UIDE
Powered by
Arizona State University

Maestría en
SISTEMAS DE INFORMACIÓN

Mención **Inteligencia de Negocios y Analítica de Datos Masivos.**

Tesis previa a la obtención del título de Magíster en Sistemas de Información mención Inteligencia de Negocios y Analítica de Datos Masivos.

AUTOR: Bolívar Alvarez Muñoz

Carlos Rosero Cumbillo

Ricardo Aimacaña Gualpa

Edwin Chacón Cajamarca

TUTOR: Paulina Vizcaíno Imacaña

ANÁLISIS PREDICTIVO DE VENTAS BASADO EN MODELOS DE MACHINE LEARNING PARA EL NEGOCIO DE GENÉTICA AVÍCOLA

APROBACIÓN DEL TUTOR

Yo, Paulina Vizcaíno Imacaña, certifico que conozco los autores del presente trabajo siendo la responsable exclusiva tanto de su originalidad y autenticidad, como de su contenido.

Paulina Vizcaíno Imacaña
DIRECTOR DE TESIS

MAESTRÍA EN SISTEMAS DE LA INFORMACIÓN, MENCIÓN EN INTELIGENCIA DE NEGOCIOS Y ANALÍTICA DE DATOS MASIVOS

CERTIFICACIÓN DE AUTORÍA

Nosotros, Bolívar Alvarez Muñoz, Carlos Rosero Cumbillo, Ricardo Aimacaña Gualpa, Edwin Chacón Cajamarca, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada.
Cedo mis derechos de propiedad intelectual a la Universidad Internacional del Ecuador, para que sea publicado y divulgado en Internet, según lo establecido en la Ley de Propiedad Intelectual, su reglamento y demás disposiciones legales.

....

Bolívar Alvarez Muñoz Carlos Rosero Cumbillo

C.I.: 1719468496 C.I.: 1710156959

....

Ricardo Aimacaña Gualpa Edwin Chacón Cajamarca

C.I.: 1721060620 C.I.: 1725126880

DEDICATORIA

Dedicamos este trabajo, a nuestras familias y seres queridos, quienes nos han brindado su apoyo incondicional durante todo este tiempo, el cual ha sido lleno de desafíos y muy gratificante camino hacia la obtención de esta maestría. Su paciencia, comprensión y constante motivación ha sido el pilar fundamental que nos ha sostenido a lo largo de todo este proceso. Este logro es tanto nuestro como de todos ustedes.

AGRADECIMIENTOS

Queremos agradecer, en primer lugar, a Dios, por ser nuestra luz y fortaleza. Sin su guía y bendición, no habríamos logrado completar este importante reto en nuestras vidas.

Nuestro más sincero agradecimiento a nuestros profesores, cuyas enseñanzas, guía, dedicación y paciencia han sido fundamentales en nuestra formación académica. Sus conocimientos y experiencias han sido de inspiración para ir más allá de nuestros propios límites.

Agradecemos también a nuestros compañeros de clase, con quienes compartimos no solo conocimientos, sino también largas noches de trabajo, esfuerzo y cansancio, superando desafíos y celebrando victorias. Esta experiencia, aunque exigente, nos unió y convirtió este viaje en uno colaborativo y enriquecedor. Finalmente, extendemos nuestro agradecimiento a la empresa que nos brindó el acceso a herramientas, recursos e información, permitiéndonos consolidar lo aprendido y aplicar nuestras habilidades en un escenario real.

Este trabajo es un testimonio del esfuerzo conjunto, y del deseo de seguir creciendo y aprendiendo.

INDICE

Tabla de Contenido

LISTA DE FIGURAS

RESUMEN

Este proyecto pretende generar predicciones de finanzas basándose en la data histórica de las ventas en el negocio avícola. Usaremos aprendizaje automático para probar tres modelos y encontrar el menor error porcentual en la predicción del segundo semestre del año 2024. Primero se pretende crear un análisis exploratorio de la data de ventas para proceder a limpiar y cargar la misma data en dataframes de Python para su procesamiento. Posteriormente se implementará modelos de regresión lineal, random forest y red neuronal para conseguir unas predicciones de ventas del segundo semestre del año 2024. Finalmente se calculará el error para elegir la mejor predicción que se ajuste a nuestra data y poder simplificar la toma de decisiones en las predicciones de ventas.

ABSTRACT

This project aims to generate financial predictions based on historical sales data in the poultry business. We'll use machine learning to test three models and find the smallest percentage error in the second semester of 2024-year prediction. First, create an exploratory analysis of the sales data to proceed to clean and load the same data into Python dataframes for processing. Subsequently, linear, random forest and neural network regression models will be implemented to achieve sales predictions for the year 2024. Finally, the error will be calculated to choose the best prediction that fits our data and be able to simplify decision-making in the sales predictions of each coming year.

PALABRAS CLAVE

Aprendizaje automático, machine learning, ML, regresión lineal, random forest, red neuronal, aprendizaje profundo, aprendizaje no supervisado, open-source, ERP, finanzas, exactitud, overfiting, ganancia, serie temporal, dataframe, neurona, capas, entrenamiento, epoch, error cuadrático medio, error absoluto medio, proyección.

CAPITULO 1

INTRODUCCIÓN

El Negocio Avícola (NA) posee organizaciones comerciales enfocadas al alimento balanceado, huevo comercial, pollito de cría y materias primas; el proceso de ventas lo soporta un ERP que puede generar informes, que ayudan al área comercial para tomar decisiones, presupuestos y planificación. Las predicciones de ventas se hacen según los datos históricos y la experiencia de las personas de mercadeo y la gerencia del negocio. El presupuesto año a año lo ideal es no bajarlo, sino mantenerlo o incrementarlo.

Considerando el contexto previo de nuestra empresa, proponemos un proyecto de innovación centrado en el análisis predictivo de ventas para los próximos años.

Es fundamental reconocer que el éxito de una compañía radica en su capacidad para anticipar, planificar e implementar eficazmente sus productos o servicios en el mercado, con el objetivo de maximizar las ventas. Para lograrlo, es necesario desarrollar presupuestos y planificaciones que permitan a todas las áreas operar de manera eficiente, entregando productos y servicios en excelentes condiciones y dentro de plazos concretos. En este contexto, nuestro proyecto adquiere relevancia; basándonos en los datos históricos de ventas, podemos construir modelos predictivos que nos ayuden a anticipar las ventas futuras, considerando el crecimiento experimentado en los últimos años.

Actualmente, contamos con datos postpandemia del Negocio Avícola que utilizaremos para predecir las ventas. La información recopilada desde 2022, 2023 y primer semestre del 2024 como base para probar y ajustar nuestros modelos.

El proyecto se estructura en varias etapas. En primer lugar, realizaremos un análisis y limpieza exhaustiva de los datos, organizando y filtrando la información de ventas, clientes, artículos, cantidades e importes. A continuación, procederemos a crear y entrenar diversos modelos utilizando los datos depurados. Durante el análisis, evaluaremos, compararemos y ajustaremos los modelos para minimizar el error en las predicciones. Finalmente, con un modelo ajustado a nuestras necesidades, analizaremos las predicciones de nuevos datos y extraeremos conclusiones relevantes para el área de ventas.

CASO DE ESTUDIO

Antecedentes

El Negocio Avícola (NA) fue fundada en la década de 1970 con un solo galpón y 10 empleados, la compañía ha crecido significativamente hasta contar con más de 75 galpones, una planta de alimento balanceado y su propia incubadora, empleando a más de 450 familias. Su visión es ser la mejor opción para los clientes, y su misión es producir y comercializar productos agroindustriales de manera eficiente con un excelente capital humano y valores empresariales sólidos. En el último trimestre de 2021, la empresa fue absorbida por una entidad mayor, lo que ha aumentado su presencia, mejorado la atención a sus clientes y sostenido su crecimiento.

OBJETIVOS

General

- Utilizar un modelo predictivo de ventas para impulsar el giro de negocio y maximizar su proyección de ganancias al año 2024

Especifico

- Investigar un modelo de Machine Learning viable que genere predicciones mediante datos históricos del giro del negocio

- Cambiar el modelo tradicional basado en experiencias a un modelo predictivo en Machine Learning en la creación del presupuesto de ventas.

- Generar informes de presupuestos basado en modelos de ML para la toma de decisiones del año 2024

- Definir y optimizar modelos de Machine Learning que mejor se adapten a la empresa y a su giro de negocio.

ALCANCE DEL PROYECTO

El proyecto tiene como finalidad el desarrollo, entrenamiento y testeo de modelos predictivos de machine learning en python, se planea desarrollar un máximo de 3 modelos para analizar y comparar los resultados obtenidos definiendo el mejor modelo.

Requisitos del Proyecto

- El modelo debe ser desarrollado en Python.
- El modelo debe tener un porcentaje de acierto mayor al 80% con datos de entrenamiento desde el 2022 hasta el 2023, y datos del primer semestre del 2024 como muestra de revisión.

Límites del proyecto

- El modelo se ajustará a la predicción del segundo semestre del 2024 basándonos en datos históricos de ventas correspondientes a los años 2022, 2023 y primer semestre del 2024.
- No incluye el despliegue en producción del modelo.

Impacto de negocio

Las tecnologías modernas, como lo es el machine learning, permiten a las empresas obtener ventajas competitivas y optimizar sus operaciones. La aplicación eficaz del presente proyecto permitirá a la empresa del Negocio Avícola (NA) definir de manera más precisa las proyecciones de ventas para los siguientes años. Esto representa un impacto muy significativo en el modelo de negocio en varios aspectos. Las proyecciones que se obtendrán de los modelos de machine learning serán la clave para simplificar la toma de decisiones, mejorar las estrategias de mercado e incluso adelantarse a la competencia. Como resultado la empresa verá mejoras positivas en su gestión y participación de mercado.

Problemática

ANÁLISIS PREDICTIVO DE VENTAS BASADO EN MODELOS DE MACHINE LEARNING PARA EL NEGOCIO AVÍCOLA

Stakeholders y áreas del negocio

Las predicciones de ventas del Negocio Avícola se hacen según los datos históricos y la experiencia de las personas del área directamente involucrada que es Mercadeo y la Gerencia del Negocio. El presupuesto anual de ventas se elabora con el 80% de los clientes fiables a quienes se les consulta un aproximado de sus necesidades y se le incrementa un porcentaje consensuado; para los no fiables se estima en base a las compras del año en curso. Lo ideal es no bajar el presupuesto respecto al año anterior, sino mantenerlo o incrementarlo. La necesidad de tener un presupuesto que se ajuste a los datos históricos y se pueda proyectar para añso futuros es de vital importancia para la empresa Negocio Avícola. Considerando el contexto previo de la empresa, proponemos un proyecto de innovación centrado en cambiar el modelo tradicional basado en experiencias por un análisis predictivo de ventas utilizando modelos de ML.

Fuentes de información

El proyecto tiene como finalidad el desarrollo, entrenamiento y testeo de modelos predictivos de machine learning en Python, se planea desarrollar un máximo de 3 modelos para analizar y comparar los resultados obtenidos definiendo el mejor modelo. Los modelos se los extraerá de investigaciones y publicaciones previas en áreas de finanzas y machine learning.

La data se la extraerá del módulo ERP de la empresa teniendo en cuenta un histórico correspondiente del 2022 y 2023.

El ERP es un sistema basado en una base de datos Oracle que tiene varios módulos que permiten la gestión de la empresa en sus distintas áreas como:

- Gestión económico-financiera, contabilidad general-analítica, tesorería, caja, presupuestos, etc.
- Gestión de entradas/salidas de artículos, así como la logística, y gestión de almacenes.
- Gestión de compras en materias primas, productos, elementos de inmovilizado y/o servicios.
- Gestión de todos los procesos relacionados con la venta-distribución a clientes.
- Generación de los documentos digitales con su firmado digital, almacenamiento y envío al cliente.

- Definición de la estructura productiva, programación, producción, planificación, seguimiento.
- Gestión de mantenimiento preventivo/correctivo, gestión de reparaciones internas/externas.
- Gestión de RRHH desde la parte económica, control y cumplimiento de norma laboral existente.
- Gestionar la calidad de productos/procesos con sus certificaciones y normativas.
- Herramientas de extracción de datos, conexión e integración con otros sistemas.

Justificación

Las proyecciones de ventas son una herramienta fundamental para la toma de decisiones estratégicas en la empresa. Permiten a los gerentes y directivos anticipar el comportamiento del mercado y tomar medidas para alcanzar los objetivos de ventas establecidos. Con lo expuesto anteriormente, un proyecto de innovación que permita predecir el crecimiento basado en históricos es de gran ayuda para cualquier compañía en términos de precisión, y reducción de costos en planificación.

Este documento presenta la justificación de la proyección de ventas del Negocio Avícola durante el período 2024; la proyección se basa en un análisis exhaustivo de datos históricos, tendencias del mercado y factores externos que podrían afectar las ventas.

CAPITULO 2

Marco Teórico

Machine Learning

Es un conjunto de algoritmos que se utilizan en el campo de las ciencias de la computación para definir sistemas que no son programados para responder de manera estática, en lugar de eso, su respuesta se basa en los datos históricos de problemas resueltos del dominio de aplicación que se procesan, a medida que los sistemas procesan más datos, más complejo es su aprendizaje y más acertada es su respuesta. (Mahesh, 2018).

Aprendizaje Supervisado

En el campo de la inteligencia artificial, el aprendizaje supervisado es un cambio que experimenta un sistema basado en un conjunto de datos de ejemplo que tienen su respectivo resultado, los sistemas aprenden de estos ejemplos y basan sus predicciones en los resultados entregadas. A diferencia del aprendizaje no supervisado donde los datos de aprendizaje no son etiquetados y no tiene un resultado. (Chaviano Arteaga).

Los modelos ML que se utilizarán para el análisis de predicción de ventas son los siguientes.

Regresión Lineal

La regresión lineal es una técnica estadística utilizada para modelar la relación entre una variable dependiente (Y) y una o más variables independientes (X). Esta relación se expresa a través de una ecuación lineal, lo que permite hacer predicciones sobre el valor de Y en función de los valores de X. (Montgomery, Peck, & Vining, 2021)

Los objetivos a aplicar de la regresión lineal son:

- **Predicción:** Estimar valores futuros de la variable dependiente basados en valores conocidos de las variables independientes.
- **Explicación:** Identificar la fuerza y dirección de la relación entre las variables, y determinar qué variables independientes tienen un mayor impacto sobre la variable dependiente.
- **Modelado:** Crear un modelo matemático que represente la relación entre las variables, permitiendo su análisis y comprensión. (Montgomery, Peck, & Vining, 2021)

Los modelos de regresión lineal tienen dos tipos:

- **Simple:** Una sola variable independiente (X) se utiliza para predecir la variable dependiente (Y).
- **Múltiple:** Múltiples variables independientes (X1, X2, ...) se utilizan para predecir la variable dependiente (Y). (Montgomery, Peck, & Vining, 2021)

Ecuación de la Regresión Lineal Simple

La ecuación general de la regresión lineal simple es:

$$Y = \beta 0 + \beta 1X + \varepsilon$$

- **Y:** Variable dependiente
- **β0:** Intercepto (valor de Y cuando X es igual a 0)
- **β1:** Pendiente (cambio en Y por cada unidad de cambio en X)
- **X:** Variable independiente
- **ε:** Error aleatorio (Montgomery, Peck, & Vining, 2021)

Supuestos de la Regresión Lineal

- **Linealidad:** La relación entre las variables es lineal (Draper & Smith, 1981).
- **Independencia:** Los errores son independientes entre sí (Draper & Smith, 1981).
- **Homocedasticidad:** La varianza de los errores es constante para todos los valores de X (Draper & Smith, 1981).
- **Normalidad:** Los errores se distribuyen normalmente. (Draper & Smith, 1981)

Procedimiento General

1. **Recopilación de datos:** Se recolectan datos sobre las variables de interés.
2. **Exploración de datos:** Se realiza un análisis exploratorio de los datos para identificar patrones, relaciones y posibles problemas.
3. **Estimación de los parámetros:** Se utilizan métodos estadísticos (como mínimos cuadrados ordinarios) para estimar los valores de los parámetros β0 y β1.
4. **Evaluación del modelo:** Se evalúa la calidad del modelo mediante estadísticas como el coeficiente de determinación (R^2), el error estándar de la estimación y pruebas de hipótesis.
5. **Interpretación de los resultados:** Se interpretan los resultados del modelo en el contexto del problema en estudio. (Draper & Smith, 1981)

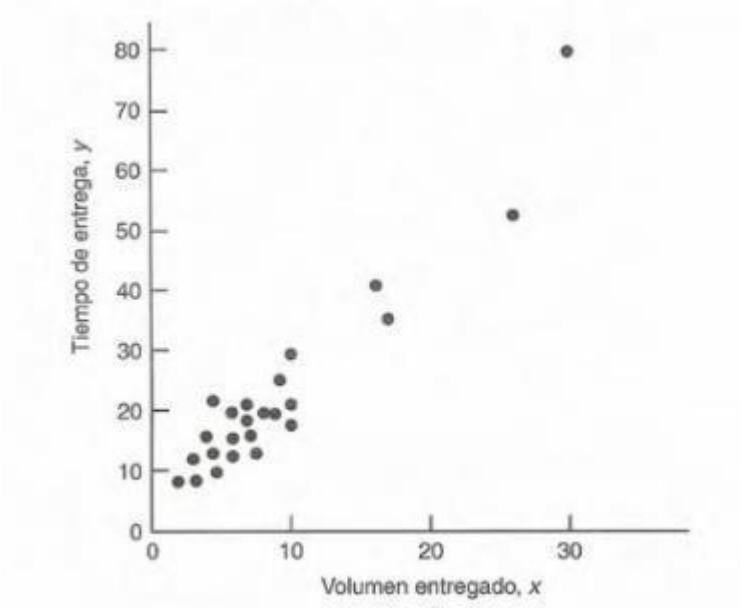


Ilustración 1 : Diagrama de dispersión para el volumen entregado *(Montgomery, Peck & Vining, 2021)*

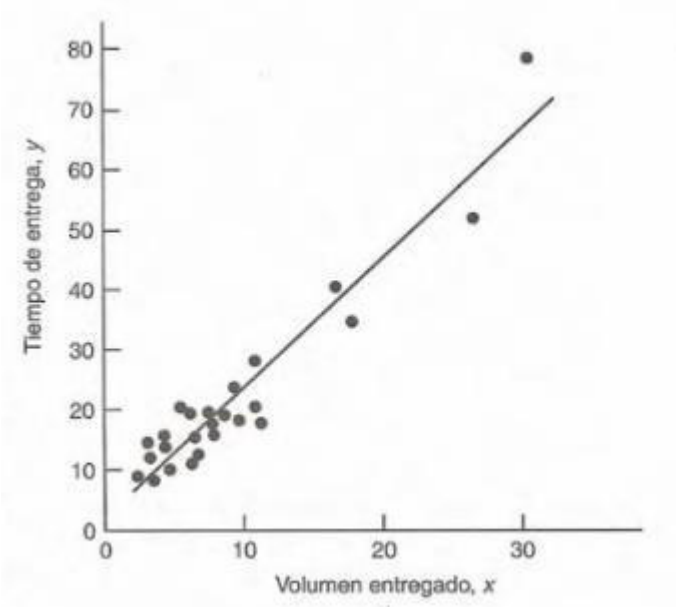


Ilustración 2: Relación rectilínea entre el tiempo de entrega y el volumen entregado (Montgomery, Peck & Vining, 2021)

Árbol de Decisiones

En el área avícola, la asertividad en la proyección de ventas es importante, debido a la naturaleza perecedera de los productos, la variabilidad en la demanda y los ciclos de producción de las materias primas, y en ciertos casos la complejidad del entorno a nivel país es necesario contar con herramientas que permitan la toma de decisiones. Los métodos tradicionales de

predicción, aunque útiles, a menudo no permiten la visualización de la complejidad de los datos de ventas actuales. En este contexto, los árboles de decisión aparecen como una técnica efectiva dentro del aprendizaje automático (ML), siendo una herramienta poderosa para modelar las complejas interacciones entre múltiples variables que afectan las ventas (Breiman et al., 1984).

Fundamentos de los Árboles de Decisión

Los árboles de decisión son un tipo de modelo predictivo que se utiliza tanto en problemas de clasificación como de regresión. Su popularidad dentro del mundo del ML se debe a su capacidad para manejar tanto datos categóricos como continuos, es una excelente alternativa por su alta precisión y explicabilidad; además de su capacidad para capturar relaciones no lineales entre las variables predictoras y la variable objetivo (Hastie, Tibshirani, & Friedman, 2009)(Patricio Fernandez, Machine Learning(ML) – Tipos de algoritmos I, T-SII_21_001043_01.pdf; https://eig.brightspace.com/d2l/le/content/146821/viewContent/1084329/View).

Un árbol de decisión se lo representa de forma jerárquica para una mejor interpretación, este contiene los nodos y las hojas como se muestra la Figura 3.

Árbol de decisión representada de forma jerárquica.

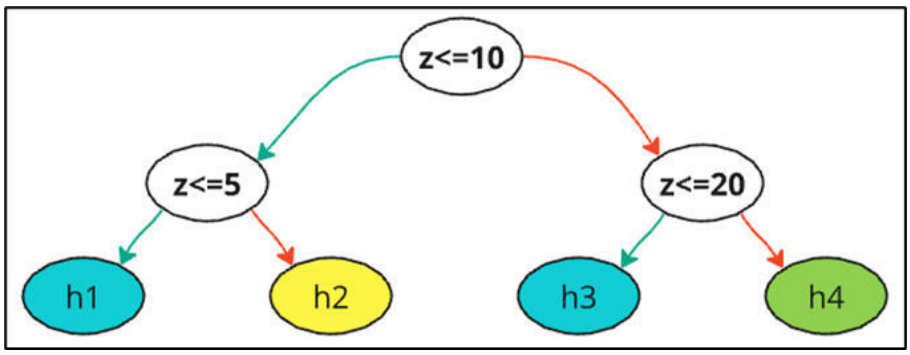


Ilustración 3: Árbol de decisiones de forma jerárquica

En un árbol de decisión, los datos se dividen de forma recursiva en subconjuntos cada vez más pequeños a través de decisiones binarias basadas en el valor de una o más características. Este proceso de particionamiento recursivo continúa hasta que se cumple un criterio para detenerlo como: profundidad máxima del árbol, un número mínimo de muestras en una hoja o una reducción mínima en la impureza (James et al., 2013). Cada nodo del árbol representa una característica del conjunto de datos, y cada rama representa un valor de esa

característica; las hojas del árbol tendrán las predicciones finales; que, en el caso de un problema de regresión, son valores continuos, como las ventas proyectadas.

Una de las fortalezas más importante de los árboles de decisión es su explicabilidad; dado que las decisiones dentro del árbol son explícitas y basadas en reglas claras, los árboles de decisión permiten entender cómo las distintas variables afectan las ventas, lo que facilita la toma de decisiones estratégicas (Quinlan, 1986). Por ejemplo, en el sector avícola, un árbol de decisión podría revelar que las ventas de pollitos bebé son significativamente más altas durante ciertos meses del año, o que las promociones tienen un impacto considerable en la demanda de alimento balanceado.

Ventajas y Limitaciones de los Árboles de Decisión en la Proyección de Ventas

Ventajas:

1. Interpretabilidad: Los árboles de decisión son fáciles de entender y visualizar, esto facilita la interpretación de los resultados por parte de quienes toman las decisiones empresariales. Esta característica es especialmente útil en la proyección de ventas, donde es importante comprender no solo las predicciones, sino también los factores que las impulsan (Quinlan, 1986).
2. Flexibilidad: Pueden manejar fácilmente variables categóricas como numéricas, sin la necesidad que los datos estén escalados. Esto es particularmente útil en el sector avícola, donde los datos pueden incluir una mezcla de variables continuas (como precios y cantidades) y categóricas (como familias de productos o regiones geográficas).
3. Captura de Interacciones Complejas: Los árboles de decisión permiten capturar interacciones no lineales entre las variables predictoras, lo que les facilita modelar relaciones complejas que no saltan a la vista en un análisis tradicional.

Limitaciones:

4. Propensión al Sobreajuste: Los árboles de decisión son propensos al sobreajuste, especialmente si no se controla su crecimiento a través de ciertos criterios. Esto puede resultar en un modelo que se ajusta perfectamente a los datos de entrenamiento, pero que no logra generalizar bien cuando se alimenta al modelo con datos nuevos. Se pueden utilizar técnicas para mitigar el sobreajuste como: la poda del árbol, la

implementación de modelos de ensamblado como Random Forest o Gradient Boosting (Breiman, 2001).

5. Inestabilidad: Los árboles de decisión son sensibles a pequeñas variaciones en los datos con los que se alimenta el modelo. Un cambio pequeño en los datos de entrenamiento puede resultar en un árbol completamente diferente a uno anterior y esto puede afectar la estabilidad de las predicciones; esto se puede mitigar utilizando modelos de ensamblado que promedian las predicciones de múltiples árboles para reducir la variabilidad (Hastie et al., 2009).
6. Limitación en la Captura de Relaciones Lineales: Aunque los árboles de decisión son excelentes para capturar relaciones no lineales, pueden no ser tan buenos para modelar relaciones lineales simples; para estos casos, modelos más simples como la regresión lineal podrían ser más adecuados.

Mejoras del Modelo y Técnicas Avanzadas

Para sobrellevar las limitaciones expuestas de los árboles de decisión, se pueden implementar varias técnicas avanzadas como:

- La poda es una técnica que reduce el tamaño del árbol eliminando nodos que tienen muy poco poder predictivo, lo que permitirá el prevenir el sobreajuste (Breiman et al., 1984).
- Uso de algoritmos de ensamblado como el Random Forest y el Gradient Boosting, que generan múltiples árboles de decisión y combinan sus predicciones para mejorar la precisión y la estabilidad del modelo.

El Random Forest construye una colección de árboles de decisión independientes mediante el uso de diferentes subconjuntos de los datos de entrenamiento y diferentes subconjuntos de características para cada árbol como se ilustra en la Figura 4. Las predicciones finales se realizan promediando las predicciones individuales de todos los árboles, lo que generalmente resulta en un modelo más fuerte y menos propenso al sobreajuste (Breiman, 2001).

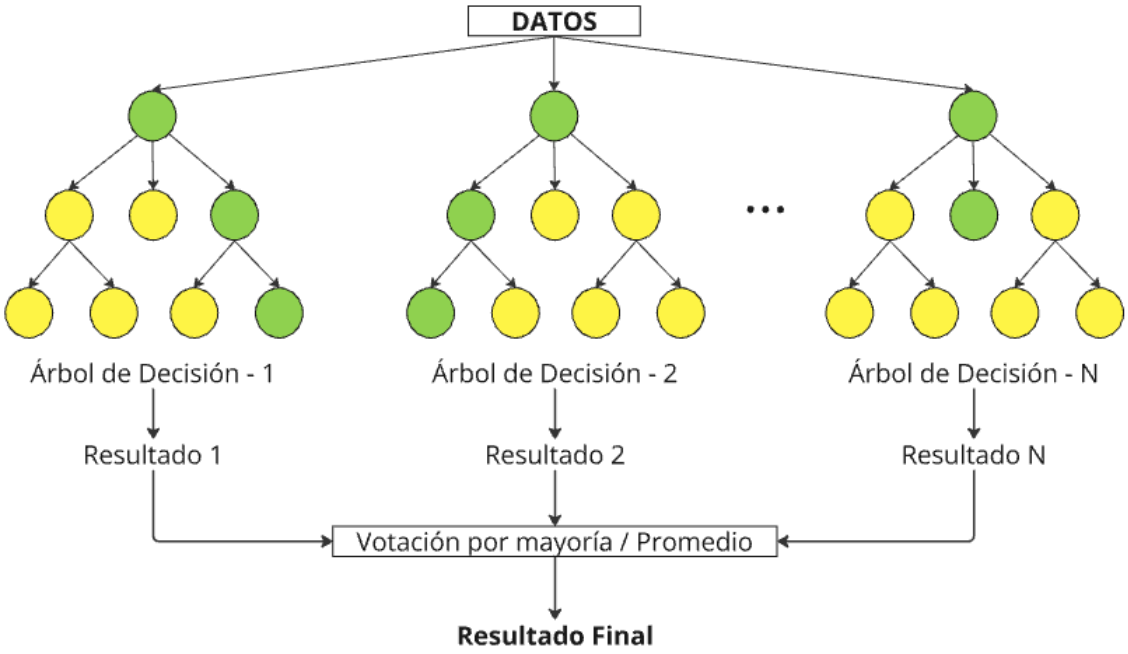


Ilustración 4 Representación de un random forest

Nota. El resultado final corresponde al promedio de las predicciones de los árboles existentes.

El Gradient Boosting es otra técnica que construye árboles de decisión de forma secuencial, donde cada árbol intenta corregir los errores cometidos por el árbol anterior, de esta forma se puede capturar relaciones más complejas y a menudo produce predicciones más precisas que un solo árbol de decisión tradicional (Friedman, 2001).

Aplicabilidad en la Proyección de Ventas en el Negocio Avícola

El sector avícola en Ecuador, al igual que en otros países de la región, está sujeto a una variedad de factores que influyen en las ventas. Estos factores incluyen la estacionalidad, las condiciones económicas, las políticas gubernamentales, procesos electorales, las fluctuaciones en los precios de los insumos, y las preferencias de los consumidores. Los árboles de decisión son particularmente adecuados para modelar este tipo de escenarios complejos, ya que pueden manejar múltiples variables predictoras y capturar interacciones no lineales entre ellas (Breiman et al., 1984).

Red Neuronal

Las redes neuronales son herramientas de la ciencia de datos, específicamente de la rama del machine learning, que imitan funciones de procesamiento del cerebro humano. (Amazon Web Services, n.d.)

Este tipo de modelo usa nodos alineados en capas dependiendo de que tan complejo sea los datos por manejar, cada nodo se interconecta con un umbral especifico y una ponderación que se ajusta según el procesamiento de la data, este proceso tiene similitud a la sinapsis humana la cual tratan de replicar (Chollet, 2018)

El objetivo de esta mímica es lograr un aprendizaje profundo o deep learning en donde los datos siguen un flujo de redes en capas. (Google Cloud, n.d.) Se le considera un método de aprendizaje no supervisado el cual crea un sistema adaptable que permite al modelo aprender de sus errores y dar un resultado más exacto.

Es importante mencionar que el aprendizaje profundo tiene la necesidad de gran cantidad de datos para que el aprendizaje modifique los umbrales y ponderaciones que intercomunican los nodos y disminuya el error resultante. Mas aún tenemos que tener precauciones como normalizar y analizar bien los datos para que el aprendizaje no sufra de problemas como overfitting o que la tendencia se pierda en una distribución sin un aprendizaje convergente.

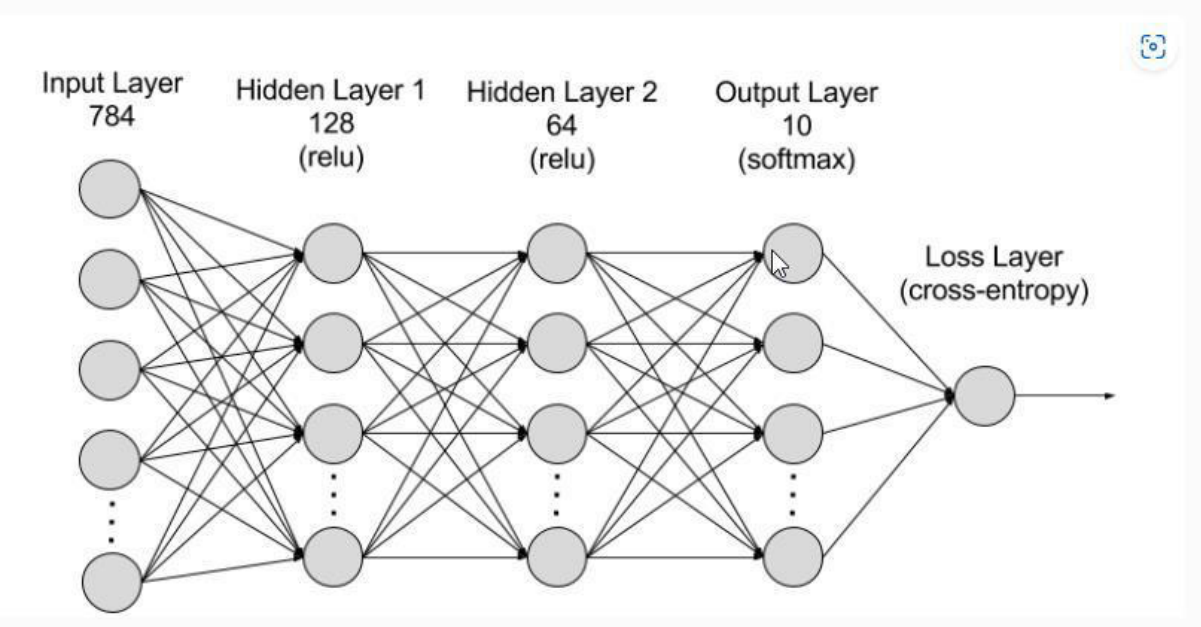


Ilustración 5: Red Neuronal(IBM, n.d.).

En la ilustración anterior vemos un ejemplo grafico de una red neuronal. Tenemos claro que cada circulo es una neurona y se organiza por capas. Las conexiones de las neuronas se crean con algoritmos de entrenamiento como la relu, sofmax o en nuestro caso tanh. Cabe mencionar que cada algoritmo tiene especificaciones como rangos para el entrenamiento, o el tipo de análisis como regresión o clasificación.

En nuestro proyecto el uso de las redes neuronales se sujetará a cinco capas para la predicción de las ventas tomando en cuenta la data de los clientes y su antigua facturación. Usaremos un algoritmo de entrenamiento para regresiones llamado tanh o tangente hiperbólica, tiene un rango centrado en 0 a 1 absoluto. Si la data es suficiente, se espera que el algoritmo disminuya el error cuadrático medio.

También cabe mencionar que por la complejidad de este algoritmo es muy necesario plantear bien la forma de alimentar la data en el modelo y considerar aprendizaje supervisado para disminuir la complejidad de los cálculos.

LIBRERIAS OPENSOURCE

SCIKIT-LEARN

Scikit-Learn es una biblioteca y api de código abierto que integra varios módulos de machine learning para uso en problemas de aprendizaje supervisado y no supervisado (Pedregosa et al., 2011). Esta biblioteca cuenta con varios de los algoritmos más usados a nivel educativo como comercial, y al ser opensource ha contribuido en gran cantidad al desarrollo del machine learning (Pedregosa et al., 2011). Sus algoritmos están diseñados para realizar regresiones, clasificaciones y agrupamientos de datos de manera simple, utilizando la menor cantidad posible de dependencias externas en Python (Scikit-learn developers, n.d.). Entre los algoritmos y funciones más populares tenemos regresión lineal, árbol de decisión, procesamientos gaussianos y clustering para clasificación. Actualmente en el área de machine learning, scikit-learn ha contribuido enormemente al desarrollo y perfección de modelos y nos atrevemos a afirmar que en cualquier proyecto de Python al respecto siempre se tiene modelos implementados con esta librería.

KERAS

Keras es una biblioteca de código abierto para la aplicación de modelos de aprendizaje profundo. Su principal fortaleza es el diseño enfocado en implementación y el entrenamiento de una manera fácil y limpia (TensorFlow,n.d). Además, se tiene una implementación modular y personalizable en bloques.

Keras es un api de alto nivel de tensoflow (TensorFlow,n.d) por lo que permite una integración completa con esta librería y sus diferentes funciones. Esto quiere decir implementación en Python y acceso a diferentes algoritmos de machine learning. Finalmente, al ser un proyecto open source cuenta con documentación abierta al público que facilita su personalización y manejo; en la actualidad, Keras es muy utilizado para implementar redes neuronales.

Análisis PEST

1. Políticos

Las normas y regulaciones son administradas y ejecutadas por el Ministerio de Agricultura y Ganadería y para ello tenemos a Agrocalidad que es la instancia encargada de la definición y ejecución de políticas de control y regulación para la protección y el mejoramiento de la sanidad animal, sanidad vegetal e inocuidad alimentaria. (Agencia de Regulación y Control Fito y Zoosanitario. (s.f.). *Misión y visión*. Agrocalidad.)

Regulaciones de productos alimenticios como el maíz que está relacionado directamente con alimento balanceado (AB) y que corresponde entre el 50% y el 60% del costo de fabricación del alimento balanceado.

Normativas y políticas de seguridad alimentaria definidas por Agrocalidad.

Ley Orgánica para la Protección y Defensa de los Derechos de los Animales en debate en la asamblea nacional. (Asamblea Nacional del Ecuador. (2024, 9 de agosto). *Comisión aprobó informe para primer debate del proyecto de ley*)

2. Económicos

En el Ecuador los cambios económicos siempre son un factor para tomar en cuenta. En los últimos años protestas sociales han causado una crisis económica grave.

Además, desde la publicación del Decreto Ejecutivo No. 198 se ha incrementado el iva al 15% (Servicio de Rentas Internas, n.d.) y pese a que la carne de pollo no grava el impuesto (Servicio de Rentas Internas, 2013), su venta si se ve afectada por el incremento en los insumos básicos para su mantenimiento.

La salida del subsidio a los hidrocarburos igual repercute en la cadena de suministros y logística necesaria para la industria. Como recomendación es un margen de error en las proyecciones con un aproximado del +-5% de incremento para cubrir la afectación de los cambios económicos en el proyecto (Quipu, s.f.).

A favor del Ecuador el uso del dólar (Banco Central del Ecuador, 2023) da una seguridad ya que evitamos deflación y devaluación de una moneda local. Lo cual nos da una certeza para inversiones y su rentabilidad económica.

Finalmente se debe considerar que el empleo bruto en Ecuador según el INEC se ubicó en un 63,6 (Instituto Nacional de Estadística y Censos, 2021) lo cual se ha mantenido en rangos iguales en los últimos dos años. Esta cifra nos puede dar idea de que tan bueno sería implementar operaciones y la forma de contratación ya que la facturación de servicios también es una opción.

3. Sociales

- Demografía, envejecimiento de la población: - En Ecuador apenas el 11% de la población tiene más de 60 años. Según las proyecciones del Instituto Nacional de Estadísticas y Censos (INEC,2024), de los 17,5 millones de las personas que viven en el país, hay 1,9 millones que superan esa edad. - El 38% de la población en Ecuador está compuesta por niños y adolescentes. El 31% son adultos jóvenes, es decir, personas entre 20 y 39 años. Y el otro 20% tiene entre 40 y 59 años. Eso significa que el 69% de la población es joven y que el 31% está por la edad de adulto mayor.

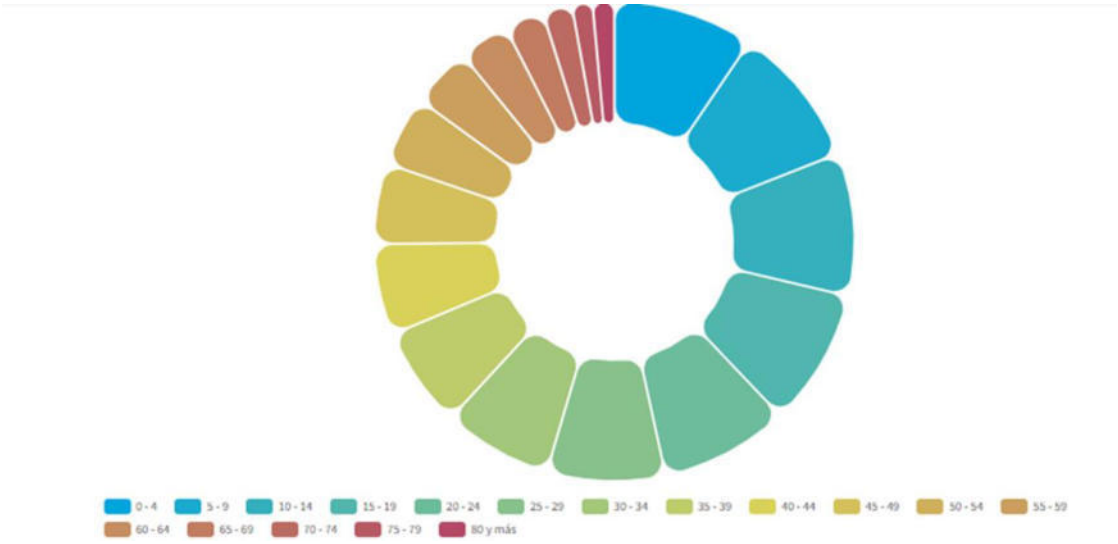


Ilustración 6 Envejecimiento de la población (INEC, 2024)

- Cultura y valores:

Consciencia sobre el bienestar animal: Existe una creciente preocupación por el bienestar animal, lo que puede llevar a una mayor demanda de productos avícolas provenientes de granjas con prácticas más éticas. (Humanitaria, 2024)

- Tendencias alimentarias: Dietas vegetarianas y veganas, así como alergias alimentarias, pueden afectar la demanda de productos avícolas. (Humanitaria, 2024)

Salud y conciencia ambiental:

Consumo de proteínas alternativas: El aumento de la popularidad de proteínas alternativas, como las proteínas vegetales, puede afectar la demanda de productos avícolas. (Humanitaria, 2024)

- Cambio climático: Los eventos climáticos extremos y la escasez de agua pueden afectar la producción avícola y, por lo tanto, la oferta y los precios de los productos. (Humanitaria, 2024)

4. Tecnológicos

Utilización de un ERP que permite una gestión de las actividades empresariales diarias. Herramientas como librerías y modelos de Python de tipo opensourse (Python Software Foundation, n.d.) para el procesamiento y limpieza de datos, tienen un alto grado de importancia y utilidad. Manejo e interpretación correcta de modelos de machine learning aplicado a nuestro conjunto de datos, provee alta disponibilidad de predicciones ajustables a las necesidades de nuestra empresa. Herramienta actualizada como Power BI que nos permite visualizar datos de forma interactiva y amigable al usuario final (Microsoft, n.d.).

Arquitectura

El proyecto tiene como finalidad el desarrollo, entrenamiento y testeo de modelos predictivos de machine learning en Python, se planea desarrollar un máximo de 3 modelos para analizar y comparar los resultados obtenidos definiendo el mejor modelo. Los modelos se los extraerá de investigaciones y publicaciones previas en áreas de finanzas y machine learning. La data se la extraerá del ERP que tiene la empresa, teniendo en cuenta un histórico desde el 2022, 2023 y el primer semestre del 2024.

La propuesta de la arquitectura para el proyecto esta esquematizada en la gráfica que se muestra a continuación, en donde se lo ordena por origen de datos, procesamiento y presentación.

Arquitectura del proyecto

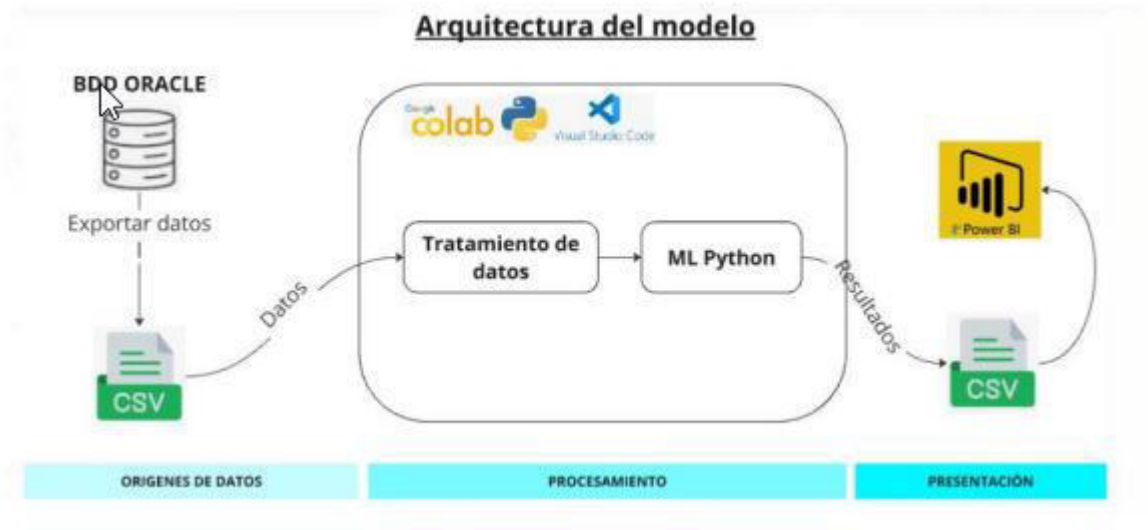


Ilustración 7: Arquitectura del proyecto

La arquitectura propuesta para el proyecto surge de un ERP cuya base de datos está sobre Oracle que nos proporcionará un archivo en formato .csv, el cual será procesado mediante un modelo de Machine Learning desarrollado con herramientas de Visual Studio Code, Google Colab, y el lenguaje de programación Python; finalmente se generará el nuevo archivo .csv con los resultados y que podría ser presentada en un aplicativo de visualización de datos.

CAPITULO 3

Metodología

Regulación y protección de datos

La información necesaria para el proyecto está en un ERP, esta se maneja por códigos para clientes, artículos, provincias, tipos de clientes, etc. La información sensible corresponde a campos de descripción y esa información no se utilizará para analizar proyección de ventas.

Un ejemplo de la información con códigos es la siguiente:

										ENE. 2023		FEB. 2023
Cód. Org. Comercial	Cód. Cliente	Cód. Artículo	Cód. Factura	Cód. F.Factura	F.Factura	Cód. Forma pago	Cód. Presentación	Cód. Provincia cliente	Cód. Tipo Cliente	Importe Neto	Cantidad Vendida	Importe Neto
01	00248	001454	Factura:	F.Factura:	27/03/2023	C30T	UND	2PQ1	C09	0.00	0.00	0.00
01	00248	001454	Factura:	F.Factura:	04/04/2023	C30T	UND	2PQ1	C09	0.00	0.00	0.00
01	00248	001454	Factura:	F.Factura:	02/05/2023	C30T	UND	2PQ1	C09	0.00	0.00	0.00

Ilustración 8: Exploración de datos ERP

Un extracto de la consulta que se hace al ERP y los tipos de datos que se obtienen se los detalla a continuación.

```
SELECT oc.codigo_org_comer "Cód. Organización Comercial",
       oc.nombre "Organización Comercial",
       cl.codigo_rapido "Cód. Cliente",
…
FROM provincias pv,
       tipos_cliente tc,
      formas_cobro_pago fp,
…
WHERE     vx.empresa = '02'
       AND cgc.codigo_grupo = '01'
      AND vx.centro_contable = cgc.codigo_centro
…
GROUP BY oc.codigo_org_comer,
       oc.nombre,
      cl.codigo_rapido,
...
```

Campo	Tipo de dato	Observación
Cód. Organización Comercial	VARCHAR2 (5 Char)	Código de la organización comercial que le corresponde a la factura
Cód. Cliente	VARCHAR2 (15 Char)	Código de cliente a quien se emite la factura

Cód. Artículo	VARCHAR2 (30 Char)	Código del artículo vendido
Cód. Familia	VARCHAR2 (15 Char)	Código de familia a la que corresponde el artículo
Familia	VARCHAR2 (40 Char)	Nombre de la familia a la que corresponde el artículo
Cód. Factura	VARCHAR2	Valor constante porque corresponde a un reporte de facturas
Factura	VARCHAR2	Es un valor que se arma con información de la factura: (vx.organizacion_comercial ‖ '/' ‖ vx.ejercicio_factura ‖ '/' ‖ vx.numero_serie_fra ‖ '/' ‖ vx.numero_factura)
Cód. F.Factura	VARCHAR2	Valor constante porque corresponde a f.factura
F.Factura	DATE	Fecha en la cual fue emitida la factura
Cód. Forma pago	VARCHAR2 (4 Char)	Código de la forma de pago
Forma pago	VARCHAR2 (50 Char)	Descripción de la forma de pago
Cód. Presentación	VARCHAR2 (5 Char)	Código de presentación del artículo facturado
Presentación	VARCHAR2 (40 Char)	Descripción de la presentación del artículo vendido.
Cód. Canton cliente	VARCHAR2 (4 Char)	Código del cantón donde reside el cliente
Canton cliente	VARCHAR2 (40 Char)	Nombre del cantón donde reside el cliente
Cód. Tipo Cliente	VARCHAR2 (3 Char)	Código del tipo de cliente asignado.
Tipo Cliente	VARCHAR2 (50 Char)	Nombre del tipo de cliente asignado
Cantidad	NUMBER (19,4)	Cantidad facturada al cliente
Importe	NUMBER (19,4)	Importe ó monto facturado al cliente

Al no existir datos sensibles para el proyecto, no habría inconvenientes por el tema de protección de datos.

Si por algún requerimiento puntual adicional e importante, es necesario incorporar datos sensibles, las personas que deben definir los campos sensibles a utilizar sería el Gerente de Mercadeo en conjunto con el administrador del ERP; la técnica que se podría utilizar sería mediante una función hash md5, o el enmascaramiento de la data “RUC 17xxxxxxxx001”.

Planteamiento Agile.

Vamos a aplicar el marco de trabajo Scrum, la naturaleza iterativa y flexible de Agile, junto con su énfasis en la comunicación y la colaboración, se adapta bien a un entorno de trabajo virtual y a las dinámicas de un equipo pequeño como el nuestro que es de 4 personas.

Aplicando el marco de Scrum, proponemos sprints de dos semanas con un planning el día sábado, reportes por mensaje para complementar las reuniones diarias, y un entregable el último sábado para terminar el sprint y comenzar la nueva fase. El proyecto se dividirá entre todos los integrantes y será supervisado por un scrum elegido por el grupo.

Herramientas:

1. Comunicación:
    a. WhatsApp: Para comunicación diaria y rápida
    b. Google Meet ó Zoom: Para reuniones de planificación y revisión.
2. Gestión de Proyectos:
    a. Jira: Para gestionar y seguir el progreso de las tareas y planificación del sprint.
3. Complementarias
    a. Office365: Para manejo de tablas de datos y elaboración del documento final en formato .docx y .pdf.

PREPARACION DE DATOS

Tras tener un acuerdo de confidencialidad, se revisa los campos en el módulo ERP para exportarlos y alimentar los modelos de ML. En esta actividad se revisó la facturación bruta de los años 2022, 2023 y el primer semestre del 2024. Debido a una restructuración de la empresa no podemos contar con la facturación de años anteriores pero el proyecto queda abierto para una retroalimentación con datos futuros.

Al ser un ERP confidencial no podemos exponer las consultas para exportar los datos ni el proceso en sí, pero una vez con los archivos exportados en formato xls se debe juntar limpiar y procesar para crear un solo dataframe. Este proceso se detalla en el desarrollo.

Construcción y entrenamiento de modelos

TESTEO DE LOS MODELOS

KPI´s

Los indicadores claves de desempeño serán:

El modelo ML utilizado debe tener un porcentaje de acierto mayor al 80% con datos de entrenamiento desde el 2022 hasta el 2023, y datos del primer semestre de 2024 como muestra de revisión.

- Obtención de un modelo a partir de conjunto de datos con un error cuadrático medio entre 0 y 1 para el análisis de las ventas a lo largo del año 2024
- Detección de variables en los tipos de cliente para apuntalar las ventas y tener un plan de acción para mejorar las ventas del 2024 con un porcentaje de al menos el 3 % o superior a la inflación del año, teniendo en cuanta un margen de error del +-5%
- Seguimiento de evolución de ventas durante las reuniones trimestrales para el área de ventas y gerente de negocio, mediante reportería mensual entregada del año en curso para la reducción de tiempo en un 20% dedicado en la toma de decisiones.

Desarrollo

Extracción, tratamiento y carga de datos

Para esta sección se procedió a tomar la información exportada del sistema ERP. Editándola con Python se procedió a cargar, limpiar y trasformar el dataframe a un modelo apto para alimentar modelos de machine learning.
El primer paso que se dio en la creación del modelo fue la carga de los extractos en formato

xls a un dataframe de pandas. Cabe mencionar que desde ya se creó un dataframe de entrenamiento, año 2022 y 2023, y uno de control primer parcial del año 2024.

```
df_2022 = pd.read_excel('/content/vtas_facturacion_2022.xls')
df_2023 = pd.read_excel('/content/vtas_facturacion_2023.xls')
df_2024 = pd.read_excel('/content/vtas_facturacion_S1_2024.xls')

frames = [df_2022, df_2023]
df_join = pd.concat(frames)
print(df_join.head())

num_filas = len(df_join)
print(f"Número total de filas 1: {num_filas}")
```

Ilustración 9: Python importe de datos

El segundo paso fue la limpieza de datos. En esta sección tenemos varias trasformaciones del dataframe para tener data apta para alimentar los modelos de machine learning. Se procede a dar formato a las columnas del dataframe para poder llamarlas de forma más didáctica y con sentido.

```
[ ] df_join.columns = df_join.iloc[3].values.tolist()
    print(df_join.columns)

    df_2024.columns = df_2024.iloc[3].values.tolist()
    print(df_2024.columns)
    # df = df.iloc[4:]
```

Ilustración 10: Python formato de columnas

A continuación, eliminamos toda la meta data que no corresponda al conjunto de entramiento, en estos valores de metadata incluimos casillas vacías, encabezados y textos varios que son parte del formato de exportación de los libros xls del ERP.

```
print(len(df_join))
df_join = df_join[pd.to_numeric(df_join['Cód. Organización Comercial'], errors='coerce').notnull()]
print(len(df_join))


print(len(df_2024))
df_2024 = df_2024[pd.to_numeric(df_2024['Cód. Organización Comercial'], errors='coerce').notnull()]
print(len(df_2024))
```

Ilustración 11: Python eliminación de filas vacías

Ahora eliminamos las columnas con data irrelevante y que no aportaría a nuestro modelo como son las columnas de encabezados, títulos, valores únicos, y códigos sin valores para mapeo.

```
columns_to_drop = ['Cód. Cliente', 'Cód. Familia', 'Cód. F.Factura', 'Cód. Presentación',
                   'Cód. Canton cliente', 'Presentación', 'Cód. Presentación', 'Factura',
                   'Forma pago','Cód. Factura','Cód. Tipo Cliente']
df = df_join.drop(columns=columns_to_drop)

print(df.head())
print(len(df_join))
print(df.columns)

df_control = df_2024.drop(columns=columns_to_drop)
print(df_control.head())
print(len(df_control))
print(df_control.columns)
```

Ilustración 12: Python drop de columnas no relevantes

Trasformamos las columnas a un formato apropiado para que no sean objetos sin formato. A los numéricos se le da int64 o float, y a las fechas formato date.

```
print(df.info())
df['Importe'] = pd.to_numeric(df['Importe'])
df['F.Factura'] = pd.to_datetime(df['F.Factura'])
df['Cantidad'] = pd.to_numeric(df['Cantidad'])
print(df.info())
```

Ilustración 13: Python formato de datos

Finalmente eliminamos las filas correspondientes a proformas marcadas como valores negativos en el dataframe, aparte redondeamos los decimales de las columnas float para estandarizar a valores de dos cifras decimales.

```
df = df[df['Importe'] >= 0]
df = df.round({'Importe': 2})
print(df.info())

df_control = df_control[df_control['Importe'] >= 0]
print(df_control.info())
```

Ilustración 14: Python eliminación de datos negativos

En todos nuestros códigos se puede apreciar impresiones para visualizar los cambios de forma, tamaño y formato del dataframe. Estos son añadidos a los resultados en los anexos.

Análisis exploratorio de los datos

Para el análisis se generan varios gráficos que permiten tener un entendimiento de la información y con ello una mejor visualización de las variables del negocio y como ellas interactúan.

En nuestro proyecto y análisis, el factor clave es el valor de ventas el cual en nuestro caso seria los valores a proyectar. Procedemos a graficar los diferentes campos del dataframe limpio para comprender la estructura de la data

Los principales campos para graficar son las ventas con respecto al tiempo. Es importante tener claro como es el trayecto de los datos y como avanza en el trascurso de 2022 y 2023 para marcar tendencias, y tener claro cómo nos va a quedar la proyección.

Las ventas con respecto al tiempo

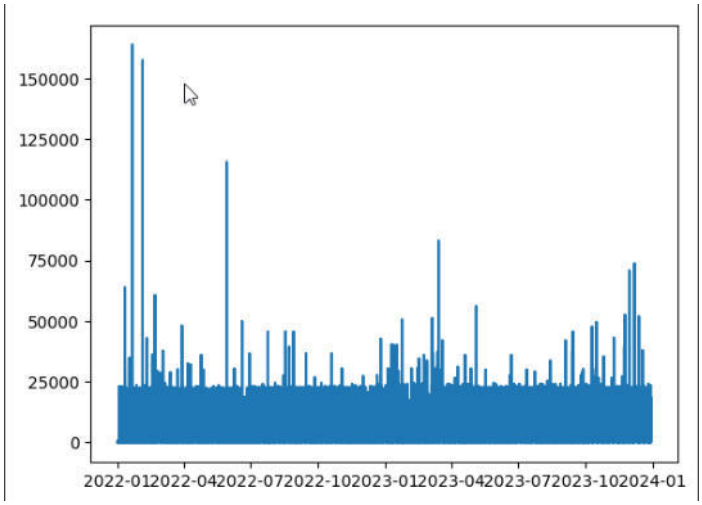


Ilustración 15: Ventas vs Tiempo

En las ventas con respecto al tiempo tenemos una clara tendencia de 25000 como media en ventas registradas, pero nos es imposible analizar y sacar conclusiones con tantos datos, por lo que procedemos a graficar las ventas agrupadas por mes.

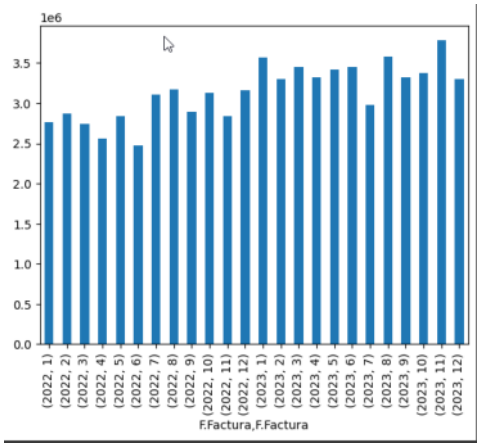


Ilustración 16: Ventas vs tiempo agrupado

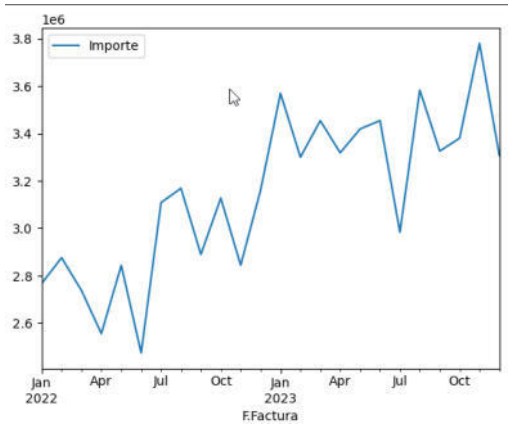


Ilustración 17: Ventas vs tiempo agrupado y acumulado

En estos grafico tenemos más control sobre la varianza y tendencia de las ventas a lo largo de 2022 y 2023. Teniendo un incremento porcentual visible entre datos de 2022 y 2023.

Así mismo siguiendo la tendencia en la proyección de 2024 tendremos alzas porcentuales en ventas que deberían cubrir por lo menos la inflación del país

Otro dato interesante que tenemos son las ventas acumuladas para ver la tendencia de crecimiento

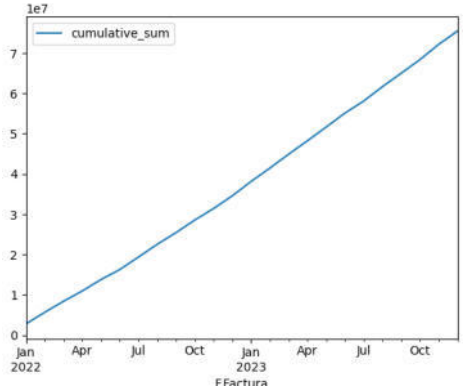


Ilustración 18: Ventas vs tiempo acumulado

Como podemos ver aquí lo importante seria la pendiente de la recta que marcaría el incremento acumulado de las ventas teniendo un claro incremento en el 2023 lo cual es directamente relacional al incremento acumulado de las ventas.

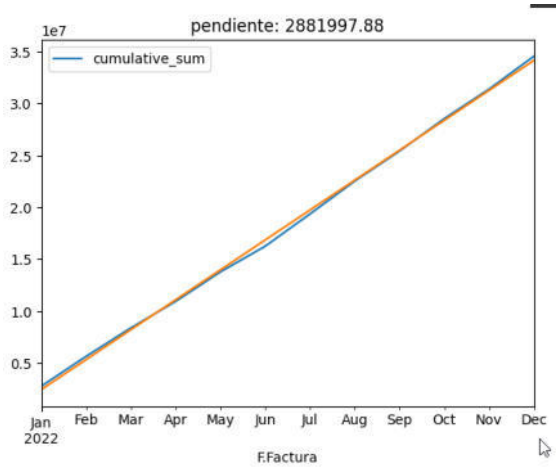


Ilustración 19 Ventas vs tiempo 2022 tendencia

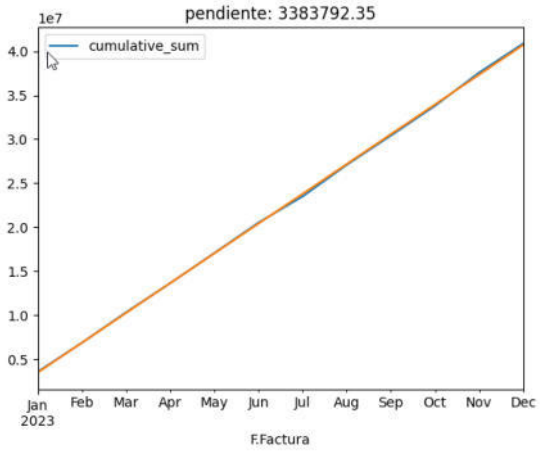


Ilustración 20 Ventas vs tiempo 2023 tendencia

Organización comercial y los importes generados.

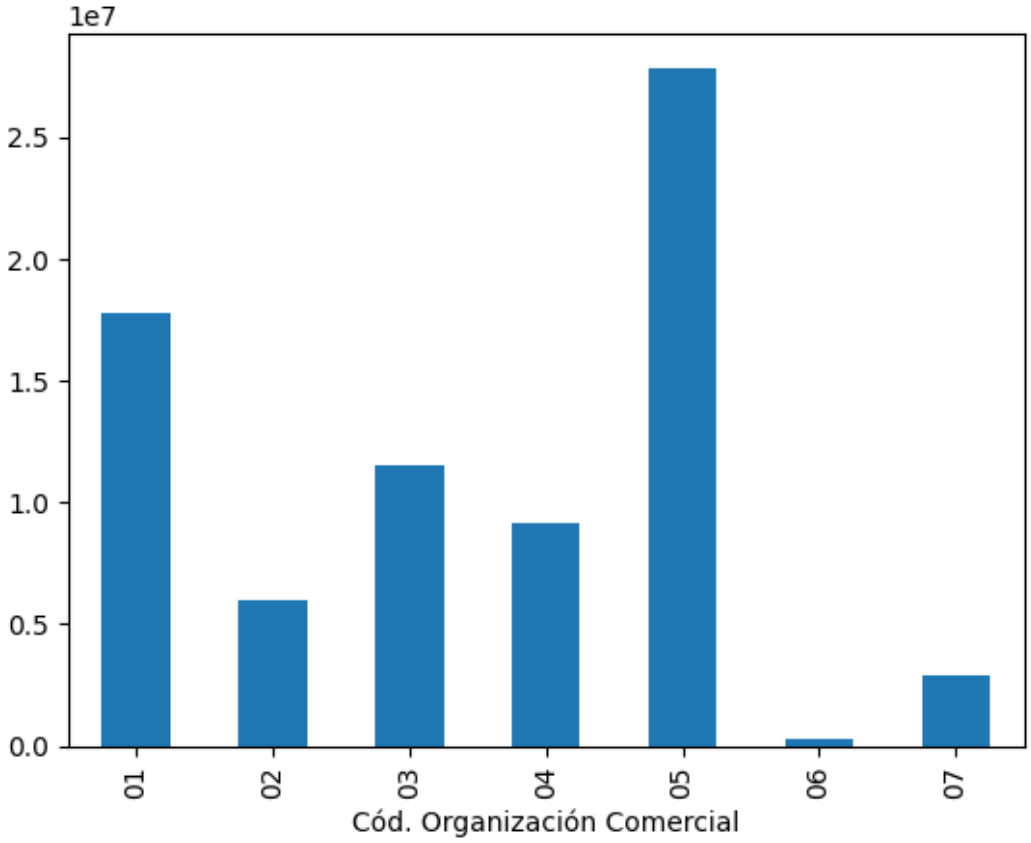


Ilustración 21: Organización comercial vs importe

El volumen de venta del Negocio Avícola está marcado por una participación preponderante de las siguientes organizaciones comerciales:

- 05 que corresponde a la incubadora generando al pollito de cría con una participación aproximada dentro del negocio del 37%
- 01 que es el alimento balanceado tanto para aves, cerdos y ganado con un 23%
- 03 Huevo Fértil que corresponde al producto para ser vendido a otras incubadoras y como materia prima para la organización comercial 05, este representa el 15%.

Las tres organizaciones comerciales nombradas en conjunto representan el 75% de las ventas totales y corresponden al core del negocio avícola

La organización que menos aporta en el volumen total corresponde a la 06 Abono, en donde se genera el abono en sacos para la venta en base a los desechos avícolas recolectados de los diferentes galpones.

Código de artículo vs importes

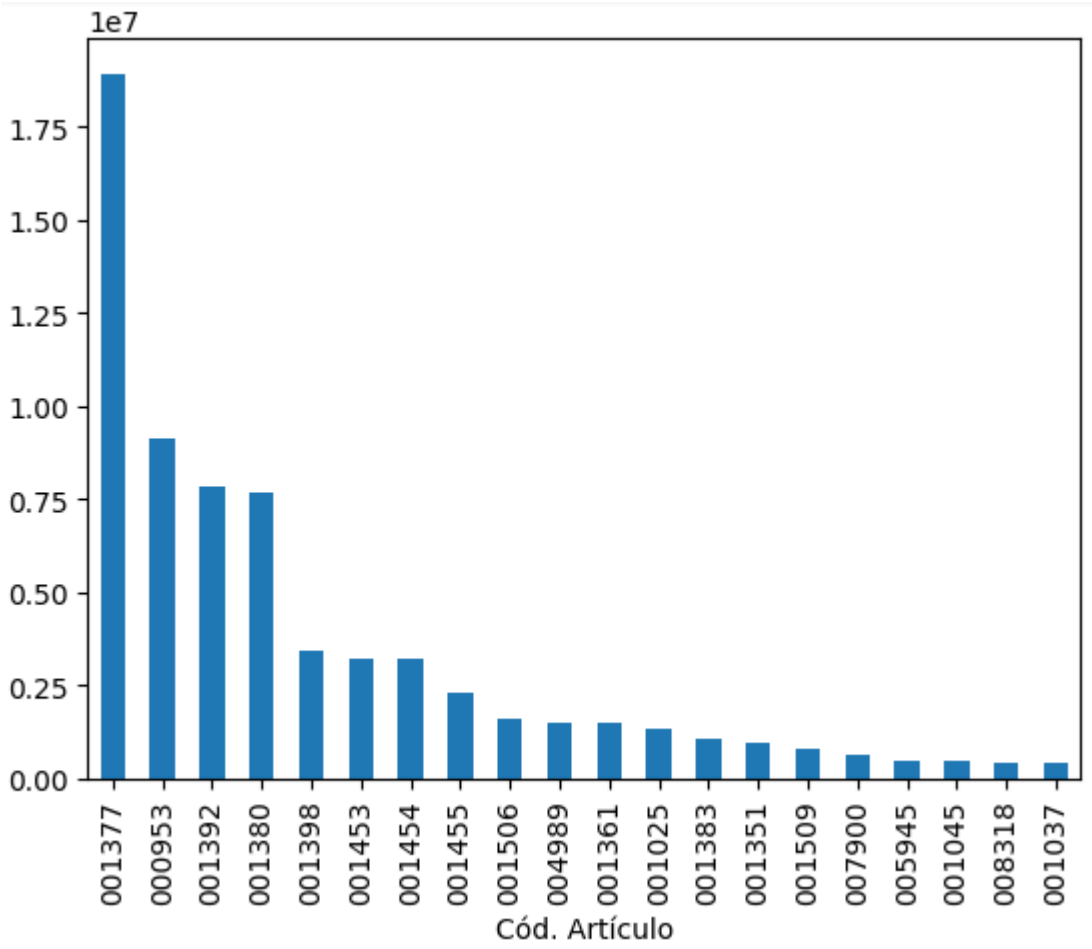


Ilustración 22: Artículos vs importe

Para una mejor claridad, se muestran los 20 artículos con mayor importe, este grupo representa el 88 % de la venta analizada.

El artículo 001377 corresponde a un pollito de cría de calidad y que es parte de la organización comercial 05, este artículo tiene una participación importante de venta dentro del negocio.

Todos los artículos de la gráfica son parte de las organizaciones comerciales que tienen la mayor participación y que pertenecen al core del negocio avícola.

Existe un caso especial, que corresponde al artículo 000953 de la organización comercial 04 Materia Prima, que ha repuntado en estos dos años y que se podría manejar adecuadamente para que pueda formar parte de un crecimiento temporal.

Tipo Cliente vs Importe

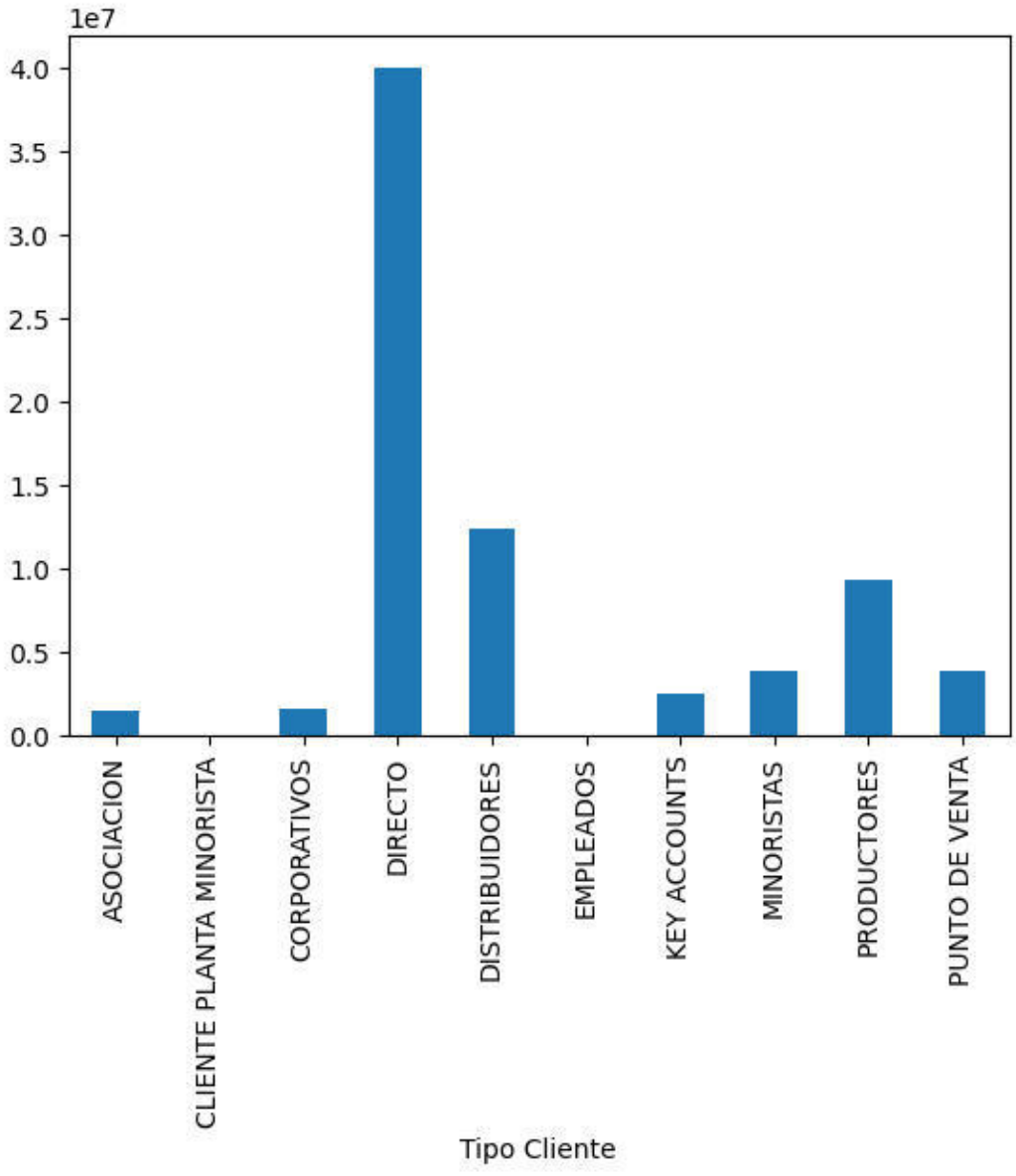


Ilustración 23: Clientes vs importe

El gráfico evidencia que los 3 tipos de clientes con mayor importe son “Directo”, “Distribuidores” y “Productores” mientras que los tipos de clientes con menor importe son “Cliente Planta Minorista” y “Empleados”. Los valores del eje y están expresados en millones de dólares.

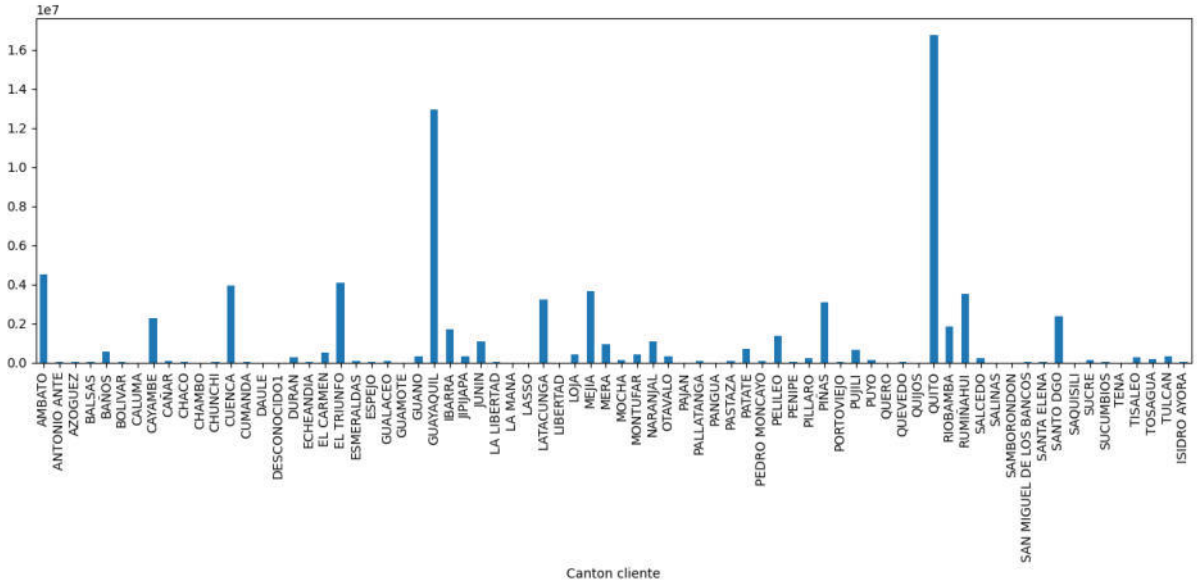


Ilustración 24: Cantones de venta vs importe

En la gráfica se evidencia que los cantones con mayor importe son Quito y Guayaquil, los cantones con proyección son Ambato, Cuenca y el Triunfo.

Análisis y modelización.

Regresión Lineal

Se importan las siguientes librerías

```
#Importación de librerías
from sklearn import metrics
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression
```

Ilustración 25: Librería regresión lineal

Se definen los conjuntos de entrenamiento y prueba.

```
# Definición características (X) y variable objetivo (y) con df (entrenamiento)
X_train = df[['Cód. Organización Comercial', 'Cód. Artículo', 'Familia', 'F.Factura', 'Cód. Forma pago', 'Canton cliente', 'Tipo Cliente', 'Cantidad']]
y_train = df['Importe']

# Definición características (X) y variable objetivo (y) con df_2024 (test)
X_test = df_control[['Cód. Organización Comercial', 'Cód. Artículo', 'Familia', 'F.Factura', 'Cód. Forma pago', 'Canton cliente', 'Tipo Cliente', 'Cantidad']]
y_test = df_control['Importe']

# Concatenar para aplicar dummy
X_combined = pd.concat([X_train, X_test])
X_combined = pd.get_dummies(X_combined)

# Separa los dataset después de aplicar dummy
X_train = X_combined.iloc[:len(X_train)]
X_test = X_combined.iloc[len(X_train):]

# Convertir datetime a represetntación numérica
for col in X_train.columns:
  if X_train[col].dtype == 'datetime64[ns]':
    X_train[col] = X_train[col].astype('int64')
    X_test[col] = X_test[col].astype('int64')

#Define la columna de agrupamiento
A_column_test = df_control['Tipo Cliente']
```

Ilustración 26: Definición de conjunto a entrenar regresión lineal

Se entrena el modelo con la data del 2022 y 2023 y se realizan predicciones con la data del 2024.

Se evalúan 4 métricas: MSE, RMSE, MAE, R-CUADRADO

```
#REGRESIÓN LINEAL

# Crear el modelo de regresión lineal
regresion_lineal = LinearRegression()

# Entrenar el modelo
regresion_lineal.fit(X_train, y_train)

# Predecir con el conjunto de prueba
y_pred = regresion_lineal.predict(X_test)

print("REGRESIÓN LINEAL")

# Evaluar el modelo utilizando el conjunto de prueba (Error cuadrático medio)
mse = mean_squared_error(y_test, y_pred)
print(f"Error cuadrático medio: {mse:.2f}")

# Calcula Raíz del Error Cuadrático Medio (RMSE)
rmse = np.sqrt(mse)
print(f'Raíz del Error Cuadrático Medio (RMSE): {rmse:.2f}')

# Calcula Error Absoluto Medio (MAE)
mae = mean_absolute_error(y_test, y_pred)
print(f'Error Absoluto Medio (MAE): {mae:.2f}')

# Calcula Coeficiente de Determinación (R²)
r2 = r2_score(y_test, y_pred)
print(f"R-cuadrado: {r2:.2f}")

# Visualizar los resultados
# Use X_test for features and y_test for target values
plt.scatter(y_test, y_pred, color='blue', label='Datos originales') # Changed X to y_test and y to y_pred
plt.plot(y_test, y_pred, color='red', label='Línea de regresión') # Changed X to y_test
plt.xlabel('y_test') # Changed X to y_test
plt.ylabel('y_pred') # Changed y to y_pred
plt.legend()
plt.show()
```

Ilustración 27: Predicción y análisis regresión lineal.

Agrupamos los valores por “Tipo de Cliente” y realizamos la tabla comparativa entre las predicciones y los valores reales.

```
# Desactivar la notación científica para los floats
pd.set_option('display.float_format', '{:.2f}'.format)

# Convertir las predicciones y la columna 'Tipo Cliente' en un DataFrame
df_pred = pd.DataFrame({'Tipo Cliente': A_column_test, 'Predicciones': y_pred})

# Agrupar el dataset de predicción
df_pred_group = df_pred.groupby('Tipo Cliente')['Predicciones'].sum().reset_index()

# Agrupar el dataset de prueba
df_control_group = df_control.groupby('Tipo Cliente')['Importe'].sum().reset_index()

# Crear el dataframe de comparación
df_comparacion = pd.DataFrame({
    'TIPO CLIENTE': df_pred_group['Tipo Cliente'],
    'PREDICCIÓN': df_pred_group['Predicciones'],
    'VALOR REAL': df_control_group['Importe'],
    'DIFERENCIA': df_pred_group['Predicciones'] - df_control_group['Importe']
})

print(df_comparacion)

# Sumar los valores PREDICCIÓN
suma_pred = df_comparacion['PREDICCIÓN'].sum()

print("PREDICCIÓN:", suma_pred)

# Sumar los valores REALES
suma_pred = df_comparacion['VALOR REAL'].sum()

print("VALOR REAL:", suma_pred)

# Sumar los valores DIFERENCIA
suma_pred = df_comparacion['DIFERENCIA'].sum()

print("DIFERENCIA:", suma_pred)
```

Ilustración 28: Resultados regresión lineal

Árboles De Decisión

Se han implementado dos modelos de árboles de decisión, utilizando DECISIONTREEREGRESSOR y RANDOMFORESTREGRESSOR. Ambos modelos se describen a continuación:

Para ambos modelos se realiza lo siguiente:

Se importan las siguientes librerías que serán utilizados en los dos modelos:

```
#Importación de librerías
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn import metrics
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error
```

Ilustración 29: Librerías árbol de decisión

Se definen los conjuntos de entrenamiento y prueba.

```
# Definición características (X) y variable objetivo (y) con df (entrenamiento)
X_train = df[['Cód. Organización Comercial', 'Cód. Artículo', 'Familia', 'F.Factura', 'Cód. Forma pago', 'Canton cliente', 'Tipo Cliente', 'Cantidad']]
y_train = df['Importe']

# Definición características (X) y variable objetivo (y) con df_control (test)
X_test = df_control[['Cód. Organización Comercial', 'Cód. Artículo', 'Familia', 'F.Factura', 'Cód. Forma pago', 'Canton cliente', 'Tipo Cliente', 'Cantidad']]
y_test = df_control['Importe']

# Concatenar para aplicar dummy
X_combined = pd.concat([X_train, X_test])
X_combined = pd.get_dummies(X_combined)

# Separa los dataset después de aplicar dummy
X_train = X_combined.iloc[:len(X_train)]
X_test = X_combined.iloc[len(X_train):]

# Convertir datetime a represetntación numérica
for col in X_train.columns:
  if X_train[col].dtype == 'datetime64[ns]':
    X_train[col] = X_train[col].astype('int64')
    X_test[col] = X_test[col].astype('int64')

#Define la columna de agrupamiento
A_column_test = df_control['Tipo Cliente']
```

Ilustración 30: Definición de conjuntos árbol de decisión

Decision Tree Regressor

Se define el modelo con los siguientes parámetros:

- max_depth=50, la profundidad máxima será de 50.
- min_samples_split=5, número mínimo de muestras necesarias para dividir un nodo.
- min_samples_leaf=4, número mínimo de muestras necesarias para que un nodo hoja se considere válido.
- random_state=42, semilla fija del generador de números aleatorios.

Se entrena el modelo con la data del 2022 y 2023 y se realizan predicciones con la data del 2024.

Se evalúan 4 métricas: MSE, RMSE, MAE, R-CUADRADO

```
#DECISIONTREEREGRESSOR

# Crear el modelo de árbol de decisión
regressortree = DecisionTreeRegressor(max_depth=50, min_samples_split=5, min_samples_leaf=4, random_state=42)

# Entrenar el modelo con los datos de entrenamiento
regressortree.fit(X_train, y_train)

# Hacer predicciones con el conjunto de prueba
y_pred = regressortree.predict(X_test)

# Evaluar el modelo utilizando el conjunto de prueba (Error cuadrático medio)
mse = mean_squared_error(y_test, y_pred)
print(f"Error cuadrático medio: {mse:.2f}")

# Calcula Raíz del Error Cuadrático Medio (RMSE)
rmse = np.sqrt(mse)
print(f'Raíz del Error Cuadrático Medio (RMSE): {rmse:.2f}')

# Calcula Error Absoluto Medio (MAE)
mae = mean_absolute_error(y_test, y_pred)
print(f'Error Absoluto Medio (MAE): {mae:.2f}')

# Calcula Coeficiente de Determinación (R²)
r2 = r2_score(y_test, y_pred)
print(f"R-cuadrado: {r2:.2f}")
```

Ilustración 31: Predicción y análisis árbol de regresión

Agrupamos los valores por “Tipo de Cliente” y realizamos la tabla comparativa entre las predicciones y los valores reales.

```python
# Desactivar la notación científica para los floats
pd.set_option('display.float_format', '{:.2f}'.format)

# Convertir las predicciones y la columna 'Tipo Cliente' en un DataFrame
df_pred = pd.DataFrame({'Tipo Cliente': A_column_test, 'Predicciones': y_pred})

# Agrupar el dataset de predicción
df_pred_group = df_pred.groupby('Tipo Cliente')['Predicciones'].sum().reset_index()

# Agrupar el dataset de prueba
df_control_group = df_control.groupby('Tipo Cliente')['Importe'].sum().reset_index()

# Crear el dataframe de comparación
df_comparacion = pd.DataFrame({
    'TIPO CLIENTE': df_pred_group['Tipo Cliente'],
    'PREDICCIÓN': df_pred_group['Predicciones'],
    'VALOR REAL': df_control_group['Importe'],
    'DIFERENCIA': df_pred_group['Predicciones'] - df_control_group['Importe']
})

print(df_comparacion)

# Sumar los valores PREDICCIÓN
suma_pred = df_comparacion['PREDICCIÓN'].sum()

print("PREDICCIÓN:", suma_pred)

# Sumar los valores REALES
suma_pred = df_comparacion['VALOR REAL'].sum()

print("VALOR REAL:", suma_pred)

# Sumar los valores DIFERENCIA
suma_pred = df_comparacion['DIFERENCIA'].sum()

print("DIFERENCIA:", suma_pred)
```

Ilustración 32: Resultados árbol de regresión

Random Forest Regressor

Se define el modelo con los siguientes parámetros:

- n_estimators=100, número de árboles en el bosque
- random_state=42, semilla fija del generador de números aleatorios.

Se entrena el modelo con la data del 2022 y 2023 y se realizan predicciones con la data del 2024.

Se evalúan 4 métricas: MSE, RMSE, MAE, R-CUADRADO

```
#RANDOMFORESTREGRESSOR

# Crear el modelo de Random Forest para regresión
regressorrandom = RandomForestRegressor(n_estimators=100, random_state=42)

# Entrenar el modelo
regressorrandom.fit(X_train, y_train)

# Hacer predicciones con el conjunto de prueba
y_pred = regressorrandom.predict(X_test)

# Evaluar el modelo usando el error cuadrático medio (MSE)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
print(f"Error cuadrático medio (MSE): {mse:.2f}")
print(f"Raíz del error cuadrático medio (RMSE): {rmse:.2f}")

# Calcula Error Absoluto Medio (MAE)
mae = mean_absolute_error(y_test, y_pred)
print(f'Error Absoluto Medio (MAE): {mae:.2f}')

# Calcula Coeficiente de Determinación (R²)
r2 = r2_score(y_test, y_pred)
print(f"R-cuadrado: {r2:.2f}")
```

Ilustración 33: Predicción y análisis random forest

Agrupamos los valores por “Tipo de Cliente” y realizamos la tabla comparativa entre las predicciones y los valores reales.

```
# Desactivar la notación científica para los floats
pd.set_option('display.float_format', '{:.2f}'.format)

# Convertir las predicciones y la columna 'Tipo Cliente' en un DataFrame
df_pred = pd.DataFrame({'Tipo Cliente': A_column_test, 'Predicciones': y_pred})

# Agrupar el dataset de predicción
df_pred_group = df_pred.groupby('Tipo Cliente')['Predicciones'].sum().reset_index()

# Agrupar el dataset de prueba
df_control_group = df_control.groupby('Tipo Cliente')['Importe'].sum().reset_index()

#print(df_2024_group)

# Crear el dataframe de comparación
df_comparacion = pd.DataFrame({
    'TIPO CLIENTE': df_pred_group['Tipo Cliente'],
    'PREDICCIÓN': df_pred_group['Predicciones'],
    'VALOR REAL': df_control_group['Importe'],
    'DIFERENCIA': df_pred_group['Predicciones'] - df_control_group['Importe']
})

print(df_comparacion)

# Sumar los valores PREDICCIÓN
suma_pred = df_comparacion['PREDICCIÓN'].sum()

print("PREDICCIÓN:", suma_pred)

# Sumar los valores REALES
suma_pred = df_comparacion['VALOR REAL'].sum()

print("VALOR REAL:", suma_pred)

# Sumar los valores DIFERENCIA
suma_pred = df_comparacion['DIFERENCIA'].sum()

print("DIFERENCIA:", suma_pred)
```

Ilustración 34 Resultados random forest

Redes Neuronales

Red neuronal 1

Se han implementado dos modelos de redes neuronales. Ambos modelos se describen a continuación:

Para ambos modelos se realiza lo siguiente:

Se importan las siguientes librerías que serán utilizados en los dos modelos:

```
#Importación de librerías
!pip install tensorflow
import tensorflow as tf
from tensorflow.keras import layers, models
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow.keras import layers, models, regularizers
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
```

Ilustración 35: Librerías redes neuronales 1

Se definen los conjuntos de entrenamiento y prueba.

```
# Definición características (X) y variable objetivo (y) con df (entrenamiento)
X_train = df[['Cód. Organización Comercial', 'Cód. Artículo', 'Familia', 'F.Factura', 'Cód. Forma pago', 'Canton cliente', 'Tipo Cliente', 'Cantidad']]
y_train = df['Importe']

# Definición características (X) y variable objetivo (y) con df_2024 (test)
X_test = df_control[['Cód. Organización Comercial', 'Cód. Artículo', 'Familia', 'F.Factura', 'Cód. Forma pago', 'Canton cliente', 'Tipo Cliente', 'Cantidad']]
y_test = df_control['Importe']

# Concatenar para aplicar dummy
X_combined = pd.concat([X_train, X_test])
X_combined = pd.get_dummies(X_combined)

# Separa los dataset después de aplicar dummy
X_train = X_combined.iloc[:len(X_train)]
X_test = X_combined.iloc[len(X_train):]

# Convertir datetime a representación numérica
for col in X_train.columns:
  if X_train[col].dtype == 'datetime64[ns]':
    X_train[col] = X_train[col].astype('int64')
    X_test[col] = X_test[col].astype('int64')

#Define la columna de agrupamiento
A_column_test = df_control['Tipo Cliente']

# Normalizar los datos
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Ilustración 36: Definición de conjuntos red neuronal 1.

Rede Neuronal 1.1

Se define el modelo con los siguientes parámetros:

- layers.Dense(128): capa completamente conectada con 128 nodos.
- activation='relu': es la función de activación ReLU (Rectified Linear Unit) que permite el aprendizaje de patrones complejos.
- input_shape=(X_train.shape[1],): especifica el número de columnas de los datos de entrada.
- layers.Dropout(0.3): El 30% de los nodos se desactivan aleatoriamente en cada paso de entrenamiento.
- Se añaden dos capas ocultas con 64 y 32 nodos.
- layers.Dense(1): la capa de salida tiene solo un nodo.

```
#RED NEURONAL PROFUNDA DE REGRESIÓN

# Define el modelo de red neuronal profunda
neuralnetwork = models.Sequential()

# Capa de entrada densa con 128 nodos
neuralnetwork.add(layers.Dense(128, activation='relu', input_shape=(X_train.shape[1],)))
neuralnetwork.add(layers.Dropout(0.3))  # Dropout para evitar sobreajuste

# Varias capas ocultas
neuralnetwork.add(layers.Dense(64, activation='relu'))
neuralnetwork.add(layers.Dropout(0.3))

neuralnetwork.add(layers.Dense(32, activation='relu'))
neuralnetwork.add(layers.Dropout(0.3))

# Capa de salida para regresión (sin activación, para predecir un valor continuo)
neuralnetwork.add(layers.Dense(1, activation='linear'))

# Compilar el modelo
neuralnetwork.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])

# Entrenar el modelo
history = neuralnetwork.fit(X_train, y_train, epochs=50, batch_size=32, validation_split=0.2)

# Hacer predicciones con el conjunto de prueba
y_pred = neuralnetwork.predict(X_test)
```

Ilustración 37: Predicción red neuronal 1.1

Se evalúan 4 métricas: MSE, RMSE, MAE, R-CUADRADO

```
# Evaluar el modelo 1.1
# Convert y_test to a NumPy array with float32 data type
#test_loss, test_mae = neuralnetwork.evaluate(X_test, y_test.astype('float32'))

#print(f"Error absoluto medio en test: {test_mae:.2f}")

print("RED NEURONAL 1.1")

# Evaluar el modelo utilizando el conjunto de prueba (Error cuadrático medio)
mse = mean_squared_error(y_test, y_pred)
print(f"Error cuadrático medio: {mse:.2f}")

# Calcula Raíz del Error Cuadrático Medio (RMSE)
rmse = np.sqrt(mse)
print(f'Raíz del Error Cuadrático Medio (RMSE): {rmse:.2f}')

# Calcula Error Absoluto Medio (MAE)
mae = mean_absolute_error(y_test, y_pred)
print(f'Error Absoluto Medio (MAE): {mae:.2f}')

# Calcula Coeficiente de Determinación (R²)
r2 = r2_score(y_test, y_pred)
print(f"R-cuadrado: {r2:.2f}")
```

Ilustración 38: Análisis red neuronal 1.1

Agrupamos los valores por “Tipo de Cliente” y realizamos la tabla comparativa entre las predicciones y los valores reales.

```
# Desactivar la notación científica para los floats
pd.set_option('display.float_format', '{:.2f}'.format)

# Convertir las predicciones y la columna 'Tipo Cliente' en un DataFrame
df_pred = pd.DataFrame({'Tipo Cliente': A_column_test, 'Predicciones': y_pred})

# Agrupar el dataset de predicción
df_pred_group = df_pred.groupby('Tipo Cliente')['Predicciones'].sum().reset_index()

# Agrupar el dataset de prueba
df_control_group = df_control.groupby('Tipo Cliente')['Importe'].sum().reset_index()

#print(df_2024_group)

# Crear el dataframe de comparación
df_comparacion = pd.DataFrame({
    'TIPO CLIENTE': df_pred_group['Tipo Cliente'],
    'PREDICCIÓN': df_pred_group['Predicciones'],
    'VALOR REAL': df_control_group['Importe'],
    'DIFERENCIA': df_pred_group['Predicciones'] - df_control_group['Importe']
})

print(df_comparacion)

# Sumar los valores PREDICCIÓN
suma_pred = df_comparacion['PREDICCIÓN'].sum()

print("PREDICCIÓN:", suma_pred)

# Sumar los valores REALES
suma_pred = df_comparacion['VALOR REAL'].sum()

print("VALOR REAL:", suma_pred)

# Sumar los valores DIFERENCIA
suma_pred = df_comparacion['DIFERENCIA'].sum()

print("DIFERENCIA:", suma_pred)
```

Ilustración 39 Resultados red neuronal 1.1

Red neuronal 1.2

Se define el modelo con los siguientes parámetros:

- layers.Dense(256): capa completamente conectada con 256 nodos.
- activation='relu': es la función de activación ReLU (Rectified Linear Unit) que permite el aprendizaje de patrones complejos.
- input_shape=(X_train.shape[1],): especifica el número de columnas de los datos de entrada.
- kernel_regularizer=regularizers.l2(0.001): Aplica regularización L2 al modelo.
- layers.Dropout(0.2): El 20% de los nodos se desactivan aleatoriamente en cada paso de entrenamiento.
- Se añaden tres capas ocultas con 128, 64 y 32 nodos.
- layers.Dense(1): la capa de salida tiene solo un nodo.

```python
# Definir el modelo de red neuronal profunda mejorado
model = models.Sequential()

# Capa de entrada con regularización L2 y más nodos
model.add(layers.Dense(256, activation='relu', input_shape=(X_train.shape[1],),
                       kernel_regularizer=regularizers.l2(0.001)))
model.add(layers.Dropout(0.2))  # Reducir el dropout para permitir más flujo de información

# Capas ocultas adicionales con regularización L2
model.add(layers.Dense(128, activation='relu', kernel_regularizer=regularizers.l2(0.001)))
model.add(layers.Dropout(0.2))

model.add(layers.Dense(64, activation='relu', kernel_regularizer=regularizers.l2(0.001)))
model.add(layers.Dropout(0.2))

model.add(layers.Dense(32, activation='relu', kernel_regularizer=regularizers.l2(0.001)))

# Capa de salida para regresión
model.add(layers.Dense(1, activation='linear'))

# Compilar el modelo usando una tasa de aprendizaje más baja
optimizer = tf.keras.optimizers.Adam(learning_rate=0.0005)

model.compile(optimizer=optimizer, loss='mean_squared_error', metrics=['mae'])

# Entrenar el modelo durante más épocas y con un batch size más pequeño
history = model.fit(X_train, y_train, epochs=100, batch_size=16, validation_split=0.2)

# Hacer predicciones con el conjunto de prueba
y_pred = model.predict(X_test)
```

Ilustración 40: Predicción red neuronal 1.2

Se evalúan 4 métricas: MSE, RMSE, MAE, R-CUADRADO.

```python
# Evaluar el modelo 1.2
# Convert y_test to a NumPy array with float32 data type
#test_loss, test_mae = neuralnetwork.evaluate(X_test, y_test.astype('float32'))

#print(f"Error absoluto medio en test: {test_mae:.2f}")

print("RED NEURONAL 1.2")

# Evaluar el modelo utilizando el conjunto de prueba (Error cuadrático medio)
mse = mean_squared_error(y_test, y_pred)
print(f"Error cuadrático medio: {mse:.2f}")

# Calcula Raíz del Error Cuadrático Medio (RMSE)
rmse = np.sqrt(mse)
print(f'Raíz del Error Cuadrático Medio (RMSE): {rmse:.2f}')

# Calcula Error Absoluto Medio (MAE)
mae = mean_absolute_error(y_test, y_pred)
print(f'Error Absoluto Medio (MAE): {mae:.2f}')

# Calcula Coeficiente de Determinación (R²)
r2 = r2_score(y_test, y_pred)
print(f"R-cuadrado: {r2:.2f}")
```

Ilustración 41: Análisis red neuronal 1.2

Agrupamos los valores por “Tipo de Cliente” y realizamos la tabla comparativa entre las predicciones y los valores reales.

```
# Desactivar la notación científica para los floats
pd.set_option('display.float_format', '{:.2f}'.format)

# Convertir las predicciones y la columna 'Tipo Cliente' en un DataFrame
df_pred = pd.DataFrame({'Tipo Cliente': A_column_test, 'Predicciones': y_pred.flatten()})

# Agrupar el dataset de predicción
df_pred_group = df_pred.groupby('Tipo Cliente')['Predicciones'].sum().reset_index()

# Agrupar el dataset de prueba
df_control_group = df_control.groupby('Tipo Cliente')['Importe'].sum().reset_index()

# Crear el dataframe de comparación
df_comparacion = pd.DataFrame({
    'TIPO CLIENTE': df_pred_group['Tipo Cliente'],
    'PREDICCIÓN': df_pred_group['Predicciones'],
    'VALOR REAL': df_control_group['Importe'],
    'DIFERENCIA': df_pred_group['Predicciones'] - df_control_group['Importe']
})

print(df_comparacion)

# Sumar los valores PREDICCIÓN
suma_pred = df_comparacion['PREDICCIÓN'].sum()

print("PREDICCIÓN:", suma_pred)

# Sumar los valores REALES
suma_pred = df_comparacion['VALOR REAL'].sum()

print("VALOR REAL:", suma_pred)

# Sumar los valores DIFERENCIA
suma_pred = df_comparacion['DIFERENCIA'].sum()

print("DIFERENCIA:", suma_pred)
```

Ilustración 42: Resultados red neuronal 1.2

Red neuronal 2

En este modelo tenemos varios desafíos para poder implementar un modelo de ML funcional con redes neuronales.

Nuestro primer desafío en la creación de la red neuronal fue el transformar la data a nivel numérico y normalizarla de acuerdo con nuestros requerimientos. Revisando la data vemos que la fecha de las facturas se repite varias veces ya que no lleva un distintivo más como hora de expendio y tenemos días que no se tiene facturas creadas. Esto nos da un resultado claro, la data se puede agrupar en los días de los dos años que tenemos para saber un estimado de ventas acumuladas. Procedemos a realizar la agrupación de las ventas y completar la progresión teniendo en cuenta

que nos dará una matriz resultante de 729 días en los dos años de datos que tenemos. Cabe mencionar que los días que no tengamos facturas los llenaremos con una venta equivalente a 0. Estos valores al ser días que no realizaron facturación no distorsionan el modelo al ser una análisis y modelo predictivo de ventas a lo largo del año.

Con la data agrupada, y teniendo una dataframe con días consecutivos, la idea de crear una serie temporal es lo más fácil y mejor que podemos realizar y simplifica la creación y tuneo de nuestra red neuronal.

El problema se centrará en tomar datos de la serie y predecir el siguiente dato de la serie. Como definición, tomaremos los primeros 7 días, el equivalente a una semana, y predeciremos el 8 día y así sucesivamente con todos los datos disponibles.

A esto lo conocemos como un aprendizaje supervisado y simplifica la creación de la red neuronal, pero requiere de la creación de una matriz de aprendizaje específica para nuestra problemática.

Creación de la matriz de aprendizaje supervisado

Con la data limpia ahora vamos a realizar el primer paso que sería ordenar la data por fecha, y agrupar la data sumando todos los valores de importe del día

```
# df_rn_1 = df[['F.Factura','Tipo Cliente','Importe']].copy()
df_rn_1 = df[['F.Factura','Importe']].copy()
df_rn_1 = df_rn_1.sort_values(by='F.Factura')
print(df_rn_1.info())
print('\n\n')
print(df_rn_1.head())
print('\n\n')
print(df_rn_1)

df_rn_1 = df_rn_1.groupby('F.Factura')['Importe'].sum()
print('\n\n')
print(df_rn_1)
print(df_rn_1.info())

df_rn_1 = df_rn_1.reset_index()
df_rn_1.columns = ['F.Factura', 'Importe']
print('\n\n')
print(df_rn_1.info())
```

Ilustración 43: Creación de serie temporal

Después procedemos a completar la serie temporal de los días que no tengamos facturación con valor en 0

```
date_range = pd.date_range(start=df_rn_1['F.Factura'].min(), end=df_rn_1['F.Factura'].max())
df_complete = pd.DataFrame({'F.Factura': date_range})
print(df_complete.info())

df_rn_1 = df_complete.merge(df_rn_1, on='F.Factura', how='left')
df_rn_1['Importe'] = df_rn_1['Importe'].fillna(0)
print(df_rn_1.info())
```

Ilustración 44: Completar la serie temporal

Una vez adquirida nuestra nueva serie temporal procedemos a normalizarla. En esto un factor clave serán los valores de normalización, como tenemos máximos y mínimos en nuestros datos lo mejor será normalizar con un rango de –1 a 1 para tener un espectro mayor en la normalización y poder usar funciones de activaciones como tangente hiperbólica, el cual es un modelo muy usado para regresiones.

```
scaler = MinMaxScaler(feature_range=(-1, 1))
values = df_rn_1['Importe'].values.reshape(-1, 1)
scaler.fit_transform(values)

df_rn_1['norm_import'] = scaler.transform(df_rn_1['Importe'].values.reshape(-1, 1))
print(df_rn_1)
print(df_rn_1.info())

plt.plot(df_rn_1['F.Factura'], df_rn_1['Importe'])
plt.show()
```

Ilustración 45: Normalización de serie temporal

Para visualizar mejor la normalización procedemos a graficar y comprobamos que ahora los valores de facturación oscilan entre –1 y 1 pero su estructura es la misma.

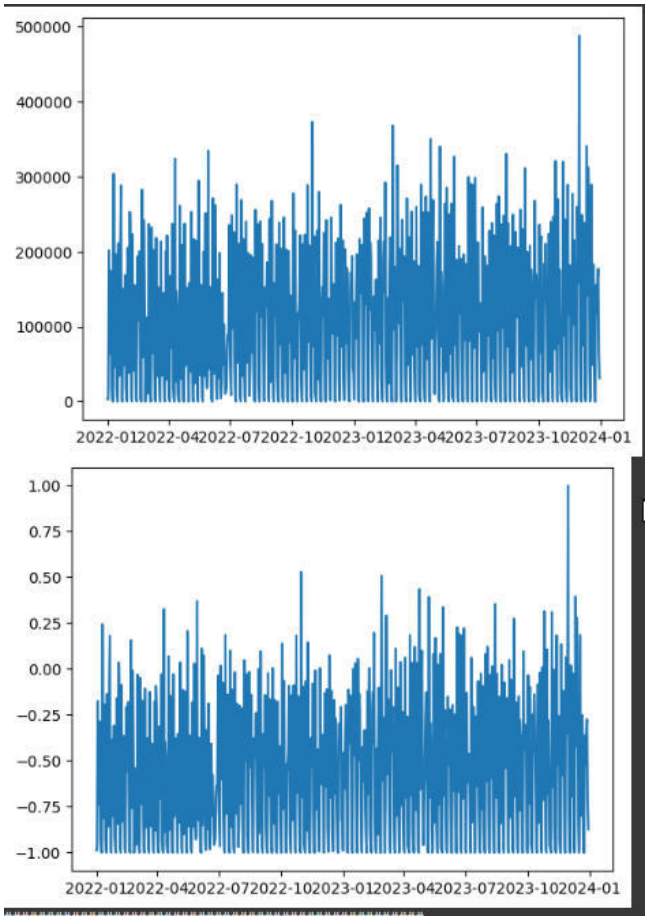


Ilustración 46: Grafica de serie temporal vs normalización

A seguir vamos con la creación de la matriz de aprendizaje supervisado que tiene la siguiente estructura

t	T+1	T+2	...	T+8
$\boldsymbol{x}$	$x(t+1)$	$x(t+2)$	...	$x(t+8)$
$\boldsymbol{x(t+2)}$	$x(t+3)$	$x(t+4)$	...	$x(t+9)$
...	...	...	...	...
$\boldsymbol{x(t+722)}$	$x(t+723)$	$x(t+724)$	...	$x(t+729)$

Usamos un lazo for para estructurar la data y crear la matriz usando la data normalizada

```
data = []
for idx in df_rn_1.index:
  if idx < len(df_rn_1)-7:
    data.append([df_rn_1['norm_import'][idx],df_rn_1['norm_import'][idx+1],df_rn_1['norm_import'][idx+2],
                 df_rn_1['norm_import'][idx+3],df_rn_1['norm_import'][idx+4],df_rn_1['norm_import'][idx+5],
                 df_rn_1['norm_import'][idx+6],df_rn_1['norm_import'][idx+7]
                 ])

df_new = pd.DataFrame(data, columns=['t', 't+1', 't+2','t+3','t+4','t+5','t+6','t+8'])
print(df_new)
```

Ilustración 47: Matriz de entrenamiento

Finalmente separamos en valores de entrada como X y valores de salida como Y

```
x = df_new.drop('t+8', axis=1)
y = df_new['t+8']
print(x.shape)
print(x.info)
print(y.shape)
print(y.info)
```

Ilustración 48: Conjunto de entrenamiento con datos de matriz

Tras tener la matriz de aprendizaje procedemos a crear nuestro modelo de red neuronal.

Los principales temas para definir son el tiempo de modelo ya sea regresión o clasificación, la cantidad de neuronas de entrada y las neuronas de salida. El resto de red y su respectivo tuneo es bastante prueba y error por lo que no se lo tomara mucho en cuenta

Tipo de análisis.

Al ser un entrenamiento supervisado en donde tomamos valores de venta de una serie temporal agrupada en una matriz de entrenamiento y determinamos el valor de salida de la octava columna de la matriz, el análisis recae en una regresión donde siete valores de entrada predicen el octavo valor de la serie temporal.

Parámetros de entrada y salida de la red neuronal

Este parámetro también lo tomamos de nuestro análisis de planteamiento del aprendizaje supervisado. Como tenemos 7 días para el análisis, la red tendrá 7 neuronas como entrada y 1 neurona como salida.

Las capas internas y las neuronas posteriores son resultado de prueba y error en donde el minimizar el error cuadrático medio y absoluto es lo principal. Tras varios intentos resulto nuestra red con 5 capas en donde 3 son las ocultas y las restantes son la entrada.

Layer (type)	Output Shape	Param #
entrada (Dense)	(None, 49)	392
oculta1 (Dense)	(None, 14)	700
oculta2 (Dense)	(None, 7)	105
salida (Dense)	(None, 1)	8

Ilustración 49: Resumen de red neuronal

En el diagrama resultante vemos la estructura de la red neuronal resultante. Se procede a compilarla y entrenarla con nuestra matriz de aprendizaje supervisado con un EPOCH de 50, lo que quiere decir que se realizaran 50 iteraciones de aprendizaje para modificar los pesos de la red neuronal.

```
model = Sequential()

#################################

model.add(Dense(49, input_dim=7, name= 'entrada',activation='tanh'))
model.add(Dense(14, name = 'oculta1', activation='tanh'))
model.add(Dense(7, name = 'oculta2', activation='tanh'))
model.add(Dense(1, name = 'salida', activation='tanh'))

#################################

model.compile(optimizer='adam',loss='mse',metrics=['mae'])
model.summary()
```

Ilustración 50: Capas del modelo RN

```
modelo_rn = model.fit(x,y,epochs=50, validation_data=(x_test, y_test))
```

Ilustración 51: Entrenamiento del RN

Posteriormente realizaremos la predicción del modelo con los valores del primer semestre de 2024

```
results=model.predict(x_test)
mse = mean_squared_error(y_test, results)
print(mse)

mae = mean_absolute_error(y_test, results)
print(mae)


plt.scatter(range(len(y_test)),y_test,c='g')
plt.scatter(range(len(results)),results,c='r')
plt.title('validate')
plt.show()

plt.plot(modelo_rn.history['loss'])
plt.plot(modelo_rn.history['val_loss'])
plt.title('Pérdida del modelo durante el entrenamiento')
plt.ylabel('Pérdida')
plt.xlabel('Época')
plt.show()
```

Ilustración 52 Predicción de RN

Y una predicción simple del semestre faltante.

```
# print(y_test.iloc[-1])
# print(x_test.iloc[-1])
# print(x_test.iloc[-1].values.reshape(1, -1))

input = x_test.iloc[-1].values.reshape(1, -1)
print(input)
print(input.shape)
mt_pred = []
print(df_control_rn.tail(10))
print(df_control_rn['F.Factura'].iloc[-1])
print('\n')

for i in range(32):
  pred = model.predict(input)
  mt_pred.append(pred[-1][0])
  var_1 = input[-1][1:]
  var_1 = np.append(var_1, pred[-1][0])
  # print(var_1)
  # print('\n')
  # print(input)
  input = np.vstack((input, var_1))
  # print('\n')
  # print(input)


start_date = df_control_rn['F.Factura'].iloc[-1]
dates = pd.date_range(start=start_date, periods=len(mt_pred))
print(dates)



df_pred = pd.DataFrame({'F.Factura': dates, 'Prediccion': mt_pred})
df_pred['Prediccion'] = scaler_control.inverse_transform(np.array(df_pred['Prediccion']).reshape(-1, 1))

plt.plot(df_pred['F.Factura'], df_pred['Prediccion'])
plt.xticks(rotation=45)
plt.show()
```

Ilustración 53: Testeo de RN

Los resultados se los discutirá en los capítulos posteriores.

CAPITULO 4

Análisis de resultados

Los datos de entrenamiento y pruebas son reales, los mismos que son extraídos directamente del ERP y que corresponden a información ingresada de la actividad de la empresa en su día a día; por lo tanto, para este dataset la probabilidad de que exista outliers o datos erróneos es muy baja.

El modelo de regresión lineal está generando valores muy altos como predicción, esto nos sugiere que este modelo no es el adecuado para este caso muy puntual de predicción, porque no existe una relación lineal o hay una relación mucho más compleja.

El modelo de árboles de decisión Random Forest Regressor tiene un comportamiento mucho más aceptable si comparamos las métricas, pero al validar los totales de valor real y el valor predicho el modelo Decision Tree Regressor tiene una menor diferencia, por lo cual, es más aceptable.

El primer modelo de red neuronal tiene un comportamiento menor al esperado, siendo menos eficaz que los modelos anteriores.

El modelo de red neuronal por la forma que está estructurado complementa el análisis previo. Con este modelo creamos predicciones de las ventas con respecto al tiempo teniendo valores muy aceptables considerando que la data que alimenta el modelo es de dos años. Con nuestro modelo de red neuronal tenemos dos casos a analizar desde el punto de ventas ya que esta información daría una gran visualización a las predicciones, pero no considera la data de los anteriores modelos así que los complementaria. Finalmente detallamos que en error absoluto tenemos un caso de error absoluto de 16% con respecto a la predicción de 2024 y casos a estudiar del resto del año.

Presentación de Resultados

COMPARACIÓN DE RESULTADOS POR MÉTRICAS				
	Error cuadrático medio	Raíz del Error Cuadrático Medio (RMSE)	Error Absoluto Medio (MAE)	R-cuadrado
Regresión Lineal	5499506.34	2345.1	1151.47	0.7
Decision Tree Regressor	357514.41	597.93	82.86	0.98
Random Forest Regressor	248301.27	498.3	67.08	0.99
Red Neuronal Profunda De Regresión	1975086477	44441.95	4275.14	-105.25
Red Neuronal Profunda De Regresión Mejorada	34438258.23	5868.41	918.72	-0.85

Ilustración 54: Comparación métricas

COMPARACIÓN DE RESULTADOS POR UNIDADES MONETARIAS			
	Predicción	Valor Real	Diferencia
Regresión Lineal	18159327.4	18670854.92	-511527.5195
Decision Tree Regressor	18670446.52	18670854.92	-408.404
Random Forest Regressor	18667557.06	18670854.92	-3297.863608
Red Neuronal Profunda De	56841696	18670854.92	38170844.3
Red Neuronal Profunda De Regresión Mejorada	21669248	18670854.92	2998391.895

Ilustración 55 Comparación unidades monetarias

El análisis de resultados se basa en los siguientes criterios:

- El margen de tolerancia en el negocio es de +- 7%, es decir, una predicción aceptable se encuentra entre valores de –7% y +7% del valor real.
- Para el negocio una predicción mayor al valor real puede derivar en pérdidas y acciones forzadas.
- Para el negocio una predicción menor al valor real es manejable en el sentido que se puede negociar con el cliente el diferir las entregas de los productos, y generar más producción con la capacidad instalada de las plantas.

De las tablas comparativas de resultados se observa que el modelo con el mejor desempeño es el Decision Tree Regressor, el cual presenta una diferencia de 408 (dólares). Esta cantidad se encuentra dentro del margen de tolerancia. Además, la predicción es un valor menor al valor real lo cual es favorable según los criterios descritos.

Red Neuronal

En la red neuronal lo primeros resultados que presentamos son la comparación del set resultante. Se grafica los puntos del modelo resultante del modelo alimentado la matriz de 2024 comparado con la estructura del primer semestre de 2024 en donde tenemos un error cuadrático medio de 0.07 y un error absoluto medio de 0.16.
Al estar normalizado vemos que el error absoluto medio nos da un error del 16% que corresponden a los valores superiores y dispersos que o encajan en el modelo.

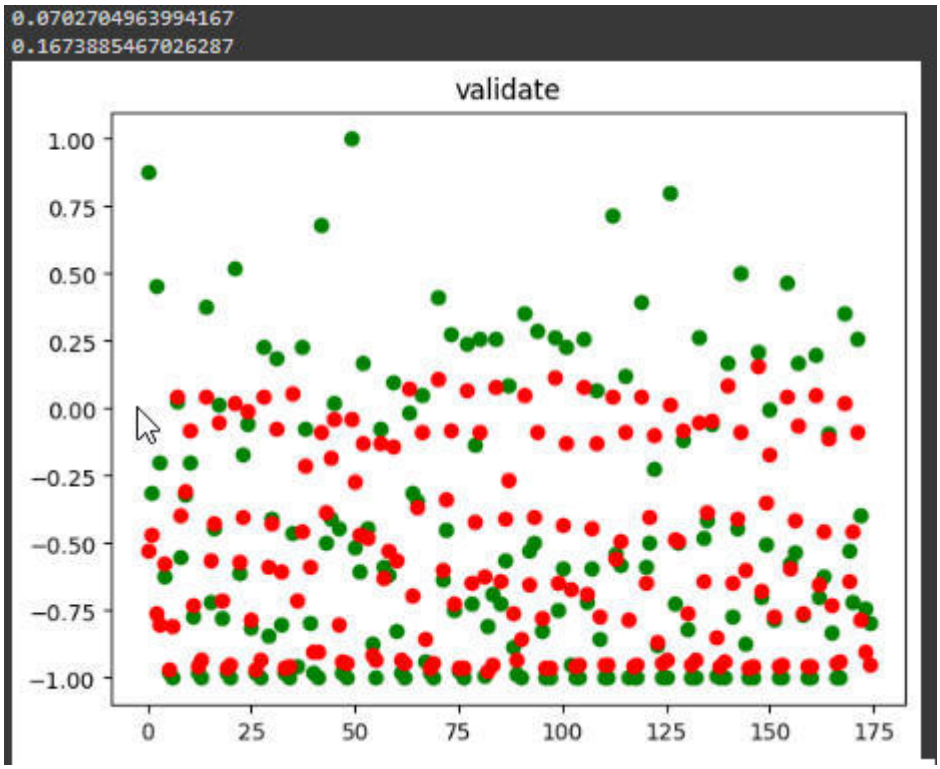


Ilustración 56: Resultado de la red neuronal vs datos del 2024

En un análisis rápido se puede apreciar fácilmente un overfiting de nuestro modelo ya que la dispersión de nuestros valores predichos no alcanza a subir a los valores fuera de la media.
Pese a que esto es un gran problema, en análisis de proyecciones de ventas es mejor manejarse con valores en rangos de la media y los valores externos superiores se los puede considerar un beneficio adicional. En nuestra empresa este error se traduciría a una falta de stock que sería un valor solventable. Si el caso fuera contrario y la media estaría en rangos superiores tendríamos un gran problema porque una sobre proyección puede afectar negativamente a la

empresa. En este caso una proyección superior que no se cumple puede traducirse en stock desperdiciado que sería costoso para nuestra empresa resultando perjudicial en todo aspecto.

Ahora para solventar el overfiting podemos reducir capas o disminuir la repetición de entrenamiento EPOCH. Pero antes revisamos las perdidas por repetición EPOCH

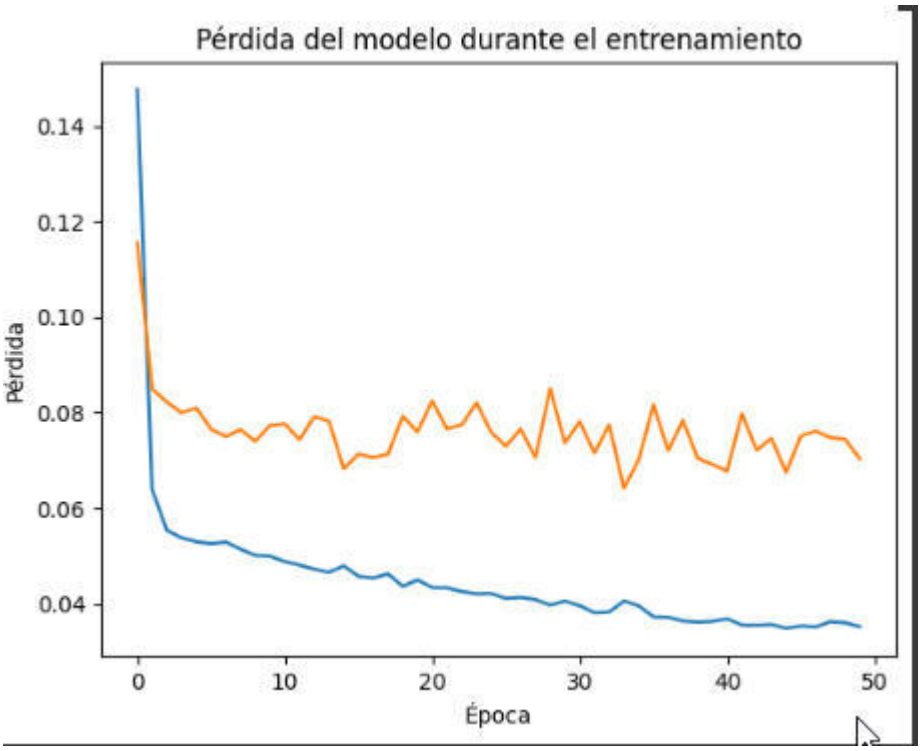


Ilustración 57: Error cuadrático medio de modelo y entrenamiento de la RN

En este gráfico apreciamos claramente que la tendencia de error medio tiene a estabilizarse con respecto a más repeticiones, pero no sería una gran diferencia ya que el resultante la línea naranja aun presenta inestabilidad con el paso de las repeticiones.
Por estas razones es preferible tener un overfiting a tener una inestabilidad mayor en los resultados.

Finalmente realizamos una predicción de nuestra data para el resto del periodo 2024, en este caso desde finales de julio al final del año.

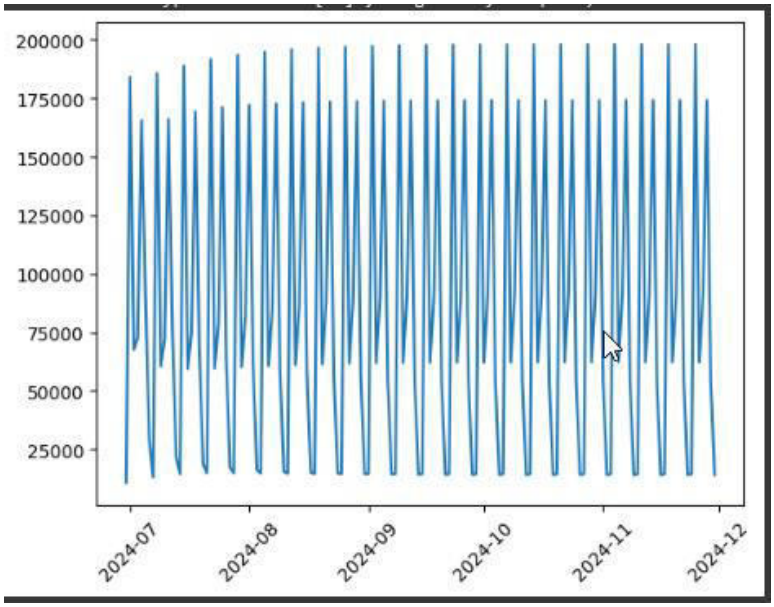


Ilustración 58: Predicción de RN con overfiting

Por nuestro overfiting tenemos una predicción repetitiva a lo largo del resto del 2024, lo cual calculando su valor medio nos da mucha información valiosa para tomar en cuenta en la proyección de ventas.

```
print(df_pred['Prediccion'].describe())
```

```
count       154.000000
mean      86510.578125
std       66815.078125
min       10732.283203
25%       15681.488770
50%       62231.667969
75%      172569.929688
max      198059.750000
Name: Prediccion, dtype: float64
```

Ilustración 59: Información de la predicción de la RN con overfiting

Si modificamos la red neuronal disminuyendo los epoch de 50 a 20 y el número de capas internas de 3 a 2 tenemos un modelo sin overfiting pero con error absoluto medio mayor

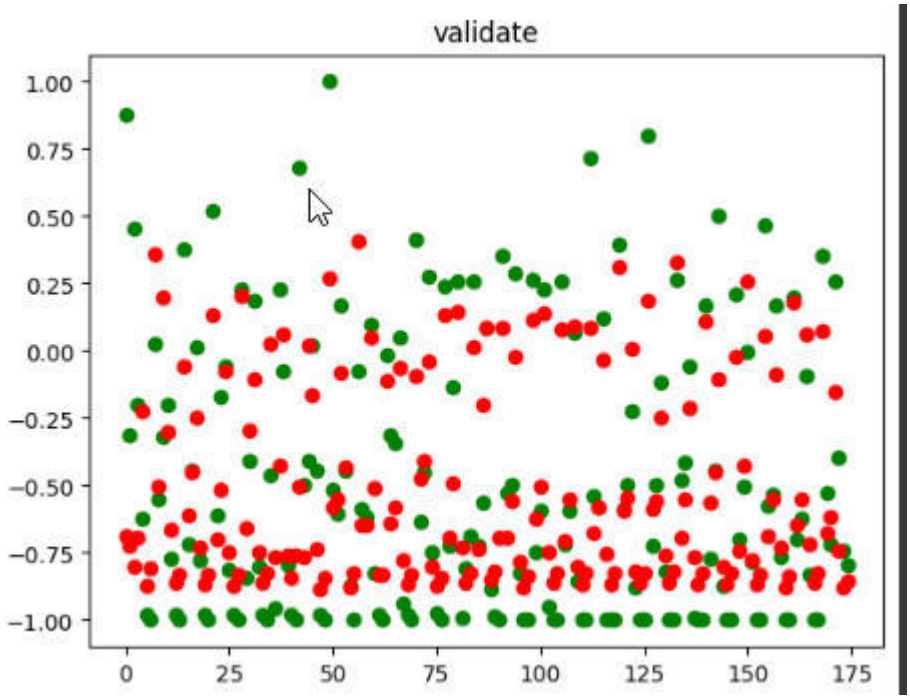


Ilustración 60: Resultado de la RN vs datos del 2024 para evitar overfiting

Esta predicción nos da un mejor ajuste a valores superiores pero el error medio y la seguridad al realizar predicción de ventas tendría un mayor error porcentual de 16% a 22%.

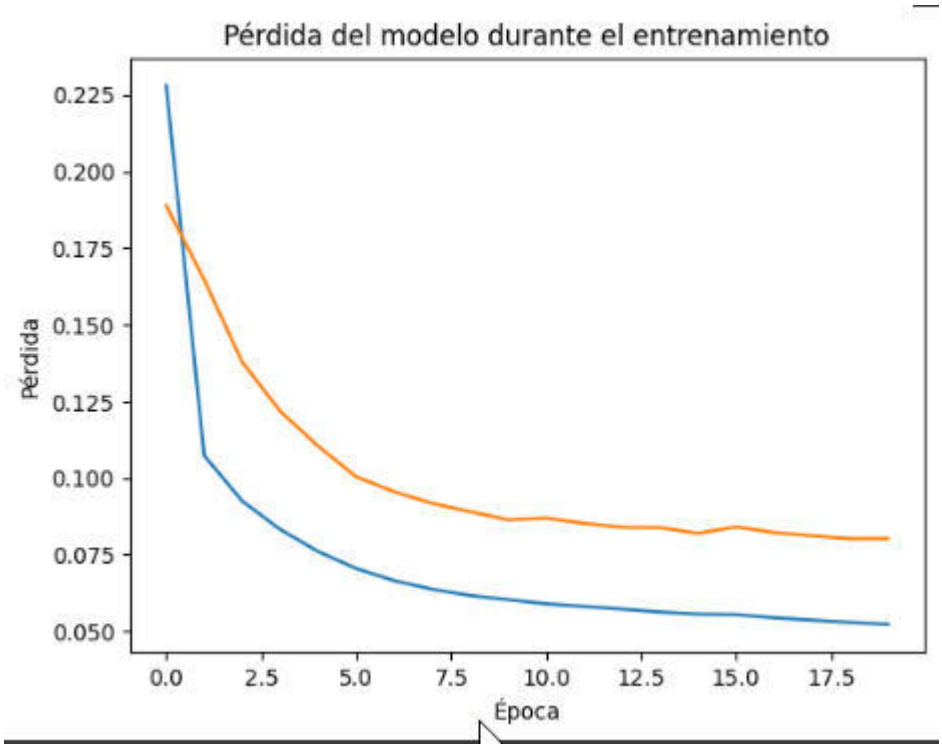


Ilustración 61: Error cuadrático medio de modelo y entrenamiento de la RN sin overfiting

Las pérdidas del error cuadrático medio tienden a estabilizarse, pero necesitaríamos más epoch lo que trae overfiting, por lo cual de ser el caso se recomienda tomar el caso anterior.

Finalmente, en cuanto a la predicción tenemos un caso muy particular, tomando en cuenta que el modelo considera picos y una distribución no uniforme, la predicción del resto del año toma una forma curiosa.

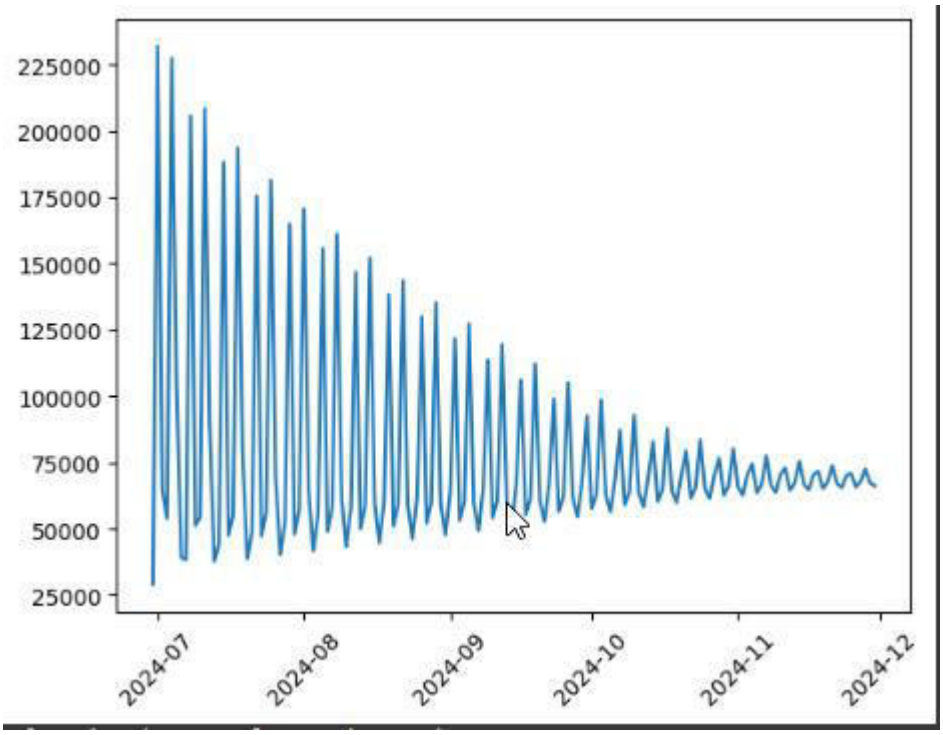


Ilustración 62: Predicción de final de 2024 sin overfiting

Una interpretación de estos resultados dice que debiéramos seguir aportando al modelo con más data de años futuros para que este modelo de red neuronal pueda determinar qué días vamos a tener los picos de ventas y una aproximación más acertada con la realidad.

```
print(df_pred['Prediccion'].describe())

count       154.000000
mean      86510.578125
std       66815.078125
min       10732.283203
25%       15681.488770
50%       62231.667969
75%      172569.929688
max      198059.750000
Name: Prediccion, dtype: float64
```

Ilustración 63: información de la predicción de la RN sin overfiting

Análisis de Predicción 2024

Con los modelos y las predicciones de todo el parcial de 2024 podemos comenzar a realizar un análisis financiero de los resultados Primero procedemos a graficar los resultados de todo el parcial 2024. Tanto para el modelo con overfiting como para el modelo sin overfiting.

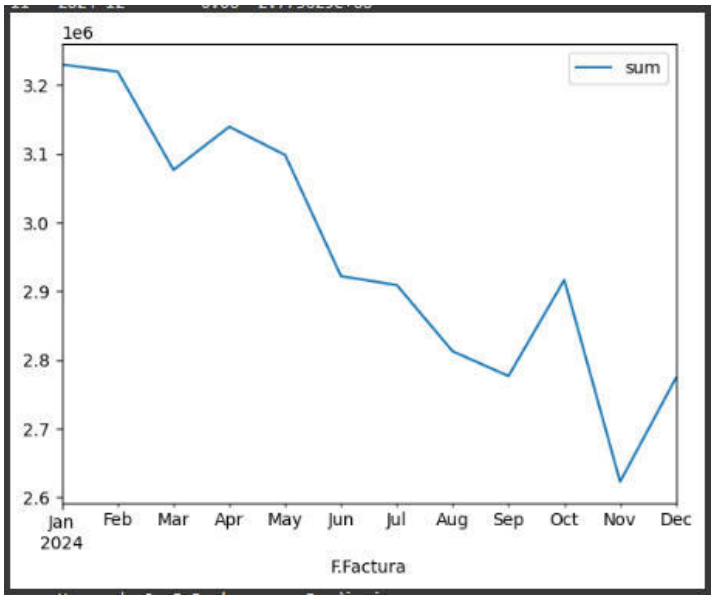


Ilustración 64: Predicción 2024 con overfting

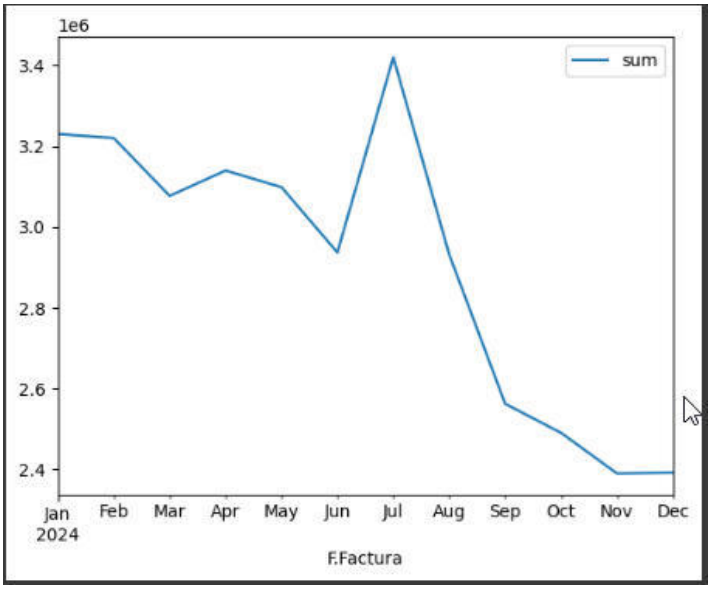


Ilustración 65: Predicción 2024 sin overfiting

En ambos modelos podemos ver que la predicción nos da una venta descendiente que continua la tendencia de venta de principios de año 2024 y contrata a la venta del 2023 en la Ilustración 17, pero está acorde a las ventas del año 2022. Esto es porque el modelo solo tiene dos años para comparar y trata de acomodar más a una media de los dos años.

Las diferencias notables entre los dos modelos son los picos que se forman tanto en el modelo de overfiting como en el sin overfiting.

En el modelo con overfiting podemos ver un resultado más parecido al final del año 2022 en donde la venta oscila los 28 millones bajando a un mínimo aproximado de 26 millones, pero recuperándose en el último parcial de diciembre.

En el modelo sin overfiting, vemos un pico alto de 34 millones lo cual englobaría la predicción creciente del año 2023 pero por la media este tiende a decrecer hasta estabilizarse en el final del año en un aproximado de ventas de 25 millones englobando las ventas bajas del año 2022 que al igual estaban en estos valores.

Tras un análisis del primer parcial de 2024, las data nos dice que no tendríamos un incremento de ventas en 2024 ya que los datos del primer semestre de 2024 tienden a la baja y

el modelo los continua según la tendencia que ha aprendido de dos años de entrenamiento. Por lo que deberíamos prepararnos para minimizar los impactos en producción, tratar de apuntar las ventas del último semestre de 2024, y ver impactos en el profit00 teniendo en cuenta que la venta media del segundo semestre del año estaría en 29 millones aproximadamente.

A nivel financiero el modelo con overfiting nos aporta más información útil ya que tenemos valores con descensos e incrementos tal cual es el comportamiento de los dos anteriores años sin una tendencia a estabilizarse. Si se prosigue esta tendencia en el 2025 no veríamos un descenso menor a 26 millones, pero tampoco veríamos incrementos de más de 30 millones. Esto minimiza el riesgo en la proyección, lo cual a nivel de finanzas es más conservador, si el departamento de ventas repunta o incrementa las ventas del último semestre podremos tener ganancias y beneficios no proyectados en la compañía.

Las conclusiones de los modelos se detallarán en el siguiente capitulo lo cual al final sería muy subjetivo al riesgo que el departamento de venta quiera atenerse y la consecutiva inversión en el programa con más datos para ir mejorando la meta data a analizar.

Análisis de rendimiento y limitaciones de los modelos

En lo que respecta a la red neuronal, tenemos un modelo preciso en lo referente al entrenamiento con los datos de 2022 y 2023, pero con una gran limitación en cuanto a la predicción de los años futuros. La falta de años de entrenamiento y el overfiting perjudican al modelo delimitando las futuras ventas en los rangos promedio existentes en 2022 y 2023, los cuales evitan que se marque una tendencia de crecimiento. Para ser más conciso esto quiere decir que la predicción de los días futuros está marcada por un rango de ventas definido, sin la posibilidad de poder predecir picos de ventas, la tendencia del modelo será a la media de las ventas de 2022 y 2023. Esto no aportara data útil ya que el promedio evita proporcionar data sensible que ayude al departamento de ventas preparar el presupuesto y acciones de ventas adecuadas. Esta limitante se solventará añadiendo más datos de entrenamiento para que se marque una tendencia de ventas en crecimiento y la red pueda predecir picos de ventas de una manera más adecuada.

En cuanto a rendimiento nuestra red neuronal da buenos resultados a corto plazo, esto quiere decir que mientras no exijamos que la red prediga más de ¼ del tiempo de entrenamiento, tendremos data relativamente útil para nuestras predicciones. Con data de dos años entrenando el modelo, actualmente la red neuronal solo podrá predecir 6 meses de datos útiles. Al finalizar el 2024 tendremos 3 años de datos para entrenar la red por lo que podremos

incrementar la proyección a 9 meses y así sucesivamente. Se estima que teniendo cinco años de datos el modelo tenga suficiente data para realizar predicciones con mayor certeza y con perturbaciones de picos de ventas sin caer en rangos promedios.

Cabe mencionar que también debemos considerar factores externos que afectan las ventas, como ejemplo ponemos las ventas del segundo semestre de 2024 que pueden verse afectadas por los apagones eléctricos que está sufriendo actualmente Ecuador. Estos factores pueden repercutir en disminución de ventas y afectar las predicciones por lo que siempre se debe analizar la data, previo al entrenamiento de los modelos.

Implementación en producción y futuro de los modelos

Se comienza aclarando que la implementación y modificación a futuro no entra en el alcance del proyecto y solo se bosqueja variables que se pueden tomar en cuenta y serian importantes para futura continuación del mismo.

El presente proyecto tiene como finalidad el realizar predicciones de presupuesto para los años futuros de nuestra compañía. Pese a ser un asunto de carácter financiero, la implementación en producción debe estar a cargo del área de TI debido al carácter técnico de los modelos. En demandas de hardware no se tiene muchas y lo mejor sería implementarlo en un cluster compartido con el ERP y con suficientes recursos para compilar entrenar y predecir redes neuronales. A nivel de automatización, los modelos deben ser entrenados anualmente para mejorar la predicción y su exactitud. Para ello requerimos realizar una integración con el módulo ERP de la compañía o crear un servicio que consuma la data de ventas de cada año y alimente los modelos por medio de APIs. Esto requiere implementar el código de limpieza y construcción de los modelos como un microservicio que exporte el modelo para que otro microservicio corra el código de la predicción.

El proceso también puede ser modificado a nivel de código sin tener una automatización o RPA (Robotic Process Automation), pero ahí requerimos aún mas de personal calificado para el entrenamiento lo cual no sería el fin. Mientras más años alimentos nuestro modelo su predicción será más exacta.

El compartir los resultados de las predicciones requiere cargar la data exportada a un reporte que grafique los datos ó subirlo directamente al ERP para que sea consumida por las áreas de ventas y financiera. Aquí el ERP puede mostrar gráficas en forma de widgets o usar reporterías como Tableu o Power BI.

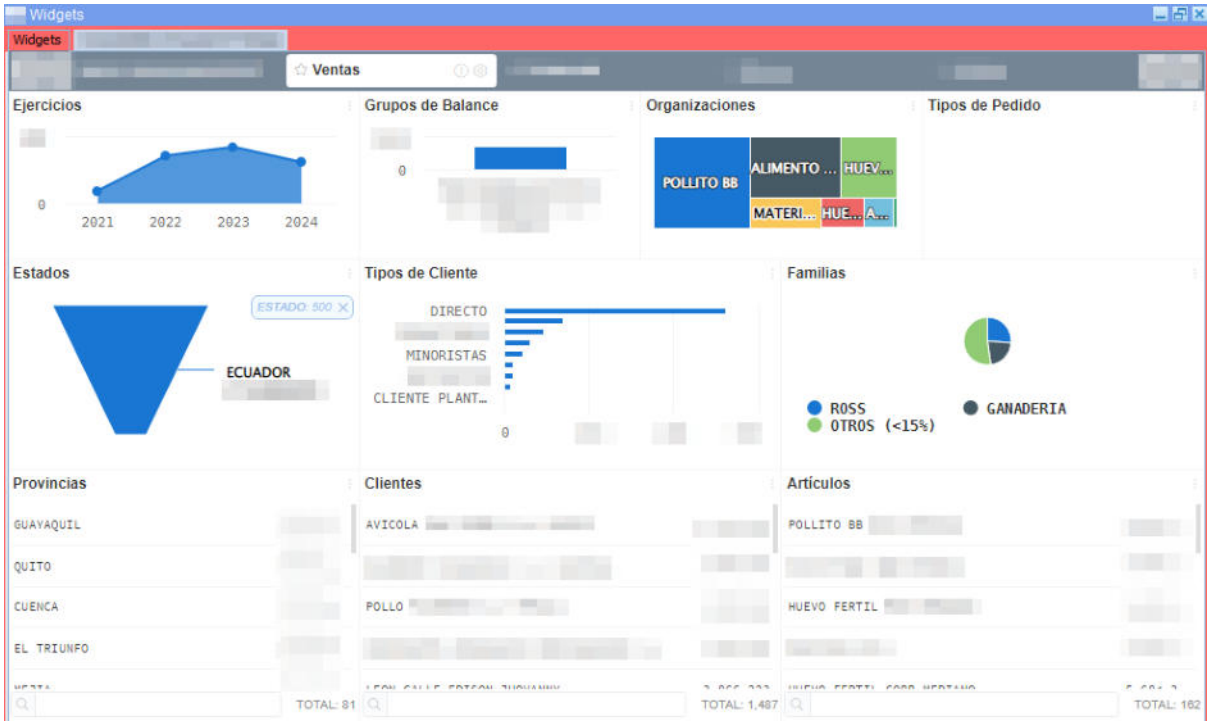


Ilustración 66: Widgets que se pueden crear en el ERP

Los reportes deberán ser revisados por el área financiera y de ser el caso modificados para definir un presupuesto adecuado para el siguiente año. Las modificaciones pueden darse a nivel de tablas compartidas de excel las cuales alimentan directamente los reportes. En estos casos el área financiera gozara de una predicción basada en datos de ventas que remplazar las aproximaciones por experiencia y ahorrara horas hombre de trabajo al mismo departamento. Con el feedback del área financiera se puede modificar los modelos o incrementar perturbaciones a la tendencia para simular eventos fortuitos que se deben considerar y ajustar.

La finalidad del proyecto es bajar la carga de trabajo al área financiera al predecir las ventas y nuestro proyecto adecuadamente implementado generaría esa disminución de horas hombre en la compañía.

CAPITULO 5

Conclusiones

- Ambos modelos requieren de más años de ventas para mejorar su predicción, pese a esto el modelo con overfiting tiene buenos resultados con un error absoluto medio de 16.73% lo que nos da un modelo con un 83% de acierto cumpliendo el kpi general que nos planteamos
- El error medio cuadrático en la predicción del primer semestre del 2024 tiene un error cuadrático que se acerca mucho al 0 con un valor de 0.07, lo que nos dice que pese a que se eliminan los valores elevados por el overfiting la media de aciertos tiene una gran precisión sin mucho error. Estos valores están dentro de lo planteado en el kpi especifico por lo que se cumple lo requerido para el proyecto pese a la falta de años para entrenar el modelo.
- De los análisis realizados a los diferentes modelos propuestos, podemos concluir que el modelo basado en arboles de decisión es el que mejor resultados entrega a la predicción de ventas.
- Si bien ambos modelos presentan resultados válidos, el modelo Decision Tree Regressor ha demostrado ser más preciso en la predicción contra el valor total real, dando así una mayor claridad entre la producción y las ventas de los meses y años posteriores.
- Por la limitada cantidad de años en ventas, el modelo con overfiting es mucho más preciso y conservador para el uso actual. Una vez se tenga más datos de ventas de diferentes años, por lo menos unos 5 años, seria de mayor utilidad bajar el número de repeticiones (epoch) al entrenar y tener un modelo que calcule mejor los picos y la tendencia de las ventas de nuestra empresa. Así evitamos el overfiting y tenemos un modelo mucho más preciso.
- A nivel del departamento de ventas, la entrega de predicciones de ventas del segundo semestre de 2024, aporta mucha data útil para obtener medidores de desempeño de las ventas, apuntalar el mercado e incrementar el beneficio, y para evaluar problemas como desperdicio de producción. Esta data sería muy útil y aportaría a las reuniones trimestrales del departamento de ventas reduciendo tiempo en la predicción y análisis en un 30% el cual era el tiempo que tomaba al departamento preparar los datos de ventas

para presentarlos. Así con la entrega de los datos y los gráficos de los mismos, cumplimos el kpi especifico relacionado a las reuniones trimestrales.

Recomendaciones

- Se recomienda continuar alimentado el modelo de la red neuronal con la data de años futuros. Según las proyecciones y análisis de los resultados mientras más años tengamos los resultados del modelo van a ser más precisos.
- El overfiting puede eliminarse reduciendo el epoch y las capas ocultas de la red neuronal; al hacer esto sacrificamos aciertos del modelo por lo que se quiere más data para compensar estas modificaciones
- Actualmente recomendamos usar los resultados con overfiting el cual nos da un modelo más conservador, pero con más similitud a los datos reales de 2024

REFERENCIAS BIBLIOGRAFICAS

- Amazon Web Services. (n.d.). *¿Qué es una red neuronal?*. Retrieved August 17, 2024, from https://aws.amazon.com/es/what-is/neural-network/
- Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.
- Google Cloud. (n.d.). *What is deep learning?*. Retrieved August 17, 2024, from https://cloud.google.com/discover/what-is-deep-learning
- TensorFlow. (n.d.). *Guía de Keras*. TensorFlow. Recuperado el 18 de agosto de 2024, de https://www.tensorflow.org/guide/keras?hl=es-419
- IBM. (n.d.). *Aprendizaje supervisado*. IBM. https://www.ibm.com/es-es/topics/supervised-learning
- Agencia de Regulación y Control Fito y Zoosanitario. (s.f.). *Misión y visión*. Agrocalidad. Recuperado el 10 de agosto de 2024, de https://www.agrocalidad.gob.ec/mision-vision/#:~:text=La%20Agencia%20de%20Regulaci%C3%B3n%20y,calidad%20de%20vida%20de%20los
- Asamblea Nacional del Ecuador. (2024, 9 de agosto). *Comisión aprobó informe para primer debate del proyecto de ley*. https://www.asambleanacional.gob.ec/es/noticia/97862-comision-aprobo-informe-para-primer-debate-del-proyecto
- Servicio de Rentas Internas. (n.d.). Impuesto al valor agregado (IVA). https://www.sri.gob.ec/impuesto-al-valor-agregado-iva
- Servicio de Rentas Internas. (2013). *Guía para contribuyentes: Bienes y servicios gravados con tarifa 0% del IVA*. Quito, Ecuador: SRI. Recuperado de https://www.sri.gob.ec/o/sri-portlet-biblioteca-alfresco-internet/descargar/38e837fe-49b1-460c-983e-efd39fd04303/Bienes%20y%20servicios%20gravados%20con%20tarifa%20cero%20porciento%20del%20IVA.pdf[1].
- Quipu. (s.f.). *Proyección financiera: Qué es, cómo hacerla y ejemplos*. Blog de Quipu. Recuperado de https://getquipu.com/blog/proyeccion-financiera/[1].

- Banco Central del Ecuador. (2023). *Informe de la evolución de la economía ecuatoriana en 2022 y perspectivas 2023*. Quito, Ecuador: Banco Central del Ecuador. Recuperado de https://contenido.bce.fin.ec/documentos/Administracion/EvolEconEcu_2022pers2023.pdf[1].
- Instituto Nacional de Estadística y Censos. (2021). *Empleo-Ene-2021*. Recuperado de https://www.ecuadorencifras.gob.ec/empleo-ene-2021/
- Instituto Nacional de Estadística y Censos. (n.d.). Censo de población y vivienda. INEC. Recuperado el 10 de agosto de 2024, de https://www.ecuadorencifras.gob.ec/censo-de-poblacion-y-vivienda/
- Fundación Humanitaria. (n.d.). Comprender el impacto de las preocupaciones por el bienestar animal en las elecciones de alimentos humanos y la viabilidad de una dieta basada en plantas. Cruelty.farm. Recuperado el 10 de agosto de 2024, de https://cruelty.farm/es/comprender-el-impacto-de-las-preocupaciones-por-el-bienestar-animal-en-las-elecciones-de-alimentos-humanos-y-la-viabilidad-de-una-dieta-basada-en-plantas/
- Python Package Index. (s.f.). *Help: Packages*. Recuperado de https://pypi.org/help/#packages
- Microsoft. (s.f.). *Get started with Power BI Desktop*. Recuperado de https://learn.microsoft.com/en-us/power-bi/fundamentals/desktop-getting-started[1].
- Patricio Fernandez, Machine Learning(ML) – Tipos de algoritmos I, T-SII_21_001043_01.pdf; https://eig.brightspace.com/d2l/le/content/146821/viewContent/1084329/View
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). Introduction to linear regression analysis. John Wiley & Sons.
- Draper, N. R., & Smith, H. (1981). Applied regression analysis. John Wiley & Sons.
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). [Diagrama de dispersión para el volumen entregado] .Recuperado de https://www.academia.edu/42811449/Introduccion_al_Analisis_de_Regresion_Lineal_Tercera_Edicion_Montgomery_Peck_Vining
- Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). [Relación rectilínea entre el tiempo de entrega y el volumen entregado] .Recuperado de

https://www.academia.edu/42811449/Introduccion_al_Analisis_de_Regresion_Lineal_Tercera_Edicion_Montgomery_Peck_Vining

- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees*. Chapman & Hall/CRC.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81-106.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45, 5-32.
- Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 29(5), 1189-1232.
- Agencia de Regulación y Control Fito y Zoosanitario. (s.f.). *Misión y visión*. Agrocalidad. Recuperado el 10 de agosto de 2024, de https://www.agrocalidad.gob.ec/mision-vision/#:~:text=La%20Agencia%20de%20Regulaci%C3%B3n%20y,calidad%20de%20vida%20de%20los
- Asamblea Nacional del Ecuador. (2024, 9 de agosto). *Comisión aprobó informe para primer debate del proyecto de ley*. https://www.asambleanacional.gob.ec/es/noticia/97862-comision-aprobo-informe-para-primer-debate-del-proyecto
- Servicio de Rentas Internas. (n.d.). Impuesto al valor agregado (IVA). https://www.sri.gob.ec/impuesto-al-valor-agregado-iva
- Servicio de Rentas Internas. (2013). *Guía para contribuyentes: Bienes y servicios gravados con tarifa 0% del IVA*. Quito, Ecuador: SRI. Recuperado de https://www.sri.gob.ec/o/sri-portlet-biblioteca-alfresco-internet/descargar/38e837fe-49b1-460c-983e-efd39fd04303/Bienes%20y%20servicios%20gravados%20con%20tarifa%20cero%20porciento%20del%20IVA.pdf[1].
- Quipu. (s.f.). *Proyección financiera: Qué es, cómo hacerla y ejemplos*. Blog de Quipu. Recuperado de https://getquipu.com/blog/proyeccion-financiera/[1].
- Banco Central del Ecuador. (2023). *Informe de la evolución de la economía ecuatoriana en 2022 y perspectivas 2023*. Quito, Ecuador: Banco Central del Ecuador.

Recuperado de https://contenido.bce.fin.ec/documentos/Administracion/EvolEconEcu_2022pers2023.pdf[1].

- Instituto Nacional de Estadística y Censos. (2021). *Empleo-Ene-2021*. Recuperado de https://www.ecuadorencifras.gob.ec/empleo-ene-2021/
- Instituto Nacional de Estadística y Censos. (n.d.). Censo de población y vivienda. INEC. Recuperado el 10 de agosto de 2024, de https://www.ecuadorencifras.gob.ec/censo-de-poblacion-y-vivienda/
- Fundación Humanitaria. (n.d.). Comprender el impacto de las preocupaciones por el bienestar animal en las elecciones de alimentos humanos y la viabilidad de una dieta basada en plantas. Cruelty.farm. Recuperado el 10 de agosto de 2024, de https://cruelty.farm/es/comprender-el-impacto-de-las-preocupaciones-por-el-bienestar-animal-en-las-elecciones-de-alimentos-humanos-y-la-viabilidad-de-una-dieta-basada-en-plantas/

- Python Package Index. (s.f.). *Help: Packages*. Recuperado de https://pypi.org/help/#packages
- Microsoft. (s.f.). *Get started with Power BI Desktop*. Recuperado de https://learn.microsoft.com/en-us/power-bi/fundamentals/desktop-getting-started[1].
- Revista SEDEN. (s.f.). Capítulo 14: Modelo de regresión lineal. Recuperado de https://www.revistaseden.org/files/14-cap%2014.pdf
- Dagnino, J. (2014). Regresión lineal. Revista Chilena de Anestesiología, 43, 143-149. Recuperado de https://www.sachile.cl/upfiles/revistas/54e63943b5d69_14_regresion-2-2014_edit.pdf
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Dubourg, V. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830. https://dl.acm.org/doi/10.5555/1953048.2078195
- Scikit-learn developers. (n.d.). *Scikit-learn: Machine learning in Python*. Retrieved September 29, 2024, from https://scikit-learn.org/stable/index.html
- Python Software Foundation. (n.d.). *Documentación de la biblioteca estándar de Python* (versión 3). https://docs.python.org/es/3/library/index.html

- Microsoft. (n.d.). *Introducción a Power BI*. https://learn.microsoft.com/es-es/power-bi/fundamentals/power-bi-overview

APENDICES Y ANEXOS

1. Resultados de la limpieza de datos

1.1 Resultado de la importación de los archivos xls a dataframes. Se imprimió un head de los dataframe para apreciar su forma y data.

```
                            Estadística De Ventas    Unnamed: 1  \
0  Agrupada por: "Org. Comercial" - "Cliente" - "...           NaN
1                                                NaN           NaN
2                                                NaN           NaN
3                      Cód. Organización Comercial  Cód. Cliente
4                                                01         00248

      Unnamed: 2    Unnamed: 3 Unnamed: 4    Unnamed: 5        Unnamed: 6  \
0            NaN           NaN        NaN           NaN               NaN
1            NaN           NaN        NaN           NaN               NaN
2            NaN           NaN        NaN           NaN               NaN
3  Cód. Artículo  Cód. Familia    Familia  Cód. Factura           Factura
4         001453        030103  GANADERIA      Factura:  01/2022/K25/1166

       Unnamed: 7           Unnamed: 8       Unnamed: 9  \
0             NaN                  NaN              NaN
1             NaN                  NaN              NaN
2             NaN                  NaN              NaN
3  Cód. F.Factura           F.Factura  Cód. Forma pago
4     F.Factura:  2022-02-04 00:00:00              C30T

                    Unnamed: 10        Unnamed: 11   Unnamed: 12  \
0                           NaN                NaN           NaN
1                           NaN                NaN           NaN
2                           NaN                NaN           NaN
3                  Forma pago  Cód. Presentación  Presentación
4  CREDITO 30 DAS TRANSFERENCIA                UND        UNIDAD

           Unnamed: 13     Unnamed: 14        Unnamed: 15   Unnamed: 16  \
0                  NaN             NaN                NaN           NaN
1                  NaN             NaN                NaN           NaN
2                  NaN             NaN                NaN           NaN
3  Cód. Canton cliente  Canton cliente  Cód. Tipo Cliente  Tipo Cliente
4                2PQ1           QUITO                C09       DIRECTO

  Unnamed: 17 Unnamed: 18
0         NaN         NaN
1         NaN         NaN
2    Año 2022         NaN
3    Cantidad     Importe
4          53     1063.19
Número total de filas 1: 51102
```

Anexo figura 1

1.2 Resultado de la modificación de columnas de nuestros dataframe. Se imprime los nombres de las columnas.

```
Index(['Cód. Organización Comercial', 'Cód. Cliente', 'Cód. Artículo',
       'Cód. Familia', 'Familia', 'Cód. Factura', 'Factura', 'Cód. F.Factura',
       'F.Factura', 'Cód. Forma pago', 'Forma pago', 'Cód. Presentación',
       'Presentación', 'Cód. Canton cliente', 'Canton cliente',
       'Cód. Tipo Cliente', 'Tipo Cliente', 'Cantidad', 'Importe'],
      dtype='object')
Index(['Cód. Organización Comercial', 'Cód. Cliente', 'Cód. Artículo',
       'Cód. Familia', 'Familia', 'Cód. Factura', 'Factura', 'Cód. F.Factura',
       'F.Factura', 'Cód. Forma pago', 'Forma pago', 'Cód. Presentación',
       'Presentación', 'Cód. Canton cliente', 'Canton cliente',
       'Cód. Tipo Cliente', 'Tipo Cliente', 'Cantidad', 'Importe'],
      dtype='object')
```

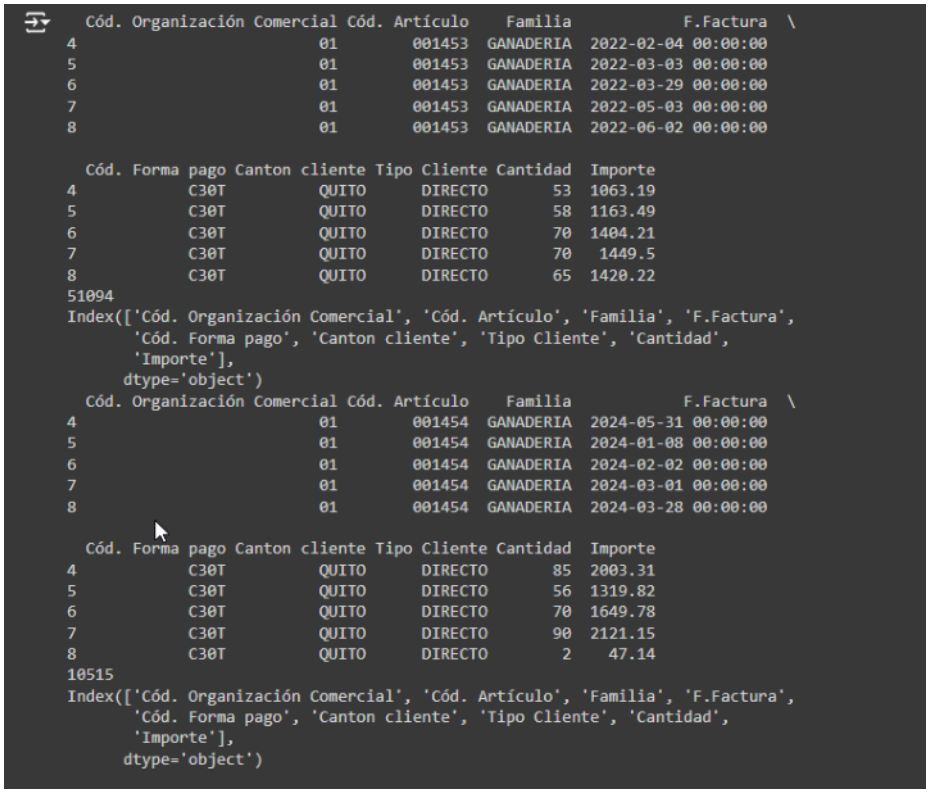
Anexo figura 2

1.3 Resultado de la eliminación de filas con metadata no consumible por los modelos ML. Se imprime el tamaño de dataframe antes y después de la limpieza.

```
51102
51094
10519
10515
```

Anexo figura 3

1.4 Resultado de la eliminación de columnas sin data relevante. Se imprime los primeros valores del dataframe , su tamaño y las columnas restantes.



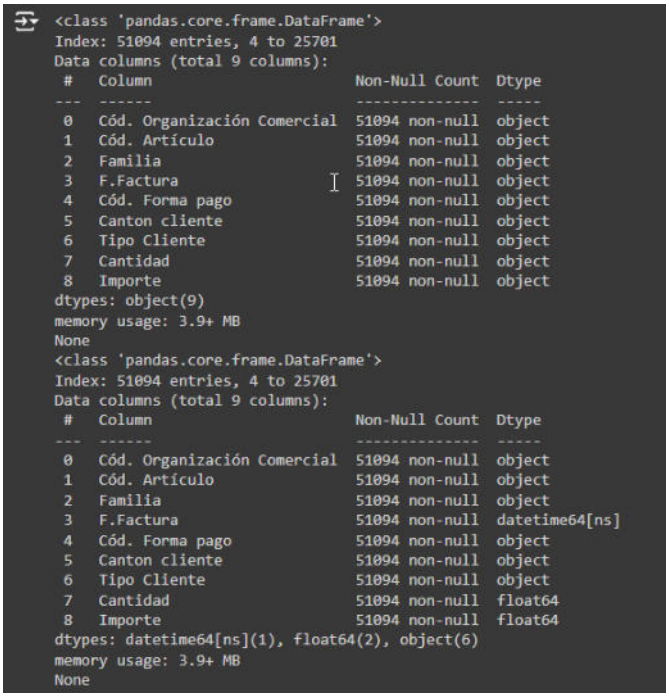
```
  Cód. Organización Comercial Cód. Artículo    Familia           F.Factura  \
4                           01        001453  GANADERIA  2022-02-04 00:00:00
5                           01        001453  GANADERIA  2022-03-03 00:00:00
6                           01        001453  GANADERIA  2022-03-29 00:00:00
7                           01        001453  GANADERIA  2022-05-03 00:00:00
8                           01        001453  GANADERIA  2022-06-02 00:00:00

  Cód. Forma pago Canton cliente Tipo Cliente Cantidad  Importe
4           C30T          QUITO      DIRECTO       53  1063.19
5           C30T          QUITO      DIRECTO       58  1163.49
6           C30T          QUITO      DIRECTO       70  1404.21
7           C30T          QUITO      DIRECTO       70   1449.5
8           C30T          QUITO      DIRECTO       65  1420.22
51094
Index(['Cód. Organización Comercial', 'Cód. Artículo', 'Familia', 'F.Factura',
       'Cód. Forma pago', 'Canton cliente', 'Tipo Cliente', 'Cantidad',
       'Importe'],
      dtype='object')
  Cód. Organización Comercial Cód. Artículo    Familia           F.Factura  \
4                           01        001454  GANADERIA  2024-05-31 00:00:00
5                           01        001454  GANADERIA  2024-01-08 00:00:00
6                           01        001454  GANADERIA  2024-02-02 00:00:00
7                           01        001454  GANADERIA  2024-03-01 00:00:00
8                           01        001454  GANADERIA  2024-03-28 00:00:00

  Cód. Forma pago Canton cliente Tipo Cliente Cantidad  Importe
4           C30T          QUITO      DIRECTO       85  2003.31
5           C30T          QUITO      DIRECTO       56  1319.82
6           C30T          QUITO      DIRECTO       70  1649.78
7           C30T          QUITO      DIRECTO       90  2121.15
8           C30T          QUITO      DIRECTO        2    47.14
10515
Index(['Cód. Organización Comercial', 'Cód. Artículo', 'Familia', 'F.Factura',
       'Cód. Forma pago', 'Canton cliente', 'Tipo Cliente', 'Cantidad',
       'Importe'],
      dtype='object')
```

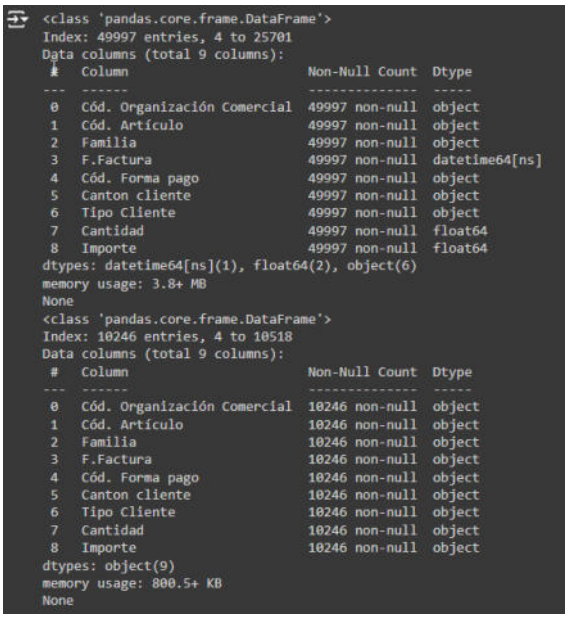
Anexo figura 4

1.5 Resultado de la trasformación del tipo de valor de cada columna. Se imprime la información de las columnas antes y después de la trasformación.



Anexo figura 5

1.6 Resultado de la eliminación de filas con valores negativos en nuestro dataframe. Se imprime la información de dataframe para apreciar las filas restantes y los formatos de las columnas restantes.



Anexo figura 6

2. *Enlace github del codigo usado* [bolivar2dam/proyecto-pbl-codigo: proyecto pbl para defensa de tesis (github.com)]