

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

TRABAJO DE TITULACIÓN

"APLICATIVO BASADO EN MACHINE LEARNING QUE DETECTA FALLOS EN SERVIDORES Y SUGIERE SOLUCIONES"

TUTOR: PhD RAFAEL MELGAREJO

AUTOR: ANDRÉS EDUARDO LARA HIDALGO

AGOSTO 2021

DEDICATORIA

Dedico este logro a mis padres César y Janneth por el apoyo, el acompañamiento y el infinito amor y enseñanzas que me han podido entregar, por hacer de mí una persona de bien. A mis hermanos Gabriel y Johanna, para ellos el amor y la infinita gratitud por el acompañamiento.

A PhD Rafael Melgarejo, Director de Tesis por su dedicación, asesoramiento y apoyo con sus amplios conocimientos profesionales y técnicos, a mis revisores por la gestión y apoyo.

Andrés Eduardo Lara Hidalgo

Tabla de Índice

1.1 Introducción 1
1.2 Ciencias Relacionadas con la Machine Learning 2
1.3 Descripción del proceso general de un proyecto de Machine Learning 3
1.4 Clustering 5
1.4.1 Definición 5
1.4.2 Técnicas de Clustering Jerárquico. 5
1.4.3 Técnicas de Clustering Particional 6
1.5 Definición y características de Reglas de Asociación 7
1.5.1 Introducción al problema 7
1.5.2 Definición del Problema 8
1.6 Definición y características de Redes Neuronales 9
2.1 Origen de los datos 11
2.2 Depuración y Estructura de los Datos 15
2.2.1 Depuración de los datos 15
2.2.2 Estructura de los Datos 16
3.1 Introducción 20
3.2 Clusterización: 20
3.2.1 Características principales 20
3.2.2 Ventajas 21
3.2.3 Desventajas 21
3.3 Reglas de Asociación: 21
3.3.1 Características principales 22
3.3.2 Ventajas 22
3.3.3 Desventajas 22
3.4 Redes Neuronales: 23
3.4.1 Características principales 23
3.4.2 Ventajas 23
3.4.3 Desventajas 24
3.5 Uso de Librerías: 24
3.5.1 Tensorflow 24
3.5.2 Keras 25
3.5.3 Scikit-learn 25
3.5.4 K-Means 25
3.6 Ejecución del proyecto: 26

4.1 Conclusiones de los modelos de Machine Learning 27
4.1.1 Conclusiones de Clusterización 27
4.1.2 Conclusiones de Reglas de Asociación 28
4.1.3 Conclusiones de Redes Neuronales 28
4.2 Ejecución conjunto dato PRONACA 29
4.3 Recomendaciones para proyectos de Machine Learning 30
Bibliografía 31
Anexos 32

Tabla de Figuras

Figura 1. Canasta de compras en un supermercado. Fuente: Robertazzi, Thomas en “Networking ad Computation” (2020) 7
Figura 2. Conexión a un servidor mediante la Remote Desktop Connection de Windows. Fuente: Elaboración Propia 11
Figura 3. Performance Monitor en ejecución. Fuente: Elaboración Propia 12
Figura 4. Creación del Data Collector Set. Fuente: Elaboración Propia 12
Figura 5. Selección del tipo de datos que se incluirán en el Data Collector Set. Fuente: Elaboración Propia. 13
Figura 6. Seleccionando los Performance Counters. Fuente: Elaboración Propia. 13
Figura 7. Data Collector Set Creado mediante el proceso detallado en este trabajo. Fuente: Elaboración Propia. 14
Figura 8. Selección de un horario para el Data Collector Set creado previamente. Fuente: Elaboración Propia. 14
Figura 9. Comando para exportar los datos recopilados en un archivo CSV. Fuente: Elaboración Propia 15
Figura 10. Datos exportados como CSV. Fuente: Elaboración Propia 15
Figura 11. Archivo CSV depurado. Fuente: Elaboración Propia. 16

Capítulo 1: Prólogo – Machine Learning

1.1 Introducción

Tal y como lo menciona Kubat (2015), entre los múltiples objetivos que puede tener un ordenador, existe un grupo específico de tareas que se centran que en estas máquinas puedan reconocer y definir ciertos objetos o conceptos, es decir, que estos ordenadores por sí mismos generen conocimiento a partir de datos brutos o, en otras palabras, que puedan aprender. Sin embargo, es evidente que las máquinas no pueden trabajar en este reconocimiento del mismo modo que un ser humano. Por lo tanto, conforme la computación ha evolucionado, se han ido creando distintos métodos para "enseñar" a los ordenadores a cumplir ciertas tareas.

Shalev-Shwartz & Ben-David (2014) afirman que ese conjunto de métodos se conoce como Aprendizaje de Máquina o Machine Learning (término en inglés) y pretende que la máquina realice una serie de operaciones de manera automática ya sea por medio de un aprendizaje denominado como "Razonamiento Inductivo" que consiste en que la máquina aprenda por medio de un conjunto de ejemplos, así como de manera menos guiada. Estos dos tipos de Aprendizaje suelen clasificarse como "Aprendizaje Supervisado" y "Aprendizaje no Supervisado". Por medio de la utilización de este medio, se busca vías alternativas para algunas actividades y que son el fundamento para estudiar esta rama de la Inteligencia Artificial. Algunas de las citadas actividades son, por ejemplo, tareas que las personas suelen realizar con cotidianeidad, pero que son muy difícil de abstraer como para ser plasmadas en un programa computacional; también se debe mencionar, como factor adicional para emplear Machine Learning, el hecho de la rigidez que presenta la construcción de software tradicional, que suele tener un propósito único y su escalabilidad no permite una diversificación de funciones.

Es por lo mencionado previamente que se busca con el presente estudio emplear ciertas técnicas derivadas de Machine Learning para el siguiente propósito:

- Desarrollo de un aplicativo que detecte fallos en servidores y sugiera soluciones.

Se busca lograr este objetivo utilizando tres métodos distintos de Machine Learning, es decir, distintos algoritmos que generen patrones y en base a estos obtener conclusiones apropiadas para generar informes dentro de una organización, y así mantener una cultura preventiva.

1.2 Ciencias Relacionadas con la Machine Learning.

Como ya se mencionó, el Aprendizaje de Máquina puede verse como una rama de la Inteligencia Artificial y se lo suele considerar como un subcampo de las Ciencias de la Computación. Por lo tanto, existen varios conceptos que se comparten. Sin embargo, no debería verse al Machine Learning simplemente como una tarea más dentro de la IA, debido sobre todo a que sus objetivos más generales no son exactamente iguales. Shalev-Shwartz & Ben-David (2014) proponen que el Aprendizaje de Máquina no busca imitar de manera automatizada un comportamiento inteligente, sino que pretende usar las fortalezas y características especiales de los ordenadores con el afán de complementar la inteligencia humana, usualmente desarrollando tareas que van (comúnmente) más allá de las posibilidades de los seres humanos. Y es menester hacer hincapié en este último punto: "ir más allá de lo que puede desarrollar una persona". Por ejemplo: el Aprendizaje de Máquina puede ayudar a detectar patrones de cientos de miles de datos agrupados en complejas estructuras de datos en un tiempo nada despreciable. Una sola persona se expondría a gastar muchas horas frente al computador para si quiera recorrer una vez los datos de uno en uno, además del cometimiento de posibles errores de interpretación.

Otra de las ciencias relacionadas es la Estadística. Cuando en Machine Learning se habla de entrenamiento se hace referencia a un conjunto de datos que son generados aleatoriamente y la tarea de la máquina (que en este caso se está entrenando para generar conocimiento) es la de procesar esos ejemplos generados de manera aleatoria para sacar conclusiones que sirvan para el entorno del que se tomaron dichos ejemplos. Esta particular definición básica de Machine Learning destaca su relación con la Estadística.

A pesar de que, en efecto, existen varias similitudes entre estas dos ciencias, debido sobre todo a sus objetivos y técnicas, existen diferencias importantes las cuales deben ser mencionadas. Shalev-Shwartz & Ben-David proveen un ejemplo bastante acertado: si un médico plantea una hipótesis de que existe una correlación entre el tabaquismo y una enfermedad cardíaca, es función del estadista ver muestras de pacientes y verificar la validez de esa hipótesis (esta vendría a ser una tarea común para probar hipótesis). Por el contrario, el Machine Learning tiene como objetivo

utilizar los datos recopilados de muestras de pacientes para obtener una descripción de las causas de la citada enfermedad cardíaca. Lo que se espera, por tanto, es que la aplicación de varias técnicas/algoritmos de manera automatizada permita descurbir patrones importantes que existen en los datos y que para una persona son díficiles de encontrar.

Las diferencias con la Estadística se pueden seguir numerando, como por ejemplo, y referenciando nuevamente a Shalev-Shwartz & Ben-David (2014), el hecho de que en ciertos casos la Estadística se interesa en los comportamientos asintóticos en donde las muestras tomadas para algún proceso no tienen un fin, mientras que el Aprendizaje de Máquina por las limitaciones que supone un ordenador, se centra únicamente en los límites de muestras finitas. También se puede mencionar el hecho de que en la Estadística se suele trabajar con supuestos en caso de no tener un modelo de datos especificado, por ejemplo se suele asimir que cierto grupo de datos tiene una distribución normal cuando no se está totalement seguro de este punto, mientras que para el Machine Learning esto no representa un problema real puesto que una máquina no necesita conocer ni asumir la distribución de los datos que se le ingresar para comenzar su proceso de aprendizaje.

De igual manera, la relación con la Matemática debe ser mencionada, especialmente en algunas áreas importantes cómo el Álgebra Lineal para cálculo de vectores, matrices, etc., Matemática Analítica para cálculo de derivadas y gradientes, Cálculo Multivariable, entre otros. Sin embargo, esta relación con la matemática será explicada más adelante de acuerdo a los métodos que serán profundizados.

1.3 Descripción del proceso general de un proyecto de Machine Learning.

Según Guo (2017), Aprendizaje de Máquina se plantea como un proceso que consta de siete etapas las cuales tienen como meta la obtención de predicciones y ese proceso es el siguiente:

1. Obtención de los datos: en este punto es cuando se recaban todo aquello que se considere importante para futuros pasos. Su importancia radica en que la calidad y cantidad de los datos determinará directamente en qué tan bueno será el modelo predictivo elegido.

2. Preparación de los datos: aquí es donde los datos se almacenan en un lugar adecuado y se preparan de cualquier manera elegida. Por ejemplo, un punto clave sería juntar datos en un mismo lugar si éstos se encuentran dispersos en varias bases de datos. También se puede generar

un orden aleatorio de modo que no afecte a los futuros pasos. En este punto también se suelen emplear técnicas de visualización de datos para encontrar posibles conexiones entre variables (aquellas que sean bastante obvias de encontrar). También se pueden encontrar errores, normalizar los datos o eliminar duplicados. Finalmente, la actividad con la que se cierra esta etapa es con la división de los datos en un grupo que sirva para entrenar el modelo y otro para evaluar el mismo. Esta división se suele realizar de modo que el grupo de entrenamiento represente a la mayoría de los datos.

3. Elección del modelo: se escoge un modelo de entre los muchos disponibles que han sido desarrollados con el paso del tiempo. En el presente estudio, además, se plantea la utilización de tres distintos modelos con el objetivo de encontrar diferencias en los resultados arrojados.

4. Entrenamiento: en este punto es cuando se busca que la máquina mejore en su habilidad de predicción que, se recuerda, es el objetivo principal establecido. Este es un subproceso cíclico en el que se busca que, con cada iteración, las predicciones arrojadas sean mejores que la vez anterior. Para esto se usa el primer grupo de datos, aquel que representaba a la mayoría de los datos recopilados.

5. Evaluación: una vez que se haya establecido que el entrenamiento fue completado, se debe evaluar si la máquina realmente aprendió. Para esto se usa el segundo grupo de datos, el más pequeño. La idea es simple: ejecutar el modelo escogido y ejecutarlo con un grupo de datos nuevo que no haya sido usado en la sección de entrenamiento y, con esto, analizar lo arrojado.

6. Mejora y Ajuste de Parámetros: una vez se obtengan los resultados, la persona a cargo de ejecutar el proceso de aprendizaje puede decidir si el modelo es mejorable. Esto se puede lograr analizando los parámetros escogidos en las primeras instancias del proceso en donde algunos pueden haber sido más importantes que otros y que luego de la evaluación se concluye necesitan más relevancia en el proceso. No existe una fórmula que asegure que un modelo mejorará de acuerdo con la inclusión o exclusión de ciertos parámetros debido a que cada grupo de datos representa un proceso diferente, por lo que la experimentación juega un papel importante aquí.

7. Predicciones: es donde se pueden generar las conclusiones y responder varias de las preguntas iniciales.

1.4 Clustering

1.4.1 Definición

Según Jain et al. (1999) Clustering es el proceso de identificar grupos o clústeres naturales dentro de datos multidimensionales que contengan alguna similitud entre sí. Dentro de las múltiples utilizaciones que se les han dado a los algoritmos de Clústering se encuentran la segmentación de imágenes, cuantificación de imágenes vectoriales, minería de datos y Machine Learning. Un clúster, que significa grupo o conjunto, se identifica usualmente por un núcleo o centro denominado "centroide". Clustering es un ejemplo de reconocimiento de patrones no supervisado, es decir, pertenece a la clasificación de Unsupervised Learning dentro de Machine Learning

Jain et al. (1999) plantean una definición de Clustering en donde dado un conjunto de datos $Z = \{z_1, z_2, \dots, z_p, Z_{N_p}\}$ donde z_p es un patrón en el espacio de características N_d dimensional y N_p es el número de patrones en Z, entonces un clúster de Z es la partición de Z en K clústeres con la forma $\{C_1, C_2, \dots, C_K\}$ cumpliendo con las siguientes condiciones:

- Cada patrón debe estar asignado a un clúster.
- Cada clúster debe tener asignado al menos un patrón.
- Cada patrón debe ser asignado a un y solo un clúster. Esto es aplicable únicamente en el caso de Hard Clustering.

Frigui & Krishnapuram (1999) sostiene que la mayoría de los algoritmos que realizan Clustering están basados en dos populares técnicas conocidas como Clustering Jerárquico y Clustering Particional.

1.4.2 Técnicas de Clustering Jerárquico.

Según Hamerly (2003), los algoritmos en esta categoría generan un árbol clúster, mejor conocido como dendograma mediante el uso de técnicas heurísticas de división o fusión. Un árbol clúster es definido como un árbol que muestra una secuencia de clústeres, con cada clúster siendo una partición del conjunto de datos. Los algoritmos que realizan una división para generar el árbol clúster son denominados "divisivos" mientras que aquellos que realizan una fusión o unión para generar dicho árbol son denominados "aglomerativos".

Hamerly (2003) también afirma que los algoritmos divisivos empiezan asignando todos los patrones a un solo clúster. Luego, el proceso de división es aplicado al clúster, partiendo los datos hasta que cada grupo generado tenga un solo patrón. Por el contrario, los algoritmos aglomerativos empiezan con cada patrón asignado a un grupo para luego fusionar los dos clústeres más similares y esto se repite hasta que todos los patrones se asignen a un solo clúster. La diferencia entre las variantes de los algoritmos aglomerativos es la forma en que se calcula cuáles son los dos clústeres más similares para poder fusionarlos. Dentro de todas estas variantes, la más comunes son las denominadas: “single link” y “complete link” o también llamados enlace único y enlace completo por sus traducciones. Los algoritmos de enlace único fusionan los clústeres cuya distancia entre sus patrones más cercanos es la más pequeña mientras que los algoritmos de enlace completo fusionan los grupos cuya distancia entre sus patrones más distantes es la más pequeña. Esta diferencia hace que los algoritmos de enlace único generen clústeres compactos y los algoritmos de enlace completo generen clústeres alargados.

Turi (2001) propone algunas de las ventajas y desventajas de Clustering Jerárquico, teniendo como dos grandes ventajas que a priori, el número de algoritmos no necesita ser especificado y que los clústeres no están sujetos a posibles condiciones iniciales. Sin embargo, las desventajas que presentan son su costo computacional es alto, por lo tanto, no son aplicables a grupos con grandes volúmenes de datos, los patrones asignados a un clúster no pueden ser transferidos a otro clúster. es posible que no puedan separar los clústeres superpuestos debido a falta de información sobre la forma o el tamaño global de los clústeres.

1.4.3 Técnicas de Clustering Particional

Hamerly (2003) también señala que los algoritmos en esta categoría dividen el conjunto de datos en un número específico de clústeres. Estos algoritmos tratan de minimizar ciertos criterios y por ello pueden ser tratados como algoritmos para optimización de problemas. Las ventajas de los algoritmos de Clustering Jerárquico son las desventajas de Clustering Particional y viceversa. Por lo tanto, citadas las ventajas y desventajas de Clustering Jerárquico se entiende mejor el motivo de la popularidad de Clustering Particional al momento de reconocer patrones. Algoritmos como el K-medias, el C-medias difuso, entre otros, son solo algunos ejemplos de las formas de aplicar Clustering Particional, cada uno con su particular aplicación de manera iterativa para lograr el objetivo antes planteado.

En el momento de la aplicación de Clustering para el presente estudio, se detallará el algoritmo como tal presentando los pasos y etapas de su ejecución.

1.5 Definición y características de Reglas de Asociación

1.5.1 Introducción al problema

Se plantea de parte de Moreno, Segrera y López (2005) que el análisis de Asociación ha sido usado en muchos campos, uno de los cuales es el ámbito de los negocios donde el descubrimiento de patrones de compra o asociación entre productos es muy útil para la toma de decisiones y efectivo para el mercadeo. En los últimos años, se ha incrementado considerablemente su uso en distintas aplicaciones como pueden ser el descubrimiento de patrones en bases de datos biológicas, extracción de conocimiento de métricas de Ingeniería de Software u obtención de perfiles de usuario para personalización de sitios web. La Asociación es también catalogada como un ejemplo de Aprendizaje No Supervisado, es decir, que los datos que se tienen no están etiquetados para un entrenamiento y solamente se conocen los datos de entrada sin tener un conjunto de datos de salida.

Por lo tanto, el objetivo de este análisis es el de encontrar relaciones interesantes que están ocultas a simple vista dentro de un gran grupo de datos. Estas relaciones que son descubiertas pueden ser representadas en forma de reglas de asociación (de ahí su nombre). A continuación, se presenta una imagen con un listado de distintos elementos que representan una canasta de compras al ir a un supermercado basados en un ejemplo propuesto por Robertazzi (2020):

TID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}

Figura 1. Canasta de compras en un supermercado. Fuente: Robertazzi, Thomas en "Networking ad Computation" (2020)

En base a esa canasta con distintos elementos, se puede generar una regla que sea de la siguiente forma: $\{Diapers\} \rightarrow \{Beer\}$.

Esta regla bien puede representar que existe una relación estrecha entre la compra de diapers (pañales) y beer (cerveza) porque muchos de los clientes de ese supermercado que

compraron pañales también compraron cerveza. Un negocio podría llevar a cabo este tipo de análisis para incrementar sus ventas.

1.5.2 Definición del Problema

Lo que realiza Robertazzi (2020) es una representación binaria con el objetivo de simplificar el problema de Reglas de Asociación. Los datos que se mostraron en la sección anterior que contenían elementos de una canasta de compras de un supermercado pueden también ser representados de manera binaria de modo que cada ítem tenga un valor binario: siendo 1 si el ítem se encuentra en la transacción de compra o en caso contrario, 0. Debido a que la presencia de un ítem en una transacción es usualmente considerada más importante que su ausencia, el ítem se convierte en una variable binaria asimétrica. Esta simplificación puede traer complicaciones en la vida real debido a que se quedan fuera algunos elementos importantes como la cantidad de ítems vendidos o el precio pagado por dichos elementos, sin embargo, esto no supone un problema para la presente explicación. Se establece que $I = \{i_1, i_2, \dots, i_d\}$ es el conjunto de todos los ítems en una canasta de compras y que $T = \{t_1, t_2, \dots, t_N\}$ es el conjunto de todas las transacciones. Cada transacción t_i contiene un subgrupo de ítems elegidos de I. En el Análisis de Asociación, una colección de cero o más ítems se denomina "conjunto de elementos" o "itemset" en inglés. Si un itemset contiene *k* ítems, es denominado *k*-itemset. Por ejemplo, el conjunto $\{Cerveza, Pañales, Leche\}$ es un ejemplo de un 3-itemset. La anchura de una transacción se define como el número de ítems presentes en una transacción. Una transacción t_j se dice que contiene un itemset X si X es un subgrupo de t_j. Por ejemplo, la transacción $\{Pan, Pañales, Cerveza, Huevos\}$ contiene el itemset $\{Cerveza, Pañales\}$ pero no el itemset $\{Pan, Leche\}$.

Por lo tanto, basados en lo que plantea Robertazzi (2020), una regla de asociación es una expresión de implicación de la forma $X \rightarrow Y$, donde X y Y son dos itemsets disjuntos, es decir que, si se realiza una intersección entre los dos itemsets, se obtiene como resultado vacío. ($X \cap Y = \emptyset$). La fortaleza de una regla de asociación puede ser medida en términos de soporte y confianza. Soporte establece qué tan seguido una regla es aplicable a un determinado grupo de datos, mientras que confianza establece qué tan frecuente los ítems en Y aparecen en las transacciones que contiene X.

1.6 Definición y características de Redes Neuronales

Gracias a Russell y Norvig (2010), se entiende que para poder hablar sobre el concepto como tal de Redes Neuronales es necesario comprender ciertos puntos básicos de los cuales se parte. En primer lugar, se debe mencionar que la actividad mental consiste fundamentalmente en la actividad electroquímica en redes de células cerebrales llamadas neuronas. Aunque esto se mantiene como una hipótesis que no termina de ser refutada o comprobada debido a la complejidad del cerebro humano, su base ha tratado de ser replicada en términos computacionales, mediante la creación de redes neuronales artificiales. Sin embargo, esta concepción se completa con un modelo matemático que fue planteado por McCulloch y Pitts en el libro "*A logical calculus of the ideas immanent in nervous Activity"* que fue publicado en el año 1943 y a partir del cual se desarrollaron varios modelos con más detalle y más realismo, tanto para las neuronas como para los sistemas dentro del cerebro que llevaron posteriormente a la creación de la Neurociencia Computacional. Por otro lado, investigadores de IA y de Estadística se empezaron a interesar por las propiedades más abstractas de las redes neuronales como su habilidad para realizar computación distribuida, su tolerancia a inputs con ruido y su habilidad para aprender.

Una red neuronal no es más que un conjunto de unidades que están conectadas entre sí, según el mismo planteamiento de Russell y Norvig (2010). Las propiedades de una red están determinadas por su topología y por las propiedades de sus neuronas. Como se dijo, las redes neuronales se componen de nodos o unidades que están conectadas directamente mediante links o enlaces. Un enlace desde *i* hasta *j* sirve para propagar la activación a_i desde *i* hasta *j*. Cada enlace tiene también un valor numérico llamado peso $w_{i,j}$ asociado con él y que determina la fuerza y la señal de dicha conexión.

De la misma manera que ocurre con los modelos de Regresión Lineal, Russell y Norvig (2010) indican que cada unidad tiene una entrada ficticia $a_0 = 1$ con un peso asociado $w_{0,j}$. Cada unidad *j* primero calcula una suma ponderada de sus entradas: $in_j = \sum_{i=0}^{n} w_{i,j} a_i$ y posteriormente, se establece necesaria la aplicación de una función de activación g a esta suma para derivar la salida: $a_j = g(in_j) = g\left(\sum_{i=0}^{n} w_{i,j} a_i\right)$. Una vez establecido el modelo matemático para neuronas individuales, es momento de determinar una manera de conectarlas mediante una red. Esto se puede realizar de dos maneras: mediante una Red Prealimentada y mediante una Red Recurrente. Una Red Prealimentada tiene conexiones únicamente en una dirección, es decir, con la forma de

un Grafo Acíclico Dirigido (DAG por sus siglas en inglés), no se forman ciclos de ninguna manera y la información se mueve solamente hacia adelante. Este tipo de red representa una función de su actual entrada y no tiene ningún otro estado más que sus mismos pesos asociados. Por otro lado, una Red Recurrente tiene como característica que alimenta sus entradas con sus propias salidas, lo que significa que los niveles de aceptación de la red forman un sistema dinámico que puede alcanzar una estabilidad, exhibir oscilaciones o inclusive mostrar un comportamiento caótico. Además, la respuesta de la red a una entrada dad depende de su estado inicial, que puede depender de inputs anteriores. Por lo tanto, las Redes Recurrentes pueden soportar memoria a corto plazo.

Capítulo 2: Obtención y Formalización de Datos

2.1 Origen de los datos

La recolección de datos agrupa todo el funcionamiento de distintos componentes de un servidor, de modo que mediante los mismos y aplicando Machine Learning se obtengan predicciones para mantener un modo de trabajo proactivo y procurar la anticipación de eventos desfavorables como el malfuncionamiento de algún equipo.

La herramienta para la recolección de datos se conoce como Performance Monitor. Lo primero que se debe realizar es una conexión remota utilizando la aplicación "Remote Desktop Connection" utilizando un usuario y contraseñas predefinidas en el servidor al que se está ingresando.

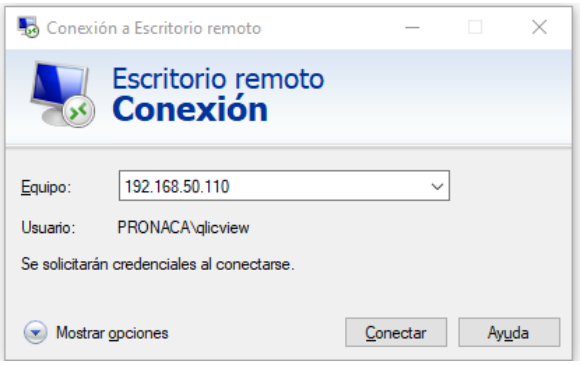


Figura 2. Conexión a un servidor mediante la Remote Desktop Connection de Windows. Fuente: Elaboración Propia

Luego de establecer la conexión al servidor, se debe buscar la mencionada aplicación "Performance Monitor" y ejecutarla como administrador para evitar problemas con alguna configuración posterior. La aplicación ejecutándose puede verse en la Figura 4. Lo que se debe buscar en este punto es el apartado de Data Collector Sets en el menú izquierdo. Aquí se puede apreciar cómo el sistema por defecto tiene asignados algunos Data Collector Sets que no son más que colecciones de datos de rendimiento del sistema, información extra del sistema o seguimiento de eventos. Aquí el usuario puede seleccionar la opción "User Defined" para crear un nuevo y personalizado Data Collector Set.

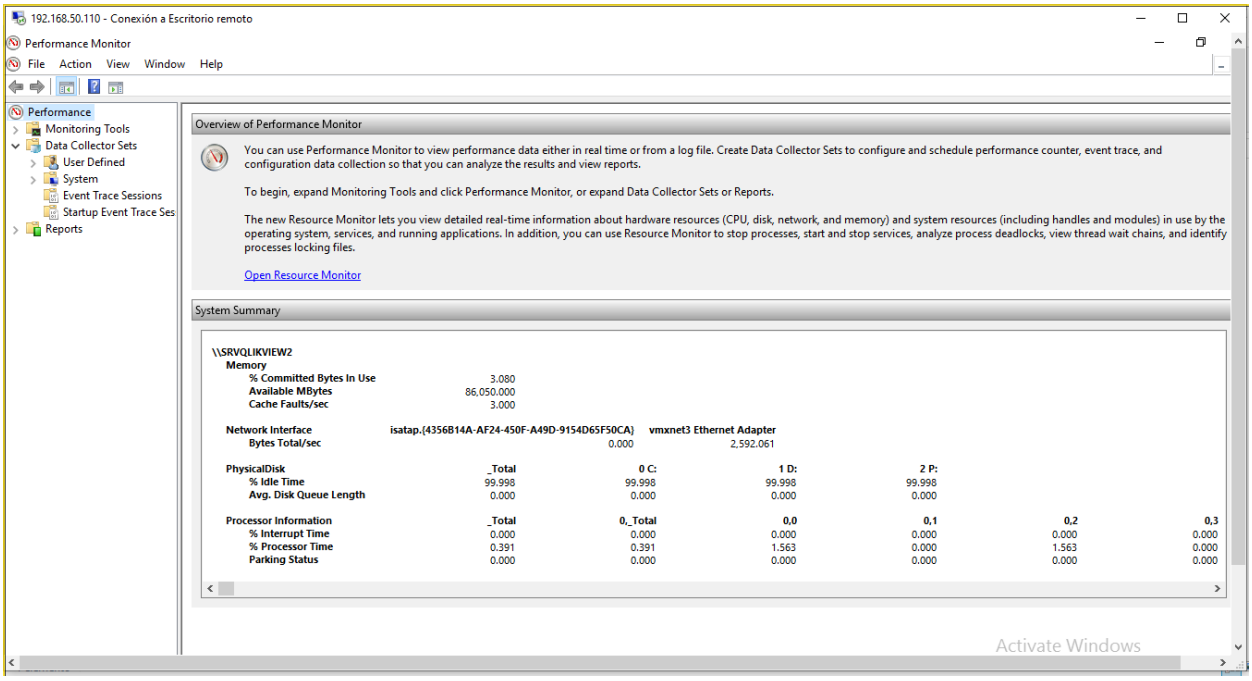


Figura 3. Performance Monitor en ejecución. Fuente: Elaboración Propia

El siguiente paso es dar clic derecho sobre la carpeta User Defined y escoger la opción New y posteriormente Data Collector Set para empezar con el proceso de crear uno nuevo. Inmediatamente la aplicación abre una nueva ventana donde se solicita un nombre para el nuevo Data Collector Set, así como si se desea generar la colección sobre un modelo preestablecido o crearlo manualmente. En este caso se elegirá crear uno manualmente.

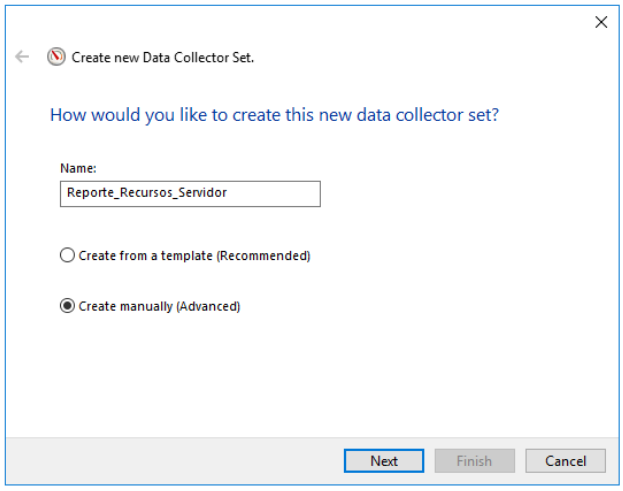


Figura 4. Creación del Data Collector Set. Fuente: Elaboración Propia

El nombre elegido puede ser uno cualquiera. Luego de dar al botón para continuar, la aplicación pregunta por el tipo de datos que se quiere recopilar. Como se mencionó previamente, un Data Collector Set puede guardar datos de rendimiento del sistema y en este caso, es la opción que se elige: “Create Data Logs” y habilitar únicamente la casilla “Performance counter”, como se muestra en la Figura 5.

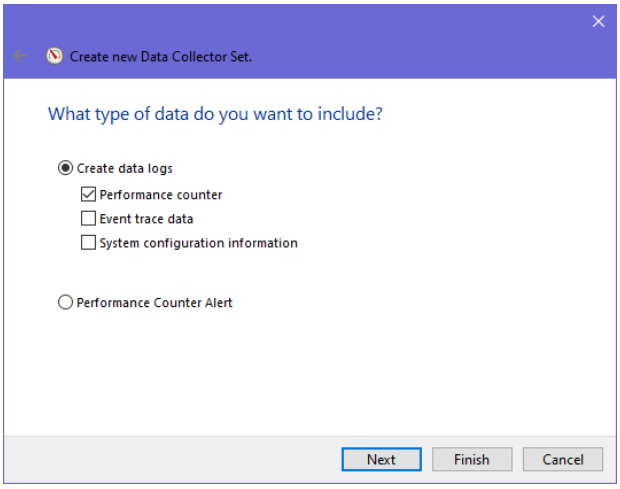


Figura 5. Selección del tipo de datos que se incluirán en el Data Collector Set. Fuente: Elaboración Propia.

A continuación, la aplicación solicita añadir los denominados “performance counters” que se encargan de medir el comportamiento de algún servicio, proceso o sección de hardware específica dentro del sistema. Los elegidos para este trabajo se presentan en la Figura 6.

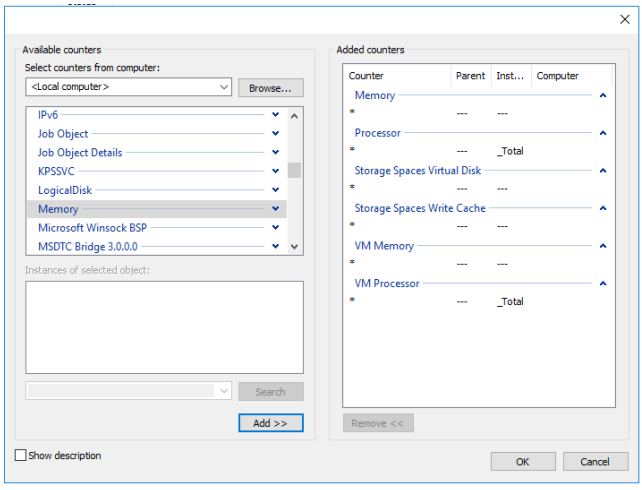


Figura 6. Seleccionando los Performance Counters. Fuente: Elaboración Propia.

Los elementos elegidos como Memoria, Procesador y el resto, serán explicados posteriormente. Una vez elegidos los elementos, se escoge un intervalo de tiempo en el que el

sistema recopilará los datos. Finalmente se escoge en qué directorio se desea guardar el archivo generado. El proceso de creación del Data Collector Set está completo y se lo encuentra junto al resto ya creados, tal y como se muestra en la figura 7. Sin embargo, aún se debe especificar el período de tiempo en el que el sistema estará recolectando los datos. Para esto se debe dar click derecho sobre el Data Collector Set y seleccionar Propiedades y posteriormente Schedule, aquí se escogerá un tiempo de inicio y finalización como se muestra en la figura 8.

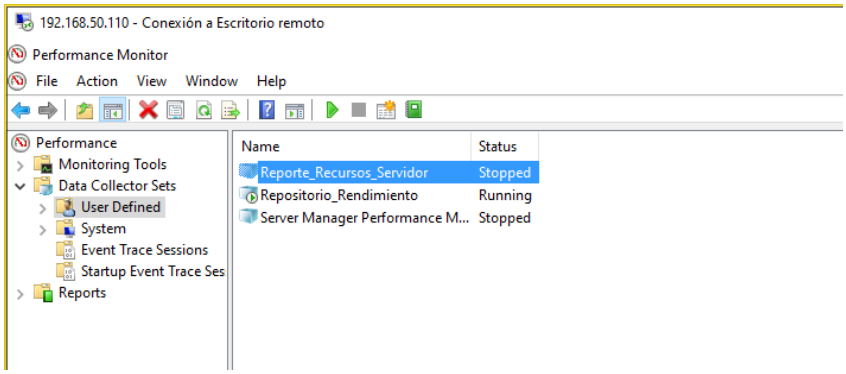


Figura 7. Data Collector Set Creado mediante el proceso detallado en este trabajo. Fuente: Elaboración Propia.

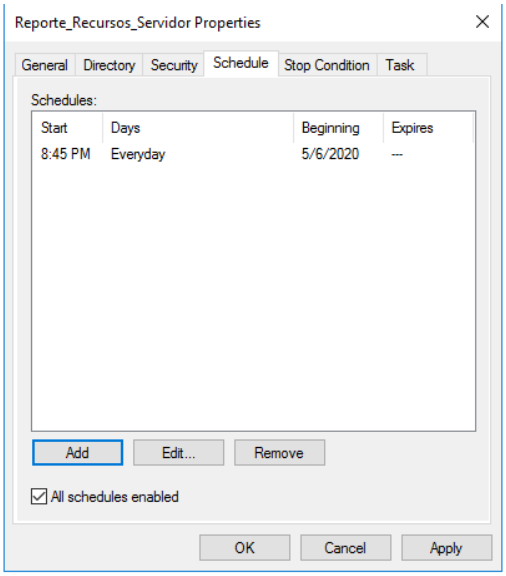


Figura 8. Selección de un horario para el Data Collector Set creado previamente. Fuente: Elaboración Propia.

Se añade el horario y se pulsa sobre el botón Apply. El Data Collector Set está listo para empezar a guardar datos del equipo. Sin embargo, aún se debe poner en marcha a este objeto para que el sistema empiece la recopilación. Para esto se da click derecho sobre el Data Collector Set creado y se selecciona la opción Start. Ahora el sistema ya se encuentra guardando los datos. Lo

único que queda por realizar es la conversión del archivo generado a un archivo mucho más amigable para poder trabajar posteriormente. Por esto, se decide que el archivo final tenga formato CSV. Mediante la consola de comandos de Windows se posiciona sobre el directorio elegido previamente para guardar el Data Collector Set y se ejecuta este comando: relog datacollector01.blg -f CSV -o logCPURAM.csv, el cual genera un archivo CSV a partir de los datos recopilados en la colección.

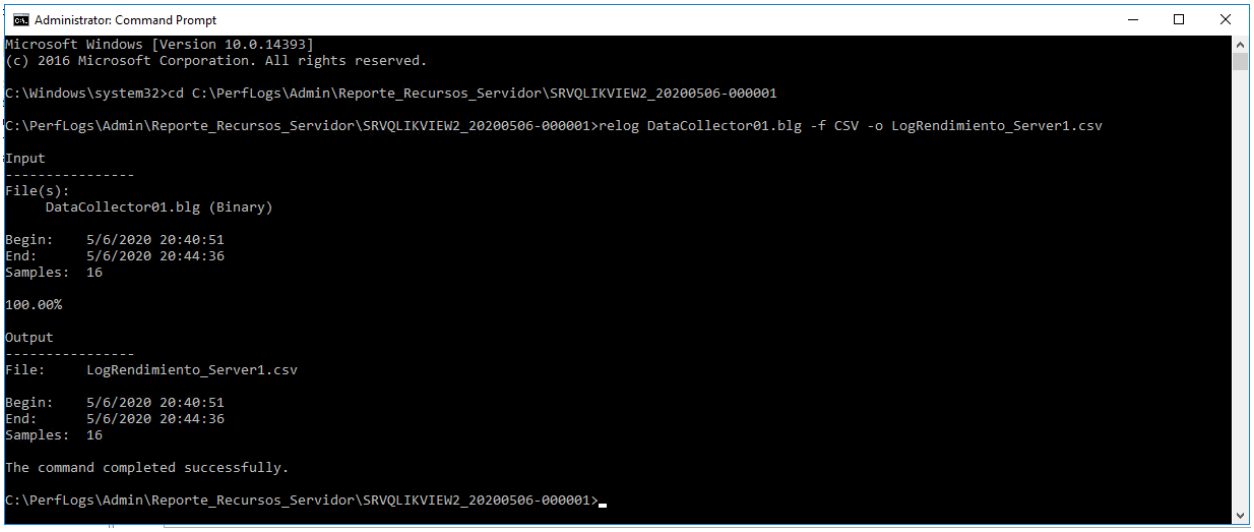


Figura 9. Comando para exportar los datos recopilados en un archivo CSV. Fuente: Elaboración Propia

2.2 Depuración y Estructura de los Datos

2.2.1 Depuración de los datos

Una vez que los datos han sido exportados, se tiene un archivo como el que se muestra en la Figura 10:

	A	B	C	D	E
1	(PDH-CSV 4.0) (SA Pacific Standard Time	\\SRVQLIKVIEW2\System\Processor Qu	\\SRVQLIKVIEW2\Processor(7)\% User T	\\SRVQLIKVIEW2\Processor(6)\% User T	\\SRVQLIKVIEW2\Processor(5)\% User T
2	42:54.0	0			
3	43:09.0	0	0.626302751	0.104383792	1.774524463
4	43:24.0	0	0.936561422	0	0.20812476
5	43:39.0	0	0.62503659	0	0
6	43:54.0	0	0.938485854	0.104276206	0.312828618
7	44:09.0	0	0.416338711	0.208169356	0.208169356
8	44:24.0	0	0.416911263	0	0.104227816
9	44:39.0	0	0.728562887	0	0.104080412
10	44:54.0	0	0.625471768	0.312735884	0.208490589
11	45:09.0	0	0.417082014	0	0
12	45:24.0	0	0.728465519	0.520332513	0.416266011
13	45:39.0	0	0.83384606	0.208461515	0
14	45:54.0	0	0.833085324	0.104135665	0.208271331
15	46:09.0	0	0.62496686	0	0
16	46:24.0	0	0.416232985	0.312174739	0.312174739
17	46:39.0	0	0.521108176	0	0
18	46:54.0	0	0.625422373	0.208474124	0.104237062
19	47:09.0	0	0.520352351	0.10407047	0.20814094

Figura 10. Datos exportados como CSV. Fuente: Elaboración Propia

Gracias a software como Excel podemos apreciar el archivo ordenado por filas y columnas, siendo las columnas todos los performances counters que fueron elegidos previamente. Se señala que cada archivo cuenta con 17280 registros o filas, debido a que se configuró el Performance Monitor para que se recabaran datos cada 15 segundos, por lo que el total de veces que el servidor recopiló datos fue de 5760 al día, cuyo valor se multiplica por el número de días que estuvo activa la aplicación, es decir en este caso tres. Este proceso fue aplicado en tres distintos servidores.

A partir de este punto se procede a realizar una depuración de los datos. Lo que se realizará es eliminar todos los valores individuales de los procesadores, es decir, los cores que tienen puesto que los valores del performance counters elegidos se despliegan por core. Esto se realiza por la diferencia de cores que tienen las CPUs de cada servidor. Además, la primera fila con los nombres de las columnas se editará para mostrar la referencia abreviada. También se procede a buscar valores que puedan estar vacíos. En la figura 11 se puede apreciar el archivo depurado correspondiente al servidor 1:

	A	B	C	D	E	F	G	H
1	Tiempo	Processor Queue Length	User Time (%)	Privileged Time (%)	Working Set	Disk Read Bytes/sec	Avg. Disk sec/Read	Current Disk Queue Length
2	43:09.0	0	0.495822678	0.52191896	67413975040	0	0	0
3	43:24.0	0	0.312187141	0.260155951	67412152320	0	0	0
4	43:39.0	0	0.221366959	0.234389054	67412180992	0	0	0
5	43:54.0	0	0.417104824	0.456208568	67412189184	0	0	0
6	44:09.0	0	0.208169356	0.234190192	67412004864	0	0	0
7	44:24.0	0	0.182399011	0.234512919	67412107264	0	0	0
8	44:39.0	0	0.247191146	0.22117071	67411279872	0	0	0
9	44:54.0	0	0.377889027	0.612441273	67411402752	0	0	0
10	45:09.0	0	0.182473047	0.208541007	67411222528	0	0	0
11	45:24.0	0	0.442282969	0.364232759	67412795392	0	0	0
12	45:39.0	0	0.312692272	0.364807651	67411566592	273.2341156	0.011126811	0
13	45:54.0	0	0.33844058	0.507660869	67411468288	0	0	0
14	46:09.0	0	0.234362906	0.22134293	67411369984	0	0	0
15	46:24.0	0	0.247138501	0.247137835	67411439616	0	0	0
16	46:39.0	0	0.195415066	0.338720648	67410132992	0	0	0
17	46:54.0	0	0.390889317	0.534214777	67411841024	0	0	0
18	47:09.0	0	0.234158225	0.20814094	67411476480	0	0	0
19	47:24.0	0	0.664750684	0.299789616	67411136512	0	0	0
20	47:39.0	0	0.221259341	0.273320401	67410694144	0	0	0
21	47:54.0	0	0.338229388	0.585396275	67410710528	0	0	0
22	48:09.0	0	0.169332354	0.182358433	67410460672	0	0	0
23	48:24.0	0	0.23453117	0.234530503	67409285120	0	0	0
24	48:39.0	0	0.299218309	0.260190268	67409305600	0	0	1
25	48:54.0	0	0.338828439	0.508242325	67415031808	0	0	0
26	49:09.0	0	0.195249488	0.234299519	67413794816	0	0	0
27	49:24.0	0	0.273620424	0.299679194	67413954560	0	0	0

Figura 11. Archivo CSV depurado. Fuente: Elaboración Propia.

2.2.2 Estructura de los Datos

Los performances counters elegidos son ciertas propiedades de los componentes más importantes de una computadora: la CPU, la memoria (RAM) y el disco. Establecer el significado de estos apartados es importante para entender cada variable en la posterior aplicación de algoritmos de Machine Learning.

La información que se muestra a continuación se basa en las publicaciones realizadas por AppAdminTools (2016), así como de las guías oficiales de Microsoft. Se debe tener en cuenta, además, que todos estos datos son orientados a sistemas Windows.

Memoria (RAM) y Cache:

- **Available MB:** el Contador de MB disponibles es el tamaño del pool de páginas disponibles en la RAM que el sistema usa para satisfacer requerimientos de nuevas páginas. Hay que recordar que una página es un bloque de memoria que posee una longitud fija y que corresponde a una única entrada en una tabla de paginación. Usualmente este valor se debe encontrar por encima del 10% de la memoria total, sin embargo, si se encuentra en valores inferiores a ese porcentaje no se puede determinar inmediatamente si la memoria del sistema es insuficiente, sino que se deben emplear otros counters como las Pages/sec para determinar esto.
- **Pool Nonpaged Bytes:** las páginas de memoria que el sistema usa se cuentan usando dos distintos counters: el Pool Nonpaged Bytes y el Cache Bytes. Este contador muestra loa cantidad de páginas residentes que no se pueden paginar en la RAM que el kernel usa.
- **Cache Bytes:** este contador muestra la cantidad de páginas residentes asignadas en la RAM que los hilos del kernel pueden abordar sin causar un Page Fault.
- **Pages/sec:** este contador tiene como propósito principal determinar si la memoria del sistema representa un cuello de botella. Cuando se observa un valor constante mayor a 20, es necesario monitorear de cerca el sistema y si muestra un valor superior a 50 es muy posible que el sistema se encuentre falto de memoria.
- **Page Reads/sec:** este contador también sirve para observar posibles problemas de memoria. La descripción oficial de Microsoft menciona que este contador es el ratio con el que el disco fue leído para resolver problemas errores de página graves y muestra el número de operaciones de lectura sin contar el número de páginas recuperadas en cada operación.
- **Page Writes/sec:** este contador hace referencia al ratio con el que se escribieron páginas al disco con el fin de liberar espacio en la memoria.

Procesos:

- **Working Set:** este contador hace referencia al número total de páginas residentes asignadas en la RAM que todos los procesos que se están ejecutando en el sistema pueden abordar sin causar un Page Fault (un error de página).

Paging File:

- **% Usage:** muestra el porcentaje de uso del archivo de página actual. Aunque no está relacionado totalmente con el performance de la máquina, es conveniente revisar que este valor no llegue nunca al 100% pues podría desembocar en errores del sistema graves.

CPU:

- **% Privileged Time:** es el porcentaje de tiempo transcurrido en el que todos los procesos han estado ejecutando código en modo privilegiado. Por ejemplo, cuando se llame a un sistema de Windows, éste usualmente se ejecutará en modo privilegiado para tener acceso a los datos privados del sistema. Aunque se puede configurar la ejecución de una aplicación en este modo, la mayor parte del porcentaje se deberá a los procesos propios del sistema operativo.
- **% User Time:** es el porcentaje de tiempo transcurrido en el que todos los procesos han estado ejecutando código en modo usuario. Este modo está designado para aplicación y varias clases de subsistemas.

Sistema:

- **Processor Queue Length:** este contador hace referencia al número de hilos en la cola del procesador. En máquinas de un solo core, un valor constante entre 5 y 10 indica que la carga de trabajo es mayor que la que el procesador puede soportar. En sistemas multinúcleo, como los tres servidores elegidos para el presente estudio, una buena regla general es dividir la longitud de la cola para el número de núcleos del sistema y usar la misma indicación anterior para comprobar qué tanta carga tiene el procesador.

Disco:

- **Disk Read Bytes/sec:** es el ratio al que los bytes son transferidos desde el disco en las operaciones de lectura.
- **Disk Write Bytes/sec:** es el ratio al que los bytes son transferidos desde el disco en las operaciones de escritura.
- **Avg. Disk Queue Length:** es el promedio de solicitudes de tanto lectura como escritura que se pusieron en cola para un disco seleccionado durante el intervalo de muestra.
- **Current Disk Queue Length:** es el número de solicitudes pendientes en el disco en el momento en que se recopilan los datos de rendimiento. Si se tienen valores constantes por encima de 5 por disco, esto puede indicar un cuello de botella en el subsistema del disco.
- **Avg. Disk sec/Read:** tiempo promedio en segundos en el que se leen datos del disco.
- **Avg. Disk sec/Write:** tiempo promedio en segundos en el que se escriben datos al disco.
- **Split IO/sec:** es la velocidad a la que las operaciones de I/O se dividieron en varias suboperaciones de I/O. Una I/O divida puede ser resultado de un tamaño demasiado grande para caber en una sola operación de I/O o que el disco sobre el que se está trabajando está fragmentado.

Una vez que se han comprendido los datos de cada columna y los datos han sido depurados, se puede empezar con el proceso de Machine Learning que se cubrirá en el siguiente capítulo. Cabe recalcar que los performance counters elegidos son significativos, pero se podrían elegir otros para determinar el rendimiento específico de alguna otra sección del sistema.

Capítulo 3: Implementación Modelos y Librerías – Machine Learning

3.1 Introducción

Machine Learning corresponde a una disciplina dentro del área de Inteligencia Artificial que parte de la creación de sistemas con aprendizaje automático, con el objetivo de descubrir complejos patrones gracias al uso de varios algoritmos que trabajan con conjuntos de datos para poder entrenarse y predecir futuros comportamientos de manera automática, por lo tanto se puede resumir que se tratan de sistemas de mejora constante y autónoma con aprendizaje no supervisado para la predicción de comportamientos futuros. En la actualidad existen diversos datos generados en empresas, incrementando exponencialmente la oportunidad de extraer información relevante para predecir comportamientos en varios ámbitos, incrementando su ventaja competitiva con el uso de algoritmos definidos y desarrollados. El presente trabajo se utilizará los modelos para la generación de patrones con los algoritmos de Clustering, Reglas de Asociación y Redes Neuronales para la predicción de comportamientos futuros con un conjunto de datos tomados de un servidor de un ambiente de producción.

3.2 Clustering:

Clustering trata de la partición o segmentación de un conjunto de datos no etiquetados en diversos grupos donde se prioriza la búsqueda de propiedades similares entre sí y que además tengan propiedades diferenciales con respecto a otros elementos para el aprendizaje automático. El nivel de complejidad dentro de un proceso de Clustering se basa en los criterios, reglas y normas para la generación de la agrupación, además de requerir un número de agrupaciones que van de acuerdo a la definición y al conjunto de datos de entrada.

3.2.1 Características principales

- Aprendizaje automatizado para el entrenamiento y descubrimiento de patrones entre un conjunto de datos con propiedades similares

- Generación de habilidades y capacidades para el análisis de grandes conjuntos de datos con un margen de error pequeño
- Agrupamiento y creación de subconjuntos de datos por medio de grafos con elementos diferenciadores con respecto a otros grupos
- Método con métricas de error con niveles aceptables durante el entrenamiento del algoritmo
- Proceso iterativo que permite interpretar los posibles errores dentro de los conjuntos de datos generados

3.2.2 Ventajas

- Depuración y selección efectiva de datos con el método de aprendizaje no supervisado
- Método efectivo para el uso de datos de gran escala
- Fácil implementación comparado con modelos de agrupamiento similares
- Adaptación al tipo de dato por medio de multiclases externas dentro del entrenamiento del modelo

3.2.3 Desventajas

- Es necesario de un algoritmo adicional para determinar cuántos clusters o agrupaciones serán las óptimas para la segmentación de información
- Requiere de un tiempo mayor comparado con otros algoritmos para determinar las características similares entre el conjunto de datos
- La agrupación del cluster depende directamente de la definición de las reglas y repercute en el resultado

3.3 Reglas de Asociación:

Reglas de asociación permite la búsqueda de comportamientos y patrones mediante el uso de reglas de asociación en base a un conjunto de datos, una regla de asociación se compone de un antecedente de un suceso y una consecuencia, por lo que un antecedente se encuentra en el conjunto de datos, mientras un consecuente nace por la combinación de un antecedente. El objetivo de las reglas de asociación es el buscar entre el conjunto de datos, comportamientos o patrones de uso recurrente con ayuda de criterios de confianza para determinar la regla con el menor margen de error, el nivel de confianza es la medida que representa el número de ocurrencias de tipo verdaderas, además, este algoritmo tiene la posibilidad de poder ajustar los modelos para obtener

un nivel de confianza mayor con el fin de determinar reglas de asociación con mayor precisión y de mayor aporte para concluir con una regla generada.

3.3.1 Características principales

- Generación de probabilidades resultantes de un conjunto de datos de alta cantidad de registros
- Búsqueda de relaciones entre datos por cada iteración definida para el algoritmo
- Vinculación de transiciones que contengan similitudes entre el conjunto de datos
- Descubrimiento de correlaciones por medio de reglas de asociación con un nivel de confianza de ocurrencia
- Identificación de asociaciones ocurrentes para concluir con reglas de asociación sobre un conjunto de datos

3.3.2 Ventajas

- Algoritmo de complejidad baja para la implementación y de fácil comprensión
- Método ideal para el uso de datos de gran escala
- Generación de reglas de asociación entre varios elementos del conjunto de datos
- Ordenamiento de reglas de asociación en base a su nivel de confianza y soporte

3.3.3 Desventajas

- Requiere de un tiempo considerable para determinar el valor para soporte debido a que usa conjunto de datos de gran escala
- Alta demanda de recursos de hardware para encontrar el número de posibles reglas por parte del algoritmo
- El modelo requiere de varias ejecuciones hasta poder ajustarse y tener un nivel de confianza y soporte aceptables

3.4 Redes Neuronales:

Redes neuronales son modelos que permiten generar aprendizaje por medio de la alteración o modificación de su estructura con la finalidad de poder resolver actividades con un nivel de complejidad alto para automatizar acciones que solo podían ser ejecutadas por personas. Una red neuronal es una mezcla de una serie de valores de entrada, mismas que luego se desplazan un nodo conocido como neurona, al mismo tiempo cada neurona contiene un dato numérico que representa su peso, al ejecutarse se realiza una modificación con el dato de entrada, los valores resultados continúan con el proceso por medio de sus capas ocultas definidas hasta llegar a un nodo salida que contiene la respuesta. Este valor final es la predicción realizada en la ejecución de la neurona artificial, una red neuronal aumenta su complejidad en función del número de capas que se definan para el modelo

3.4.1 Características principales

- Requiere de un entrenamiento previo para que la neurona artificial pueda realizar predicciones sobre un conjunto de datos
- Cada ejecución la neurona aprende y predice valores con mayor precisión incluso si se trata de un conjunto de datos diferentes al original
- Se trabaja con un modelo de capas y neuronas definidas previamente en función al conjunto de datos y a la complejidad
- Aprendizaje automatizado con una parte del conjunto de datos destinados para el entrenamiento de la red neuronal
- Cada nodo contiene un peso que corresponde a un valor numérico, este valor se realiza una operación con el dato de entrada para devolver un dato salida

3.4.2 Ventajas

- Modelo que determina patrones para la predicción de valores de un conjunto de datos de entrada
- Investigación y desarrollo en auge para nuevos modelos de redes neuronales
- Método ideal para el uso de conjunto de datos de gran escala
- Permite descubrir comportamientos que son complejos determinar por metodologías de programación clásica

3.4.3 Desventajas

- La base de los algoritmos tiene base matemática avanzada que es difícil de comprender y crear un modelo de red neuronal desde cero
- Es requerido una escala y normalización de datos para poder ejecutar el modelo de red neuronal correctamente
- El modelo requiere de varias ejecuciones hasta poder ajustarse por lo que demanda de muchos recursos de hardware en paralelo

3.5 Uso de Librerías:

En el presente trabajo se desarrollará la implementación de los 3 modelos de Machine Learning con la herramienta de Python versión 3.7.6 debido a que es un lenguaje interpretado, con semántica dinámica y de alto nivel que tomará el rol de kernel, adicional la herramienta Visual Code versión 1.59.0 como editor de código y finalmente Jupyter Notebook versión 6.3.0 para la ejecución de código por bloques para los 3 modelos. Las librerías para la implementación de los modelos escogidos se describen a continuación

3.5.1 Tensorflow

Tensorflow se trata de una biblioteca con herramientas de tipo código abierto para el desarrollo de modelos de aprendizaje automático basados en cálculos numéricos, además de contar con un ambiente colaborativo y flexible. Fue desarrollado por Google con el fin de poder ofrecer un servicio con redes neuronales en la nube para determinar patrones o comportamientos, en el presente trabajo se utilizará la API para Python para crear, construir y entrenar a la red neuronal que pueda determinar y predecir el comportamiento de los recursos de los servidores en base a su log de recursos. La ventaja de usar Tensorflow es el dinamismo a la hora de crear un modelo de red neuronal siendo más dinámico que la propia librería de Python Numpy, debido a que Tensorflow hace uso de grafos de cálculo de forma síncrona, por el contrario, Numpy solamente ejecuta las operaciones matemáticas el momento de la corrida del grafo. La versión que se está usando en el proyecto es 2.6.0, que fue lanzada y estabilizada el 15 de enero del 2020

3.5.2 Keras

Keras es un API de código abierto que permite la ejecución y aprendizaje de redes neuronales por medio de librerías de cálculos matemáticos complejos como Tensorflow o Theano, su funcion es proporcionar bloques modulares que ejecutan modelos de aprendizaje profundo en diversas plataformas modelos de predicción construidos y personalizados en diversos kernel y finalmente integrándose con servicios en la nube como Azure o Amazon AWS. Dentro del proyecto será el runtime como backend para la ejecución de Tensorflow al momento de iniciar la red neuronal, la versión que se está usando en el proyecto es 2.6.0 que fue lanzada y estabilizada el 17 de junio del 2020

3.5.3 Scikit-learn

Scikit-learn se trata de una librería open source de Python, con algoritmos que incluyen Clustering, regresiones lineales o exponenciales y de clasificación, la ventaja del uso de esta librería es la diversa cantidad de módulos y algoritmos que facilitan la construcción de modelos de aprendizaje no supervisado y automático, la mayoría de los algoritmos están optimizados por su construcción sobre núcleos particionados y en paralelo. La versión que se está usando en el proyecto es 0.24.2 que fue lanzada y estabilizada el 12 de enero del 2021

3.5.4 K-Means

K-Means se trata de una librería específica de Sklearn de tipo open source de Python, que contiene el algoritmo del aprendizaje no supervisado de Clustering, usado principalmente para la determinación de número de clusters o grupos se desea generar para categorizar el conjunto de datos de entrada. La librería además permite la iniciación tomando al azar un número de clusters y estos se van alterando hasta encontrar el óptimo para el número de registros en el conjunto de datos, luego la asignación de los datos a los grupos creados y finalmente la actualización de los clusters con respecto a los datos. La versión que se está usando en el proyecto es 0.24.2 que fue lanzada el 12 de enero del 2021

3.6 Ejecución del proyecto:

Para la ejecución de los 3 modelos de Machine Learning se utilizará un log de recursos sacados de un servidor de producción operativo y es virtualizado en el data center de PRONACA con su autorización en el anexo 1. En este log se ha logrado recuperar 17280 registros donde se registraron:

Memoria (RAM) y Cache:

- **Available MB**.
- **Pool Nonpaged Bytes.**
- **Cache Bytes**.
- **Pages/sec**.
- **Page Reads/sec**.
- **Page Writes/sec.**

Procesos:

- **Working Set.**

Paging File:

- **% Usage**.

CPU:

- **% Privileged Time**.
- **% User Time**.

Sistema:

- **Processor Queue Length**.

Disco:

- **Disk Read Bytes/sec**.
- **Disk Write Bytes/sec**.
- **Avg. Disk Queue Length**.
- **Current Disk Queue Length**.

- **Avg. Disk sec/Read**.
- **Avg. Disk sec/Write**.
- **Split IO/sec**.

Capítulo 4: Conclusiones y Recomendaciones

4.1 Conclusiones de los modelos de Machine Learning

4.1.1 Conclusiones de Clustering

Luego de la implementación del modelo con las librerías Sklearn y Keras, se logró completar un script completo y parametrizado para la corrida de la aplicación que permiten concluir lo siguiente:

- Incluir librerías Sklearn y K-Means dentro modelo de Clustering facilitó y optimizó la ejecución de operaciones matemáticas complejas durante la ejecución del script.

- El modelo de Clustering genera segmentación de datos en base a propiedades similares para su agrupación y distinción con propiedades diferenciales.

- El modelo logró analizar y agrupar el conjunto de datos del servidor virtual de PRONACA con 17280 registros.

- Clustering corresponde a un aprendizaje automático de tipo no supervisado, lo que hace que el análisis de los clusters generados sean orientados a ser subjetivos.

- K-Means es un algoritmo ideal para conocer cuál es el número de clusters es el óptimo para el modelo cluster dentro de un rango proporcionado.

- Normalizar los datos antes de integrarlos al modelo para su ejecución es buena práctica para evitar datos que salgan del rango entre 0 y 1.

- La limitante de hardware hace que los tiempos de ejecución del modelo se incremente del tiempo promedio sugerido por las librerías.

- El manejo de las librerías no requiere de un conocimiento a profundidad de las bases matemáticas del modelo de Clustering, por lo que es suficiente revisar el foro oficial de Python

4.1.2 Conclusiones de Reglas de Asociación

Del mismo modo, realizado el script funcional para reglas de asociación se puede concluir lo siguiente:

- El algoritmo Apriori extrae reglas por medio de la combinación entre antecedentes y consecuentes de todo el conjunto de datos de entrada.

- El modelo reglas de asociación encontró patrones/ocurrencias dentro del conjunto de datos del servidor virtual de PRONACA.

- Los antecedentes están presentes en el conjunto de datos y los consecuentes se forman con la combinación de este con el antecedente para generar la regla de asociación.

- La ejecución de reglas de asociación requiere de los componentes: soporte, confianza y ajuste para la validación de la certeza de las reglas generadas.

- El manejo de probabilidades resultantes de las reglas generadas permite verificar la confianza de las ocurrencias de las reglas de asociación.

- Para obtener un resultado de confianza alto es necesario realizar más de una vez el algoritmo hasta considerar el más alto de todas las ejecuciones.

4.1.3 Conclusiones de Redes Neuronales

El modelo de redes neuronales se lo armo con ayuda de las librerías Tensorflow y Keras, por lo tanto, se concluye lo siguiente:

- Redes neuronales son modelos predictivos que generan aprendizaje en cada una de las iteraciones donde viajan los datos entre capas y nodos.

- La complejidad de una red neuronal aumenta en función del número de capas ocultas y del número de nodos de entrada

- Cada nodo tiene un valor numérico de peso mismo que sirve para que la red neuronal pueda realizar los cálculos e indicar la predicción de resultado.

- Comportamientos que son complejos los puede determinar una red neuronal que a diferencia una metodología tradicional de programación no podría.

- La limitación para la ejecución de redes complejas con más de 3 capas ocultas es el hardware del equipo.

- La red neuronal pudo manejar excepciones y datos que no estaban normalizados desde un principio, a pesar de no ser obligatorio una normalización de los datos se pudo procesar.

- Tensorflow es una librería completa para el uso de cálculos matemáticos complejos con grafos y recorridos que simplifica la construcción de un modelo de red neuronal.

- Keras permite el funcionamiento nativo de Tensorflow en su runtime sin embargo su propósito es para redes neuronales más complejas dedicadas al Deep Learning

4.2 Ejecución conjunto dato PRONACA

Con la autorización de la compañía se puedo usar los tres algoritmos con datos reales de un servidor de pruebas, que contiene ya registro de uso de recursos día a día, lo que se puede concluir es lo siguiente:

- Los tres scripts desarrollados en Python funcionaron y lograron sacar el resultado predictivo acerca de los recursos del servidor, considerando que fue un conjunto de datos de 17280 registros el algoritmo tardo un mayor tiempo más que del tiempo promedio durante el desarrollo de los algoritmos.

- PRONACA cuenta con una infraestructura grande, por lo que hay servidores virtuales que pueden ser analizados con cualquier algoritmo para tratar de predecir su comportamiento.

- La ventaja de trabajar con servidores virtuales como es el caso de PRONACA permite que recursos si se disparan se puede tomar acciones ese momento y realizar un cambio sin necesidad de apagar el servidor y parar las operaciones.

- El servidor presenta un comportamiento de incremento de memoria en un lapso de dos horas y es correcto debido a que en ese servidor existe una aplicación que cada dos horas.

ejecuta una carga de información por ETL, por lo que la predicción por parte de la red neuronal de que la memoria se incrementará es correcta.

- El alcance de la aplicación es el de predecir el comportamiento de un servidor, sin embargo, se podría desarrollar una fase 2 donde se pueda considerar servidores UNIX o Linux para su extracción de logs de recursos y predicción de recursos.

- El servicio AppAdminTools de Windows Server se puede programar para que siempre esté activo tomando logs de sus recursos así se puede disponer de cualquier momento del archivo .csv para poder ser analizado por los modelos.

4.3 Recomendaciones para proyectos de Machine Learning

En el proceso deconstrucción de los modelos realizados con las últimas versiones estables para Machine Learning ha permitido exponer unas recomendaciones descritas a continuación:

- Se debe realizar un preanálisis para determinar las ventajas, desventajas de las librerías a utilizar para poder hacer una correcta elección sobra las mismas, el objetivo siempre será el facilitar o simplificar el trabajo y el autor poder enfocarse en lo crítico utilizando lo ya creado por grupos de código abierto.

- Es necesario analizar el dato que se desea tomar para poder predecir los datos puestos que deben ser datos dependientes e independientes para poder hacer un correcto entrenamiento en las redes neuronales.

- Existen diversas formas de armar modelos de Machine Learning, sin embargo, siempre es correcto ver los caminos más simples y funcionales, puesto que ahorrará tiempo que luego será aprovechado con nuevas funcionalidades o nuevos alcances.

- La distribución de tiempo para construir proyectos de Machine Learning se debe priorizar el análisis el mayor tiempo, ya que con ello se definirá mejor las actividades y serán más sencillas de realizar.

- Realizar pruebas con herramientas que tengan el mismo ámbito sirve como referencia para conocer si se está cumpliendo las actividades y los resultados esperados.

Bibliografía

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press.

Advani, V. (2020). *What is Machine Learning?* My Great Learning.

Bonaccorso, G. (2017). *Machine Learning Algorithms: Reference guide for popular algorithms for data science and machine learning.* Reino Unido: Packt Publishing Ltd.

Bonnin, R. (2017). *Machine Learning for Developers: Uplift your regular applications with the power of statistics, analytics, and machine learning.* Reino Unido: Packt Publishing Ltd.

Boschetti, A., & Massaron, L. (2016).). *Python Data Science Essentials: Become an efficient data science practitioner by understanding Python's key concepts (2nd ed.).* Reino Unido: Packt Publishing Ltd.

Deshpande, A., & Kumar, M. (2018). *Artificial Intelligence for Big Data: Complete guide to automating Big Data solutions using Artificial Intelligence techniques.* Reino Unido: Packt Publishing Ltd.

Fandango, A. (2017).). *Python Data Analysis: Data manipulation and complex data analysis with Python (2nd ed.).* Reino Unido: Packt Publishing Ltd.

Guo, Y. (2017, August 31). *The 7 Steps of Machine Learning*. Retrieved from Towards Data Science: https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e

Kubat, M. (2015). *An Introduction to Machine Learning.* Springer.

Moreno, M., Segrera, S., & López, V. (2005). Association Rules: Problems, solutions and new applications. *Universidad de Salamanca*, 8.

Natingga, D. (2017). *Data Science Algorithms in a Week: Data analysis, machine learning, and more.* Reino Unido: Packt Publishing Ltd.

Russell, S., & Norvig, P. (2010). *Artificial Intelligence - A mondern Approach.* Prentice Hall.

Anexos

Quito, 17 de Agosto del 2021

PRONACA
Te alimenta bien

CARTA DE AUTORIZACIÓN DE USO DE INFORMACIÓN

Por medio de la presente, autorizo al señor: **Andrés Eduardo Lara Hidalgo** con cédula de identidad: **1751984798** para el uso exclusivo de la información de logs de recursos del servidor de pruebas del registro de alimento de la compañía **PRONACA**, para el trabajo de titulación: "**Aplicativo basado en Machine Learning que detecta fallos en servidores y sugiere soluciones**". Quedo a sus órdenes y envío un cordial saludo.

Atentamente

RUBEN LINDEMBERG RECALDE BOLAÑOS

Ruben Lindemberg Recalde Bolaños

Director IT – Dirección de Tecnología de Información

Teléfono: (02) 3976-400 Ext. 1128

Correo: rrecalde@pronaca.com