de la función sigmoide. En esta función la sumatoria de las probabilidades es igual a 1.	$\text{Softmax}(Z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_i)}$
Swish	Función desarrollada por investigadores de Google. Es una función de suavizado, no cambia los valores de forma abrupta como lo hace ReLU. Toma en cuenta valores negativo de entrada.	$f(x) = \text{x} * \text{sigmoide}(x)$

2.5. Software

Por el largo camino que recorre el flujo de una transacción desde su inicio; pasando por todos los componentes de LA EMPRESA, componentes para ML y la visualización de la información por parte de usuarios final, se debe contar con distintas herramientas que permitan la maniobrabilidad adecuada en cada funcionalidad requerida, entre las cuales tenemos: almacenamiento, carga de datos, transformación de datos, procesamiento de modelos y visualización de datos. Estas herramientas se las puede clasificar en dos grupos que son: backoffice, aquellas que permiten el procesamiento a nivel de servidores, y frontend: aquellas herramientas a nivel de cliente, que permiten interactuar con el usuario final.

2.5.1. Componentes de Backoffice

Almacenamiento: Para el almacenamiento de datos se utilizarán bases de datos con un RDBMS que permita el manejo masivo de datos con buen rendimiento en inserción y lectura; dentro de esta se manejará el modelo de bases de datos donde residirán las tablas centrales, estas

corresponden a la información con las características seleccionadas para ser utilizadas por el modelo seleccionado de ML, así como las tablas catálogos de las entidades relacionadas como son: comercios, terminales, información geográfica, entre otros.

ETL: Como parte del proceso se deben ejecutar tareas de extracción, procesamiento y carga de datos donde se utilizará software especializado que se encargue de realizar este procedimiento de manera que me permita conectar el origen y el destino, sabiendo que estos dos extremos son bases de datos.

ML: Parte del flujo del objetivo global que busca la clasificación de los eventos se requiere utilizar un lenguaje de programación funcional, como R o Python, que permite el tratamiento de los datos y aplicación del modelo de ML seleccionado.

Procesamiento: El uso de servidores con componente como Discos Duros para almacenamiento, CPU para procesamiento lógico matemático y Memoria RAM para almacenamiento volátil de alta velocidad son el componente esencial

2.5.2. Componentes de Frontend

Servidores de aplicación: Para hospedar la aplicación web que se va a utilizar es necesario contar con un servidor de aplicaciones web que permita ejecutar aplicaciones que puedan conectarse a la base de datos y extraer los datos de manera que los paneles de visualización puedan ser actualizarse constantemente. La aplicación será desarrollada con lenguaje Java y el framework JSF, de manera que pueda servir a las consultas con lenguaje HTML, CSS y Javascript.

Cliente web: El software cliente para configurar con acciones de guardado, actualización, lectura y eliminación de registros para la clasificación de eventos, debe permitir

conectarse al servidor de aplicaciones de manera que maneje HTML, CSS y Javascript, por lo que se utilizarán navegadores web disponibles en el mercado como Google Chrome, Mozilla Firefox, entre otros.

Plataforma de Inteligencia de Negocio: Herramienta que permite la visualización y dar formato a la información, de manera que se pueda aplicar los cambios en tiempo real. En el mercado existen distintos proveedores de este tipo de software, los cuales son seleccionados en base a muchos criterios, siendo uno de los principales la capacidad económica.

CAPÍTULO 3

3. Diseño e Implementación

Partiendo del objetivo principal, que es crear un modelo que se ajuste lo mejor posible a la clasificación de los eventos que ocurran sobre los componentes principales, de esta manera se logrará reducir el tiempo y recursos utilizados para el análisis y posterior solución a los eventos, el resultado debe ser mostrado en una aplicación web donde la página que muestra la información va a ser actualizada cada X segundos al usuario encargado del monitoreo.

La aplicación web contará con paneles donde se mostrarán en gráficos la clasificación de los eventos, se contará con la distinción por cada clase, producto de la clasificación; esta información es mostrada por cada componente: enrutador de transacción, switch transaccional y core transaccional.

Para esta presentación se hace una selección previa de los gráficos a mostrar en cada panel entre los cuales se tiene las opciones de barras verticales, barras horizontales, gráfico de líneas en el tiempo, mapa de Ecuador, gráfico de áreas, entre otros.

3.1. Infraestructura de la solución

Para implementar esta solución se debe tener una visión global del proceso y todos sus componentes, estos son:

- El origen de la transacción está en los comercios, sitios web y pasarelas; estos los clientes de LA EMPRESA.
- Comunicación estable y confiable desde el origen hacia la EMPRESA, permite la trasmisión y recepción de las transacciones.

- Componentes encargados del flujo de la transacción, estos son: Enrutador de Transacciones, Switches Transaccionales y Core Transaccional.
- Procesos ETL para los datos que se utilizarán en el modelo de ML, permiten la extracción, procesamiento y carga de datos desde las bases de datos de los componentes hasta la base de datos para el modelo de ML a utilizar en la clasificación.
- Servidores y base de datos para ML, donde residen los datos provenientes de los distintos componentes y se ejecuta la clasificación de los eventos con el modelo seleccionado.
- Servidor de ML, contiene y ejecuta los procesos batch de clasificación, las ejecuciones se realizan mediante una programación calendarizada.
- Power BI, herramienta de presentación de los paneles con los datos actuales y de la clasificación por componente.

A continuación, en la **Figura 6**, se presenta el diagrama de la infraestructura de componentes y flujo de información.

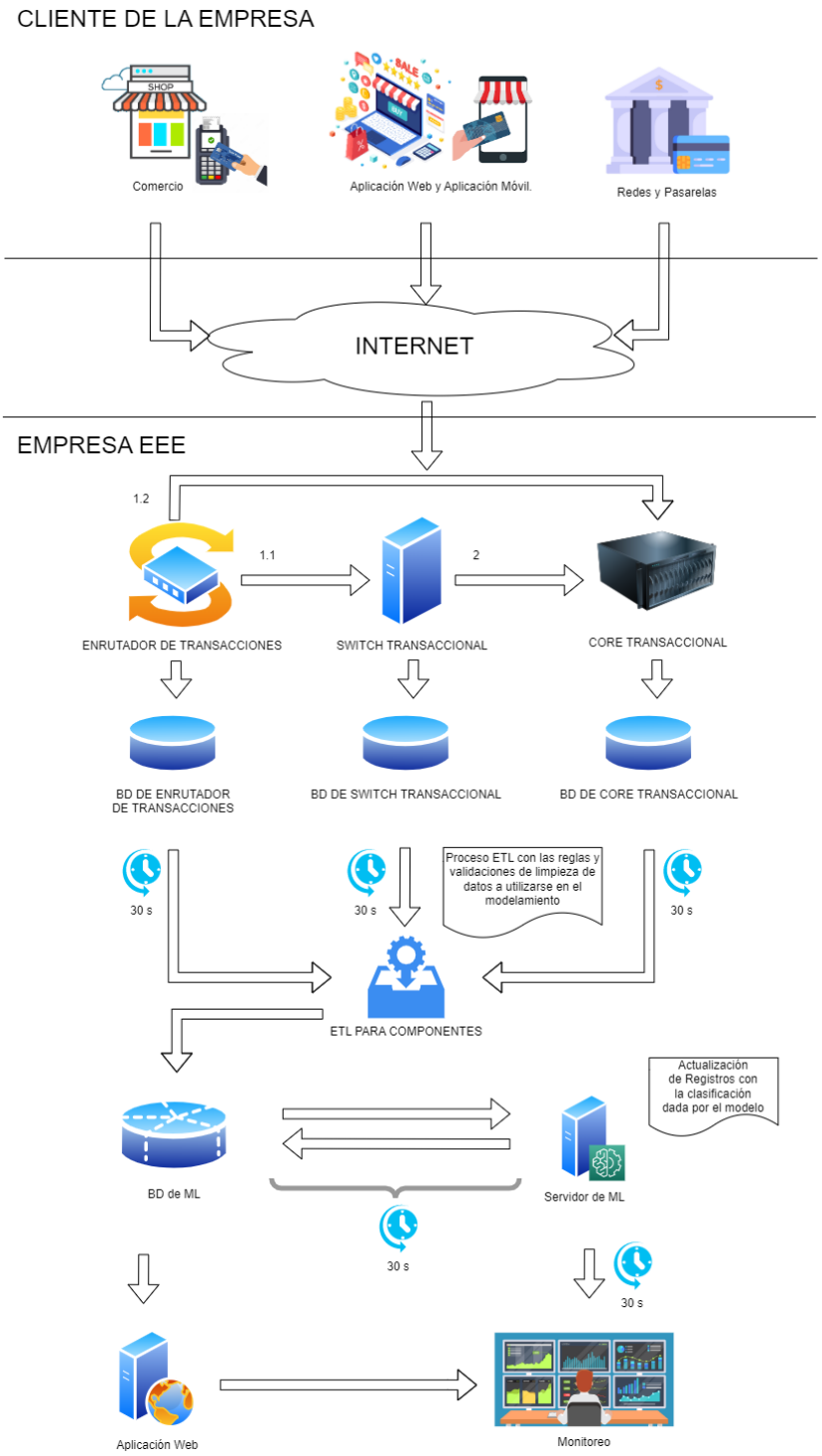


Figura 6. Infraestructura de componentes y flujo de información.

3.1.1. Cliente de LA EMPRESA

Los clientes de la empresa son aquellos que contratan un servicio que permite realizar cobros con tarjetas de crédito o pago que permitan llevar a cabo una transacción comercial, este servicio establece la comunicación con LA EMPRESA, de manera que puedan transmitir el pago y recibir una aprobación o declinación.

Los clientes poseen distintos tipos de dispositivos para efectuar cobro con tarjetas, estas son: terminal físico o POS, terminal virtual y un cliente especial llamado Redes o Pasarelas.

Tabla 2. Tipos de dispositivos de cobro disponibles en LA EMPRESA

Tipo de dispositivos de cobro	Descripción
Terminal físico o POS	Permite el uso de una tarjeta física de forma presencial o no presencial. Estos equipos tienen diferentes marcas y cada marca tiene distintos modelos.
Terminal virtual	También llamadas como botones de pago, permiten realizar el cobro en aplicaciones web o aplicaciones móviles.
Redes o Pasarelas	Es una entidad con permiso legal de ofrecer los mismos servicios que LA EMPRESA, pero deben conectarse hacia LA EMPRESA para poder llegar al autorizador.

3.1.2. Comunicación

Sin duda, la comunicación de redes es lo que permite que exista la transaccionalidad entre LA EMPRESA y sus clientes en este mundo de pagos con datos electrónicos. Los métodos de comunicación para la autorización de pagos con tarjetas fueron evolucionando a medida que la tecnología avanzaba, es así como al inicio se lo hacía por medio de llamadas telefónicas

directas al autorizador; luego, se pudo contar con terminales POS que empezaron con la comunicación mediante línea telefónica, seguido por la comunicación por redes WAN y LAN; continuó con la comunicación por GPRS de las redes celulares, enlaces directos; finamente, se cuenta con la comunicación directa a través de Internet. La **Tabla 3** detalla la información correspondiente a cada tecnología.

Tabla 3. Tipos de tecnología de comunicación para transmisión de pagos

Tipo de tecnología	Descripción
Línea telefónica o DIAL UP	Los terminales iniciales contaban con la capacidad de trasmitir la transacción mediante la comunicación al autorizador por medio de una llamada telefónica. El tiempo de la transacción tenía un promedio de 6 segundos de duración.
WAN, LAN o Wi-Fi	Con el avance de la tecnología se crearon terminales con la capacidad de crear comunicaciones con redes WAL y LAN mediante cables coaxiales, FTP, UTP y UTP. Dentro de esta misma categoría, con la aparición de la tecnología Wi-Fi, se incluyó este método entre las opciones disponibles.
GPRS	Con la aparición del servicio de comunicación satelital por medio de redes celulares se hizo posible transmitir las transacciones a través de Internet.
Enlaces directos	Para clientes que disponen de alta tecnología y una tasa alta de transaccionalidad se hace necesario tener un canal dedicado o enlaces directos que permitan tener alta velocidad de transmisión y alta seguridad.

3.1.3. Transacciones de pago

Las transacciones comerciales con tarjetas se originan en los comercios, clientes de LA EMPRESA, estos a su vez tienen clientes que poseen o son dueños de las tarjetas de crédito o débito, llamados tarjeta habiente, que usan para realizar una transacción de compra. Las

transacciones ocurren a cada segundo en todo el Ecuador, la cantidad de dueños de tarjetas multiplicados por los clientes de LA EMPRESA hace que se lleguen alrededor de 12 transacciones por segundo (TPS) en promedio en horario laborable. El tiempo máximo para completar una transacción está dada por cada autorizador, se lo mide en segundos, siendo entre 2 y 4 segundos el tiempo normal de autorización, con máximos de 10 segundos y 20 segundos de espera; el tiempo máximo de espera lo da el autorizador. La **Figura 7** muestra el flujo de la transacción a lo largo del tiempo.

A pesar de solo ser una entidad de paso de transacciones y enrutarlas al autorizador correspondiente, LA EMPRESA es calificada como entidad bancaria por lo que tiene la obligación de proveer su servicio de enrutamiento las 24 horas del día los 7 días de la semana.

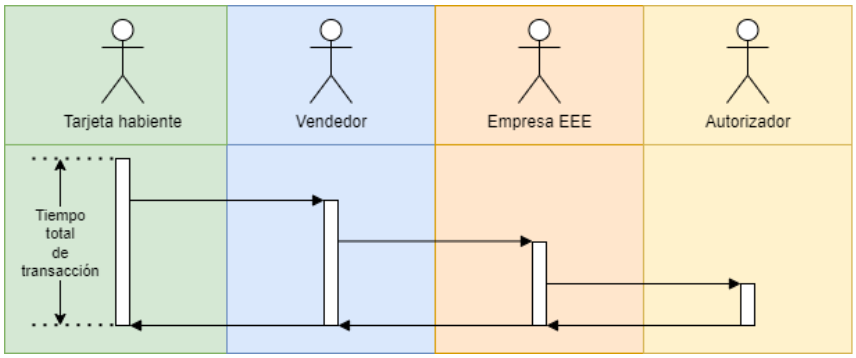


Figura 7. Modelo unificado de una transacción a lo largo del tiempo.

3.1.4. Trama de pago

Las transacciones que son enviadas hacia LA EMPRESA llegan como mensajes de redes por protocolo TCP/IP en un formato específico, estos mensajes son conocidos como TRAMA. La trama se basa en el estándar para transacciones financieras con mensaje originados por una

tarjeta de crédito, ISO 8583, con pequeñas variantes dadas por cada autorizador. Las especificaciones para esta trama vienen detalladas en la siguiente tabla.

Tabla 4. Partes del mensaje (trama) de una transacción de pago con tarjeta.

Nombre de parte	Descripción
NII	Identificador para enrutamiento enrutamiento.
Indicador de tipo de mensaje (MTI)	Longitud de 4 bytes, indica si es un mensaje de autorización, compra, manejo de archivos, reverso o anulación, conciliación, administrativo, entre otros.
Mapa de bits (bitmap)	Longitud de 8 bytes, cada bit indica con 1 o 0 si los campos del mensaje se encuentran presentes o no, respectivamente. El mapa de bit primario indica los primeros 64 campos, el secundario indican la presencia de los campos del 64 al 128.
Campos de mensaje	De longitud variable, contienen la información con los campos indicados en el mapa de bits y corresponden a datos de la transacción entre los cuales se encuentran: fecha de la transacción, hora de la transacción, montos de transacción, números de cuentas, entre otros.

3.1.5. Enrutador de Transacción

Al entrar las transacciones a LA EMPRESA las recibe el hardware dedicado y especializado en el tratamiento de tramas ISO 8583, este tiene la capacidad de descomponer la TRAMA de manera que extrae los datos para enrutar la transacción al siguiente componente, que puede ser el switch transaccional o el core transaccional, como se aprecia en la **Figura 8**.

Este componente tiene su propio sistema y base de datos, donde residen los datos de la transacción que le corresponde analizar, estos serán extraídos, procesados y cargados en la base

de datos para el procesamiento de ML. Por temas de cumplimiento de seguridad y protección de datos sensibles no guarda información relacionada a la tarjeta de pago.

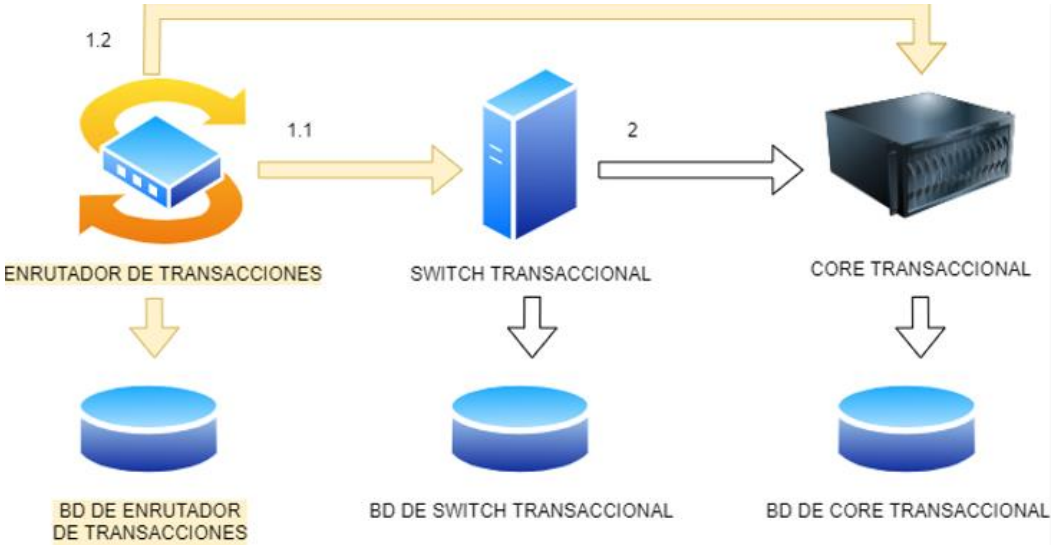


Figura 8. Transacción originada en el enrutador de transacciones a través de los componentes de LA EMPRESA.

3.1.6. Switch Transaccional

Luego de haber sido procesada la transacción por el enrutador de transacciones y enrutada al Switch Transaccional, la trama llega a este sistema que está estrechamente ligado a la marca del terminal físico o POS, incluye la funcionalidad de identificación de datos propietarios incrustados por parte del terminal donde se usa la tarjeta.

Por seguridad, también incluye las llaves para el intercambio la trama, la cual viaja encriptada desde el terminal y es desencriptada para ser enviada al Core Transaccional.

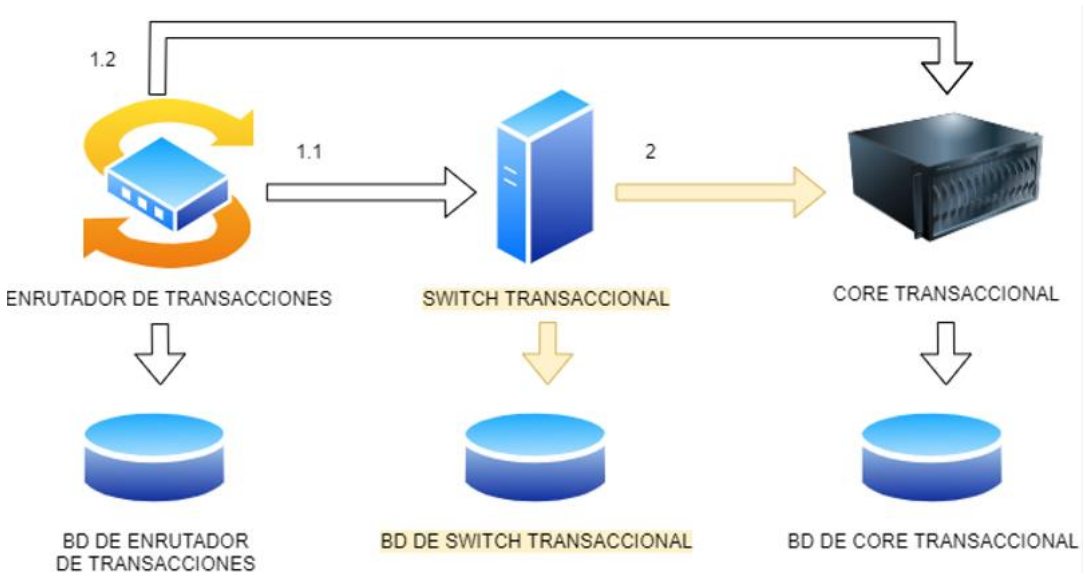


Figura 9. Transacción originada en el switch transaccional a través de los componentes de LA EMPRESA.

Este componente tiene la capacidad de almacenar datos de la transacción de una forma más extendida al enrutador de datos, respetando los lineamientos de seguridad y confidencialidad de estos. Así mismo se encarga de darle el formato ISO 8583 adecuado para que el Core Transaccional, que es el siguiente componente en recibir la TRAMA, sea capaz de entenderla bajo sus estándares definidos.

3.1.7. Core Transaccional

Este es el componente final, como se lo puede apreciar en la **Figura 10**, previo a enviar la trama de una transacción al autorizador, se encarga de aplicar las validaciones y transformaciones propias para cada autorizador dentro del estándar ISO 8583. Tiene todas las configuraciones de conexión y codificación con las que se debe enviar la trama hacia el autorizador.

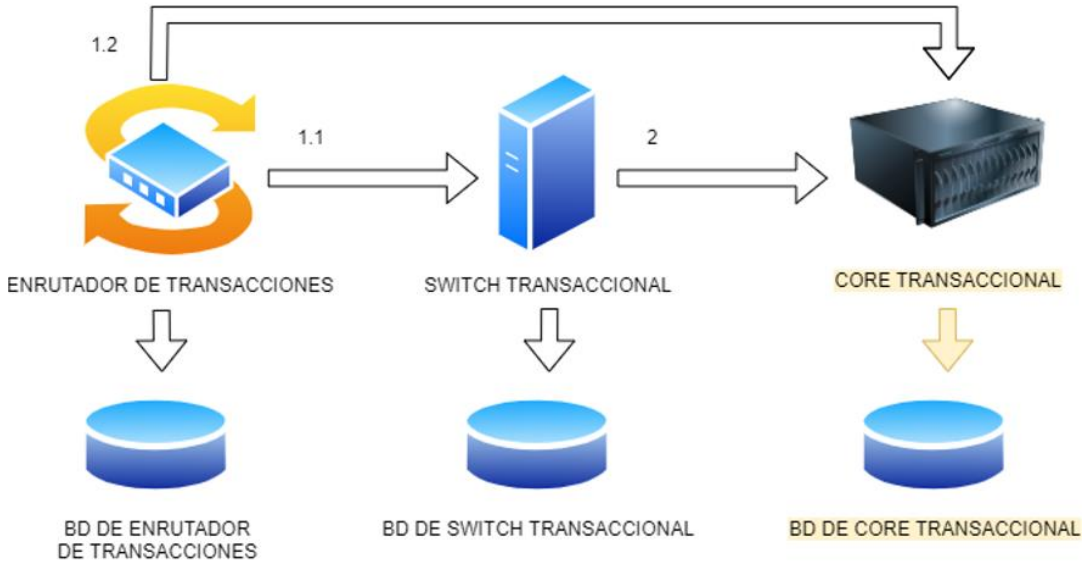


Figura 10. Transacción originada en el *core* transaccional a través de los componentes de LA EMPRESA.

Al igual que los componentes anteriores la seguridad de acceso hacia y desde este servidor es alta, este es el único sistema autorizado por LA EMPRESA como por los autorizadores para realizar el envío de transacción al exterior para aprobación o declinación.

3.1.8. Recursos para revisión de eventos

Cada área dentro del departamento de tecnología es responsable de identificar si el evento que está ocurriendo pertenece a los equipos o sistemas de los cuales son responsables, las áreas son:

- Networking, encargados de los equipos de redes y su comunicación como el enrutador de transacciones.
- Analistas de POS, responsables de la revisión del comportamiento de los terminales físicos o POS de todas las marcas, así como del comportamiento en los Switches Transaccionales de cada marca.

- Analistas de core, son encargados del análisis de tramas transacciones que fluyen por los switches transaccionales y core transaccional en casos de eventos.
- Producción, encargados de los servicios y sistemas que sirven en ambiente de producción y monitoreo respectivo, entre los cuales se encuentra el core transaccional, en caso de eventos sobre este último, son los responsables del análisis y solución.
- Analistas de botón de pago, llamados cuando existen eventos en las transacciones con los terminales virtuales de Datafast.

Cada área tiene un líder encargado dar ejecutar el análisis y soporte en caso de eventos con los recursos designados para este tipo de tareas, es así como se cuenta con 2 recursos de Networking, 2 de Analistas de Core, 1 analista de POS, 1 analista de Botón de Pago de Datafast y 3 para Producción quienes deben analizar los datos que se obtienen desde las aplicaciones, que a su vez las obtienen de las bases de datos de cada componente, para realizar el análisis con los datos que se extraen; este análisis puede llevar minutos, horas o días, dependiendo de la gravedad de evento se puede contar con un tiempos menor o mayor para entregar la solución, resultado del análisis.

3.2. Identificación y recolección de datos

Los datos nacen en cada componente; el flujo de una transacción a través de los componentes inicia en el enrutador de transacciones; luego continua hacia el switch transaccional o hacia el core transaccional; finalmente, en caso de haber ido por el switch transaccional, su siguiente paso será el core transaccional.

Cada componente tiene su propia base de datos, donde se registran los datos que serán utilizados para en análisis, estos serán extraídos con validaciones y condiciones básicas

requeridas que identifiquen los eventos, luego serán cargados en la base de datos donde se aplicará el modelo de clasificación, la **Figura 11** expone los procesos *ETL* de cada base de datos de los componentes.

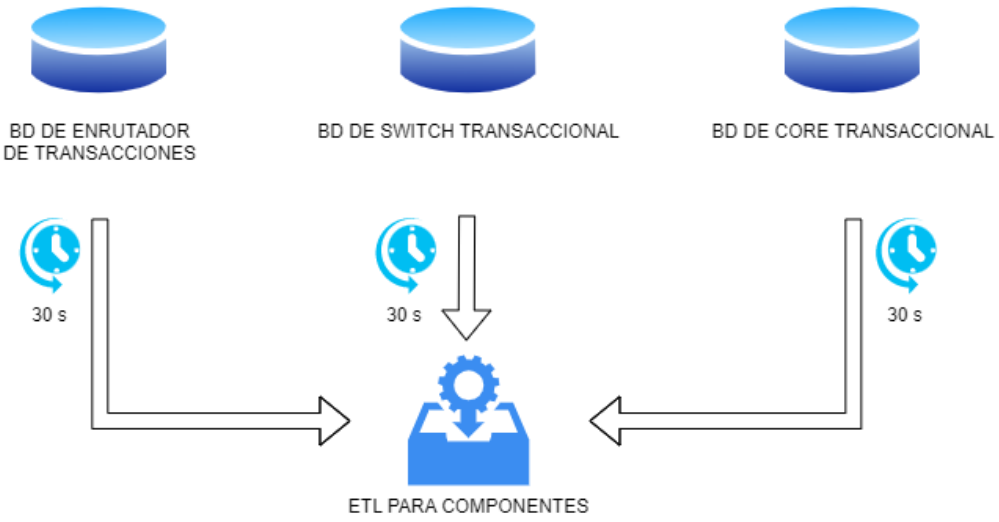


Figura 11. Procesos *ETL* desde los componentes hacia la BD de ML

3.3. Preprocesamiento de datos

Para el preprocesamiento de datos, se cuenta con las bases de datos de cada componente y las bases de datos de los sistemas relacionados que tiene información de: comercios, terminales, información geográfica, entre otros. Los datos son extraídos desde su origen y cargados en la base de datos para el análisis con el modelo de *Machine Learning* (*ML*) a seleccionar. El modelo destino donde residirán los datos es el que se aprecia en la **Figura 12**.

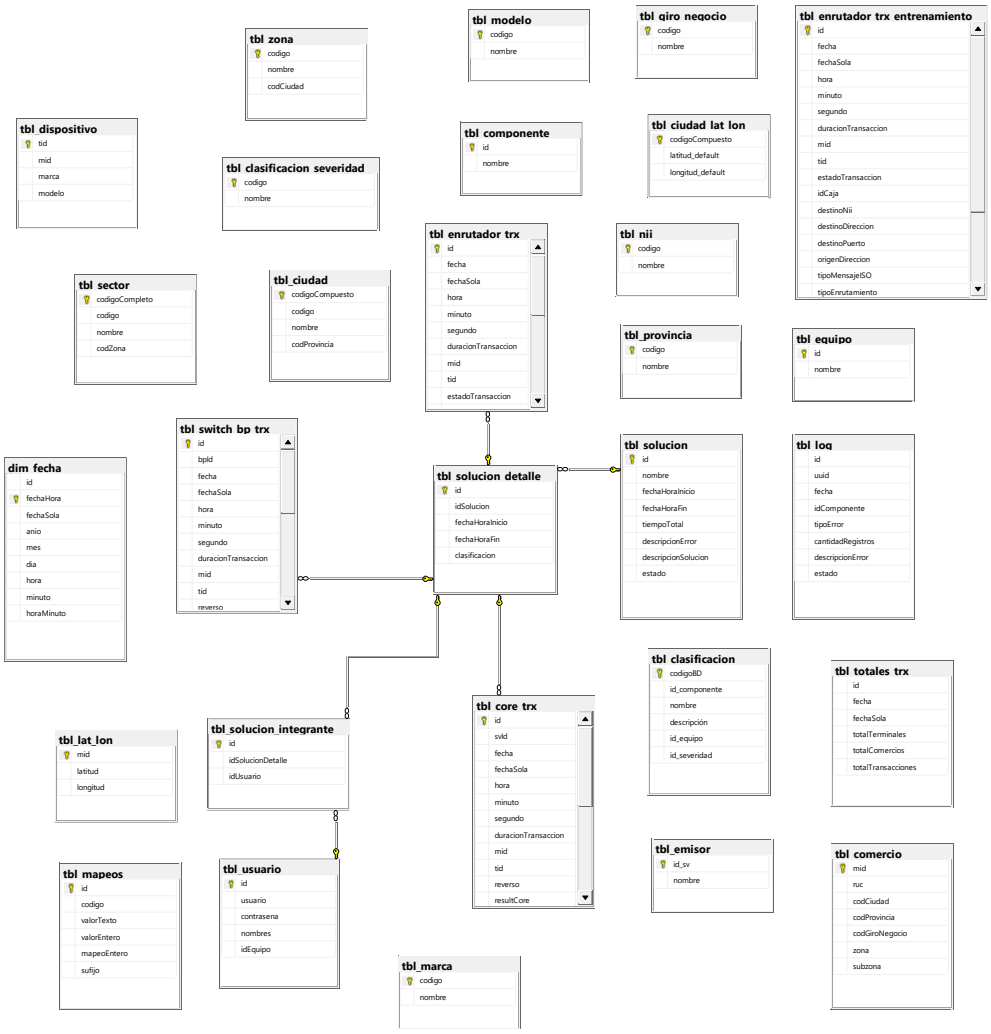


Figura 12. Modelo de base de datos y clasificación de eventos con modelo de ML

3.4. Aplicaciones

Para este proyecto donde se busca seleccionar el mejor modelo que se utilizará en la clasificación de los eventos se contará con software que LA EMPRESA tiene licenciamiento y que no se requiere incurrir en gasto extra, así como aquellos que son de uso libre para las distintas tareas como almacenamiento de datos, extracción, transformación, carga de

información, limpieza de datos, aplicación de los modelos y para la aplicación web son los siguientes:

3.4.1. SQL Server (BD)

Es un Sistema de Administración de Bases de Datos Relacionales (RDBMS), desarrollado por Microsoft, usa lenguaje Transact-SQL, implementación estándar de SQL para operaciones de manipulación de datos (DML) y definición de objetos de datos (DDL).

A pesar de tener 5 ediciones para su uso: Enterprise, Developer, Standard, Express y SQL Azure; en este proyecto se utilizará la edición Standard, ya que se la utilizará para almacenamiento de datos y sus respectivas consultas, en batch y transaccionales.

3.4.2. SQL Server Integration Services (SSIS)

Es un componente o rol de SQL Server con herramientas diversas para la extracción, transformación y carga de datos con distintas fuentes y destinos. Se lo utilizará para extraer la información de las bases de datos de los componentes, transformarlos en el formato adecuado y finalmente, pasar la data a la base de datos destino donde se clasificará cada evento.

3.4.3. Python

Lenguaje de programación de alto nivel multiparadigma, con manejo de programación orientada a objetos, con librerías especializadas en el uso de modelos de Machine Learning. Su sintaxis simplista y fácil de entender hace que la curva de aprendizaje sea máxima en menor tiempo. Se lo utilizará para la aplicación de limpieza de datos y aplicación de modelos de Machine Learning.

3.4.4. Jakarta Server Faces (JSF)

Jakarta Server Faces (JSF) es framework para soportar aplicaciones basadas en Java usando Java Enterprise Edition (JEE) y Java Server Pages (JSP), se lo utilizará para la creación de la aplicación Web para la visualización interactiva de los resultados del modelo de Machine Learning a utilizar.

3.4.5. Power BI

Es la plataforma de Inteligencia de Negocios para análisis de datos de Microsoft, permite la creación de visualizaciones interactivas con capacidad aplicar inteligencia empresarial. Una característica es el consumo y manejo de datos masivos. Cuenta con 7 componentes, que son: Power BI Desktop, Power BI Service, Power BI Mobile App, Power BI Gateway, Power BI Embedded, Power BI Report Server y Power BI Visual Marketplace; para este proyecto se utilizará el componente Desktop para la interacción con el usuario para la visualización de reportes.

3.5. Exploración y validación de datos

Una parte fundamental al momento de implementar cualquier modelo de ML es el análisis y exploración de datos, en este caso se hace necesario identificar características que puedan surgir del análisis primario. Para este es proyecto se decidió utilizar el modelo seleccionado en los registros de todos los eventos con errores que surgen donde se tenga que prestar algún tipo de atención por parte de los recursos de las distintas áreas.

3.6. Modelización del aprendizaje automático

Uno de los objetivos del proyecto para lograr la clasificación de eventos por medio de aprendizaje supervisado. Luego de la etapa de manejo de datos donde se recolectaron datos desde las distintas fuentes, limpieza y carga de datos, viene la etapa de análisis donde se implementaron 3 algoritmos de Machine Learning de manera que se pueda evaluar y comparar los resultados de la precisión de cada uno, para esto se procedió a implementar Árboles de Decisión (DT) (LEE, CHEON, & HWANG, 2022), Máquina de Vector de Soporte (SVM) y Redes Neuronales Artificiales (NN); para esta implementación y análisis se utilizó el componente Core Transaccional.

La preparación previa a la implementación de los algoritmos seleccionados fue estructurada de la siguiente manera (AL-OBEIDAT & EL-ALFY, 2017):

1. Extracción de datos desde las bases de datos del componente, para esta tarea se utilizó la BD del core Transaccional (MILLER, 2017).
2. Creación de libro código de programación en Python, con la importación de librerías como Pandas, Matplotlib y Seaborn a utilizar para el manejo de datos, así como una visualización inicial de la distribución de datos en las distintas características.
3. Carga de datos en el libro de Pyhton, creación de características derivadas de las inicialmente cargadas; asignación de tipos de variables que corresponde al tipo de datos de manera que se haga un uso eficiente de la RAM.
4. Ejecución de un Análisis Exploratorio de Datos (EDA) mediante la codificación de etiquetas de las características, así como la estandarización de datos de manera que se pueda ejecutar un análisis multivariante y verificar la correlación, lo que nos ayuda a identificar las características con mayor representación.

5. Creación de diccionarios de datos para la codificación de etiquetas, estos diccionarios estarán guardados en la base de datos creada para el Sistema de Clasificación de Eventos con ML.
6. Luego de la codificación de etiquetas, se procede con la aplicación de *One Hot Encoding* y *Label Encoding* para un mejor modelamiento de los algoritmos, para esto se usa los valores de diccionarios indicados en el numeral previo.
7. El paso inicial para la implementación de cada algoritmo empieza con la separación del conjunto de datos que se van a utilizar para entrenamiento, validación y pruebas del modelo.
8. Se selecciona un modelo entre los algoritmos DT, SVM y NN; luego de probar varios modelos cada algoritmo queda aquel que tenga la precisión más alta para utilizarse en la comparación de algoritmos de clasificación.
9. Finalmente, de los 3 modelos: 1 de DT, 1 de SVM (HASSANBAKI GARABAGHI, BENZER, BENZER, & CAGLAN GUNAL, 2022) y 1 de NN (SAFFAR & KALHOR, 2023), se selecciona el que tenga la mayor precisión para clasificar los eventos de los componentes Enrutador de Transacciones, Switch Transaccional y Core Transaccional.

3.6.1. Árbol de Decisión (DT)

La implementación del algoritmo de Árbol de Decisión empieza con el uso de un Clasificador de Árbol de Decisión donde el criterio de división está dado por el índice de Gini; luego se utiliza una búsqueda de los mejores parámetros de manera que podemos decidir entre la utilización de 6, 9, 10, 12, 15, 18 nivel de profundidad; mínimo de muestras:3 y calificación: ROC AUC (COLEDANI, ANSELMI, & ROBUSTO, 2023).

3.6.2. Máquina de Vectores de Soporte (SVM)

Para la implementación del algoritmo SVM se utilizaron 3 kernels: Lineal, Polinomial y RBF. En cada kernel se hizo las implementaciones con distintos hiper parámetros llegando a los de mejores resultados luego de varias iteraciones (WATT, BORHANI, & KATSAGGELOS, 2020).

El modelo con mejor precisión fue el que se aplicó kernel RBF, siendo este el seleccionado para la comparación con los algoritmos de DT y NN.

3.6.3. Redes Neuronales (NN)

En la implementación de Redes Neuronales se utilizó la librería Torch para Python con el método funcional para la aplicación de la función de activación de las neuronas entre capas. Los modelos se iteraron con variación en las neuronas de entrada, cantidad de capas, por último, la cantidad de neuronas por capas. El modelo con mejor precisión tiene 28 neuronas de entrada que corresponden a las características, en las parte intermedia u oculta se utilizaron cinco capas y una última capa, la de resultados con 13 neuronas que son la cantidad de clases totales. La función de activación utilizada en todos los modelos fue ReLu (GERÓN, 2019).

3.6.4. Técnicas y modelos multivariantes

Dentro del análisis multivariante se ejecutó un Análisis de Componentes Principales (PCA), llevándolo a las distintas dimensiones, en la siguiente tabla se puede apreciar cómo el porcentaje de representación va aumentando a medida que se usa mayor cantidad de componentes. Se obtuvo que se requiere un mínimo de 13 dimensiones para tener una representación de más del 95%.

3.7. Presentación de datos en Power BI

La EMPRESA trabaja actualmente con Inteligencia de Negocio (BI), para esto posee licencias de Power BI que fueron adquiridas con anticipación, este es usado por las distintas áreas para la visualización de paneles con datos proveniente de los cubos modelados por al área de BI. Al ser una herramienta práctica y de excelente funcionalidades en la presentación datos y manipulación de dimensiones, además de contar con las licencias, es la aplicación seleccionada para la visualización de datos. El departamento de NOC, que opera 24 x 7, de LA EMPRESA será el encargado de utilizar los paneles con la información del monitoreo.

CAPÍTULO 4

4. Análisis De Resultados

4.1. Análisis exploratorio de datos

LA EMPRESA abarca muchos parámetros de medición, entre los cuales están: la cantidad de transacciones exitosas, es decir que cumple su flujo completo de ida y regreso, entre los terminales físicos o virtuales y los autorizadores; la cantidad de transacciones que se procesan por segundo.

Estas dos mediciones serán las principales referencias para utilizarse en el cálculo del mejoramiento del servicio.

4.1.1. Exploración de datos

La empresa está consciente que el lapso de mayor concentración de transacciones es a partir de las 12:00 hasta las 18:00, generalmente agrupan el 50% de las transacciones de un día, siendo estos el primer y tercer cuartil; se lo puede apreciar en la **Figura 13**.

Figura 13. Transacciones por segundo promedio de los 3 componentes.

La **Figura 14** nos muestra otro dato que también se tiene conocimiento, alrededor del 75% de la cantidad de transacciones recae sobre 5 ciudades, de mayor a menor tenemos las siguientes ciudades: Quito, Guayaquil, Cuenca, Samborondón y Daule.

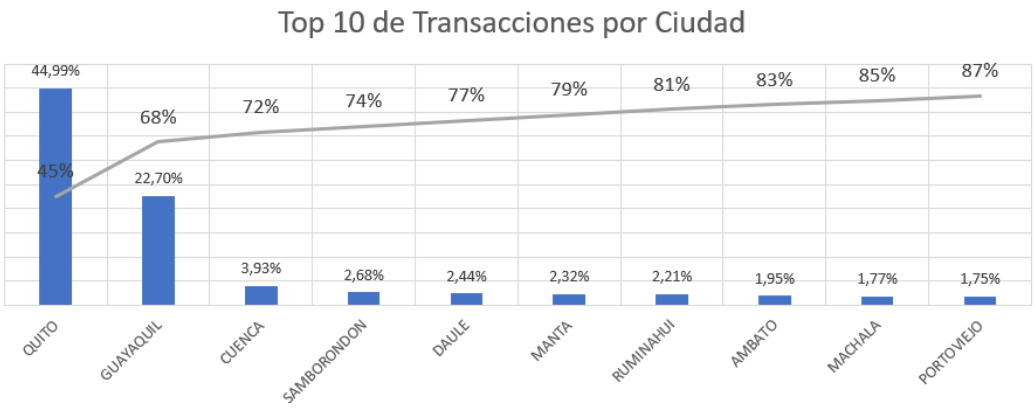


Figura 14. Porcentajes transaccionales en las mayores 10 ciudades de Ecuador.

La cantidad de transacciones por Autorizador es otro dato sobre el cual se sabe que las cifras van a ser generalmente constante, donde el Autorizador 1 abarca casi un promedio de 48% de las transacciones totales por día, como se observa en la **Figura 15**, es decir casi la mitad de las transacciones que pasan por La EMPRESA, se redirigen hacia este. También se sabe que entre las 4 primeros autorizadores llegan al 98% de transacciones en promedio cada día.

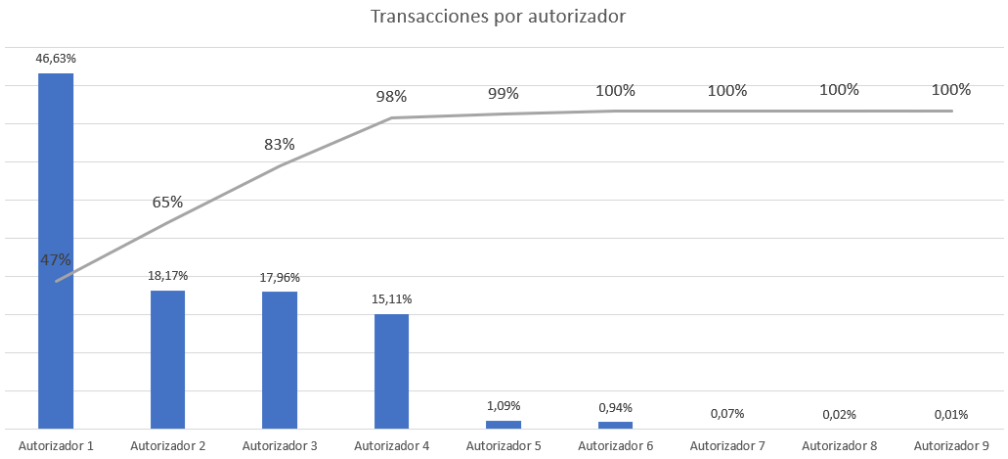


Figura 15. Cantidad de transacciones por autorizador.

4.1.2. Exploración de datos de Enrutador de Transacciones

Con los datos proporcionados por la matriz de correlación que se observa en la **Figura 16**, se visualiza que, en el Enrutador de Transacciones las características con mayor correlación son: Estado de Transacción, Tipo de Mensaje ISO y Códigos de Procesamiento ISO.

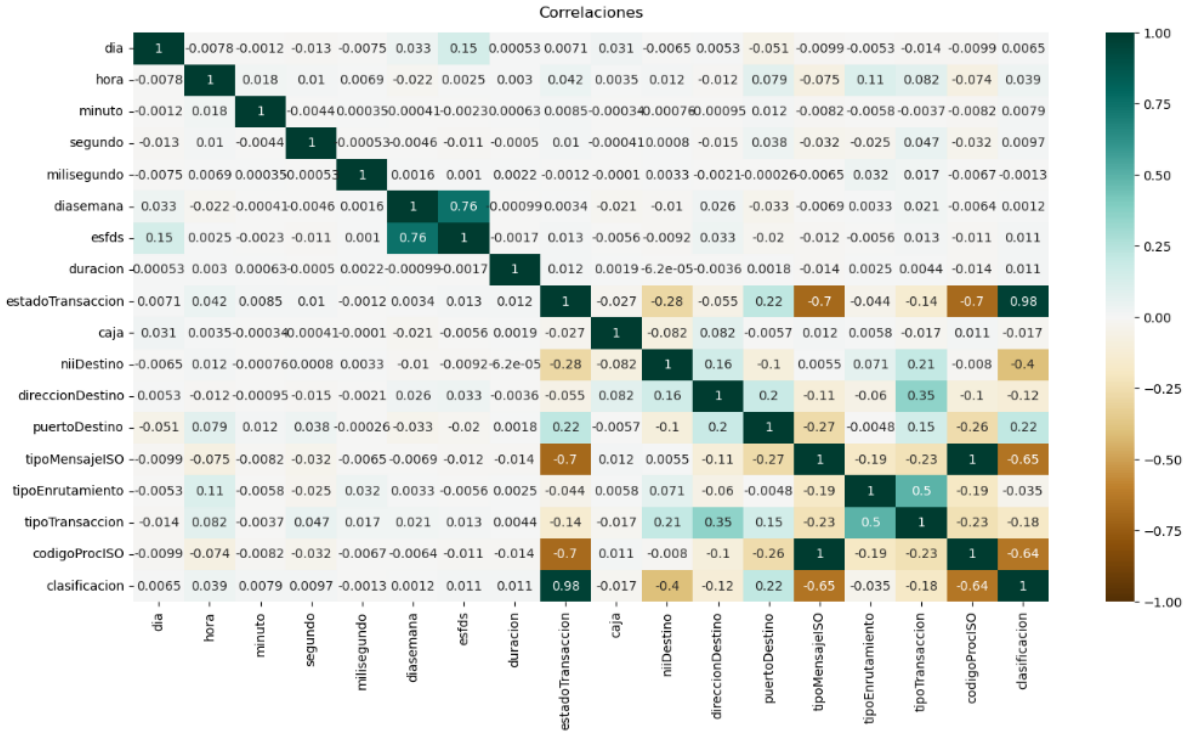


Figura 16. Matriz de correlación de las características del Enrutador de Transacciones

Se tomaron como muestra los datos de enero, febrero y marzo de 2023, de manera que podamos conocer los porcentajes de cada Estado de Transacción, que es la respuesta que el enrutador tiene para cada transacción entrante, en la **Figura 17** se observa que el código 54 “Conexión establecida, no envía mensaje de transacción en la petición” la respuesta que abarca más de la mitad de todas las transacciones de la muestra.

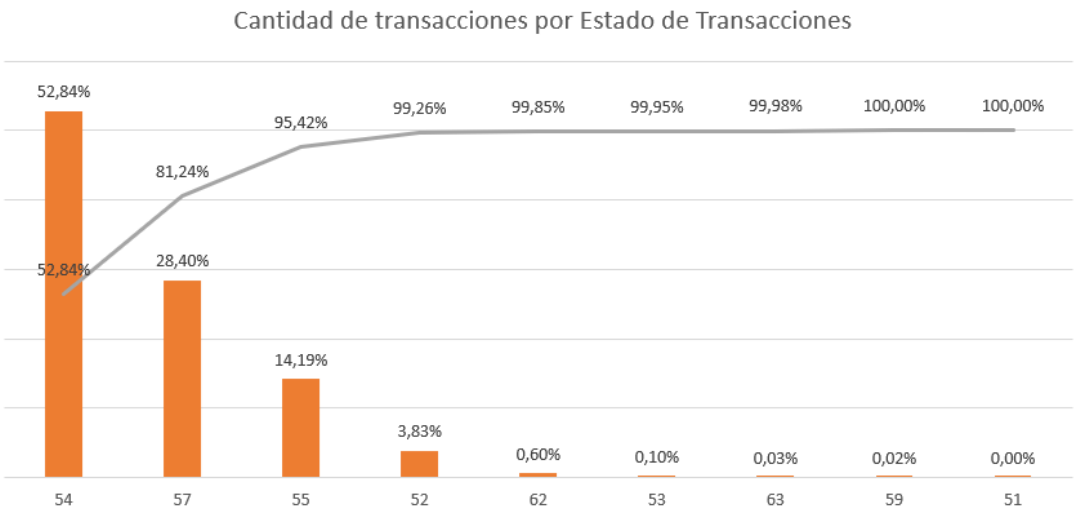


Figura 17. Porcentajes de transacciones de enero a marzo / 2023 de los Estados de Procesamiento.

Otro dato importante dentro de las características que se observa en la **Figura 18** es la alta correlación del Código de Procesamiento ISO; nos indica el tipo de transacción que se va a realizar, los valores están entre efectivo o los múltiples tipos de crédito que existen. Podemos observar que el porcentaje más alto de las transacciones con errores son aquellas que no tiene definido el Código de procesamiento.

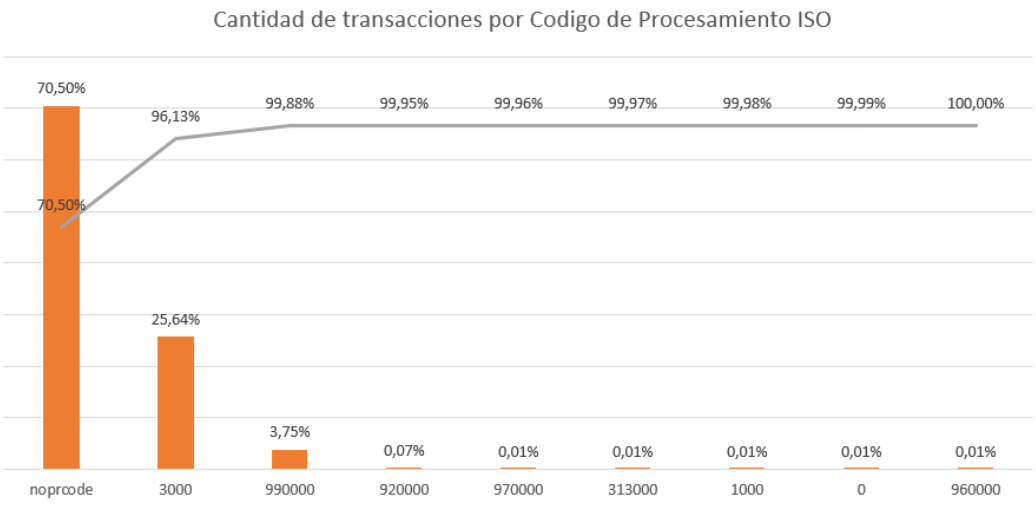


Figura 18. Porcentajes de registros de enero a marzo / 2023 de los Códigos de Procesamiento.

4.1.3. Exploración de datos de Switch Transaccional

En la **Figura 19** podemos observar que las características del Switch Transaccional con mayor correlación entre sí son: Duración, Resultado del Switch y Resultado del Emisor.

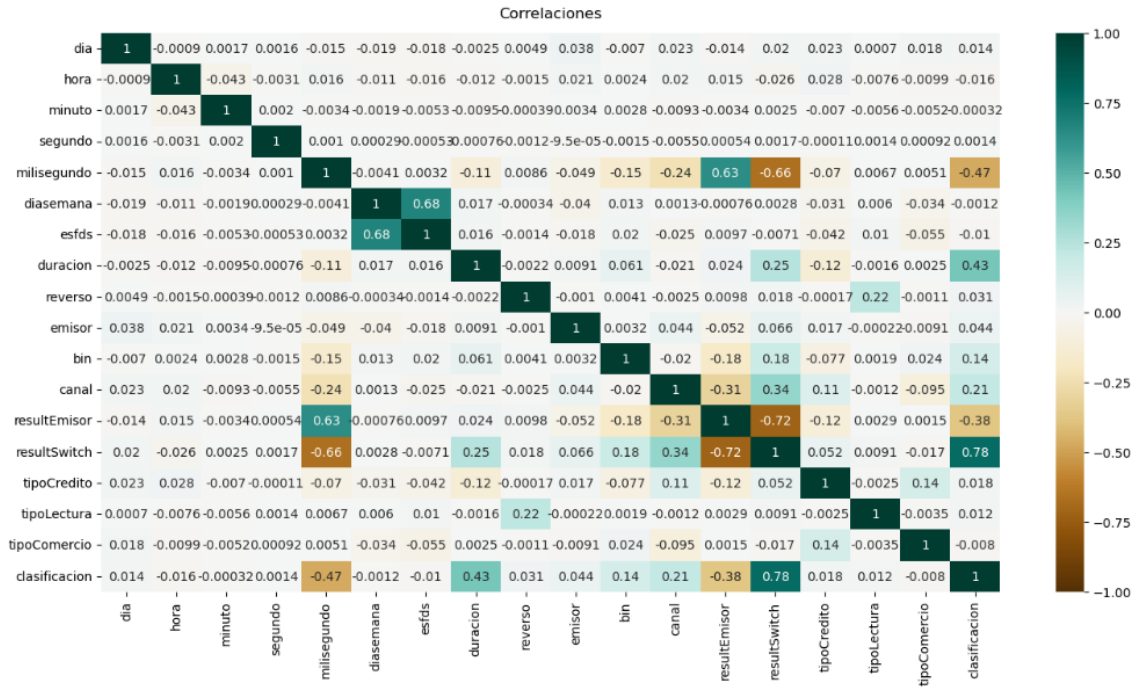


Figura 19. Matriz de correlación de las características de Switch Transaccional

El lapso de análisis para los datos fue de enero, febrero y marzo de 2023, el porcentaje de cada Resultado de Switch se lo puede apreciar en la **Figura 20**, siendo el que tiene código 000.200.000 - Transacción OTP esperando por respuesta el que representa el 61% del total de transacciones en este componente; también que, de un total de 33 posibles códigos de respuesta, 20 representan el 98% del total de respuestas.

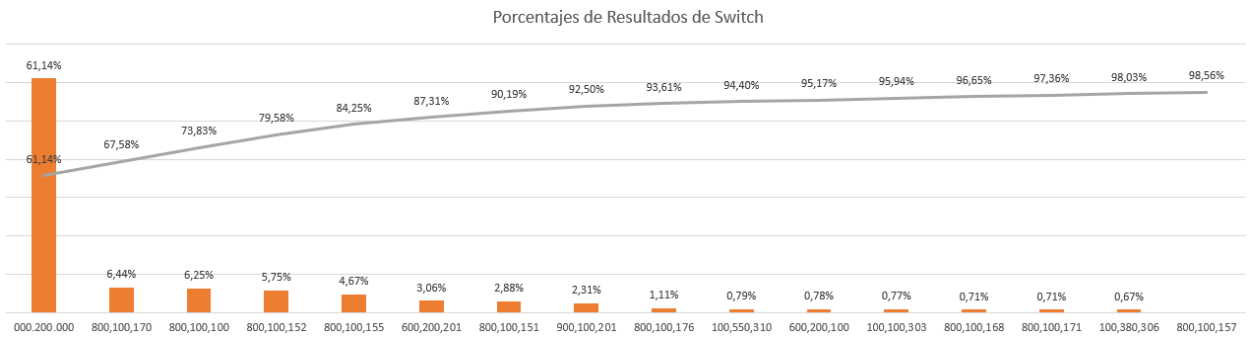


Figura 20. Porcentajes de registros de enero a marzo de 2023 para los Resultado de Switch Transaccional.

4.1.4. Exploración de datos de Core transaccional

De la misma manera que se lo hizo con el Enrutador de Transacciones y Switch Transaccional, se ejecutó el análisis de correlación de las características provistas por el Core Transaccional; en la **Figura 21** se distingue que todas las características tienen una correlación muy baja, siendo Standin y Emisor o Autorizador las más altas.

Figura 21. Matriz de correlación de las características de Core Transaccional

La muestra seleccionada para los datos del Core corresponde a los meses enero y febrero del año 2023, en este caso se ha tomado la característica Emisor, visible en la **Figura 22**, es la institución que responde si la transacción fue autorizada o declinad, como ya fue mencionado, el autorizador 1 tiene la mayor cantidad de transacciones.

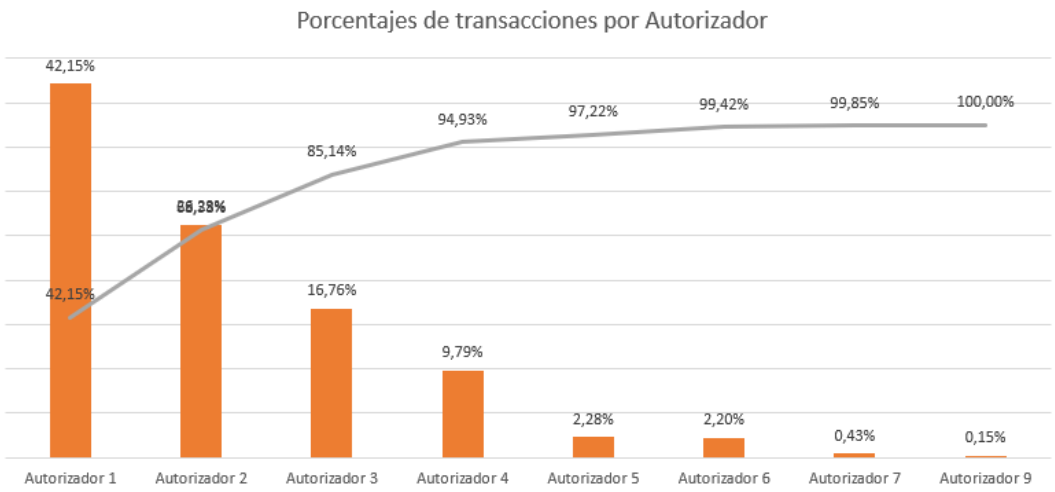


Figura 22. Porcentajes de los registros por autorizadores de enero y febrero del año 2023.

4.2. Análisis de Componentes Principales (PCA)

Los resultados del PCA, aplicado a los datos de entrenamiento del Enrutador de Transacciones, que contienen 1’120.821 muestras y 18 características indica que con 13 componentes pasamos el 95%, lo cual es aceptable; en la **Tabla 5** se puede apreciar el porcentaje de representación para cada calculo donde se varía la cantidad de componentes.

Tabla 5. Resultados de la representación en base a la cantidad de componentes del PCA.

Cantidad de componentes	Representación por cada componente	Representación total de componentes (Varianza total explicada)
2	[19.23649193, 13.32880611]	3.57 %
3	[19.23649246, 13.32880593, 9.8636984]	42.43 %
4	[19.23649246, 13.32880582, 9.86369866, 7.75003298]	50.18 %
8	[19.23649246, 13.32880624, 9.86369975, 7.75003804, 6.4523462, 5.63945157, 5.56755968, 5.55978377]	73.40 %
13	[19.23649246, 13.32880624, 9.86369975, 7.75003804, 6.4523462, 5.63945157, 5.56755968, 5.55978377, 5.54074703, 5.47153946, 5.2591623, 4.72139353, 2.11973529]	96.51 %

4.3. Evaluación de los 3 modelos

Una vez seleccionado el modelo con la mejor precisión de cada algoritmo, se procede a comparar los resultados de las distintas ejecuciones.

4.3.1. Arboles de Decisión: Análisis de rendimiento, precisión y matriz de confusión

El primer algoritmo implementado fue Arboles de Decisión (DT), los resultados de precisión de las iteraciones sobre los distintos modelos se detallan en la tabla a continuación.

Tabla 6. Lista de resultados de la precisión con el modelo seleccionado en el algoritmo de Árboles de Decisión.

Tipo	Muestra	Precisión
Validación	142.677	99.9005%

Prueba	118.897	99.9008%

En la matriz de confusión de la **Figura 23**, se puede apreciar que la precisión por cada clase está al 100% en 11 de 13 clases y que en las dos restantes la imprecisión es un número muy pequeño para la cantidad de registros de cada clase.

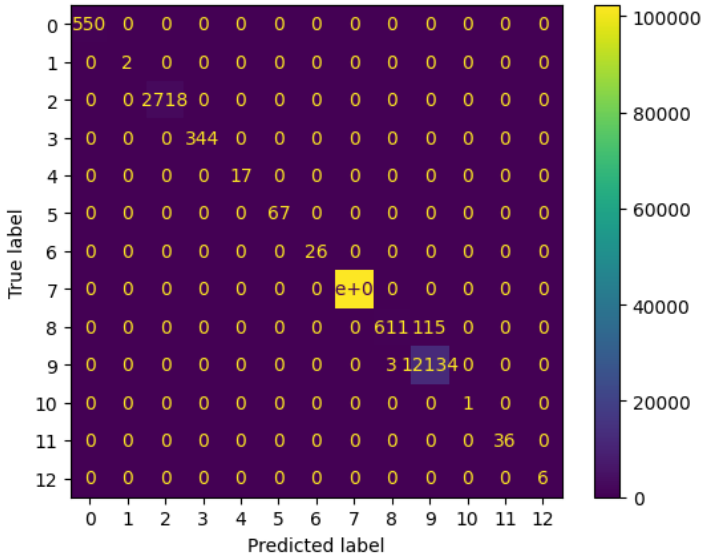


Figura 23. Matriz de confusión de la predicción del modelo seleccionado para el algoritmo de Arboles de Decisión.

Las métricas de errores: Raíz del Error Cuadrático Medio (RMSE), Error Cuadrático Medio (MSE) y Error Absoluto Medio (MAE), confirman la alta precisión arriba obtenida.

Tabla 7. Métricas del modelo seleccionado para el algoritmo de Arboles de Decisión.

Métrica	**Valor**
MAE	0.000992
MSE	0.000992
RMSE	0.031503

El modelo fue seleccionado a través de la búsqueda mediante la combinación de hiper parámetros con GridSearchCV, obteniendo el modelo con la mejor precisión, se lo puede apreciar en la **Figura 24**.

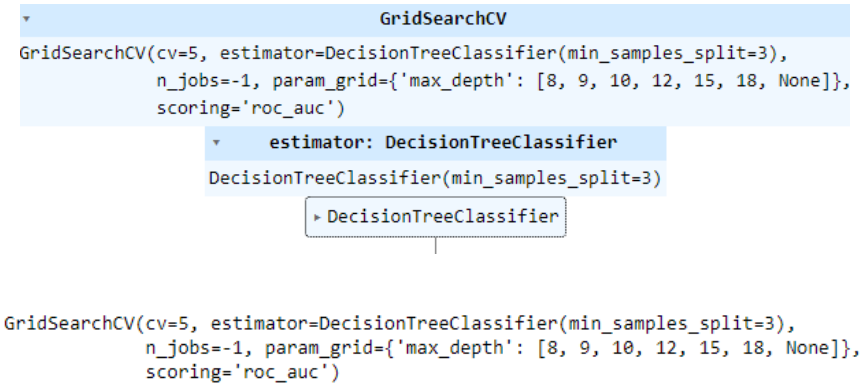


Figura 24. Búsqueda de modelo con el algoritmo Arboles de Decisión.

El modelo seleccionado del Árbol de Decisión seleccionado lo podemos apreciar en la **Tabla 8**.

Tabla 8. Modelo seleccionado de Árbol de Decisión de Clasificación.

Parámetro	Valor
Costo de complejidad Alpha	0.0
Medida de impureza	Gini
Profundidad máxima	8
Características máximas	No
Mínimo contenido de hoja	1
Máxima contenido de hoja	3
Estrategia de división	Mejor

Con la función export_graphviz de la librería de sklearn.tree se obtiene el gráfico del árbol de decisión con sus parámetros en cada nodo, en la **Figura 25** se puede apreciar la profundidad de 8 niveles con las distintas ramas. Para mayor detalle ver del **Anexo 4** al **Anexo 9**.

Figura 25. Gráfico del modelo seleccionado de Arboles de Decisión.

4.3.2. Máquina de Vectores de Soporte

El segundo algoritmo implementado fue Máquina de Vectores de Soporte (SVM), los resultados de precisión se detallan en la **Tabla 9**.

Tabla 9. Lista de valores de precisión obtenidos en los conjuntos de datos de validación y prueba.

Tipo	Muestra	Precisión
Validación	142.677	89%
Prueba	118.897	88.5%

En la matriz de confusión, de la **Figura 26**, se puede apreciar que la precisión está dada por acertar en la clase cuatro en un número muy elevado de registros, pero la imprecisión en el resto de las clases hace que la clasificación no sea tan confiable.

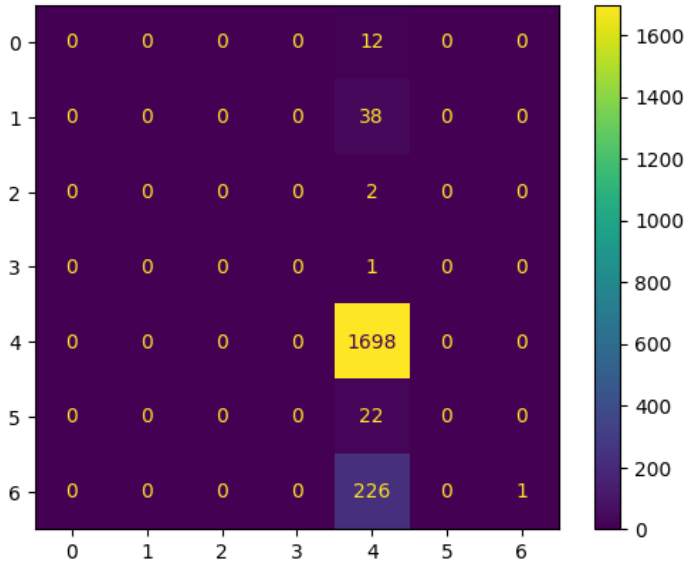


Figura 26. Matriz de confusión de la predicción del modelo seleccionado para el algoritmo de Máquina de Vectores de Soporte.

Las métricas de errores: MAE, MSE y RMSE confirman los resultados con baja precisión que se obtuvieron en la ejecución de la predicción.

Tabla 10. Métricas del modelo seleccionado para el algoritmo de Máquina de Vectores de Soporte.

Métrica	**Valor**
MAE	0.379
MSE	1.250
RMSE	1.118

El modelo con la mejor precisión fue seleccionado luego de ejecutar los entrenamientos y pruebas en el algoritmo de Máquina de Vectores de Soporte. El detalle de los resultados se lo puede apreciar en la **Tabla 11**.

Tabla 11. Lista de los mejores modelos para cada kernel en entrenamiento con SVM.

Kernel	Hiper parámetros	Tiempo de entrenamiento	Precisión
Lineal	'C': 1.0, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'degree': 3, 'gamma': 'scale', 'kernel': 'linear', 'shrinking': True, 'tol': 0.001, 'verbose': False	Se llegó a un máximo de 24 horas y el proceso no terminaba.	Indeterminada
Polinomial	'C': 1.0, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'degree': 5, 'gamma': 'scale', 'kernel': 'poly', 'random_state': 30, 'shrinking': True, 'tol': 0.001, 'verbose': False	Aproximadamente 2 horas por iteración de cada modelo.	84.95%
RBF	'C': 100.0, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'degree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'random_state': 30, 'shrinking': True, 'tol': 0.001, 'verbose': False	Aproximadamente 1 hora por iteración de cada modelo.	89.00%

El modelo seleccionado para SVC es el que se observa en la **Tabla 12**.

Tabla 12. Modelo seleccionado para algoritmo de Máquina de Vectores de Soporte.

Parámetro	Valor
Parámetro de regularización C	100
Valor gamma	Scale

Función de decisión	OVR (para n clases)
Kernel	RBF
Tolerancia para detención	1.0

4.3.3. Redes Neuronales: Análisis de rendimiento, precisión y matriz de confusión

El tercer algoritmo implementado fue Redes Neuronales (NN), de los tres modelos usados para el entrenamiento, se seleccionó uno. El resultado de precisión del modelo seleccionado se detalla en la siguiente tabla**.**

Tabla 13. Resultados de precisión del modelo seleccionado del algoritmo NN.

Tipo	Muestra	Precisión
Prueba	118.800	86%

Luego de la ejecución del entrenamiento y aplicación del modelo en los datos de pruebas, se obtuvieron los resultados de precisión para cada modelo, de entre los cuales se seleccionó el que tiene la mayor precisión. Los detalles se los puede apreciar en la **Tabla 14**.

Tabla 14. Lista de los mejores modelos para cada entrenamiento con NN.

Modelo	Hiper parámetros	Tiempo de entrenamiento	Precisión
Entrada: 28 neuronas Capas Ocultas: 1 Salida: 13 clases Dropout: 25%	n_epochs: 200 sgd_lr: 10, 1.0, 0.5, 0.05	Aproximadamente 2 horas por cada entrenamiento.	0%
Entrada: 28 neuronas	n_epochs: 200	Aproximadamente 6	40.70%

Capas Ocultas: 3 Salida: 13 clases Dropout: 25%	sgd_lr: 1.0, 0.5, 0.05	horas.	
Entrada: 28 neuronas Capas Ocultas: 5 Salida: 13 clases Dropout: 25%	n_epochs: 200 sgd_lr: 0.5, 0.05	Aproximadamente 12 horas.	86.00%

El modelo seleccionado se lo puede apreciar en la **Tabla 15**.

Tabla 15. Modelo seleccionado para entrenamiento de algoritmo de Redes Neuronales

Capa	Tipo	Entrada	Salida	Dropout
1	Entrada	28	26	25%
2	Interna	26	26	25%
3	Interna	26	26	25%
4	Interna	26	26	25%
5	Interna	26	26	25%
6	Interna	26	26	25%
7	Salida	26	13	

4.4. Mediciones del Negocio

Para el análisis y comparación de los resultados que están ligados al negocio se utilizarán la cantidad de eventos por mes, cantidad de recursos, horas por recurso utilizadas para el análisis y representación económica por la falta de servicio, esta última se representa con la suma de la

cantidad de transacciones por segundo (TPS) que ocurrieron en el evento más las multas en las que pueda incurrir LA EMPRESA por la ausencia de servicio.

4.4.1. Cantidad de transacciones por segundo

La principal medición para el negocio, referente a la transaccionalidad, son las transacciones por segundo (TPS), este parámetro es utilizado tanto para los cálculos monetarios como para la parte técnica. Se tomará este parámetro para representar la tasa transaccional (RTT) de LA EMPRESA del tiempo que no tenga los servicios activos.

$$\boldsymbol{RTT} = TiempoSinServicio * TPS \quad (1)$$

En la **Figura 27** se puede observar el comportamiento de las TPS en el lapso de 24 horas.

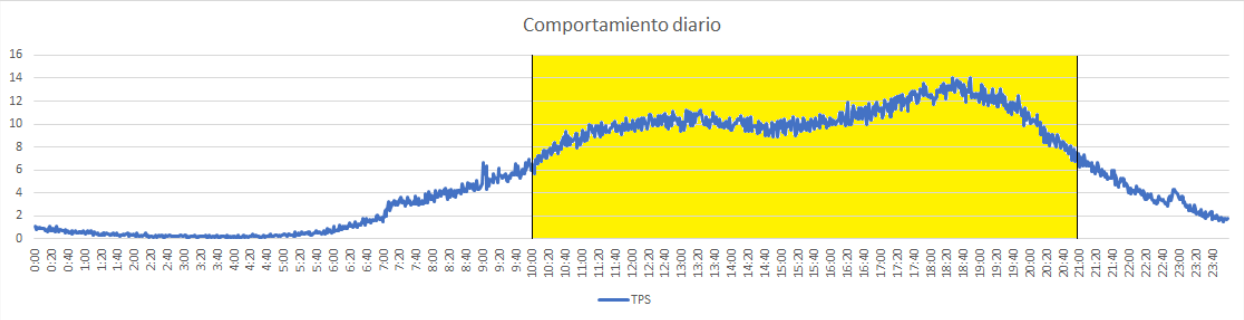


Figura 27. Comportamiento promedio de TPS por día

4.4.2. Representación Monetaria

La parte monetaria es la parte que mueve todos los procesos en LA EMPRESA, el conocer cuanto representa cada segundo que el servicio de transaccionalidad no esté activo. Esta parte se divido en ingreso y egresos; el primero se lo obtiene por las TPS que fluyan por LA EMPRESA, el segundo se da por el tiempo que no fluyan las transacciones por LA EMPRESA junto a las multas que se puede obtener de parte de las entidades gubernamentales de control si

es que el tiempo donde no se esté dando servicio pase de 1 hora por responsabilidad de LA EMPRESA.

$$\boldsymbol{Pérdida\ Monetaria} = RTT * ValorMonetario \quad (2)$$

4.4.3. Recursos y horas

Para cada evento se tiene que recurrir a los equipos especialistas de las distintos componentes por donde se enrutan las transacciones, se dispone de 3 recursos de Networking, 2 recursos de Analistas de Core y 2 recursos para el Core Transaccional. Para calcular las horas-hombre, se tomará en cuenta la suma de horas multiplicada por la cantidad de recursos que se activan en los eventos.

$$\mathbf{Horas - Hombre} = 6 \text{ recursos / evento} \quad (3)$$

4.4.4. Cantidad de eventos por mes

Para tener una estimación de la cantidad de eventos se tomarán los datos de la bitácora global de eventos sucedidos a lo largo de un año y se solicitará al experto de cada área indicar el tiempo promedio utilizado para resolverlos. Este parámetro de lo utilizará para contar una tasa de eventos por mes (EPM).

$$\boldsymbol{EPM} = \frac{TotalEventosAño}{12} \quad (4)$$

4.4.5. Cálculo de un evento

En el momento que sucede un evento donde se presentan problemas masivos, llegando a la pérdida de transaccionalidad, se activa el soporte de los recursos de las distintas áreas. Vamos a tomar los valores de un evento promedio para conocer la representación monetaria en dos

escenarios: ambiente actual sin el clasificador de eventos y ambiente con el clasificador de eventos. El detalle de las cifras que implican la atención se puede apreciar en la **Tabla 16**.

Tabla 16. Diferencia de la representación monetaria de un evento promedio.

SIN CLASIFICADOR DE EVENTOS	**CON CLASIFICADOR DE EVENTOS**
RTT (1 hora) = 1 hora * 12 tps RTT (1 hora) = 3600 s * 12 tps RTT (1 hora) = 46200 transacciones	RTT = 10 m + 10 s * 12 tps RTT = 610 s * 12 tps RTT = 7320 transacciones
PM = 43200 trx * $ 0,005 / trx PM = $\frac{\$\ 216}{\text{hora}} = \frac{\$\ 0{,}06}{\text{segundo}}$	PM = 43200 trx * $ 0,005 / trx PM = $\frac{\$\ 37}{\text{hora}} = \frac{\$\ 0{,}01}{\text{segundo}}$
PMR = 6 recursos * $ 8 hora / recurso PMR = $\frac{\$\ 48}{\text{hora}}$	PMR = 6 recursos * $ 1,33 minuto / recurso PMR = $\frac{\$\ 8}{\text{hora}}$
PMT = $ 216 / hora + $ 48 / hora PMT = $\frac{\$\ 264}{\text{hora}}$	PMT = $ 37 / hora + $ 8 / hora PMT = $\frac{\$\ 45}{\text{hora}}$

Como se puede apreciar en el detalle de las cifras y valores monetarios, la pérdida de valores monetarios de un escenario donde se cuenta con el clasificador de eventos versus el escenario donde no se cuenta con el clasificador de eventos tiene una relación de 5.8 a 1, lo que representa una reducción de 82.95%.

4.5. Visualización de resultados de clasificación para cada modelo.

Finalmente, como resultado de la aplicación del algoritmo seleccionado se puede contar con la clasificación de los eventos en el panel de monitoreo. Se lo ha dividido en 3 páginas: General, Clasificador y Soluciones.

En la página General, que se puede apreciar en la **Figura 28**, se presentan los datos de los eventos con las distintas dimensiones que actualmente se usan en el monitoreo diario de LA EMPRESA, mostrando ciertos datos de todas las áreas que son llamadas antes un evento. Se tienen datos como: cantidad de eventos por hora, transacciones por bines, transacciones por ciudades, transacciones por componente, cantidad de eventos por ubicación geográfica, entre otros.

Figura 28. Datos de eventos de los 3 componentes del flujo transaccional.

En la página Clasificador, mostrada en la **Figura 29**, se presentan los eventos clasificados, obtenidos mediante la aplicación del modelo seleccionado y entrenado; la clasificación de cada evento se muestra en los 3 componentes: Enrutador de transacciones,

Switch Transaccional y Core Transaccional; también se incluyen datos extras como: Lapsos sin clasificar, cantidad de transacciones sin clasificar, fecha inicial sin clasificar y última fecha sin actualizar. Cada clasificación está directamente relacionada con un grupo de recursos y un nivel de severidad.

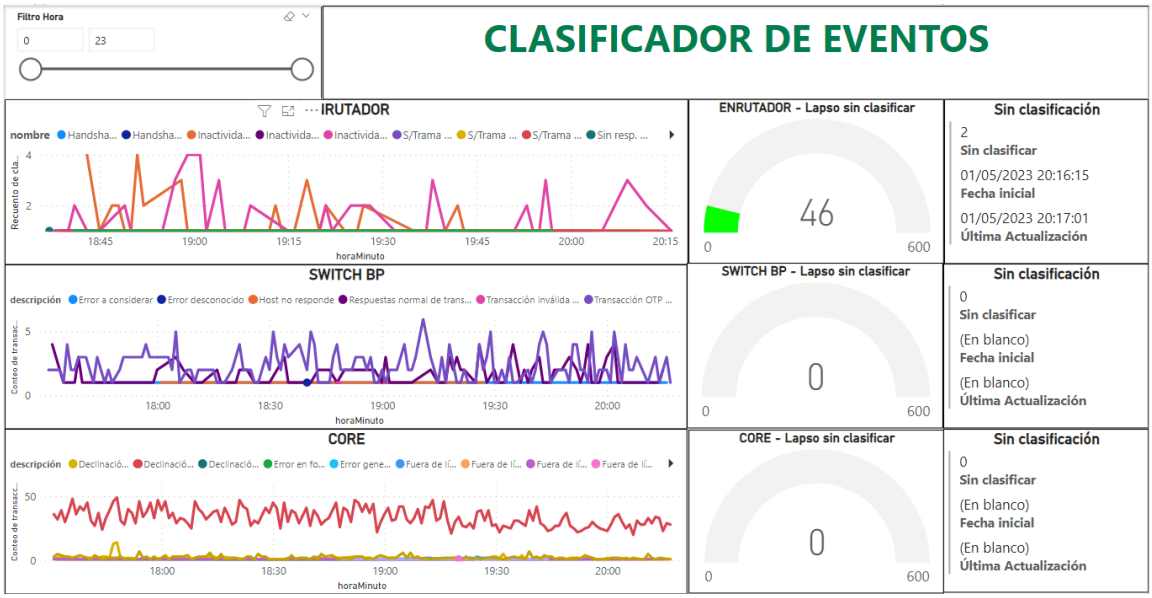


Figura 29. Clasificador de eventos con ML para los componentes Enrutador de Transacciones, Switch Transaccional y Core Transaccional.

Finalmente, en la tercera página se tiene Soluciones, **Figura 30**,se tiene las acciones que fueron ejecutadas por los recursos a los que se les asignaron la revisión, aquí se puede apreciar el nombre que se le dio al evento, fecha desde que se reportó el incidente, fecha final de la revisión, comentarios de las recursos, recursos que intervinieron en la revisión, soluciones aplicadas, entre otros datos. Estos datos, serán tomados en futuros cálculos para medir la precisión y rendimiento de la herramienta desarrollada mediante la comparación del resultado del análisis y la clasificación dada por el algoritmo.

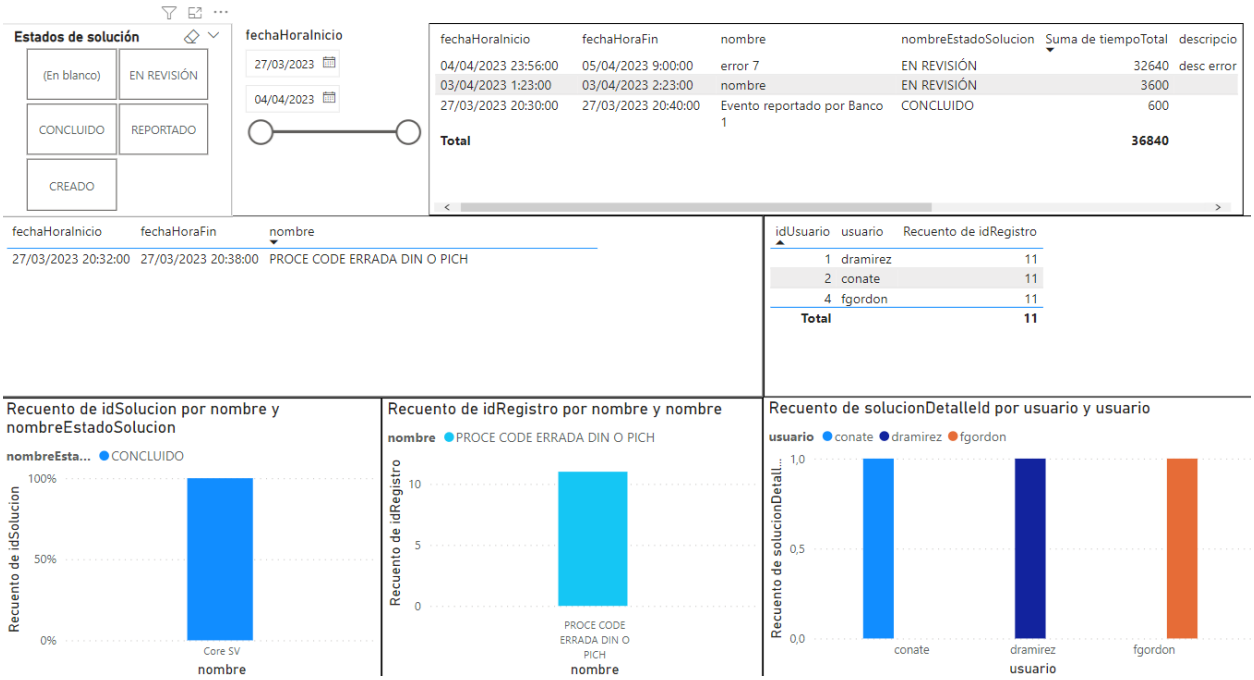


Figura 30. Soluciones aplicadas por los recursos a los eventos presentados.

5. CONSLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

La ejecución del proyecto incluía realizar el análisis de los 3 algoritmos: DT, SVM y NN, en los 3 componentes para comparar el rendimiento de cada uno y aplicar el que mejor resultado tenga en cada componente; más el esfuerzo, tiempo y recursos necesarios para aplicar estos algoritmos en el primer componente fue muy alto, por lo que se decidió usar el algoritmo seleccionado de este en los otros dos componentes.

El tratarse de componentes que manejan datos similares, la precisión del modelo seleccionado en el Clasificador de Arboles de Decisión resultó ser alta en los 3 componentes.

El uso de una clasificación categórica permitió tener valores granulares, dando paso a subclasificaciones donde se puede identificar nivel de severidad del evento, recurso que tiende el evento o equipos de recursos por cada componente.

El haber creado un modelo de base de datos que abarque la configuración de valores para todas las características, así como sus respectivas etiquetas en la técnica de Label Encoding, ayuda a redireccionar de forma dinámica la clasificación de eventos, ya sea en eventos existentes o eventos no considerados.

5.2. Recomendaciones

Se recomienda seguir probando nuevos modelos de los algoritmos ya probados, así como la aplicación de nuevos algoritmos de clasificación para ampliar el abanico de posibilidades de la clasificación de eventos.

Con los eventos ya clasificados y con la identificación de recursos a asignar para el análisis, se recomienda agregar el envío de notificaciones a los involucrados.

Al ser esta, una herramienta de monitoreo, se recomienda mejorar la visualización de alertas con severidad Alta.

Analizar la creación de una aplicación que pueda automatizar acciones de atención primer nivel, como: reinicio de servicios en los componentes afectados, extracción de logs de los componentes,

Finalmente, se recomienda actualizar los modelos es una frecuencia mensual; lapso definido en LA EMPRESA como tiempo estándar para cambios aplicaciones de monitoreo en producción.

6. REFERENCIAS

7. AL-OBEIDAT, F., & EL-ALFY, E. (Octubre de 2017). Hybrid multicriteria fuzzy classification of network traffic patterns, anomalies, and protocols. *Pers Ubiquit Comput*, 15. doi:10.1007/s00779-017-1096-z

8. AMAT RODRIGO, J. (Febrero de 2017). *Ciencia de datos*. Obtenido de https://www.cienciadedatos.net/: https://www.cienciadedatos.net/documentos/33_arboles_decision_random_forest_gradient_boosting_c50

9. BONACCORSO, G. (2018). *Machine Learning Algorithms* (2 ed.). Packt Publishing Ltd.

10. BOSLAUGH, S. (2012). *Statistics in a Nutshell* (2 ed.). O'Reilly Media, Inc.

11. C.H., S., POKALA, P. K., BOLISETTI, R., & BALASUBRAMANI, S. (2022). Analysis of Credit Card Fraud Detection using Machine Learning Techniques. *ICCES*, 5. doi:10.1109/ICCES54183.2022.9835751

12. COLEDANI, D., ANSELMI, P., & ROBUSTO, E. (Febrero de 2023). Machine learning-decision tree classifiers in psychiatric assessment: An application to the diagnosis of major depressive disorder. *Psychiatry Research*, 7. doi:10.1016/j.psychres.2023.115127

13. DANGETI, P. (2017). *Statistics for Machine Learning.* Packt Publishing.

14. DILEEP, M., NAVANEETH, A., & ABHISHEK, M. (Febrero de 2021). A Novel Approach for Credit Card Fraud Detection using Decision Tree and Random Forest Algorithms. *IEEE*, 4. doi:10.1109/ICICV50876.2021.9388431

15. GERÓN, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2 ed.). O'Reilly Media, Inc.

16. HASSANBAKI GARABAGHI, F., BENZER, R., BENZER, S., & CAGLAN GUNAL, A. (Noviembre de 2022). Effect of polynomial, radial basis, and Pearson VII function kernels in support vector machine algorithm for classification of crayfish. *Ecological Informatics*, 8. doi:10.1016/j.ecoinf.2022.101911

17. KIRK, M. (2017). *Thoughtful Machine Learning with Python.* O'Reilly Media, Inc.

18. LANTZ, B. (2015). *Machine Learning with R* (2 ed.). Packt Publishing Ltd.

19. LAROSE, D. T., & LAROSE, C. D. (2014). *Discovering knowledge in data: An Introduction to Data Mining.* (2 ed.). Wiley.

20. LEE, C.-K., CHEON, Y.-J., & HWANG, W.-Y. (Marzo de 2022). Least Squares Generative Adversarial Networks-Based Anomaly Detection. *IEEE Access*, 11. doi:10.1109/ACCESS.2022.3158343

21. LIU, J., SONG, X., ZHOU, Y., PENG, X., ZHANG, Y., LIU, P., . . . ZHU, C. (Noviembre de 2021). Deep anomaly detection in packet payload. *Neurocomputing*, 14. doi:10.1016/j.neucom.2021.01.146

22. LIU, Y. H. (2020). *Python Machine Learning By Example* (3 ed.). Packt Publishing Ltd.

23. MILLER, J. D. (2017). *Statistics for Data Science.* Packt Publishing Ltd.

24. MUKHERJEE, I., SAHU, N. K., & SAHANA, S. K. (Diciembre de 2021). Simulation and Modeling for Anomaly Detection in IoT Network Using Machine Learning. *International Journal of Wireless Information Networks*, 17. doi:10.1007/s10776-021-00542-7

25. NAZARATHY, Y., & KLOK, H. (2021). *Statistics with Julia: Fundamentals for Data Science, Machine Learning and Artificial Intelligence.* Springer Nature Switzerland. doi:10.1007/978-3-030-70901-3

26. ROSS QUINLAN, J. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, Inc.

27. SAFFAR, M., & KALHOR, A. (Marzo de 2023). Evaluation of Dataflow through layers of convolutional neural networks in classification problems. *Expert Systems With Applications*, 14. doi:10.1016/j.eswa.2023.119944

28. SAKARKAR, G., PATIL, G., & DUTTA, P. (2021). *Machine Learning Algotithm using Python Programming.* Nova Science Publishers, Inc.

29. SRINIVAS, M., SUCHARITHA, G., & MATTA, A. (2021). *Machine Learning Algorithms and Applications.* Scrivener Publishing LLC.

30. WATT, J., BORHANI, R., & KATSAGGELOS, A. K. (2020). *Machine Learning Refined: Foundations, Algorithms, and Applications* (2 ed.). Cambrige University Press. doi:10.1017/9781108690935

7. ANEXOS

Anexo 1

Dataset de Enrutador de Transacciones.

Campo	Tipo de Dato	Descripción
EstadoTransaccion	entero	Indica si la transacción es válida o con error.
DuracionTransaccion	entero	Tiempo que dura la transacción, en segundos.
HoraInicio	fecha/tiempo	Hora en que inicia la transacción.
HoraFin	fecha/tiempo	Hora en que inicia la transacción.
IdCaja	texto	Identificación única de la caja que enruta la transacción, son 4 en total.
DestinoNii	entero	Destino identificado por ID
DestinoDireccion	texto	IP o número de teléfono de destino.
DestinoPuerto	entero	Puerto de destino de la transacción.
MID	texto	Identificador del comercio que envía la transacción.
TID	texto	Identificador del terminal que envía la transacción.
Ciudad	texto	Ciudad origen de la transacción.
TipoMensaje	texto	Tipo de mensaje a procesar.
TipoEnrutamiento	texto	Tipo de enrutamiento del mensaje, por NII o Inteligente.
TipoTransaccion	texto	Tipo de tecnología: TCP, TLS o DIAL.
CodigoProcesamiento	texto	Código de procesamiento.
ReferenciaTransaccion	texto	Número de identificador de la transacción por MID, TID, Fecha, Emisor.
Clasificacion	texto	Clasificación dada al evento.

Anexo 2

Dataset de Switch Transaccional.

Campo	Tipo de Dato	Descripción
DuracionTransaccion	entero	Tiempo que dura la transacción, en segundos.
HoraInicio	fecha/tiempo	Hora en que inicia la transacción.
HoraFin	fecha/tiempo	Hora en que inicia la transacción.
MID	texto	Identificador del comercio que envía la transacción.
TID	texto	Identificador del terminal que envía la transacción.
Result	texto	Código de respuesta dado por el core.
Ciudad	texto	Ciudad origen de la transacción.
TipoSwitch	texto	Tipo de switch transaccional que procesa la transacción.
TipoTerminal	texto	Tecnología que usa el terminal que envía la transacción.

MarcaTerminal	texto	Marca del terminal que envía la transacción.
Clasificacion	texto	Clasificación dada al evento.

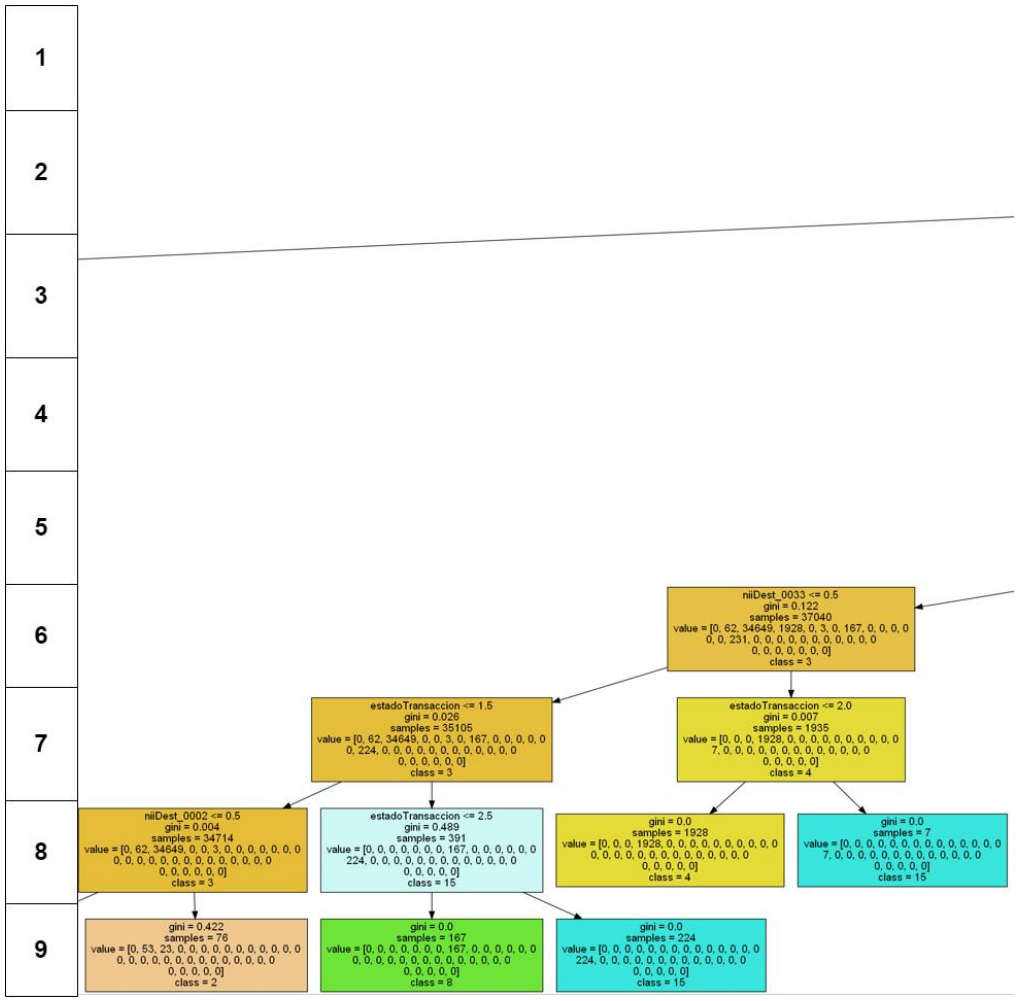
Anexo 3

Dataset de Core Transaccional.

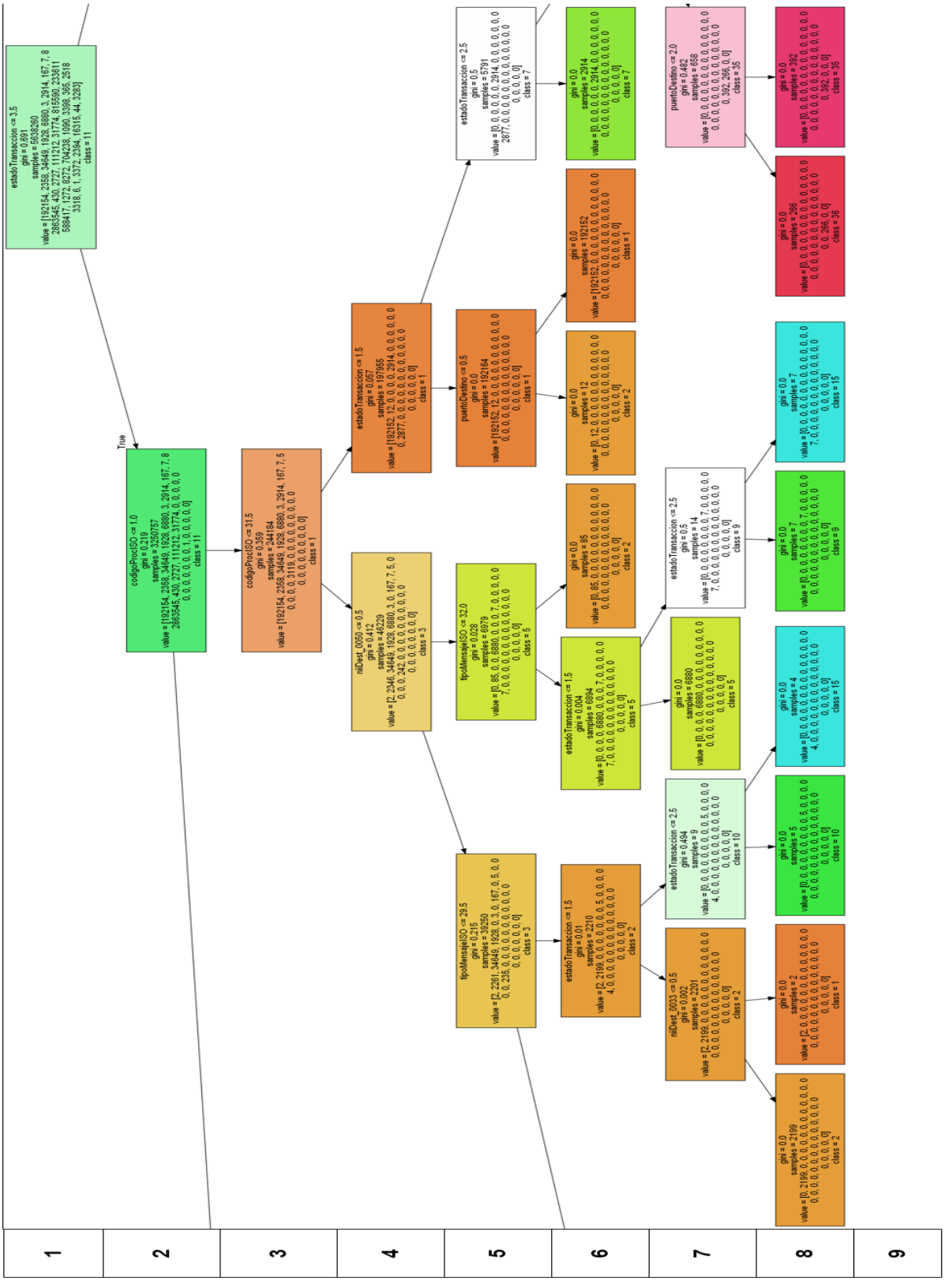
Campo	**Tipo de Dato**	**Descripción**
Fecha	fecha	Fecha de la transacción.
Hora	entero	Hora de la transacción.
DuracionTransaccion	entero	Tiempo que dura la transacción, en segundos.
MID	texto	Identificador del comercio que envía la transacción.
TID	texto	Identificador del terminal que envía la transacción.
ResultCore	texto	Código de respuesta dado por el Banco.
ResultEmisor	texto	Código de respuesta dado por el Banco.
Emisor	texto	Banco que autoriza la transacción enviada.
BIN	entero	Identificador de la marca de la tarjeta.
Standin	lógico	Indica si la transacción es procesada en linea o standby.
FechaCampo15	fecha	Fecha para contabilizar transacción.
PinBlock	lógico	Indica si tarjeta usa PIN de seguridad
ProcessingCode	texto	Tipo de transacción
TipoComercio	entero	Tipo de comercio.
Zona	texto	Zona a la que asigna LA EMPRESA al comercio.
Subzona	texto	Subzona, dentro de una zona, a la que asigna LA EMPRESA al comercio.
Ciudad	texto	Ciudad origen de la transacción.
Provincia		Provincia origen de la transacción.
MarcaTerminal	texto	Marca del terminal que envía la transacción.
ModeloTerminal	texto	Modelo del terminal que envía la transacción.
Clasificacion	texto	Clasificación dada al evento.

Anexo 4

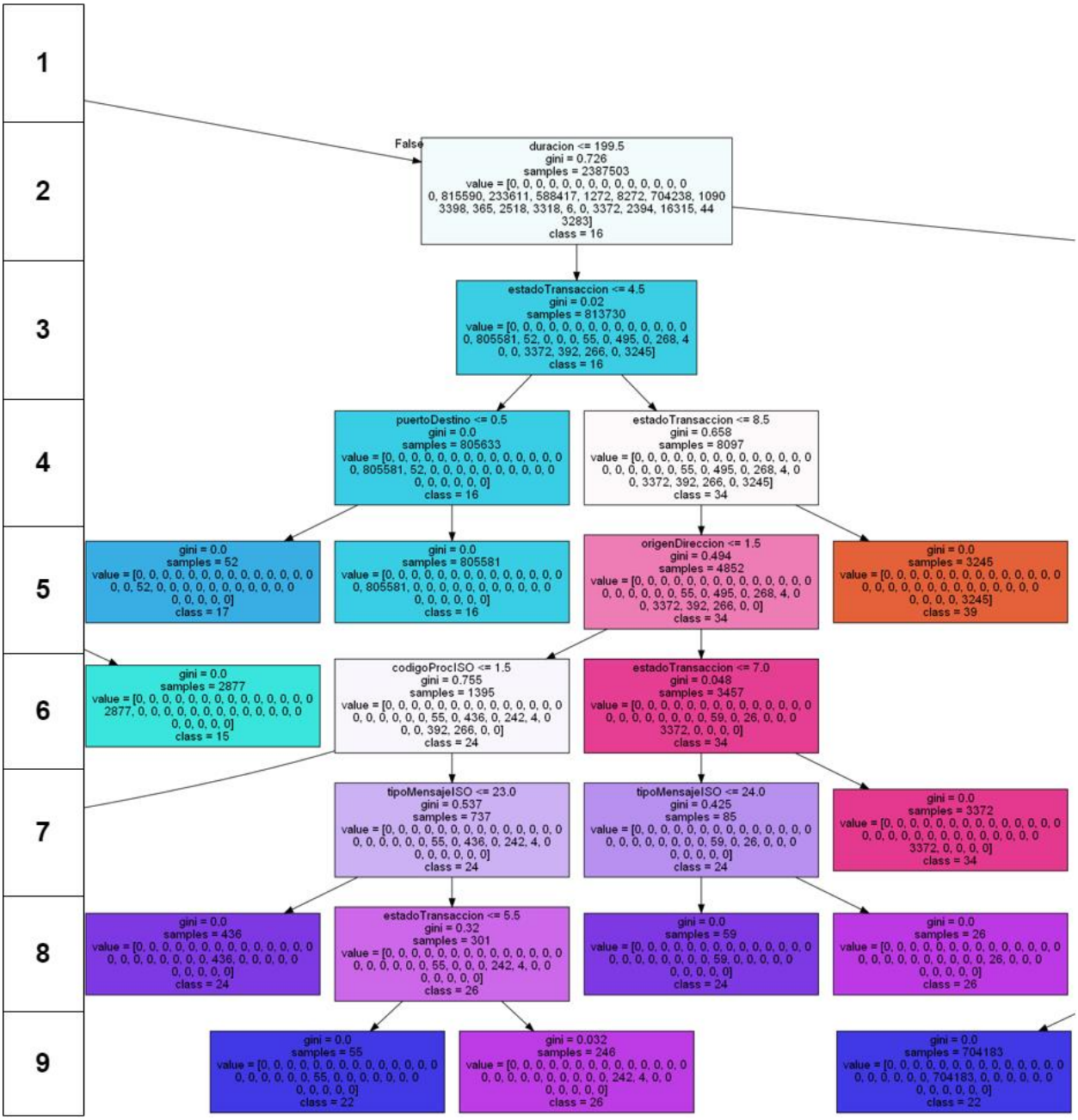
Anexo 5


1
2
3
4
5
6
7
8
9
niiDest_0033 <= 0.5
gini = 0.122
samples = 37040
value = [0, 62, 34649, 1928, 0, 3, 0, 167, 0, 0, 0, 0, 0, 231, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = 3
estadoTransaccion <= 1.5
gini = 0.026
samples = 35105
value = [0, 62, 34649, 0, 0, 3, 0, 167, 0, 0, 0, 0, 0, 224, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = 3
estadoTransaccion <= 2.0
gini = 0.007
samples = 1935
value = [0, 0, 0, 1928, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = 4
niiDest_0002 <= 0.5
gini = 0.004
samples = 34714
value = [0, 62, 34649, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = 3
estadoTransaccion <= 2.5
gini = 0.489
samples = 391
value = [0, 0, 0, 0, 0, 0, 0, 167, 0, 0, 0, 0, 0, 224, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = 15
gini = 0.0
samples = 1928
value = [0, 0, 0, 1928, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = 4
gini = 0.0
samples = 7
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = 15
gini = 0.422
samples = 76
value = [0, 53, 23, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = 2
gini = 0.0
samples = 167
value = [0, 0, 0, 0, 0, 0, 0, 167, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = 8
gini = 0.0
samples = 224
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 224, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
class = 15

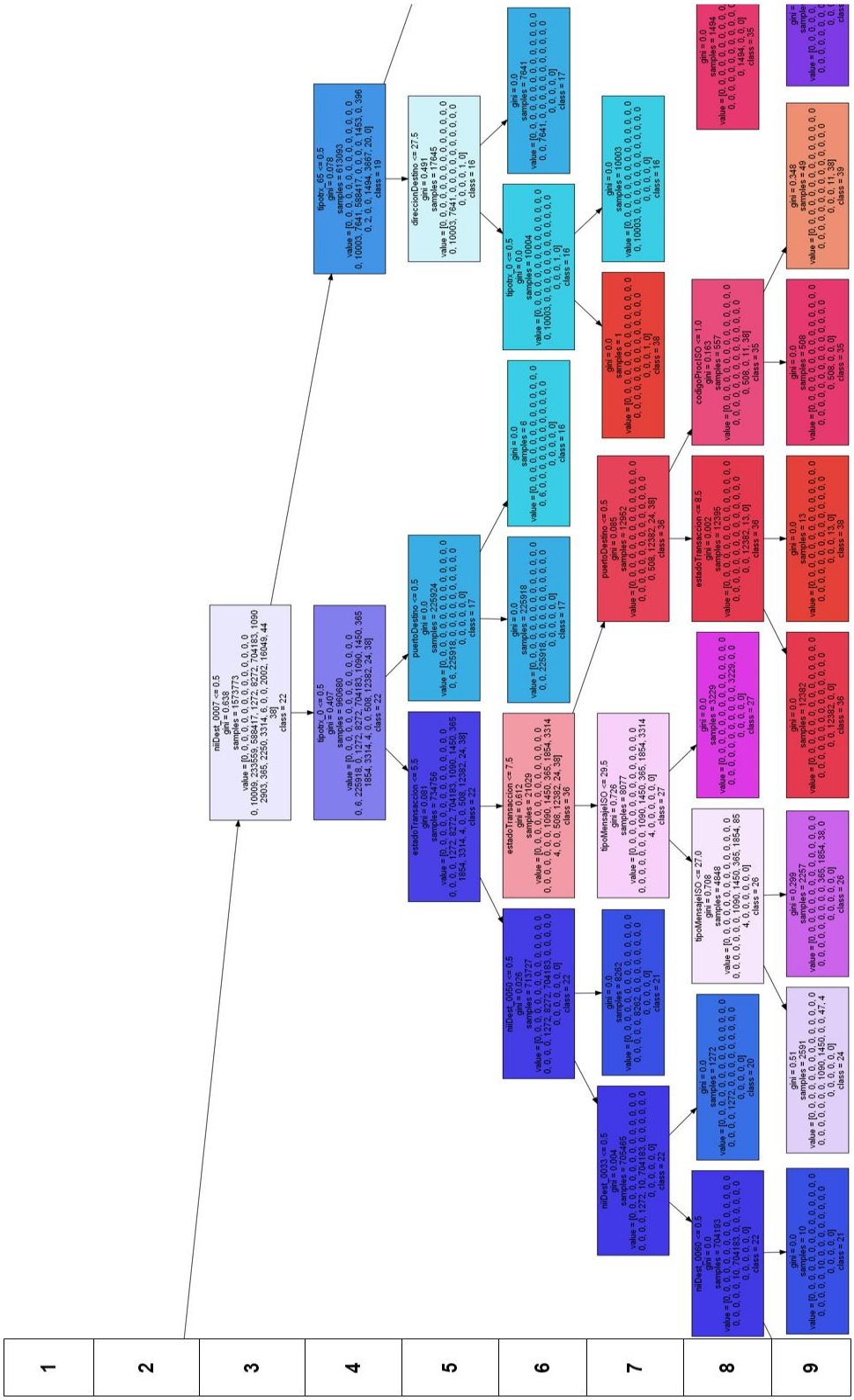
Anexo 6



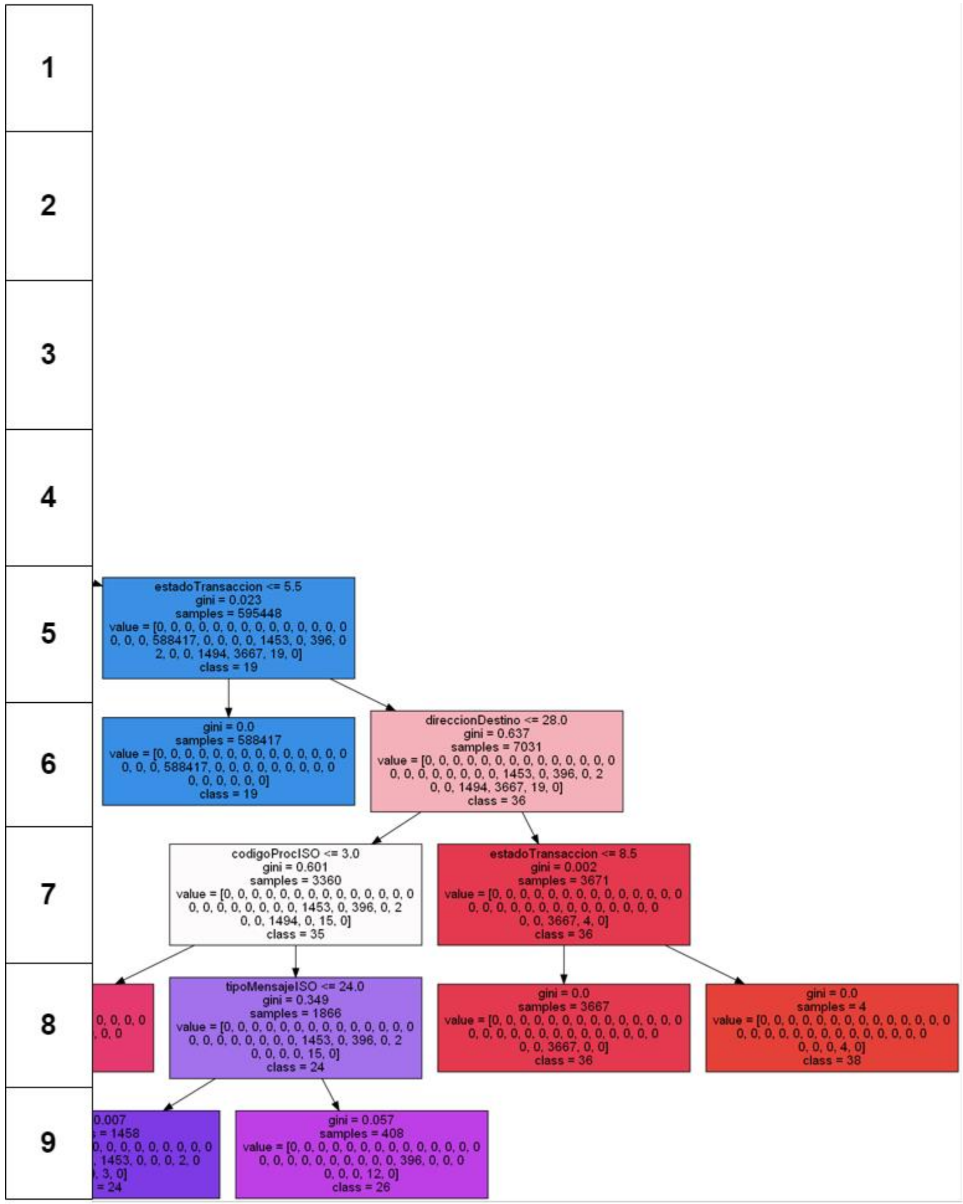
Anexo 7


1
2
3
4
5
6
7
8
9
False
duracion <= 199.5
gini = 0.726
samples = 2387503
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 815590, 233611, 588417, 1272, 8272, 704238, 1090
3398, 365, 2518, 3318, 6, 0, 3372, 2394, 16315, 44
3283]
class = 16
estadoTransaccion <= 4.5
gini = 0.02
samples = 813730
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 805581, 52, 0, 0, 0, 55, 0, 495, 0, 268, 4
0, 0, 3372, 392, 266, 0, 3245]
class = 16
puertoDestino <= 0.5
gini = 0.0
samples = 805633
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 805581, 52, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0]
class = 16
estadoTransaccion <= 8.5
gini = 0.658
samples = 8097
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 55, 0, 495, 0, 268, 4, 0
0, 3372, 392, 266, 0, 3245]
class = 34
gini = 0.0
samples = 52
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 52, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0]
class = 17
gini = 0.0
samples = 805581
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 805581, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0]
class = 16
origenDireccion <= 1.5
gini = 0.494
samples = 4852
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 55, 0, 495, 0, 268, 4, 0
0, 3372, 392, 266, 0, 0]
class = 34
gini = 0.0
samples = 3245
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 3245]
class = 39
gini = 0.0
samples = 2877
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
2877, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0]
class = 15
codigoProcISO <= 1.5
gini = 0.755
samples = 1395
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 55, 0, 436, 0, 242, 4, 0
0, 0, 392, 266, 0, 0]
class = 24
estadoTransaccion <= 7.0
gini = 0.048
samples = 3457
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 59, 0, 26, 0, 0, 0
3372, 0, 0, 0, 0]
class = 34
tipoMensajeISO <= 23.0
gini = 0.537
samples = 737
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 55, 0, 436, 0, 242, 4, 0
0, 0, 0, 0, 0, 0]
class = 24
tipoMensajeISO <= 24.0
gini = 0.425
samples = 85
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 59, 0, 26, 0, 0, 0
0, 0, 0, 0, 0]
class = 24
gini = 0.0
samples = 3372
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
3372, 0, 0, 0, 0]
class = 34
gini = 0.0
samples = 436
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 436, 0, 0, 0, 0, 0
0, 0, 0, 0, 0]
class = 24
estadoTransaccion <= 5.5
gini = 0.32
samples = 301
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 55, 0, 0, 0, 242, 4, 0, 0
0, 0, 0, 0, 0]
class = 26
gini = 0.0
samples = 59
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 59, 0, 0, 0, 0, 0
0, 0, 0, 0, 0]
class = 24
gini = 0.0
samples = 26
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0]
class = 26
gini = 0.0
samples = 55
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 55, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0]
class = 22
gini = 0.032
samples = 246
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 242, 4, 0, 0
0, 0, 0, 0, 0]
class = 26
gini = 0.0
samples = 704183
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 704183, 0, 0, 0, 0, 0
0, 0, 0, 0, 0]
class = 22

Anexo 8



Anexo 9


1
2
3
4
5
6
7
8
9
estadoTransaccion <= 5.5
gini = 0.023
samples = 595448
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 588417, 0, 0, 0, 0, 1453, 0, 396, 0
2, 0, 0, 1494, 3667, 19, 0]
class = 19
gini = 0.0
samples = 588417
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 588417, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0]
class = 19
direccionDestino <= 28.0
gini = 0.637
samples = 7031
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 1453, 0, 396, 0, 2
0, 0, 1494, 3667, 19, 0]
class = 36
codigoProcISO <= 3.0
gini = 0.601
samples = 3360
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 1453, 0, 396, 0, 2
0, 0, 1494, 0, 15, 0]
class = 35
estadoTransaccion <= 8.5
gini = 0.002
samples = 3671
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 3667, 4, 0]
class = 36
0, 0, 0, 0
, 0, 0
tipoMensajeISO <= 24.0
gini = 0.349
samples = 1866
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 1453, 0, 396, 0, 2
0, 0, 0, 0, 15, 0]
class = 24
gini = 0.0
samples = 3667
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 3667, 0, 0]
class = 36
gini = 0.0
samples = 4
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 4, 0]
class = 38
0.007
= 1458
0, 0, 0, 0, 0, 0, 0, 0, 0
, 1453, 0, 0, 0, 2, 0
, 3, 0]
= 24
gini = 0.057
samples = 408
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 396, 0, 0, 0
0, 0, 0, 12, 0]
class = 26