



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

OPTIMIZACIÓN DEL PROCESO DE SOLICITUD DE PEDIDOS PARA
FRANQUICIAS DE FARMACIAS

TESIS PARA OPTAR AL GRADO DE MAGISTER
EN TECNOLOGÍAS DE LA INFORMACIÓN

MARCELO ALEJANDRO MIRANDA WILLIAMS

PROFESOR GUÍA:
JORGE PEREZ ROJAS

MIEMBROS DE LA COMISIÓN:

AIDAN HOGAN
JEREMY BARBAY
MARCOS SEPÚLVEDA FERNÁNDEZ

SANTIAGO DE CHILE
2020

Resumen

Esta tesis trata la problemática de dos locales de franquicia de farmacia en la búsqueda de obtener un mejor margen (diferencia entre compra y venta), sin incluir sueldos, arriendos y gastos operaciones. Dentro de las variables que inciden en el margen, solamente la solicitud de pedido de mercadería es gestionable por parte del franquiciatario, dado que los precios, el proveedor y la publicidad es manejada por el franquiciante (es decir, el franquiciatario no puede innovar). Inicialmente el margen es de 22% y el objetivo es aumentar en un 8% para obtener un margen del 30%, considerando que teóricamente el margen ofrecido por la franquicia es 35%.

Entonces, para lograr el objetivo general se plantea un objetivo específico de crear un modelo de aprendizaje automático que pueda predecir la venta de la siguiente semana y en base a ello realizar el pedido. Para ello entrenamos un modelo de regresión con una matriz de features obtenida de datos históricos que considera las estaciones del año y las ventas semanales por producto. El modelo finalmente elegido es en base a redes neuronales profundas que entregó el mejor resultado en la medida de desviación estándar promedio (DEP) y es reentrenado mensualmente para asumir las variaciones de las ventas y el clima.

La medida de desviación estándar promedio (DEP) obtenida en los locales es: 0,78 en local de Santa Rosa y 0,95 en local de Vespucio Norte.

La efectividad del modelo en los últimos 5 meses logrando una mejora del 7% en el primer local (en adelante “local de Santa Rosa”) y del 5% en el segundo local (en adelante “local de Vespucio Norte”), lo que no logra el objetivo de aumento del 8% del margen. El motivo de no logro del objetivo del 8% esperado, se encuentra en la estrategia agresiva de venta que se llevó a cabo durante los meses de junio, julio y agosto, donde en adición a la cantidad predicha se aumentó el pedido en un 2% en local de Santa Rosa y en un 5% en local de Vespucio Norte sin generar el efecto esperado de aumentar ventas.

Como conclusión, el trabajo ha sido desafiante para encontrar un modelo de aprendizaje automático que tuviese mayor efectividad y junto con ello modificar el flujo de pedido. El resultado si bien no cumplió el objetivo permitió un avance de 5 y 7 puntos en el margen de cada farmacia respectivamente, lo que nos ha generado mayores ganancias como empresa.

Agradecimientos

En este trabajo, que ha sido bastante duro, quiero entregar mis agradecimientos a las personas que me han apoyado en estos dos años para lograr el objetivo de finalizar la tesis de grado de magister en tecnología de información. En primer lugar, quiero dar un especial agradecimiento a mi familia, mi esposa que siempre me impulsó a finalizar, además de ayudar en las tareas del hogar mientras mi dedicación estaba en la tesis, también a mis hijos pequeños por entender que algunos fines de semana el papá debía dedicarse a terminar el trabajo.

Por lo anterior, a ellos (esposa e hijos) dedico mi trabajo.

Por otra parte, un gran agradecimiento a mi profesor guía Jorge Perez, quien me ayudó bastante a dilucidar como debía abordar la tesis, así como también a entregarme conocimiento y alternativas para lograr el modelo que mejoró el margen de la empresa, así como también al cuerpo docente de la Universidad de Chile, en especial a mi tutor Daniel Perovich que me ayudó e impulsó a finalizar esta tesis en el período de dos años académicos.

Por ello vuelvo a dar las gracias a todos.

Tabla de contenido

1. Capítulo 1: Introducción	1
2. Capítulo II: Contexto de operación actual de la empresa	3
2.1 Descripción de la empresa	3
2.2 Operación del pedido de mercadería	4
2.3 Proceso de solicitud de pedido actual	5
3. Capítulo III: Marco teórico	7
3.1 Análisis de datos	7
3.2 Aprendizaje supervisado	7
3.3 Aprendizaje no supervisado	13
3.4 Aprendizaje reforzado	17
3.5 Redes Neuronales	18
4. Capítulo IV: Predicción de pedido a solicitar semanalmente	25
4.1 Problema a Abordar	25
4.2 Desarrollo de la solución	26
4.2.1 Manejo de inventario convencional	26
4.2.2 Análisis de datos con aprendizaje de máquinas	28
4.3 Verificaciones del modelo	36
5. Capítulo V: Conclusión	38
Bibliografía	40
Apéndices	41
Apéndice A: Código creación de la red neuronal	41
Apéndice B: Código creación de la red neuronal	43
Apéndice C: Gráficas de resultado red neuronal	45

1. Capítulo 1: Introducción

La problemática que se busca resolver en la tesis es mejorar el margen de ganancia de dos franquicias de farmacia ubicadas en distintas comunas de Santiago: franquicia Santa Rosa ubicada en la comuna de San Joaquín y franquicia Vespucio Norte ubicada en la comuna de Recoleta. Ambas pertenecientes a la empresa Sociedad De Inversiones Negocios Asociados SPA. (Alpha Business).

Para dar el contexto a la problemática, es necesario revisar la situación del mercado de las empresas farmacéuticas, que ha estado liderado por las tres grandes marcas que tienen el 42,99% del mercado (Cruz Verde, Ahumada y Salcobrand). Después se encuentra Farmacias De Similares con 4,88%, y un 52,13% corresponde a farmacias independientes y de otras marcas. En cuanto a la venta, las farmacias de cadena (incluyendo a Farmacias De Similares) reciben 89,7% de las ventas del mercado, dejando un 10,3% para las farmacias independientes (Fuente: Estudio de Georesearch, julio 2018). Dentro de este mercado se encuentran las franquicias que tienen razón social particular o comercial pero que hacen parte de la marca, y se deben regir por los estándares del franquiciante. Las cadenas de farmacias que se franquician son: Cruz Verde y Farmacias De Similares (Farmacias SIMI), donde en Farmacias SIMI el 50% de sus locales son franquicias con ventas que aumentan año a año en promedio un 20%, lo que propone un atractivo proyecto de inversión.

En términos contractuales, el franquiciante no permite elegir el proveedor, manejar los precios de venta y determinar la publicidad, pero como empresa (Pyme), es responsabilidad del franquiciatario la contratación del personal, gastos operativos y administración de los locales. Bajo esta figura el franquiciante provee los productos de farmacia con un 35% promedio de descuento (lo que constituye el margen teórico de la diferencia entre compras y ventas). Teniendo en cuenta lo anterior, las dos franquicias alcanzan un margen mensual del 22%, esto genera pérdidas de ventas, en el acto que un cliente solicita un producto que no tenemos en stock, lo que además de impedir la venta inmediata provoca con cierta probabilidad que el cliente no vuelva si en la competencia encuentra con más frecuencia el producto.

Considerando el contexto anterior, se tiene que la única variable gestionable es la solicitud de pedido de productos, dado que, el franquiciatario puede gestionar qué productos solicitar y la cantidad de cada uno de ellos, con la restricción de una cantidad mínima a solicitar (que se llama empaque) semanalmente o cada dos semanas (dependiendo de la indicación del franquiciante). El pedido en sí, se realiza un día en específico que impone el franquiciante, lunes para la franquicia Santa Rosa y miércoles para la franquicia Vespucio Norte, donde la información de los productos a pedir es cargada en el sistema de franquiciante en base al juicio experto del director técnico de cada local, quién con su experiencia, tomando en consideración del período del año, las ventas actuales, indica los productos a solicitar y la cantidad de cada uno de ellos (considerando la cantidad mínima de producto a pedir). Una vez enviada la solicitud, el pedido llega en dos días (miércoles para local de Santa Rosa y viernes de para local de Vespucio Norte).

Entonces para lograr la mejora se fija un objetivo general que es de disminuir el error en el proceso de solicitud de pedido en aproximadamente un 8%, lo que permitirá lograr un margen del 30%. Para lograr el objetivo general se tienen los siguientes objetivos específicos:

1. Generar un modelo supervisado de aprendizaje automático que permita proponer un pedido en cantidad de productos óptimo para cada semana.

2. Entregar una planilla en Excel que indique al director técnico el pedido a solicitar, incluyendo lo entregado por el modelo, junto con el inventario y empaques (cantidad mínima de producto a pedir), de tal manera que pueda agregar, modificar o eliminar productos dentro en un 5% para considerar ventas seguras no predichas.

En base al contexto y los objetivos planteados se hace la pregunta ¿Cómo logramos mejorar el margen en base a optimizar la única variable posible de gestionar que es la solicitud de pedido?. Para responder esta pregunta se debe tener en cuenta todo el contexto en el que se mueve una franquicia con el fin de generar mayores ganancias a la empresa y no tener un sobre-stock de productos que pueden permanecer en el tiempo sin vender en local, pudiendo llegar a vencer, lo que impacta seriamente en los costos, dado que el franquiciante sólo devuelve el 50% del producto vencido.

Para lograr la optimización, el proyecto se realizará bajo una metodología ágil que permite realizar entregas parciales, mejorando la predicción en cada entrega para lograr el margen objetivo. Para ello, en este documento de tesis el proceso para lograr la solución se explicará en tres capítulos.

El primer capítulo abordará el funcionamiento de la empresa Alpha Business y de la relación con la marca franquiciante, lo que entregará el contexto de dónde y cómo se mueve la empresa en su operación actual.

En un segundo capítulo, se realiza un enfoque del marco teórico que soporta a la solución propuesta, donde se exploran las definiciones, técnicas y prácticas que sirven de base para lograr la operación modelo que optimiza el margen. Se incluyen argumentos de análisis de datos, machine learning, redes neuronales y deep learning.

En el tercer capítulo, se desarrolla la solución utilizando técnicas de machine learning, presentando los resultados y las verificaciones que permiten demostrar que la respuesta ha mejorado los márgenes de la empresa, junto con lograr una mayor efectividad en la venta, aportando así a aumentar el beneficio de la empresa.

El último capítulo está dedicado a la conclusión, que revelará la respuesta concreta a la pregunta planteada, la fórmula aplicada hoy y el futuro que se espera en una mejora continua.

El proyecto como tal se refiere a una primera versión que se utiliza en el contexto de la solución y que plantea una mejora continua en base al análisis periódico de datos para ajustar el modelo a las condiciones cambiantes que se producen en un negocio tan desafiante como es la venta de medicamentos.

2. Capítulo II: Contexto de operación actual de la empresa

La empresa Farmacéutica De Similares (Farmacias SIMI) es una empresa mexicana constituida el 8 de septiembre de 1997, que inició operaciones en nuestro país en mayo de 2005, y su característica principal es especializarse en los principios activos de los productos, lo que permite obtener un mejor precio y calidad, independiente de las marcas de laboratorio (Fuente: farmacias SIMI, <https://www.farmaciasdoctorsimi.cl/historia>). Este modelo de negocio ha permitido a la organización posicionarse como la cuarta empresa farmacéutica en Chile, con más de 200 locales en todo el país.

La empresa ha franquiciado cerca de 100 farmacias en el país (que son parte de las 200 farmacias totales). Las franquicias cuentan con la particularidad (que las distingue de las corporativas) que existen dueños y representantes legales independientes de la corporación SIMI, y que obtienen el derecho a utilizar la marca por 5 años (renovable). Las obligaciones contractuales de los franquiciatarios para con Farmacias SIMI, entre otras, son las siguientes:

1. Utilizar la imagen corporativa en el diseño y construcción de la farmacia.
2. Considerar a un proveedor único que es la misma empresa Farmacéutica De Similares (Droguerías SIMI).
3. Cumplir las normas legales exigidas por las entidades reguladoras (ISP y Seremis).
4. Cumplir en sala de ventas con la forma de atención al público indicada por Farmacias SIMI (se atiende personas no números).
5. Cumplir con un 90% de los productos disponibles en sala de ventas.

Bajo este marco se desempeña la empresa Sociedad De Inversiones Negocios Asociados SPA (Alpha Business) que será el objeto del proyecto.

2.1 Descripción de la empresa

La empresa Sociedad De Inversiones Negocios Asociados SPA, Alpha Business por su alias, es una empresa SPA que nació en agosto de 2016 dedicada al rubro de la venta de medicamentos con giro de “Farmacias independientes” con una inversión inicial de \$35.000.000. La organización en su fecha de nacimiento logró la compra de la primera franquicia de SIMI que es Santa Rosa (F0170) ubicada en la comuna de San Joaquín. Al siguiente año (octubre de 2017) considerando los buenos resultados la empresa logró adquirir la franquicia ubicada en la plataforma superior del metro Vespucio Norte comuna de Recoleta, por lo que en la actualidad cuenta con dos locales en dos comunas diferentes:

1. Santa Rosa 2 (F0170): Ubicada en la comuna de San Joaquín y cuenta con los siguientes horarios:
 - Lunes a viernes: de 9:00 a 20:30 hrs.
 - Sábado: de 10:00 a 19:30 hrs.
2. Vespucio Norte (F0190): ubicada en la comuna de Recoleta en la plataforma superior del metro Vespucio Norte y cuenta con los siguientes horarios:
 - Lunes: de 7:30 a 20:30 hrs.
 - Martes a viernes: 7:30 a 20:00 hrs.
 - Sábado: 8:00 a 16:30 hrs.

Las dos franquicias en su operación conjunta de dos años logran un volumen aproximado de venta de 30 millones de pesos mensuales y cuentan con siete trabajadores contratados en total, entre los que se cuenta con:

1. Dos Químicos Farmacéuticos (QF): de acuerdo con la disposición del Instituto de Salud Pública (ISP), toda farmacia debe contar con un Químico farmacéutico como director técnico todo el período en que la farmacia se encuentre abierta. Existen un QF por local.
2. Cuatro vendedores: estos se dividen en dos vendedores por local con turnos definidos distribuidos en la totalidad de la jornada laboral.
3. Una administradora: encargada de gestionar recursos, turnos, ingresos y egresos.

2.2 Operación del pedido de mercadería

La operación de las farmacias consta de un flujo de negocio que parte con la solicitud de productos y finaliza con la venta del producto al cliente final, todo bajo la normativa que impone tanto la autoridad (Instituto de Salud Pública (ISP (<https://www.ips.gov.cl/>)) y Farmacias De Similares).

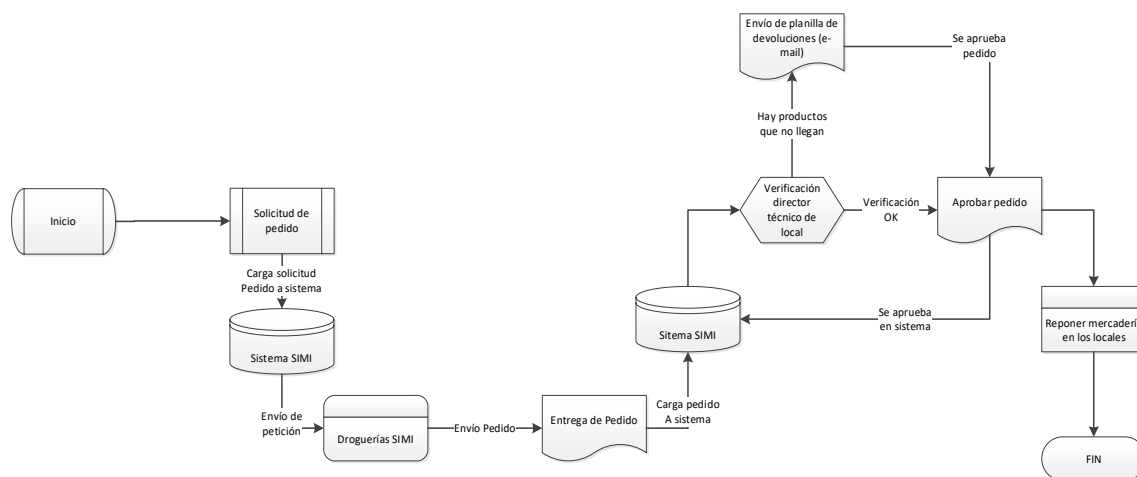


Figura 1: Flujo de compra venta de mercadería

Como muestra la figura anterior el flujo comprende lo siguiente:

1. Solicitud de Pedido: esta etapa consta de varios procedimientos para solicitar el pedido como son:
 - a. Estimación del monto para el pedido de mercadería: la estimación se realiza en base a estadísticas de ventas históricas del período, venta de la semana anterior, monto del último pedido que quedará de remanente. Esta estimación es entregada por el franquiciatario al director técnico de la farmacia (Químico Farmacéutico).
 - b. El director técnico realiza la solicitud de pedido incluyendo los productos necesarios de acuerdo con su experiencia.
 - c. Ingreso al sistema: se realiza la digitación del pedido en el sistema de Farmacias SIMI, y este se envía como definitivo a Droguerías SIMI para su carga.
2. Entrega de Pedidos: Droguerías SIMI realiza carga del pedido considerando:
 - a. Lo solicitado por el local.

- b. Los empaques (cantidad mínima de producto que se despacha), es decir, si hemos pedido 10 y la cantidad mínima de pedido es 20, Droguerías SIMI envía 20.
 - c. El stock de los productos en droguería: Droguerías SIMI revisa el pedido solicitado y envía a farmacia los productos que cuenta en stock junto con los productos nueva inclusión (que los envía sin pedirlos).
 - d. Droguerías SIMI factura el pedido y lo despacha a los puntos de venta.
3. Verificación director técnico de local: el director técnico recibe el pedido y revisa las existencias contra factura. Si existen diferencias, gestiona el procedimiento de “Faltante de Almacén”, que consiste en indicar qué productos no arribaron a la farmacia, para solicitar una nota de crédito. Paralelamente también el director técnico informa al supervisor zonal y al franquiciatario, de los productos que son solicitados y no despachados, y los productos que no son solicitados e igualmente despachados. Finalmente se aprueba el pedido en sistema y se carga al inventario del local.
4. Reponer mercadería en los locales: el director técnico repone la mercadería en la farmacia, de acuerdo con el estándar indicado por el ISP y Farmacias SIMI. Con ello los vendedores comienzan el proceso de venta al público, lo que finaliza el flujo.

Los actores involucrados en los procesos son:

- Vendedores de local
- Director técnico
- Administrador farmacias
- Franquiciatario
- Supervisor directo de la franquicia
- Gerente de franquicias
- Central de despacho

Con este detalle, en el siguiente apartado pasaremos a detallar de cómo se realiza actualmente la solicitud de pedido

2.3 Proceso de solicitud de pedido actual

En este apartado, se revisa el proceso actual de solicitud de pedido que es nuestra variable a optimizar para lograr una mejora sustancial en nuestro margen. Entonces en la actualidad el proceso para la solicitud de pedido se realiza en base al “Juicio Experto” del director técnico de cada local (químico farmacéutico) y bajo un esquema muy poco definido, donde el director técnico tiene todas las facultades de agregar o quitar productos en base a su experiencia técnica y conocimiento del lugar de venta, siempre con un monto predefinido que entrega el gerente general de la empresa. Ahora cada local tiene un día diferente para pedir, entonces:

1. Santa Rosa 2 (F0170): el día de pedido es todos los lunes y la entrega de la mercadería es los miércoles.
2. Vespucio Norte (F0190): el día de pedido es todos los miércoles y la entrega de la mercadería es los viernes.

La siguiente figura muestra el flujo del pedido actual general para los locales.

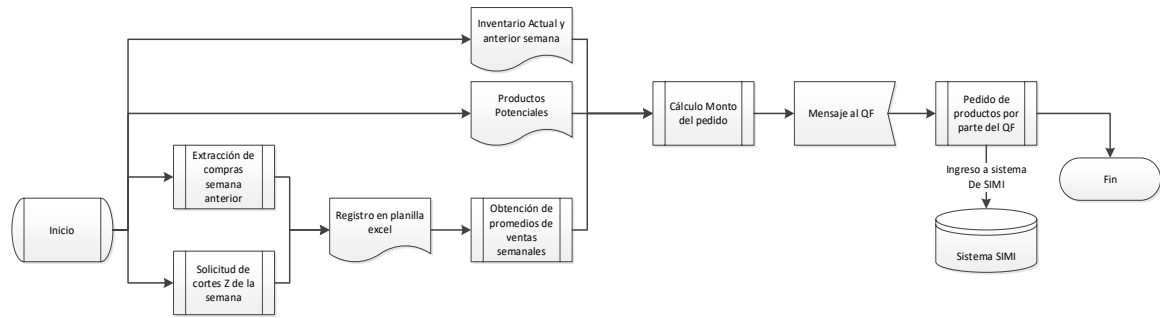


Figura 2: Flujo de pedido para cada local

Como muestra la figura anterior, el flujo consta de las siguientes etapas:

1. Extracción de compras semana anterior: Se registran los montos de las facturas de la semana anterior, de manera de tener en cuenta el monto que se ha solicitado para la semana.
2. Solicitud de cierre de ventas diarios (cortes Z) de la semana: se verifican los cortes Z de la semana que corresponden a las ventas diarias para saber el monto semanal vendido.
3. Registro en planilla excel: Se registran los datos de compras semana anterior y cortes Z en planilla Excel.
4. Obtención de promedios de ventas semanales: una vez registrados en planilla se calcula el promedio de venta de cada día y se realiza una suma que da una primera estimación de la venta de la próxima semana.
5. Productos potenciales: Se ingresan en la planilla los productos que se han solicitado por los clientes en la semana, que no tenemos en stock en sala de venta y que a su vez Droguerías SIMI lo tiene en stock.
6. Inventario Actual y de la semana anterior: Se refleja en la planilla el inventario actual al día de solicitud de pedido y el inventario al día de pedido de la semana anterior, se realiza una resta entre ellos para saber si el inventario ha aumentado (lo que significa que la venta ha disminuido esta última semana y al contrario si el inventario ha disminuido, significa que la venta ha aumentado en la última semana).
7. Cálculo de monto del pedido: Considerando la “Obtención de promedios de ventas semanales”, “Inventario Actual y de la semana anterior” y sumando los “productos potenciales” se realiza el cálculo del monto de pedido, que se envía como mensaje al químico farmacéutico, de manera que se tome como el máximo posible a pedir.

Todos estos puntos los realiza la dirección de la empresa sin intervención del químico farmacéutico o personal del local, enviado el monto comienza el trabajo del químico farmacéutico.

8. Pedido de productos por parte de un químico farmacéutico: una vez obtenido el monto, el químico farmacéutico en base a su experiencia y considerando: los productos que no tienen stock suficiente para la venta en sala, productos potenciales, empaques (cantidad mínima a pedir de cada producto) y faltantes de inventario de Droguerías SIMI. Este pedido se digita y confirma en el sistema y confirma en el sistema SIMI, lo que determina el fin del proceso.

3. Capítulo III: Marco teórico

El marco teórico (cuya fuente es el libro Python Machine Learning, segunda edición, 2019, Deep learning with Python, 2018 y Neural Network Methods for Natural Language Processing, 2017) en que se enmarca esta tesis engloba conceptos de análisis de datos como data mining y machine learning que nos ayudarán a determinar la mejor optimización de la solicitud del pedido considerando historia y cálculos.

3.1 Análisis de datos

En este apartado veremos las técnicas para analizar esos datos para lograr determinar patrones que nos permitan aprender de ellos y, con una alta probabilidad predecir acontecimientos futuros. Dentro de este espectro podemos encontrar tres tipos de aprendizaje automático como son: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje reforzado cuyas diferencias fundamentales se muestran en la siguiente figura:

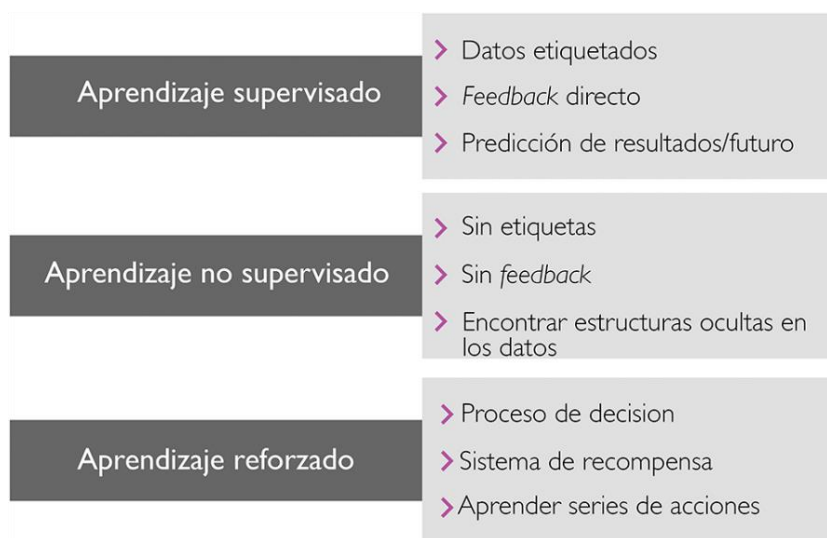


Figura 5: Tipos de aprendizaje.

Nota. Imagen adaptada del libro Python Machine Learning, segunda edición, 2019.

A continuación, se revisa cada uno de ellos, deteniéndonos en particular en el aprendizaje supervisado y no supervisado, junto con las redes neuronales que servirán de base para el trabajo.

3.2 Aprendizaje supervisado

El objetivo principal del aprendizaje supervisado es aprender un modelo a partir de datos de entrenamiento etiquetados, que nos permite hacer predicciones sobre datos futuros o no vistos. El término supervisado se refiere a un conjunto de muestras donde las salidas (etiquetas) son conocidas y en base a ellas se puede entrenar el modelo para predecir salidas futuras.

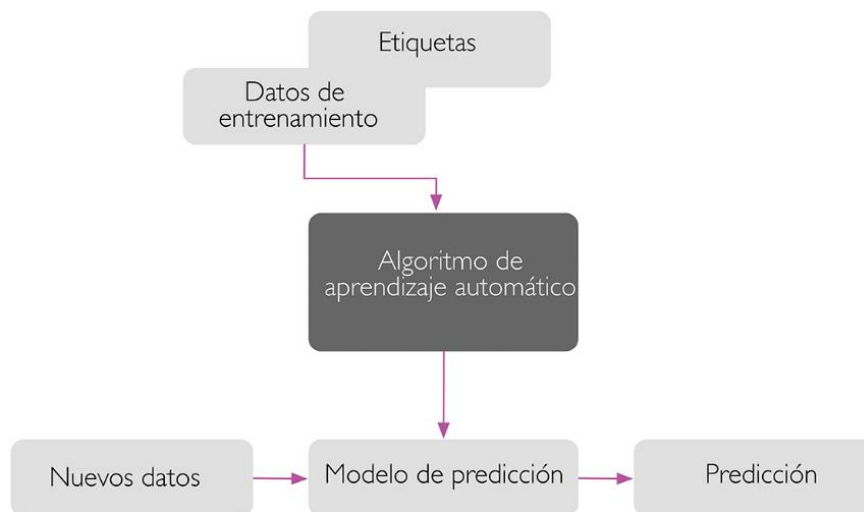


Figura 6: Flujo de operación del algoritmo supervisado.

Nota. Imagen adaptada del libro Python Machine Learning, segunda edición, 2019.

Dentro del aprendizaje supervisado podemos encontrar dos categorías de problemas: problemas de clasificación y problemas de regresión.

1. Clasificación

Es una subcategoría del aprendizaje supervisado cuyo objetivo es predecir las etiquetas discretas de clase categórica de nuevas instancias, basadas en observaciones. Un ejemplo típico de clasificación binaria es la detección de correo no deseado, donde el algoritmo de aprendizaje automático, aprende un conjunto de reglas para distinguir entre dos posibles clases que son correo deseado (0) o correo no deseado (1). Sin embargo, la clasificación no sólo tiene naturaleza binaria, sino también puede contar con varias etiquetas de clase para el conjunto de datos de entrenamiento; esto se conoce como clasificación multiclase. Un ejemplo típico de este tipo de clasificación es el reconocimiento de un carácter manuscrito, donde se recoge un conjunto de datos de entrenamientos que corresponden a ejemplos manuscritos de cada letra del alfabeto (etiquetados con la letra correcta), entonces el modelo predictivo será capaz de predecir la letra correcta con cierta precisión.

Para mayor entendimiento del concepto de clasificación se ilustra la clasificación binaria en la siguiente figura, donde contamos con 30 muestras de entrenamiento; 15 de estas muestras están etiquetadas como clase negativa (signo menos) y 15 etiquetadas como clase positiva (signo más). En este caso, el conjunto de datos es bidimensional, donde tenemos dos valores asociados X_1 y X_2 . Entonces aplicamos el algoritmo de aprendizaje supervisado para aprender la regla que puede separar las dos clases (se representa la separación por una línea discontinua) y clasificar nuevos datos dentro de cada categoría.

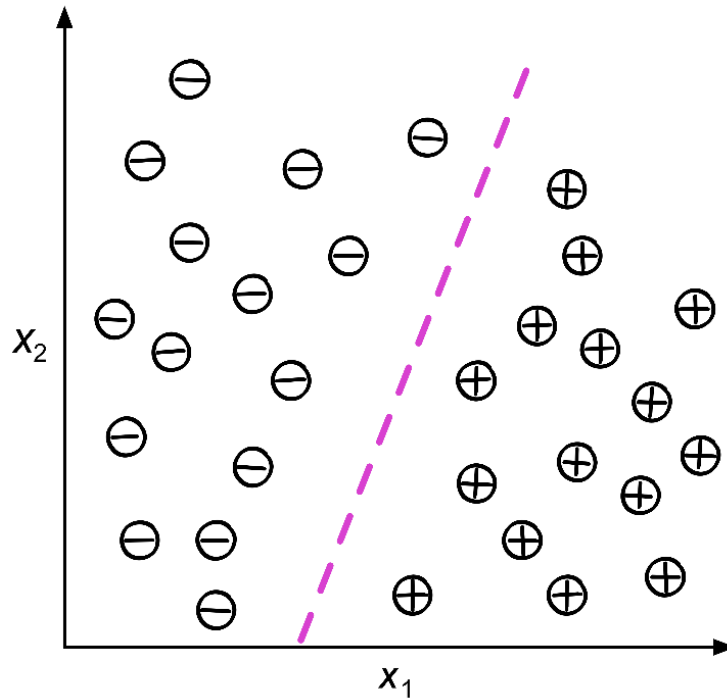


Figura 7: resultado algoritmo de aprendizaje con clasificación binaria
 Nota. Imagen adaptada del libro *Python Machine Learning, segunda edición, 2019*.

Algunos de los algoritmos de aprendizaje más robustos y conocidos para clasificación son regresión logística, máquina de vectores de soporte y árboles de decisión, los cuales se revisan sólo como una introducción:

Regresión logística

La regresión logística es un modelo que funciona muy bien en clases separables lineales. Como modelo probabilístico la razón de probabilidades se describe como $\frac{p}{(1-p)}$, donde p significa la probabilidad del evento positivo. El término evento positivo no significa necesariamente bueno, sino que se refiere al evento que queremos predecir. Si pensamos en el evento positivo como una etiqueta de clase $y = 1$, entonces podemos definir la función logit, que es el algoritmo de la razón de probabilidades:

$$\text{logit}(p) = \log\left(\frac{p}{(1-p)}\right)$$

Ahora para predecir la probabilidad de que una determinada muestra pertenezca a una clase concreta, se utiliza la función sigmoid, cuya fórmula es:

$$\phi(Z) = \frac{1}{1 + e^{-Z}}$$

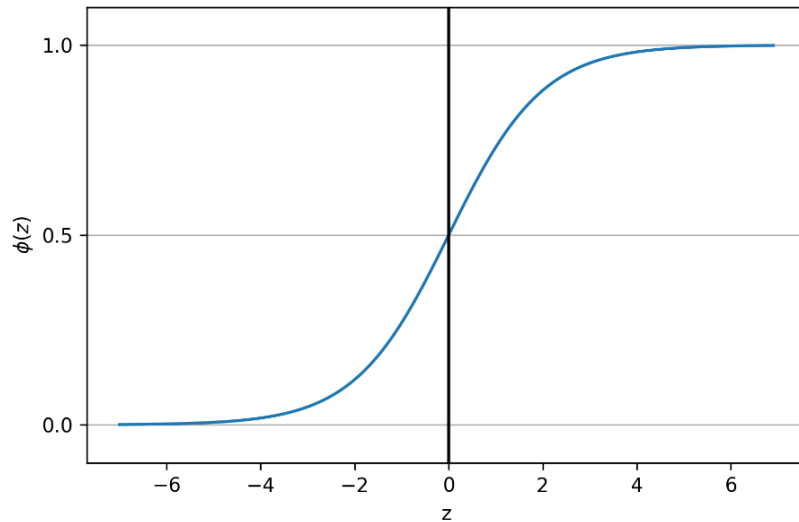


Figura 8: función sigmoide de probabilidad de predicción

Nota. Imagen adaptada del libro *Python Machine Learning, segunda edición, 2019*.

Entonces la probabilidad predicha se puede convertir después en un resultado binario mediante la función umbral (donde 1 es el evento que queremos predecir)

$$Y = \begin{cases} 1, & \text{if } \phi(Z) \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

Máquina de Vectores de Soporte (SVN)

La SVM tiene por objetivo maximizar el margen que se define como la distancia entre el hiperplano de separación (límite de decisión) y las muestras de entrenamiento que están más cerca de ese hiperplano (que también se denominan vectores de soporte). Esto se puede ver en la siguiente imagen:

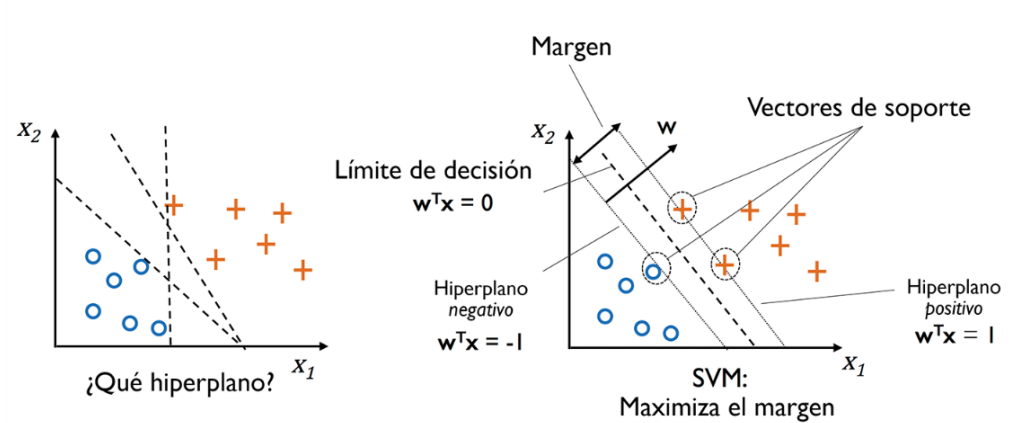


Figura 9: Aprendizaje con SVN

Nota. Imagen adaptada del libro *Python Machine Learning, segunda edición, 2019*.

La función objetivo de la SVN pasa a ser la maximización del margen bajo la restricción de que muestras están clasificadas correctamente:

$$\mathbf{w}_0 + \mathbf{w}^T \mathbf{x}^{(i)} \geq 1 \text{ if } y^i = 1$$

$$\mathbf{w}_0 + \mathbf{w}^T \mathbf{x}^{(i)} \leq -1 \text{ if } y^i = -1$$

Para $i = 1 \dots N$

Árboles de decisión

Como su nombre lo sugiere podemos pensar en este modelo como una descomposición de nuestros datos mediante una toma de decisiones basada en formulación de una serie de preguntas. Si nos basamos en las características de nuestro conjunto de datos de entrenamiento, el modelo de árbol de decisión aprende de una serie de preguntas para deducir las etiquetas de clase de la muestra. Utilizando el algoritmo de decisión, empezamos en la raíz del árbol y dividimos los datos en la característica que resulta en la mayor ganancia de información, en este proceso de iteración podemos repetir el procedimiento de división en cada nodo hijo hasta que las hojas sean puras. Para limitar la profundidad de árbol y no provocar un sobreajuste, una buena opción es podar el árbol ajustando el límite a su profundidad máxima.

En la figura como ejemplo, se muestra un árbol de decisión para decidir una actividad en un día concreto:

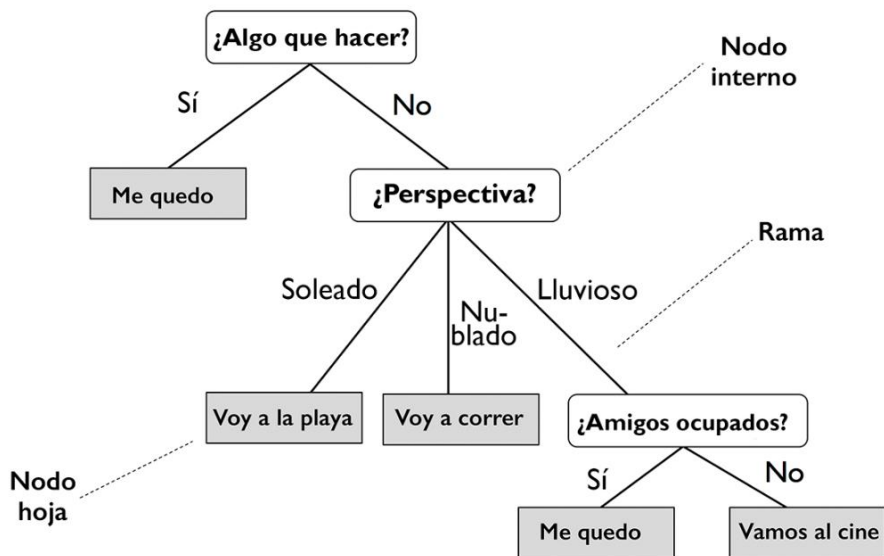


Figura 10: Árbol de decisión

Nota. Imagen adaptada del libro Python Machine Learning, segunda edición, 2019.

La función objetivo es maximizar la ganancia de información en cada división y se define de la siguiente forma:

$$\text{Gan Inf}(S, A) = \text{Entropia}(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} \text{Entropia}(S_v)$$

Donde,

S: es una colección de objetos

A: son los atributos de los objetos

V(A): Conjunto de valores que A puede tomar

2. Regresión

Corresponde a la predicción de resultados continuos, también conocido como análisis de regresión. En este análisis tenemos variables predictoras (explicativas) y una variable de respuesta continua (resultado o destino), por lo que se tiene que encontrar una relación entre estas variables que nos permita predecir un resultado. Por ejemplo, si queremos predecir los resultados del examen de selectividad de matemáticas de los alumnos, si el análisis indica que existe relación entre el tiempo que han pasado estudiando para la prueba y los resultados finales, podríamos utilizarla como dato de entrenamiento para aprender un modelo que utilice el tiempo de estudio para predecir resultados de la prueba para futuros estudiantes.

En la siguiente figura se ilustra el concepto de regresión lineal. Dada una variable predictora x y una variable de respuesta y, se aplica una línea fina a este dato, que minimiza la distancia entre los puntos de la muestra y la línea. Entonces se utiliza la intersección y la pendiente aprendidas de este dato para predecir la variable resultado del nuevo dato:

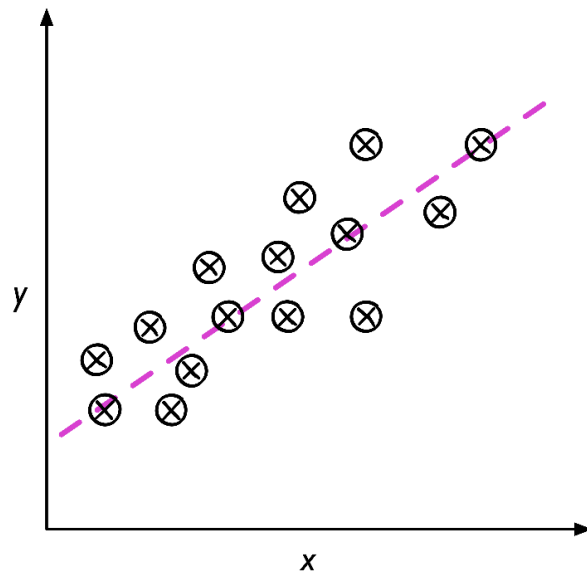


Figura 11: Análisis de regresión lineal.

Nota. Imagen adaptada del libro Python Machine Learning, segunda edición, 2019.

A continuación, describimos dos algoritmos de aprendizaje para el análisis de regresión: regresión de árboles de decisión y regresión de bosques aleatorios.

Regresión de árboles de decisión

Las fórmulas de la función objetivo para maximizar la ganancia de información es la misma que en un árbol de decisión (vistas en el apartado 3.1.1.1 Clasificación). Pero en el análisis de regresión del árbol de decisión la función de costo corresponde a la minimización del error, una de las utilizadas es MSE (error cuadrático medio por sus siglas en inglés) que es designado como la varianza dentro del nodo, razón por la cual el criterio de división también es conocido como reducción de varianza, de modo que lo que se pretende en este modelo es minimizar la varianza a un valor dado.

Regresión con bosques aleatorios (Random forest)

Un bosque aleatorio es una técnica conjunta que combina múltiples árboles de decisión y por consiguiente tiene un rendimiento de generalización mejor que un árbol de decisión individual, debido a la aleatoriedad que ayuda a la varianza del modelo. Otras ventajas son que son menos sensibles a los outliers en el conjunto de datos y el único parámetro que precisa manejar es el número de árboles en el conjunto.

3.3 Aprendizaje no supervisado

En el aprendizaje sin supervisión trata con datos sin etiquetar o datos de estructura desconocida. Con las técnicas del aprendizaje sin supervisión, podemos explorar la estructura de los datos para extraer información significativa sin ayuda de una variable de resultado conocida o una función de ganancia.

En este contexto, una técnica exploratoria de datos es el **agrupamiento** que nos permite organizar un montón de información en subgrupos significativos llamados clúster, sin tener conocimiento previo de los miembros del grupo, entonces cada clúster se compone por datos con cierta semejanza, por lo que también se puede decir que se trata de una clasificación sin supervisión. El agrupamiento es una excelente técnica para estructurar información y su aplicación, por ejemplo, permite a los vendedores descubrir grupos de clientes basados en sus intereses, con el objetivo de segmentar campañas de marketing.

La siguiente figura muestra la aplicación de un agrupamiento para organizar datos sin etiquetar, donde se puede encontrar tres clústeres basados en la similitud de sus características X_1 y X_2 .

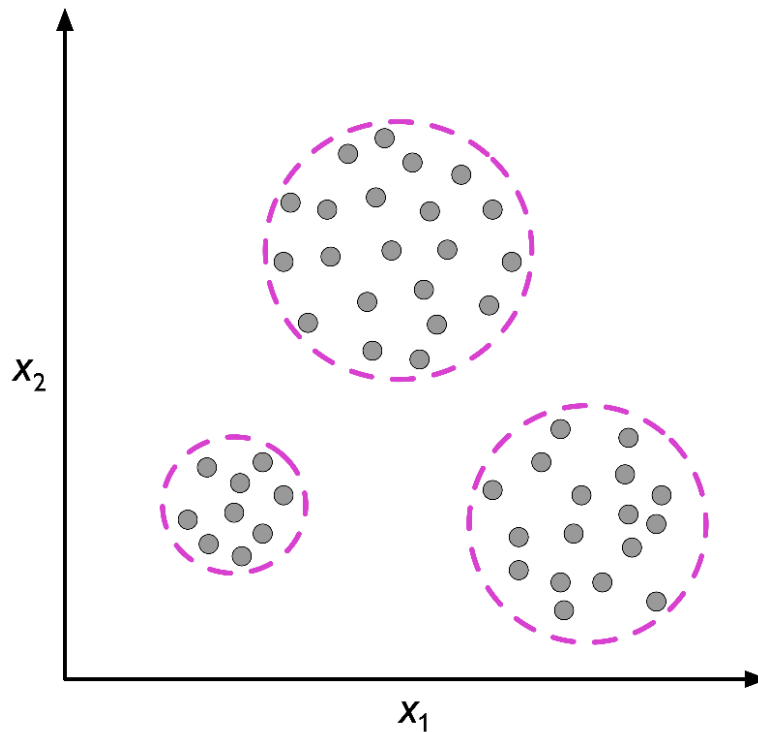


Figura 12: Agrupamiento de datos no supervisados

Nota. Imagen adaptada del libro Python Machine Learning, segunda edición, 2019.

Algunos de los algoritmos más populares para agrupamientos que ayudan a organizar los datos en estructuras significativas son: k-means y árboles de agrupamiento jerárquico.

K-means

El k-means es un algoritmo cuyo agrupamiento es basado en prototipos. Esto significa que cada grupo o clúster está representado por un prototipo que puede ser el promedio (centroide) de puntos similares con características continuas o el punto más representativo o que aparece con más frecuencia (mediodo) en el caso de las características categóricas. Este algoritmo es muy bueno para identificación de grupos en forma esférica. El inconveniente de este algoritmo es que le debemos decir a priori el número de grupos (k).

Algunas técnicas para encontrar el número óptimo de clúster para este método son:

Método del codo (elbow)

Es una de las métricas usada para comparar resultados de la **distancia media entre los puntos de datos y su centroide**. Como el valor de la media disminuirá a medida de aumentemos el valor de k, se debe utilizar la distancia media al centroide en función de k y encontrar el «punto codo», donde la tasa de descenso se aplana. Aquí vemos una gráfica a modo de ejemplo, donde podemos observar que el “codo” o el número óptimo de clúster es 3:

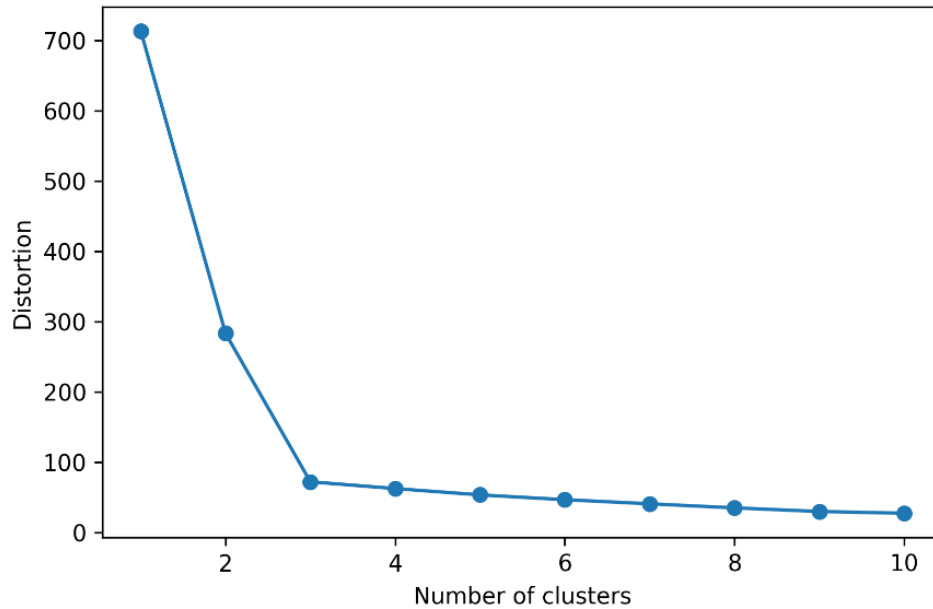


Figura 13: Método del codo

Nota. Imagen adaptada del libro Python Machine Learning, segunda edición, 2019.

Método de silueta

El análisis de la silueta mide la calidad del agrupamiento o clustering, con respecto a la distancia de separación entre los clústeres. Nos indica que tan cerca está cada punto de un clúster a puntos de los clústeres vecinos. Esta medida de distancia se encuentra en el rango $[-1, 1]$, donde un valor alto indica un buen clustering. Los coeficientes de silueta **cercanos a 1** indican que **la observación se encuentra lejos de los clústeres vecinos**. Un valor del coeficiente de **0** indica que **la observación está muy cerca o en la frontera de decisión entre dos clústeres**. **Valores negativos** indican que esas muestras quizás estén asignadas al **clúster erróneo**.

El coeficiente de silueta es calculado de la siguiente forma:

$$S = \frac{b - a}{\max(a, b)}$$

En la gráfica de ejemplo se muestra la evaluación de tres clústeres con el método de silueta, y como vemos, no se acercan a 0 lo que indica que es un buen agrupamiento:

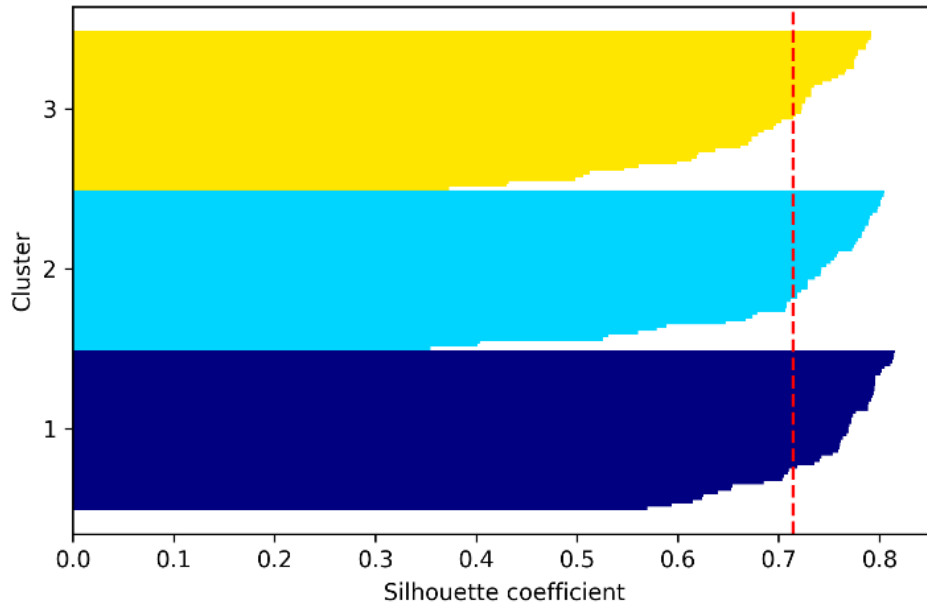


Figura 14: Método de silueta

Nota. Imagen adaptada del libro Python Machine Learning, segunda edición, 2019.

Tomando en consideración los métodos del codo y de silueta podemos determinar con mayor precisión el número óptimo de clúster (k) que debemos indicar al algoritmo k-means para ejecutar y obtener los agrupamientos finales.

Árboles de agrupamiento jerárquico

El Clustering Jerárquico (agrupamiento jerárquico o “Hierarchical Clustering” en inglés), es un algoritmo de clúster jerárquico que agrupa los datos basándose en la distancia entre cada uno y buscando que los datos que están dentro de un clúster sean los más similares entre sí. En una representación gráfica los elementos quedan anidados en jerarquías con forma de árbol.

Se pueden definir dos tipos de clustering jerárquico dependiendo de la dirección en la que el algoritmo ejecute el agrupamiento:

1. Tipo Aglomerativo: Empezamos a agrupar desde cada elemento individual. Al inicio cada punto o dato está en un clúster separado. A cada paso, los dos clústeres más cercanos se fusionan. Estas fusiones de clústeres se siguen produciendo de forma sucesiva generando una jerarquía de resultados de clustering. Al final del proceso solo queda un único clúster que aglutina todos los elementos.
2. Divisible: Comenzamos a la inversa, partimos de un único clúster que aglomera todos los datos y vamos dividiendo en clústeres más pequeños.

Este tipo de agrupamiento, para mayor claridad, se puede representar en un dendograma, donde se muestra que los datos son parecidos de acuerdo con su distancia euclidiana. Esto se puede ver en las hojas finales del árbol. En la figura de ejemplo si se considera la menor distancia euclidiana los datos más parecidos son el dato5 y dato2.

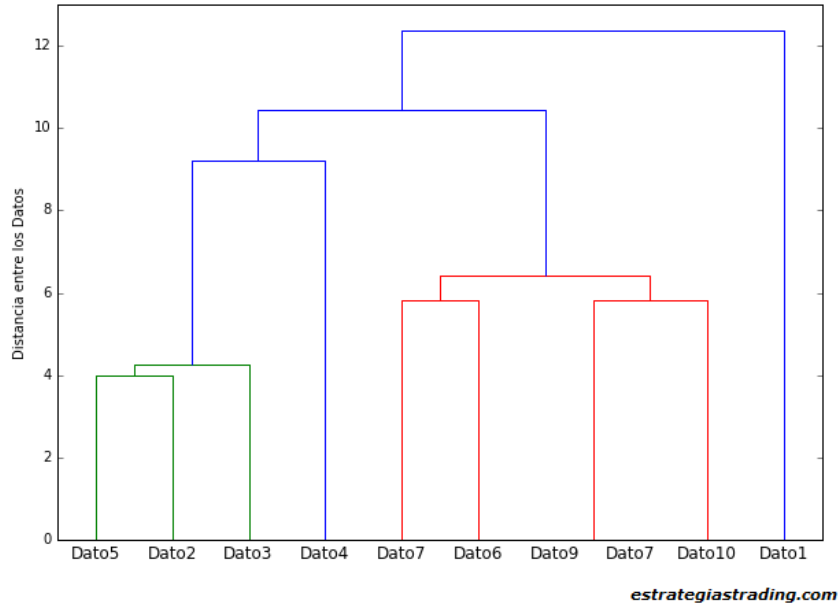


Figura 15: Dendrograma

Nota. Imagen adaptada del libro Python Machine Learning, segunda edición, 2019.

3.4 Aprendizaje reforzado

No nos detendremos en este tipo de aprendizaje. Sólo comentaremos su objetivo, que es desarrollar un sistema (agente) que mejore el rendimiento basado en interacciones con el entorno. Como la información sobre el estado actual del entorno normalmente también incluye una señal de recompensa, se puede pensar en el aprendizaje reforzado como un campo relacionado con el aprendizaje supervisado. A través de su interacción con el entorno, un agente puede utilizar el aprendizaje reforzado para aprender una serie de acciones que maximicen esta recompensa mediante un enfoque experimental de ensayo-error o una planificación deliberativa.

Un conocido ejemplo de aprendizaje reforzado es un motor de ajedrez, donde el agente elige entre una serie de movimientos según el estado del tablero (el entorno) y la recompensa se puede definir como “ganas” o “pierdes” al final del juego.

La siguiente figura esquematiza a grandes rasgos el proceso que realiza el aprendizaje reforzado, donde el agente intenta maximizar la recompensa mediante una serie de interacciones con el entorno:

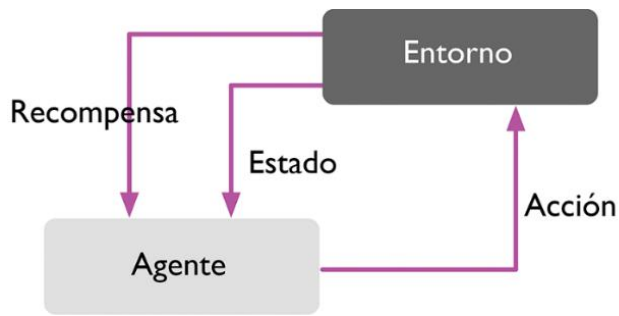


Figura 16: Aprendizaje reforzado

Nota. Imagen adaptada del libro *Python Machine Learning, segunda edición, 2019*.

3.5 Redes Neuronales

Las neuronas como tal son células nerviosas interconectadas en el cerebro que participan en el proceso y la transmisión de señales eléctricas y química como ilustramos en la siguiente figura:

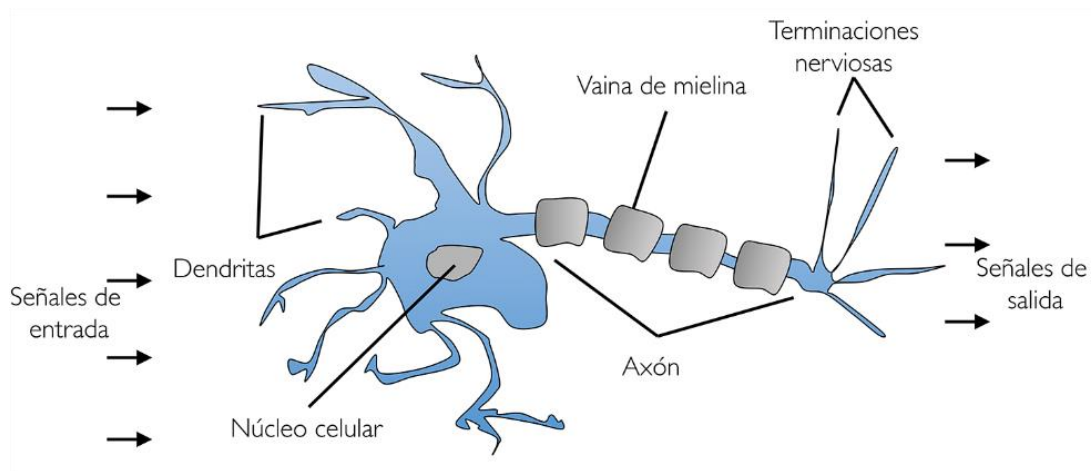


Figura 17: Neurona biológica

Nota. Imagen adaptada del libro *Python Machine Learning, segunda edición, 2019*.

Inspirado por el proceso, Frank Rosenblatt (1958) creó el perceptrón, que es un algoritmo de reconocimiento de patrones basado en una red de aprendizaje de computadora de dos capas, que utilizaba adición y sustracción simples. Rosenblatt también describe circuitería que no está en el perceptrón básico, tal como el circuito de o-exclusiva, un circuito que no se pudo procesar por redes neuronales antes de la creación del algoritmo de propagación reversa de Paul Werbos (1975), algoritmo que se puede utilizar para predecir si una muestra pertenece a una clase o no.

Siguiendo los principios de Rosenblatt, la i -ésima neurona artificial estándar en el algoritmo del perceptrón consta de los siguientes elementos:

- Conjunto de entradas o vector de entradas x , de n componentes.
- Conjunto de pesos sinápticos w_{ij} . Representan la interacción entre la neurona pre-sináptica j y la post-sináptica i .

- Regla de propagación $d(w_{ij}, x_j(t))$: proporciona el potencial post-sináptico, $h_i(t)$.
- Función de activación $a_i(t) = f(a_i(t - 1), h_i(t))$: proporciona el estado de activación de la neurona en función del estado anterior y del valor post-sináptico.
- Función de salida $F_i(t)$: proporciona la salida $y_i(t)$, en función del estado de activación.

Las señales de entrada y salida pueden ser señales binarias (0, 1 – neuronas de McCulloch y Pitts), bipolares (-1, 1), números enteros o continuos, etc. La regla de propagación suele ser una suma ponderada del producto escalar del vector de entrada y el vector de pesos:

$$h_i(t) = \sum w_{ij} * x_j$$

También se usa a menudo la distancia euclidiana entre ambos vectores:

$$h_i(t) = \sum (x_j * w_j)^2$$

Las funciones de activación que se pueden utilizar en la red neuronal se muestran en la siguiente figura:



Figura 18: Funciones de activación de una red neuronal

Nota. Imagen adaptada del libro *Python Machine Learning, segunda edición, 2019*.

Las funciones de salida son las siguientes:

- Ninguna: este es el tipo de función más sencillo, tal que la salida es la misma que la entrada. Es también llamada función identidad
- Binaria:

$$\begin{cases} 1, & \text{si } act_i \geq \epsilon_i \\ 0, & \text{de lo contrario} \end{cases}$$

Donde ϵ_i es el umbral.

- Regresión: donde la salida tiene que determinar un valor en base a la muestra

Con las fórmulas y funciones de activación podemos esquematizar en un diagrama simple el concepto de perceptrón, donde una muestra X se combina con pesos W para calcular la entrada a la red. A continuación, la entrada a la red pasa por una función umbral que genera una salida binaria -1 o 1; la etiqueta predicha de la muestra. Durante la fase de aprendizaje, esta salida se utiliza para calcular el error de la predicción y actualizar los pesos.

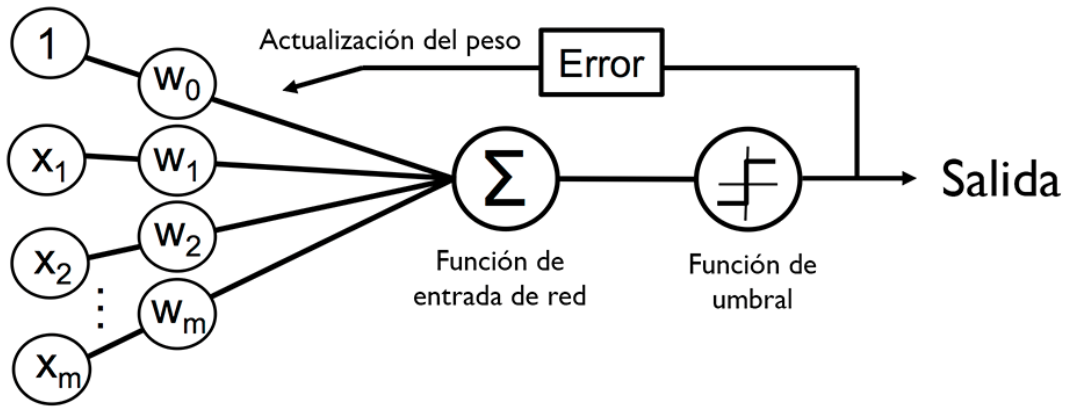


Figura 19: Modelo de perceptrón

Nota. Imagen adaptada del libro *Python Machine Learning, segunda edición, 2019*.

Aprendizaje en redes neuronales

Para el aprendizaje en las redes neuronales primero se valida la arquitectura de una red neuronal que se forma en base al perceptrón. En esta arquitectura se pueden distinguir las siguientes capas:

- De entrada: es la capa que recibe directamente la información proveniente de las fuentes externas de la red.
- Ocultas: son internas a la red y no tienen contacto directo con el entorno. El número de niveles ocultos puede variar entre cero y un número elevado. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, las distintas topologías de redes neuronales.
- De salidas: transfieren información de la red hacia el exterior.

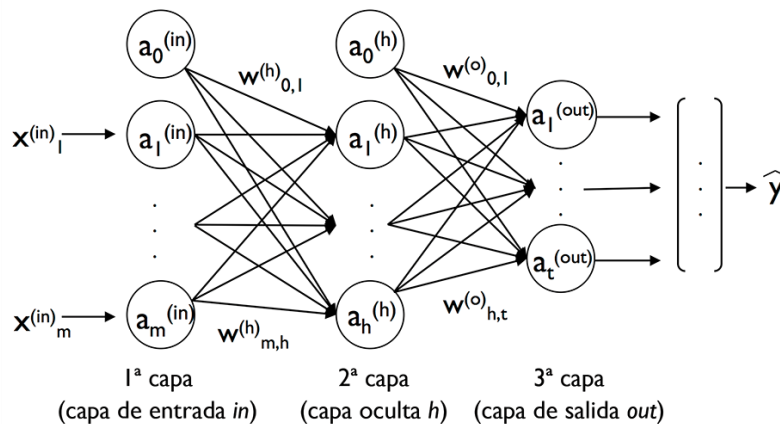


Figura 20: La figura muestra una red neuronal multicapa, con 1 capa oculta y una capa de salida.

Nota. Imagen adaptada del libro *Python Machine Learning, segunda edición, 2019*.

Entonces una red neuronal debe aprender a calcular la salida correcta para cada arreglo o vector de entrada en el conjunto de ejemplos. Este proceso de aprendizaje se

denomina proceso de entrenamiento o acondicionamiento. El conjunto de datos (o conjunto de ejemplos) sobre el cual este proceso se basa es, por ende, llamado conjunto de datos de entrenamiento.

La topología de la red y las diferentes funciones de cada neurona (entrada, activación y salida) no pueden cambiar durante el aprendizaje, mientras que los pesos sobre cada una de las conexiones si pueden hacerlo; el aprendizaje de una red neuronal significa esencialmente adaptación de los pesos. En otras palabras, el aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de esta pasa a tener un valor distinto de cero. De la misma manera, una conexión se destruye cuando su peso pasa a ser cero.

Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por lo tanto, se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables ($\frac{dw_{ij}}{dt} = 0$). Un aspecto importante respecto al aprendizaje de las redes neuronales es el de conocer cómo se modifican los valores de los pesos, es decir, cuáles son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red aprenda una nueva información. En una red neuronal al igual que los otros algoritmos podemos distinguir aprendizaje supervisado y aprendizaje no supervisado.

Otro criterio que se puede utilizar para diferenciar las reglas de aprendizaje se basa en considerar si la red puede aprender durante su funcionamiento habitual o si el aprendizaje supone la desconexión de la red, es decir, su inhabilitación hasta que el proceso termine. En el primer caso, se trataría de un aprendizaje on-line, mientras que el segundo es lo que se conoce como off line.

Cuando el aprendizaje es off line, se distingue entre una fase de aprendizaje o entrenamiento y una fase de operación o funcionamiento, existiendo un conjunto de datos de **entrenamiento y un conjunto de datos de prueba**, que serán utilizados en la correspondiente fase. Además, los pesos de las conexiones permanecen fijos después que termina la etapa de entrenamiento de la red. Debido precisamente a su carácter estático, estos sistemas no presentan problemas de estabilidad en su funcionamiento. La siguiente figura muestra que la brecha entre el entrenamiento y la precisión de la validación aumenta cuantas más épocas entrenamos la red.

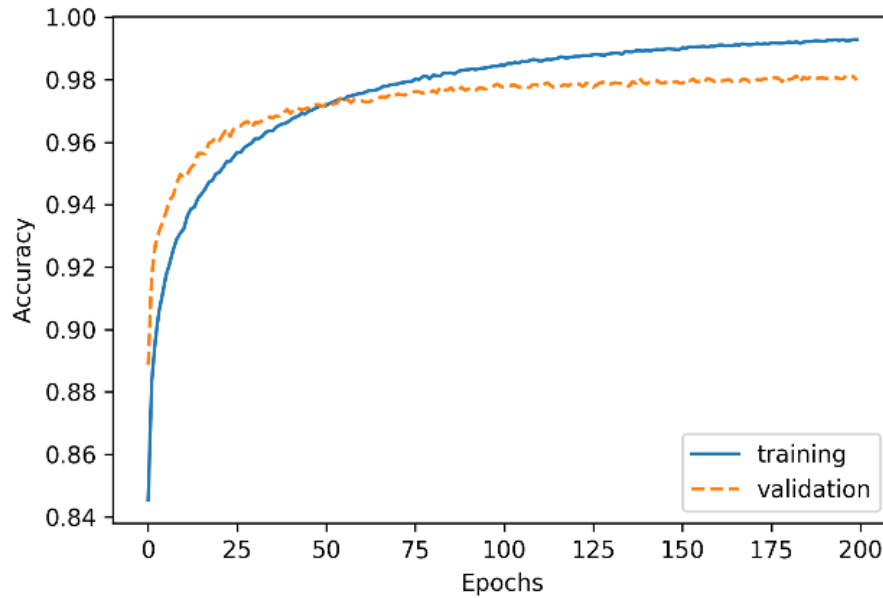


Figura 21: curva de entrenamiento y testing en una red neuronal

Nota. Imagen adaptada del libro Python Machine Learning, segunda edición, 2019.

El proceso de aprendizaje como tal, se basa en disminuir la función de error, considerando métodos de optimización como descenso de gradiente. A nivel de fórmulas matemáticas tenemos las siguientes:

1. Función de costo (error): corresponde a la función que identifica el error en el entrenamiento, donde, x son los datos de entrada, Θ los pesos y h_{θ} la función de propagación:

$$h_{\theta}(x) \in \mathbb{R}^k \quad \left(h_{\theta}(x)\right)_i = i^{\text{th}} \text{ output}$$

$$J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m \sum_{k=1}^K y_k^{(i)} \log \left(h_{\theta}(x^{(i)}) \right)_k + \left(1 - y_k^{(i)} \right) \log \left(1 - \left(h_{\theta}(x^{(i)}) \right)_k \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{L-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} \left(\theta_{ji}^{(l)} \right)^2$$

2. Descenso de gradiente: función que minimiza el error de la función de costo para lograr el menor error en la predicción.

$$\min_{\theta} J(\theta)$$

La siguiente figura esquematiza el descenso de gradiente con respecto a las iteraciones que se realizan en la red neuronal. Como se puede apreciar cada iteración busca bajar en la curva de costo para minimizar los mismos, aquello lo realiza modificando los pesos.

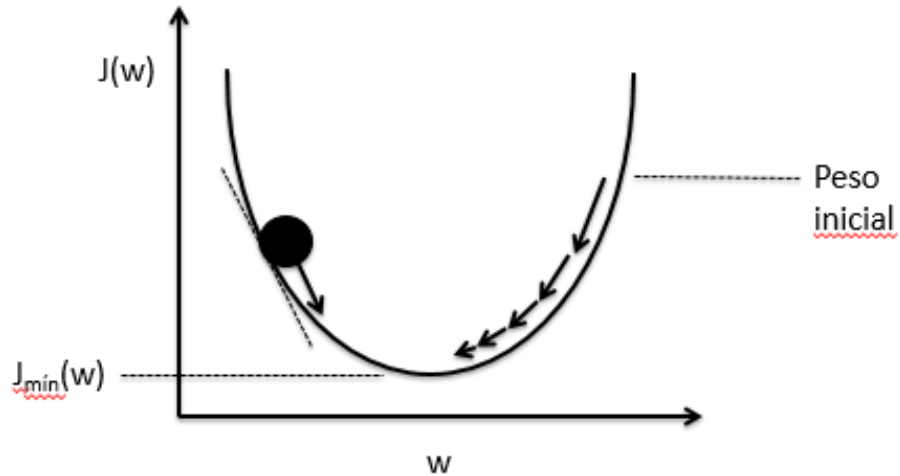


Figura 22: Esquema de descenso de gradiente

Nota. Imagen adaptada del libro Python Machine Learning, segunda edición, 2019.

A nivel de programación se encuentran componentes como: **Keras y Tensorflow** para crear y operar estas redes neuronales.

Dentro del aprendizaje de redes neuronales también aparecen los conceptos de machine learning y deep learning que pasaremos a definir para una completitud de conceptos que nos aportan en este proyecto:

1. **Machine learning:** se basa en algoritmos de aprendizaje, que se pueden clasificar, según la tarea que realizan, en: regresión, clasificación, etc. o bien, en árboles de decisión, modelos lineales, polinomiales y redes neuronales.
2. **Deep learning:** es un caso particular de red neuronal que se caracteriza por tener múltiples capas de neuronas conectadas entre sí, que han mostrado en la práctica tener mejores capacidades de aprendizaje, aunque en realidad no es el número de capas lo que define el aprendizaje profundo, el concepto subyacente al aprendizaje profundo es el procesamiento de los datos de forma jerárquica, es decir, el uso de redes neuronales para obtener representaciones cada vez más significativas de los datos mediante el aprendizaje por capas. Entonces, cada capa extrae características de un nivel cada vez más alto hasta llegar a su respuesta final. Al ir profundizando en la red, estas funciones más simples se van combinando para buscar relaciones más complejas como puedan ser partes concretas de auto (puertas, ruedas, ventanas). En un siguiente paso se identificaría las partes de auto completas (carrocería, por ejemplo), y por último se identificaría a la imagen que corresponde a un auto.

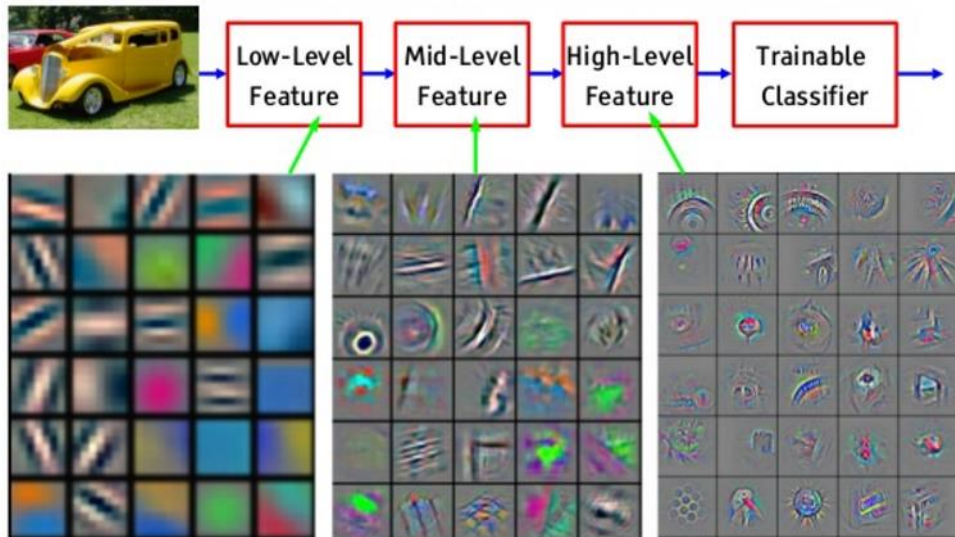


Figura 23: Reconocimiento de automovil

*Nota. Imagen adaptada de Tensorflow on Ubuntu 18.04,
url: https://informatics.sydney.edu.au/news/tf_on_linux/*

En este aparatado se han revisado todos los conceptos que nos permitirán entregar una base teórica al proyecto de tesis, donde hemos partido desde el concepto de análisis de datos en base a análisis supervisados y no supervisados para llegar a las redes neuronales, con ello ya se puede comenzar el capítulo del desarrollo de la “Predicción de pedido a solicitar semanalmente”

4. Capítulo IV: Predicción de pedido a solicitar semanalmente

En este apartado se detalla cómo se llevó a cabo el proceso para encontrar un algoritmo que logre mejorar la estimación de la solicitud del pedido. Para ello primero se define el problema a abordar para continuar con el desarrollo de la solución en sus distintas etapas:

4.1 Problema a Abordar

El problema por abordar en detalle se encuentra en el proceso de solicitud de pedido de mercadería, que es realizado por el director técnico en base a su juicio experto. Este problema ha tenido un impacto final en la pérdida de un volumen ventas de un 13%, lo que significa un bajo margen de 22%, **siendo que el margen teórico óptimo es del 35%**. Las pérdidas del volumen de ventas se generan en el acto en un cliente solicita un producto que no tenemos en la estantería, lo que además de impedir la venta inmediata provoca con cierta probabilidad que el cliente no vuelva a la farmacia dado que en la competencia podría encontrar con más frecuencia el producto.

Una solución explorada es el solicitar un sobre stock a principios de cada mes de mercadería, pero no ha ayudado, debido a que la génesis del problema se encuentra en que la **solicitud del pedido de mercadería no es óptima**, por lo que si se solicita mayor cantidad de producto el porcentaje de error aumenta, generando productos inmovilizados que se van acumulando sucesivamente (capital inmovilizado) que terminan por vencer (pérdida del 50% del valor del producto). Esto, disminuye el margen mensual (diferencia entre la venta y la compra de productos) quedando este en un 22%.

En el estudio del problema se han identificado los siguientes factores, considerados debilidades al momento de realizar la solicitud de pedido:

- 1 Cantidad de días para venta del pedido: se refiere al período en que se deben vender los productos solicitados en el pedido. En la actualidad se generan pedidos para venta en 7 días y 15 días, pero hay posibilidad de generar pedidos por más espacio de tiempo (siempre informando antes a Farmacias SIMI). Este punto hoy es analizado en cuanto a montos totales de las últimas ventas.
- 2 Ventas históricas: se refiere a ventas del período considerando, años anteriores, meses anteriores, semanas anteriores y volumen necesario para absorber gastos y mejorar rentabilidad. Estas ventas se dividen en ventas lunes, donde los productos cuentan con un 25% de descuento del precio lista y ventas no lunes, donde los productos se venden a precio lista. Actualmente el estudio se realiza parcialmente en base a juicio experto.
- 3 Quiebres de stock en la recepción de pedidos: Se refiere a la mercadería que no es entregada en el local, siendo que fue solicitada.
- 4 Paquetes de productos: Se refiere a productos que solamente se deben solicitar por cantidad mínima de 20 o 40 unidades. Esta selección de productos cambia constantemente. Este punto está sujeto a errores por manejos manuales de la información.
- 5 Comuna de la farmacia: El dato se refiere a la ubicación geográfica de la farmacia, lo que sugiere diferencia en la rotación de algunos productos de una comuna a otra. En la actualidad el estudio se realiza en base a juicio experto.

4.2 Desarrollo de la solución

Para la solución del problema se exploraron varias fórmulas, partiendo de las convencionales hasta llegar a la predicción mediante machine learning, que es el algoritmo final que se presenta como la solución real al problema. A continuación, se revisan cada una de las soluciones exploradas.

4.2.1 Manejo de inventario convencional

La primera solución explorada consiste en el manejo de inventario en forma convencional, donde el proceso de solicitud de pedido se desarrolla en base a las ventas del año anterior en la misma semana, considerando:

1. Obtención de las ventas del mismo número de semana del año anterior en la base de datos MySQL con consultas directas.
2. Los productos que no se encuentran vigentes en inventario, se eliminan.
3. El precio de venta actualizado del producto.
4. El empaque, que corresponde a una planilla con al pedido mínimo por cada producto.
5. Los productos potenciales, que corresponde a una planilla con los productos que el cliente solicita y que no se encuentran en sala de venta.
6. Inventario actual, se considera si el producto se encuentra en inventario y en qué cantidad, para no pedir y generar un sobre-stock.

El proceso descrito se automatizó mediante para entregar un archivo excel con los productos a solicitar como el siguiente:

Código Producto	Nombre Producto	Cantidad Propuesta	Inventario	Venta antes de llegada del Pedido	Empaque	Cantidad Final a Pedir	Monto	No se encuentra en catálogo	Stock en SIMI
1008	CONDON LUBRICADOS 3PZAS SIMICONDON	4	6	3	48	48	42240	FALSO	No Informado
1194	LEVODROPROPIZINA 30MG/5ML 120ML	2	8	1	1	0	0	FALSO	No Informado
1195	LATANOPROST 50MCG/ML SOL OFT 2.5ML	2	3	1	1	0	0	FALSO	No Informado
130	MEBENDAZOL 100MG 6COMP	3	18	0	1	0	0	FALSO	No Informado
1403	LOSARTAN/HIDRO 50/12.5MG 30COMP	2	6	0	5	0	0	FALSO	No Informado
141	METRONIDAZOL 500MG 10 OVU	6	19	2	5	0	0	FALSO	No Informado
1547	CUBREBOCA TRIPLE PLIEGUE PAQ 5PZAS	3	0	2	10	10	9600	FALSO	No Informado
175	RANITIDINA 300MG 10GRAG	10	9	5	10	10	8000	FALSO	No Informado

Como se puede ver en la tabla, la columna “Cantidad a pedir” es la que entrega la cantidad de producto a solicitar, esto junto al conocimiento experto del director técnico, permite realizar la solicitud de pedido en el sistema.

Los resultados de esta solución para solicitar pedido fue una mejora en los márgenes de un 3%, es decir, fue un aporte a la venta, pero tuvo los siguientes inconvenientes que no le permitió lograr la efectividad necesaria:

1. Al considerar solamente la venta del año anterior, se pierde el aumento general de las ventas, dado que no se considera el aumentando de las últimas semanas, por lo que tiende a ser una estimación muy plana que refleja que se quiere vender lo mismo que el año pasado. Esto fue contraproducente, dado que como las ventas aumentaron, dejaron de encontrarse en stock los productos necesarios.
2. No considera la estacionalidad: si bien la estacionalidad es una variable que no manejaremos como tal, si pasa a ser relevante dado que se considera el mismo número de semana del año pasado que teóricamente debió estar en la misma estación, pero no siempre es así y además con la tendencia cambiante del clima no lograba estimar los productos necesarios contando con un quiebre de stock importante en las farmacias.

Los resultados obtenidos en esta solución se resumen en la siguiente tabla.

Local	Mes	Compra	Venta	Margen
Santa Rosa	diciembre 2018	9.726.686	16.325.213	40%
Santa Rosa	enero 2019	12.590.694	14.757.499	19%
Santa Rosa	febrero 2019	9.936.710	12.869.905	23%
Santa Rosa	marzo 2019	11.753.239	16.001.089	27%
Santa Rosa	abril 2019	15.777.573	19.020.072	17%
Margen promedio local Santa Rosa:				25%
Vespucio Norte	diciembre 2018	15.918.448	21.167.452	25%
Vespucio Norte	enero 2019	14.068.218	20.308.810	28%
Vespucio Norte	febrero 2019	13.051.538	16.856.227	19%
Vespucio Norte	marzo 2019	14.258.276	22.144.217	31%
Vespucio Norte	abril 2019	19.726.099	23.432.360	16%
Margen promedio local Vespucio Norte:				24%

En el cuadro si bien la solución muestra una mejora, esta es menor lo esperado (30%). En particular, el local de Santa Rosa existe un margen del 40% (superior al 35% teórico), esto es producto de que el mes anterior (noviembre 2018) se vendió menos mercadería de la comprada, entonces la mercadería en exceso de noviembre se vende en diciembre de 2018 y por consiguiente este último mes se compra menos mercadería

generando el exceso en el margen. Por lo anterior, se considera un promedio de 5 meses.

Entonces como el objetivo es mejorar el margen por lo menos en un 8%, se continúa explorando soluciones con técnicas predictivas para obtener mejores resultados en la solicitud de pedido de manera que las ventas sean más acordes al pedido realizado.

4.2.2 Análisis de datos con aprendizaje de máquinas

Siguiendo con el objetivo de lograr predecir el pedido de la siguiente semana, se realiza el entrenamiento de un modelo en base a los datos históricos para predecir el pedido futuro, para ello se consideran los siguientes puntos:

1. Como primer paso se genera la data que se entrenará en base a lo siguiente:
 - a. Ventas: por día de la semana, mes, año, código de producto y cantidad
 - b. Producto: la descripción del producto, el precio y si está disponible en catálogo.
 - c. Productos potenciales: se carga planilla (generada por el personal de la farmacia) con productos que el cliente pide y que no se encuentran en sala de ventas. Planilla cuenta con la fecha, el código de producto y la cantidad.
 - d. Reemplazo de productos: se carga planilla (generada por el personal de la farmacia) con los códigos productos nuevos que han reemplazado a productos obsoletos históricamente vendidos.
 - e. Estaciones del año: se consideran las 4 estaciones del año, pero con meses cerrados, es decir:
 - Verano: diciembre, enero y febrero
 - Otoño: marzo, abril y mayo
 - Invierno: junio, Julio y agosto
 - Primavera: septiembre, octubre y noviembre
 - f. Se eliminan las ventas de productos que nunca se han vendido, dado que no tiene sentido incluirlos ya que la predicción siempre será 0, así como también los productos obsoletos que no tienen un reemplazo por no poder comprarlos, para finalmente agrupar la venta de producto vigente por estaciones del año, año, número de semana y código de producto obteniendo el siguiente cuadro (ejemplos de datos obtenidos):

verano	otoño	invierno	primavera	año	Número de semana	Producto	Cantidad
0	0	0	1	2016	39	1098	2
0	0	0	1	2016	39	1194	1
0	0	0	1	2016	39	1547	1
0	0	0	1	2016	39	1648	1
0	0	0	1	2016	39	175	2
0	0	0	1	2016	39	AR0082	2
1	0	0	0	2019	12	CH0267	2
1	0	0	0	2019	12	CH4002	5
1	0	0	0	2019	12	CH4010	23

2. Con los datos obtenidos en el primer paso, se realiza el ordenamiento y transformación para obtener la matriz de features y de target en base a lo siguiente:
 - a. Se define el parámetro de cantidad de semanas históricas de venta por producto que quiero evaluar, por ejemplo: si definimos el parámetro en 8 y si estamos en la semana 20 del año 2019, definimos que el valor de venta de la semana 20 es el target y los valores de venta de features que determinan la venta de la semana 20 son las ventas de las semanas 19, 18, 17, 16, 15, 14, 13 y 12. Así sucesivamente vamos formando la matriz con todos los datos disponibles.
 - b. Se completa la matriz de feature con las 4 primeras columnas que indican la estación del año de la semana evaluada en binario, es decir, el 1 indica la estación del año (por ejemplo: Verano sería [1,0,0,0]), y las siguientes columnas hacia la derecha identifican las ventas en cada semana por producto (cada producto-semana es una fila).
 - c. Se generan dos matrices de features y dos de target, donde las primeras son con las ventas exactas por producto en cada semana y las segundas corresponden al trabajo con paquetes de venta, que significa que un producto se vende por 5 unidades como mínimo, esto para hacer el entrenamiento consecuente con el esquema de empaque que propone el franquiciante en el pedido (cantidad mínima a pedir por producto). Los siguientes ejemplos muestran cómo se ven las matrices.

Matriz de features y target con evaluación de 8 semanas históricas y ventas exactas:

Features											Target	
0	0	1	0	1	3	17	9	10	9	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	3	1	0	2	2	2	1	1	3
0	0	1	0	0	0	0	0	3	1	0	1	0
0	0	1	0	8	5	3	7	7	5	5	3	1
0	0	1	0	0	0	1	1	1	1	0	0	1
0	0	1	0	0	1	1	4	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	1	1	3	2	1	0	2
0	0	1	0	2	0	0	0	1	2	0	1	0

Donde las columnas de izquierda a derecha son:

1. Verano
2. Otoño
3. Invierno
4. Primavera
5. Venta producto X Semana_evaluada - 8
6. Venta producto X Semana_evaluada - 7
7. Venta producto X Semana_evaluada - 6

8. Venta producto X Semana_evaluada - 5
9. Venta producto X Semana_evaluada - 4
10. Venta producto X Semana_evaluada - 3
11. Venta producto X Semana_evaluada - 2
12. Venta producto X Semana_evaluada - 1

En el ejemplo, si tomamos la tercera fila de la matriz de features y de target, se tiene:

Features												Target
0	0	1	0	3	1	0	2	2	2	1	1	3

Continuando con el ejemplo de la tercera fila, para las matrices de features y target con paquetes de venta (venta mínima 5 productos), queda como sigue:

Features												Target
0	0	0	0	1	0	0	0	0	0	0	0	1

Finalmente, ya obtenidos los datos de las dos matrices de features (1 para ventas totales y otra para paquetes de venta), se procede a la carga de cada una de ellas desde la base datos lo que demora un tiempo superior a las 3 horas, donde al finalizar se generan archivos con la información cargada para dejar disponible al entrenamiento. Las matrices se generan por cada local, dado que hay diferencias de comuna lo que influye en la venta de medicamentos por farmacias.

3. En el tercer paso se determina el problema, donde en este caso es un problema de regresión. Dado que se requiere saber la cantidad a solicitar de cada producto la siguiente semana, esta cantidad no es fija, sino que varía dependiendo del comportamiento de las ventas en el tiempo, por lo que se debe buscar la cantidad de venta por producto.
4. El siguiente paso realiza un escalamiento de los valores de la matriz de features con el objetivo de eliminar el sesgo de valores muy grandes (escalamiento elegido en la transformación a valores numéricos en rangos entre 0 y 1) para posteriormente dividir los datos para contar con un set de entrenamiento, validación y de prueba. En este caso se tiene cerca de 100.000 ejemplos por lo que se considera un 80% de los datos como entrenamiento, un 10% para los datos de validación y un 10% para prueba. La división de los datos en validación y prueba se realiza para que los datos de validación ayuden a minimizar el error en el entrenamiento, antes de testear el modelo con los datos de prueba.
5. Con las matrices de features y target listas en el siguiente paso se elige el modelo más apropiado a la distribución de los datos. Para lograr determinar el mejor modelo, se realizan entrenamientos con varios modelos con el objetivo de minimizar la pérdida con la medida del Error Medio Absoluto (MAE), elegida debido a que la venta es un número entero positivo lo que indica que es un término absoluto (no contamos por ejemplo con productos que se vendan fraccionados) y

el optimizador “Adam”. Una vez finalizado el entrenamiento para determinar el mejor modelo se establece la medida de “desviación estándar promedio (DEP)” que es calculada en base al promedio de las desviaciones estándar que existe entre el valor real y el valor predicho del set de testing.

A continuación, se muestra una tabla con los resultados de los modelos entrenados, donde el valor determinante es la menor desviación estándar promedio (DEP):

Modelo	Local Santa Rosa		Local Vespucio Norte	
	Ventas exactas	Paquetes de ventas (5)	Ventas exactas	Paquetes de ventas (5)
Support Vector Regression	DEP: 6,68	DEP: 1,51	DEP: 7,88	DEP: 1,73
Árbol de decisión regresivo	DEP: 7,43	DEP: 1,63	DEP: 8,61	DEP: 1,85
Regresión logística	DEP: 6,53	DEP: 1,49	DEP: 7,77	DEP: 1,72
Random Forest	DEP: 8,75	DEP: 2,54	DEP: 9,97	DEP: 2,11
ADA Boosting Regresor sobre el modelo de árboles de decisión regresivo	DEP: 12,29	DEP: 13,38	DEP: 19,46	DEP: 3,89

Como muestra el cuadro el mejor de estos modelos considerando la menor desviación estándar promedio (DEP) es:

Local Santa Rosa	Local Vespucio Norte
Regresión logística: Para ventas exactas con un DEP de 6,53 Para paquetes de ventas con un DEP de 1,49	Regresión logística: Para ventas exactas con un DEP de 7,77 Para paquetes de ventas con un DEP de 1,72

6. Buscando mejorar el DEP también se entrenan las matrices de features y target con el modelo de redes neuronales donde se determina lo siguiente:

- a. Primero se decide la arquitectura de red a utilizar, aquella se ajusta en base a pruebas para que nos entregue la mejor desviación estándar promedio (DEP). Después de esta sensibilidad se logra la arquitectura final.
- b. Segundo se determinan las épocas (o ciclos que debe realizar la red) y un stop para la red. Para este entrenamiento se tiene un total de 200 épocas y un stop de que si no disminuye el error desde el menor encontrado en 25 épocas el entrenamiento debe terminar considerando que ya no se puede disminuir el error.

Con la arquitectura y épocas se entrena la red neuronal. Para ello utilizamos código en python que se puede encontrar en el apéndice A: Código creación de la red neuronal, apéndice B: Código entrenamiento de la red neuronal y las gráficas generadas en el apéndice C: Gráficos de resultados red neuronal. El resultado de la ejecución con el modelo de redes neuronales, en base a la desviación estándar promedio (DEP) se muestra en el siguiente cuadro:

Arquitectura	Local Santa Rosa		Local Vespucio Norte	
	Ventas exactas	Paquetes de ventas (5)	Ventas exactas	Paquetes de ventas (5)
tres capas ocultas de 24, 240 y 24	DEP: 4,33		DEP: 4,44	DEP: 0,95
tres capas ocultas de 12, 120 y 12	DEP: 4,43	DEP: 0,79	DEP: 4,32	
Cuatro capas ocultas de 12, 120, 240 y 12		DEP: 0,78 Precisión: 68,9%		DEP: 0,96 Precisión: 65,4%

Como muestra el cuadro de entrenamiento con redes neuronales se logra disminuir el DEP, entonces el mejor modelo final que aplico a la predicción es:

Local Santa Rosa	Local Vespucio Norte
Arquitectura de cuatro capas ocultas de 12, 120, 240 y 12. Para paquetes de ventas con un DEP de 0,78	Arquitectura de tres capas ocultas de 24, 240 y 24. Para paquetes de ventas con un DEP de 0,95

Para ambos locales se guarda el modelo para posteriormente ejecutarlo con la predicción real.

7. En el último paso se realiza la predicción real para la siguiente semana, donde siguientes son las actividades:
- a. Cargar la matriz con los ejemplos de la semana en curso, bajo los mismos 12 features que se han cargado para la matriz a entrenar como son: las cuatro primeras columnas se identifican en binario (1 y 0) la estación del año, desde la columna 5 a la columna 13 son las semanas anteriores a la actual (es decir, columna 13 = semana actual -1 y columna 5= semana actual -8).
 - b. Se predice la venta por producto de los días entre que se solicita el pedido y que llega físicamente el pedido de la siguiente forma:
 - En el caso del local de Santa Rosa, el pedido se realiza el lunes y llega el miércoles, entonces para ello se predice la venta del mismo lunes en un 50% (dado que estima que el pedido se hace con los datos de la venta de la mitad del lunes), del martes completo y la mitad del miércoles. Completado lo anterior, se ejecuta el modelo para predecir la venta de los días en espera del pedido, dato que es multiplicado por el 37% (que es el porcentaje de la venta semanal que corresponde a los días de espera de pedido). Este valor se suma a la última semana de la matriz de feature.
 - En el caso del local de Vespucio Norte, el pedido se realiza el miércoles y llega el viernes, entonces para ello debemos estimar la venta del 50% del miércoles, del jueves completo y del 50% del viernes. Al igual que para Santa Rosa, se predecir la venta de los días en espera del pedido, dato que es multiplicado por el 37%. Este valor se suma a la última semana de la matriz de feature.
 - c. Completada la matriz de feature se predice la venta de la siguiente semana y se realizan los siguientes cálculos para entregar una salida que determine el pedido por producto:
 - Se multiplica la cantidad predicha por 5, dado que el modelo aplicado es para paquetes de venta.
 - Al valor obtenido en I se resta la cantidad de productos en existencia.
 - A la cantidad obtenida en II se le suma la parte entera de la desviación estándar promedio (DEP) del modelo.
 - El valor obtenido en III, se cuadra con el empaque indicado por el franquiciante (pedido mínimo por producto), por ejemplo: si el valor en III es 35 y el empaque es de 10, se cuadra 40 para pedir.
 - Como columna adicional le agrego precio total de la compra por producto que ayuda a determinar el valor total del pedido, entonces con el ejemplo: el producto = 5, vale \$1000, entonces me indicará que el costo de comprar el producto es de \$50.000.
 - Finalmente se eliminan los productos que el franquiciante declara sin stock en base a una planilla en Excel.

La salida de todo este proceso es una planilla excel que revisa el director técnico del local, un ejemplo de la información se muestra en este cuadro:

Código producto	Nombre producto	Inventario final	empaquete	propuesta venta semana	cantidad total	precio total
1403	LOSARTAN/HIDRO 50/12.5MG 30COMP	2	0	5	5	14400
178	SALBUTAMOL SPRAY 200 DOSIS	6	0	15	15	28200
436	BROMHEXINA 4MG/5ML SOL 100ML	2	0	10	10	12000
480	CLONIXINATO DE LISINA 125MG 10COMP	0	0	10	10	12000
6	ACICLOVIR 200MG 25COMP (BE)	0	0	5	5	12000
690	OMEGA 3/6/9 60CAP SIMIOMEGA PLUS	2	0	5	5	24400
945	CEPILLO DENTAL AD 1PZA	3	0	5	5	4200
AR0008	ACIDO FOLICO 5MG 30COMP	0	5	5	5	20600
AR2015	ATORVASTATINA 10MG 30COMP (BE)	9	12	15	24	31680
AR2141	CLOTRIMAZOL 1% CREMA 20GR	2	100	15	100	56000
AR2190	AC ACETILSA 100MG 100COMP MAST	4	5	5	5	8600
AR4039	VASELINA LIQUIDA 125ML	2	5	5	5	4800
AR9010	JERINGA 5ML AGUJA 40X08	5	100	10	100	12000
BE0003	CLORFENAMINA 4MG 20COMP	15	20	25	40	14400
BE0007	KETOPROFENO 50MG 20CAP	1	10	15	20	32000
BE0010	CARVEDILOL 25MG 30COMP	5	10	10	10	38000

Con la información anterior, el director técnico, puede variar en un 5% la solicitud del pedido en base a su juicio experto sobre los siguientes puntos:

1. Producto nuevo, que es enviado por la Droguería SIMI y que fue vendido en la última semana, en este caso es muy poco probable que el modelo indique comprar el producto, pero dada la venta inicial el director técnico puede indicar que es necesario pedirlo.

2. Producto que no se vende habitualmente, pero es solicitado por el cliente, en este caso como el producto no es vendido, el modelo no lo va a predecir la petición, por lo que el director técnico lo incluye.
3. Menos habitual, pero sucede, es la variación de inicio de las estaciones, esto es con los cambios climáticos que han sucedido, hemos experimentado el verano hasta abril y la primavera desde octubre, entonces en ese caso el modelo va a predecir de acuerdo a su historia, lo que puede inducir a algún error, para ello el director técnico puede actualizar el pedido con autorización del dueño del local. **Debido a este punto también se determinó reentrenar el modelo 1 vez por mes.**

El procedimiento generado para esta nueva forma de operar en la farmacia ha modificado el flujo de pedido referenciado en el punto 6.1 figura 2 quedando de la siguiente manera:

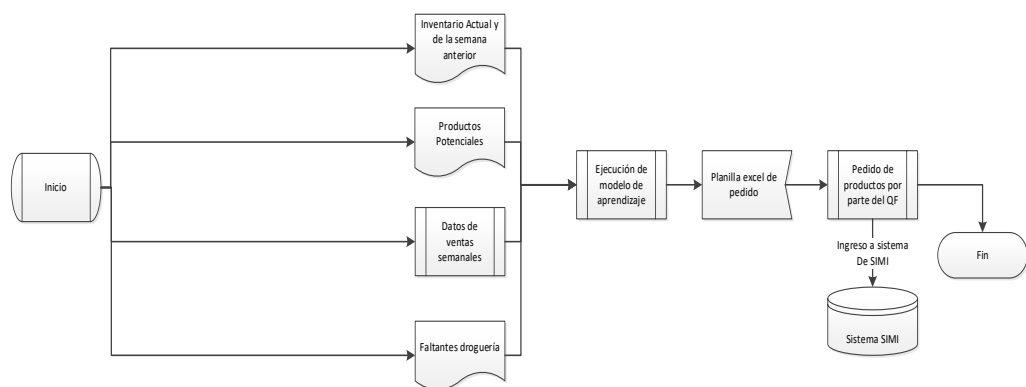


Figura 29: Flujo final de pedido

Como se muestra en la figura, el flujo consta de las siguientes etapas:

1. Obtención de ventas semanales: mediante una consulta a la base de datos se obtienen los datos de las ventas semanales para formar la matriz de features.
2. Productos potenciales: planilla de productos que se han solicitado por los clientes en la semana, que no se tiene en stock en sala de venta. Esta planilla actualiza la matriz de features.
3. Inventario Actual: se obtiene desde la base datos y se incluye para determinar la cantidad final de producto a solicitar.
4. Faltantes de stock franquiciante: planilla excel que se carga después de ejecutada la predicción para descartar pedir los productos con que no cuenta el franquiciante
5. Ejecución del modelo de aprendizaje: se realiza la ejecución del modelo de aprendizaje en base a los datos de los cuatro puntos anteriores y se obtiene como salida una planilla Excel con el pedido a solicitar.
6. Pedidos de productos por parte del director técnico: el director técnico revisa el pedido y en base a su juicio experto puede actualizar el pedido en un 5% y lo ingresa al sistema para la carga el pedido en Droguerías SIMI para su distribución.

4.3 Verificaciones del modelo

Este modelo ha estado en ejecución cada semana desde junio de 2019 y reentrenado 1 vez cada mes, debido a las condiciones cambiantes de la venta y el clima. Por lo anterior, evaluaciones del margen se realizan para los meses de mayo, junio, julio, agosto y septiembre de 2019, lo que permite comparar con valores iniciales y del modelo convencional:

Local	Mes	Compra	Venta	Margen
Santa Rosa	mayo 2019	12.975.205	19.032.813	32%
Santa Rosa	junio 2019	13.871.165	18.246.840	24%
Santa Rosa	julio 2019	16.447.153	19.680.710	16%
Santa Rosa	agosto 2019	11.198.103	18.084.470	38%
Santa Rosa	septiembre 2019	10.828.808	16.860.250	36%
Margen promedio local Santa Rosa:				29%
Vespucio Norte	mayo 2019	18.522.578	25.080.215	26%
Vespucio Norte	junio 2019	19.088.295	22.571.242	15%
Vespucio Norte	julio 2019	16.852.956	25.033.893	33%
Vespucio Norte	agosto 2019	17.734.633	22.335.870	21%
Vespucio Norte	septiembre 2019	13.010.628	22.044.950	41%
Margen promedio local Vespucio Norte:				27%

En el cuadro se observan valores de márgenes superiores al teórico que es del 35% (para el local de Santa Rosa 38% y 36% y para el local de Vespucio Norte 41%), esto (al igual que en el cuadro del método convencional) es producto de que el mes anterior se vendió menos mercadería de la comprada, entonces la mercadería en exceso se vende en el siguiente mes y por consiguiente en este último mes se compra menos mercadería generando el exceso en el margen. En el caso particular del local de Santa Rosa que cuenta con dos meses seguidos con márgenes superiores al 35% para agosto y septiembre de 2019 es producto de que la mercadería no vendida en julio 2019 (margen 16%) se vendió durante los dos meses siguiente, permitiendo comprar menos. Por lo anterior, se considera un promedio de 5 meses para una evaluación certera.

Entonces los resultados del cuadro certifica que el modelo fue muy efectivo para el local de Santa Rosa, aumentando en un **7% el margen** (un punto menos que el objetivo), lo que ha reportado mayores ganancias, en cambio para el local de Vespucio Norte el **margen ha mejorado en un 5%** (tres puntos menos que el objetivo), por lo que se ha logrado un margen general de 28% (mucho mejor que 22%) logrando un aumento de **6% del margen** inicial lo que ha sido bastante beneficioso para la empresa. Para lograr el objetivo 8% se sigue con la mejora continua del modelo.

Entre los motivos del no logro del objetivo se tienen:

Local de Santa Rosa

1. Para junio y julio se compró más mercadería apuntando a aumentar las ventas, pero siempre controlado por el modelo, es decir, el modelo indicaba una cantidad de productos X y se compra un $X+\alpha$ donde α era un máximo de un 2% sobre el total de producto indicado por el modelo, con el objetivo de mejorar las ventas. Por lo que eventualmente se puede decir que el modelo hubiese generado un aumento de margen del **9%**.

Local de Vespucio Norte

2. El bajo aumento de margen en este local es motivo de un mal resultado de una campaña agresiva para aumentar la venta en junio, julio y agosto, por lo que se compró un 5% más sobre el total del producto indicado por el modelo, por lo que teóricamente el modelo hubiese entregado un margen de **10%**.

Con los datos mostrados, se logró un aumento del margen de 6%, que ha ido en directo beneficio de las ganancias de la empresa, lo que deja conformidad con el trabajo realizado.

5. Capítulo V: Conclusión

El trabajo realizado planteó un desafío donde en primera instancia se encuentra el aprendizaje del uso de redes neuronales en aprendizaje automático. Una vez entendida la teoría se analizó el contexto de los datos de cada farmacia para determinar la mejor forma de ordenarlos, el cual finalizó con un ordenamiento en secuencia de características (features), considerando en las 4 primeras columnas las estaciones del año, las 8 siguientes las ventas por producto de semanas anteriores y el resultado o explicación de esa secuencia de semanas corresponde a las ventas de la última semana (target). En base a lo anterior, se generó una red neuronal regresiva establece y la medida de “desviación estándar promedio (DEP)” del set de pruebas como el valor a optimizar. Entonces con la información de venta del producto se logra predecir la venta de la siguiente semana, con unos valores de entrenamiento de:

- Para farmacia de Santa Rosa 2: Desviación estándar promedio de 0,78
- Para farmacia Vespucio Norte: Desviación estándar promedio de 0,95

Con los resultados del entrenamiento, se logra predecir la venta futura obteniendo mejoras en el margen:

- Para farmacia de Santa Rosa 2: Margen 29%, mejora de 7%
- Para farmacia Vespucio Norte: Margen 27%, mejora de 5%

Por lo anterior, la conclusión del trabajo es, que si bien no se logró el objetivo de aumentar en un 8% el margen de ganancia de la empresa (diferencia entre compra y ventas), se logró aumentar el margen en un 7% para local de Santa Rosa y en un 5% para local de Vespucio Norte en los últimos 5 meses, lo que deja la satisfacción de que se va en el camino correcto, tomando en consideración que el motivo del no logro del objetivo tuvo que ver más con estrategias agresivas para aumentar el volumen de venta total durante los meses de junio, julio y agosto, que hizo que se aumentara el pedido a solicitar en un 2% en Santa Rosa y en un 5% en Vespucio Norte por sobre la predicción del modelo. Creemos que si no se hubiese generado la estrategia de invierno se habría logrado un aumento de 9% del margen en Santa Rosa y de 10% en Vespucio Norte. De todas maneras, esto es mas bien una apreciación que necesitaría de un experimento controlado para ser confirmada.

Esta implementación representó un desafío de innovación en las farmacias de la franquicia; si bien muy estudiada y fundamentada con un marco teórico, logró una mejora en el margen. Además, demostró que un modelo de aprendizaje de máquinas (en particular con redes neuronales) no es un modelo solamente utilizado por grandes corporaciones y con grandes cantidades de datos, sino que también un beneficio para pymes al predecir con alta precisión lo que se venderá la siguiente semana. El modelo de aprendizaje automático se realizó de manera que pueda sea utilizado por cualquier farmacia con su correspondiente sensibilización, en cuanto a arquitectura, nodos y features y la realidad de cada local. Esto entregó una gran satisfacción por el aprendizaje de la tecnología aplicada logrando entrenar una red neuronal en una cantidad aproximada de 100.000 ejemplos, así como también entregar un valor agregado a las pymes del rubro para mejorar sus márgenes, solicitando la mercadería justa y necesaria para no tender a un sobre stock que finalmente merma las ganancias.

En lo que respecta al proceso se hizo una modificación del flujo de solicitud de pedido para los locales, donde se incluyó la ejecución del modelo como parte de él, disminuyendo la carga laboral al director técnico de farmacia, permitiéndole dedicar más esfuerzo a sus funciones y apoyar la venta de productos.

Para finalizar, existiendo una conformidad con el modelo, los pasos siguientes para una mejora continua son:

1. Continuar con los reentrenamientos mensuales del modelo, con una revisión de arquitectura, cantidad de nodos y features.
2. Entrenar con otros modelos de aprendizaje automático principalmente random forest, regresión logística y comparar.
3. Agregar o modificar características a la matriz inicial de datos a entrenar, como son: datos de ventas por producto cada dos semanas, por mes e incluir devoluciones de ventas para los períodos.
4. Cambiar los features de estaciones por la de meses del año.
5. Establecer líneas de producto, donde se determine qué productos actúan similares en su venta independientes de las variables ambientales, por lo que el modelo pueda entrenar con menos valores que discernir lo que puede minimizar el error.
6. Entrenar con modelos de redes recurrentes que consideran el tiempo en la misma arquitectura como parte de una futura evolución.

Todo lo anterior, enfocado a lograr la menor desviación estándar promedio para la solicitud de pedido de cada local, en cada instante de tiempo de acuerdo con las condiciones cambiantes del entorno, con el objetivo final de lograr el 30% de margen.

Bibliografía

Libros

1. Sebastian Raschka – Vahid Mirjalili. Python Machine Learning: Aprendizaje automático y aprendizaje profundo con Python, scikit-learn y tensorflow, segunda edición, Editorial Marombo, España, año 2019.
2. François Chollet, Deep learning with Python. Editorial Hanning. Estados Unidos, 2018.
3. Yoav Goldberg. Neural Network Methods for Natural Language Processing, Editorial: Morgan & Claypool Publishers. Estados Unidos, 2017.
4. Charu C. Aggarwal, Data Mining: The Textbook, Editorial: Springer. Estados Unidos, 2015.

Artículos

1. Carlos Alberto Ruiz. Cátedra: Informática Aplicada a la Ingeniería de Procesos – Orientación I. Tema: Redes Neuronales: Conceptos Básicos y Aplicaciones. Universidad tecnológica nacional, Rosario, 2001.
2. Alexandra Pollett y Juan Pedro Hernández. Proyecto de título “implementación de un modelo predictivo basado en data mining y soportado por sap predictive analytics en retails”, Perú, 2016.
3. Pedro Larranaga, ~ Inaki ~ Inza, Abdelmalik Moujahid. Artículo: Tema 8 - “Redes Neuronales”. Departamento de Ciencias de la Computación e Inteligencia Artificial Universidad del País Vasco–Euskal Herriko Unibertsitatea, País Vasco, 2009, [en línea] <<http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t8neuronales.pdf>>

Apéndices

Apéndice A: Código creación de la red neuronal

```
from keras.layers import Dense

from keras.callbacks import EarlyStopping

from matplotlib import pyplot as plt

from sklearn.preprocessing import normalize

import keras

es = EarlyStopping(monitor='val_loss', min_delta=0, patience=25, verbose=1,
mode='auto', baseline=None)

M = X_Train.shape[1]

N = X_Train.shape[1]

C = Y_Train.shape[1]

h1 = 12

h2 = 120

h3 = 240

h4 = 12

red_reg = keras.Sequential()

red_reg.add(Dense(h1, input_dim=N, activation='relu', name='primera_capa'))

red_reg.add(Dense(h2, activation='relu', name='segunda_capa'))

red_reg.add(Dense(h3, activation='relu', name='tercera_capa'))

red_reg.add(Dense(h4, activation='relu', name='cuarta_capa'))

red_reg.add(Dense(1, kernel_initializer='normal', name='capa_de_output'))

red_reg.compile(

loss='mean_absolute_error',

optimizer='adam',

metrics=['mean_absolute_error'])
```

red_reg.summary()

Layer (type)	Output Shape	Param #
=====		
primera_capa (Dense)	(None, 12)	156

segunda_capa (Dense)	(None, 120)	1560

tercera_capa (Dense)	(None, 240)	29040

cuarta_capa (Dense)	(None, 12)	2892

capa_de_output (Dense)	(None, 1)	13
=====		
Total params: 33,661		
Trainable params: 33,661		
Non-trainable params: 0		

Apéndice B: Código creación de la red neuronal

```
h_reg = red_reg.fit(X_Train, Y_Train,  
epochs=200,  
batch_size=12,  
validation_data=(X_Val,Y_Val),  
verbose=2,  
callbacks=[es]  
)
```

Epoch 96/200

- 10s - loss: 0.3907 - mean_absolute_error: 0.3907 - val_loss: 0.3919 -
val_mean_absolute_error: 0.3919

Epoch 97/200

- 10s - loss: 0.3899 - mean_absolute_error: 0.3899 - val_loss: 0.3910 -
val_mean_absolute_error: 0.3910

Epoch 98/200

- 11s - loss: 0.3909 - mean_absolute_error: 0.3909 - val_loss: 0.3921 -
val_mean_absolute_error: 0.3921

Epoch 00098: early stopping

Predicción y gráficas

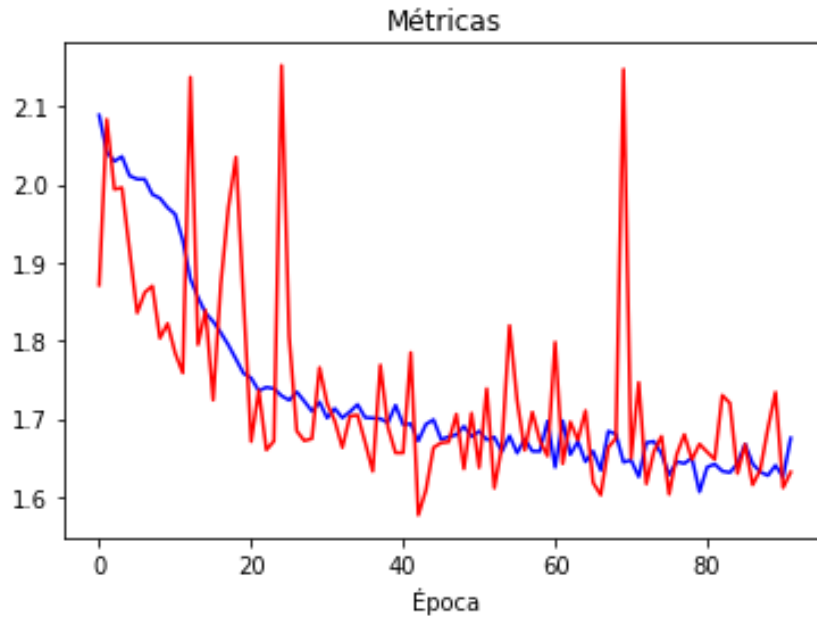
```
from sklearn.metrics import r2_score  
  
import numpy as np  
  
import locale  
  
plt.plot(h_reg.history['mean_absolute_error'], color='blue')  
  
plt.plot(h_reg.history['val_mean_absolute_error'], color='red')  
  
plt.title('Métricas')
```

```
plt.xlabel('Época')  
plt.show()  
preds = red_reg.predict(X_Test)  
diff = preds - Y_Test  
mean = np.mean(diff)  
std = np.std(diff)  
locale.setlocale(locale.LC_ALL, "en_US.UTF-8")  
print("[TEST] mean: {:.2f}, std: {:.2f}".format(mean, std))  
print('Accuracy:', r2_score(Y_Test,preds))
```


Apéndice C: Gráficas de resultado red neuronal

Local Santa Rosa para ventas exactas:

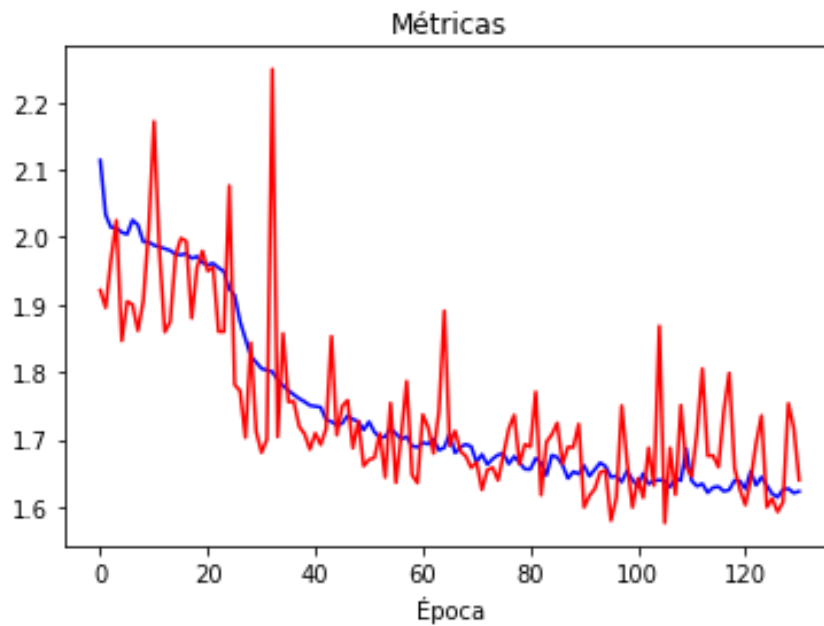
1. Con tres capas ocultas de 24, 240 y 24 neuronas respectivamente



Desviación estándar: 4,33

Precisión: 67,8%

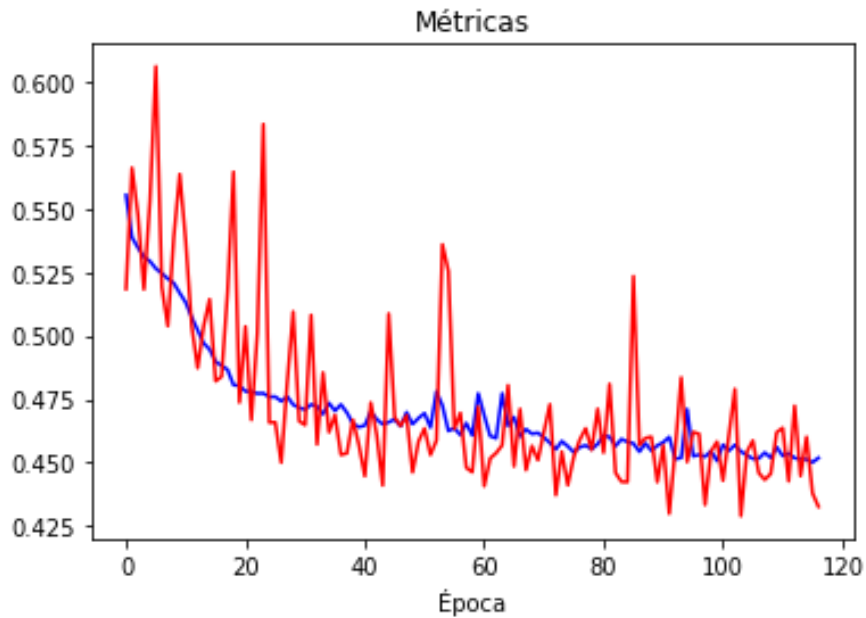
2. Con tres capas ocultas de 12, 120 y 12 neuronas respectivamente:



Desviación estándar: 4,43
Precisión: 66,3%

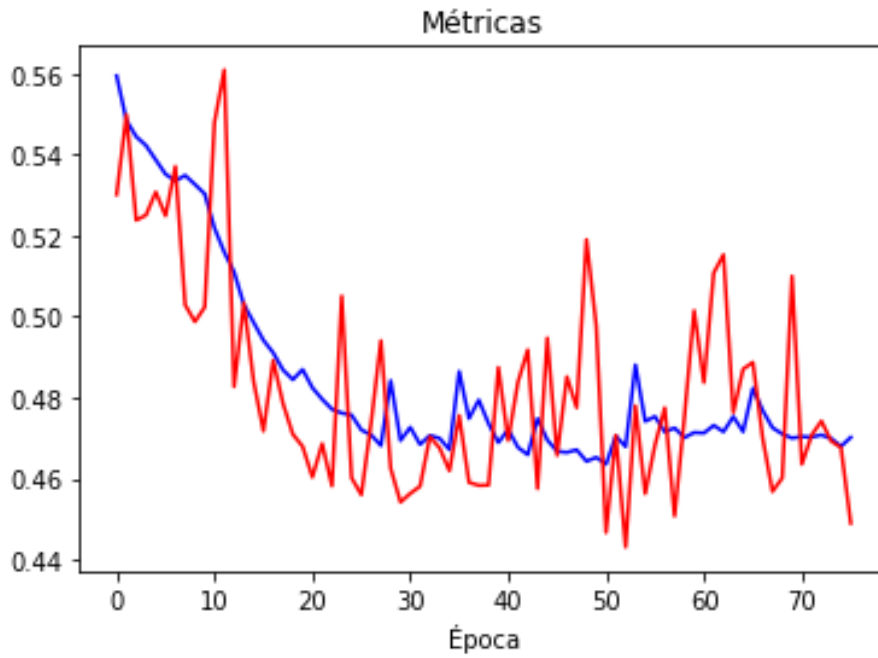
Local Santa Rosa para paquetes de venta (5):

1. Con cuatro capas ocultas de 12, 120, 240 y 12 neuronas respectivamente



Desviación estándar: 0,78
Precisión: 68,9%

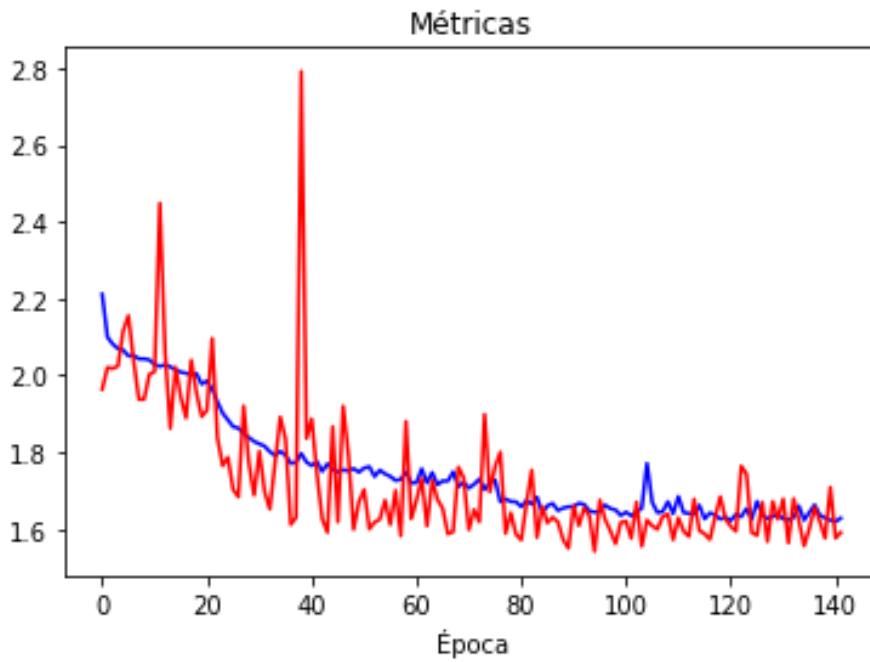
2. Con tres capas ocultas de 12, 120 y 12 neuronas respectivamente:



Desviación estándar: 0,79
 Precisión: 68,2%

Local Vespucio Norte para ventas exactas:

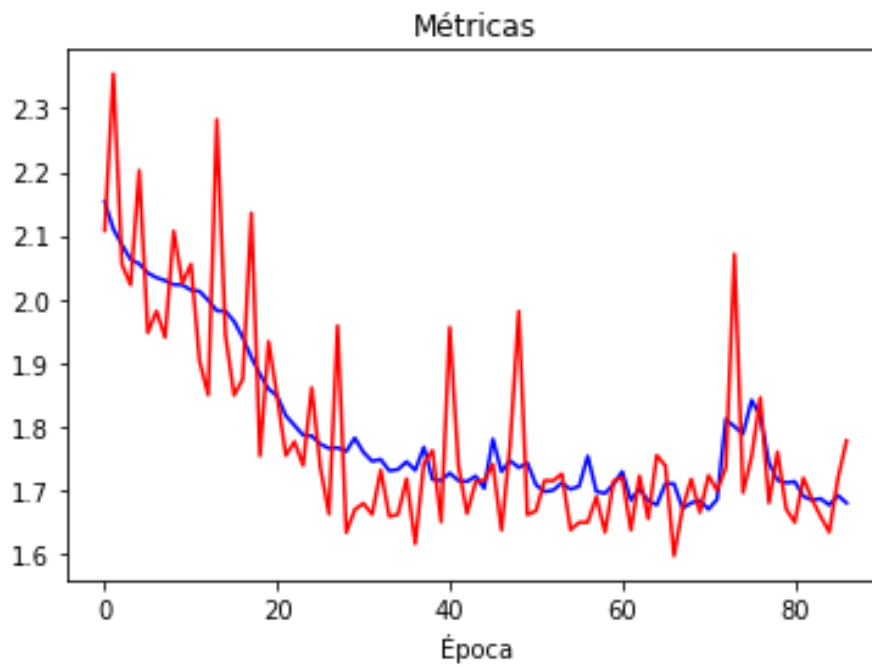
1. Con tres capas ocultas de 12, 120 y 12 neuronas respectivamente (Mejor Resultado para este punto):



Desviación estándar: 4,32

Precisión: 68,9%

2. Con tres capas ocultas de 24, 240 y 24 neuronas respectivamente:

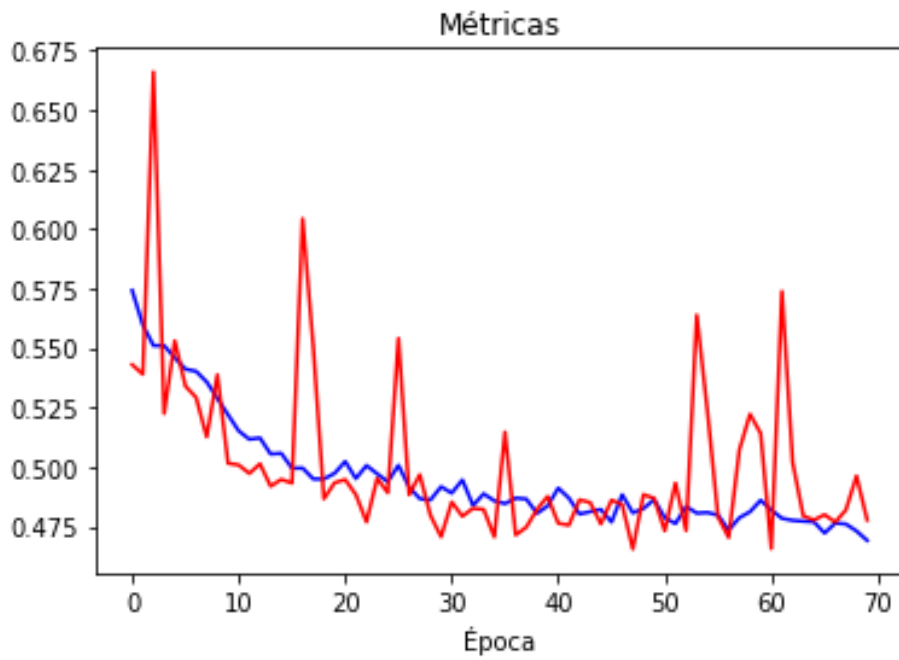


Desviación estándar: 4,44

Precisión: 66,9%

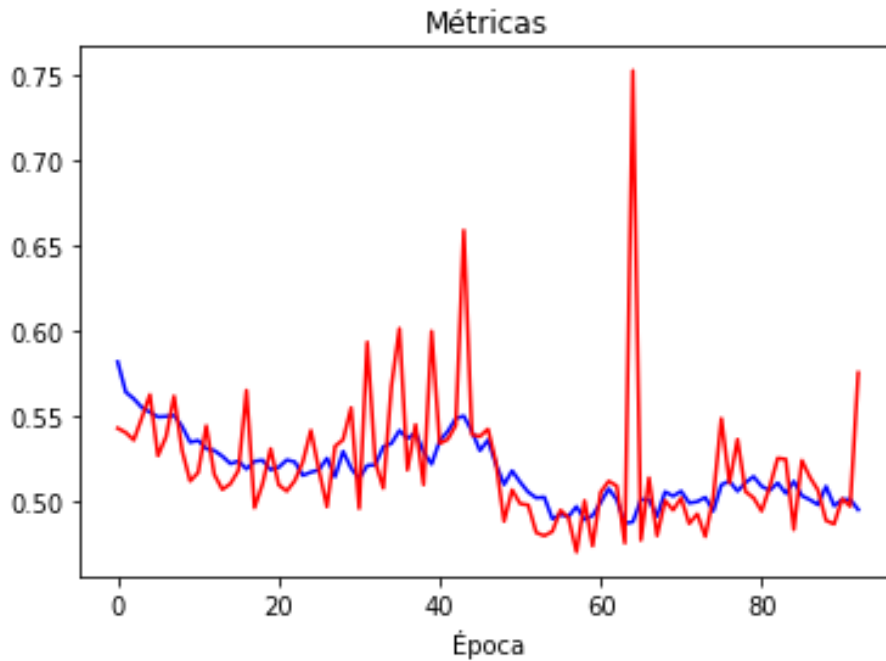
Local Vespucio Norte para paquetes de venta (5):

1. Con cuatro capas ocultas de 24, 240 y 24 neuronas respectivamente



Desviación estándar: 0,95
 Precisión: 66,6%

2. Con cuatro capas ocultas de 12, 120, 240 y 12 neuronas respectivamente:



Desviación estándar: 0,96
 Precisión: 65,4%