Desarrollo de un Sistema Predictivo del Tonelaje de Caña por Hectárea

Juan Angel Carrera Soto

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Desarrollo de un Sistema Predictivo del Tonelaje de Caña por Hectárea

Trabajo de graduación en modalidad de Trabajo Profesional presentado
por
Juan Angel Carrera Soto
para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computacion y Tecnologías de la Información

Guatemala
2024

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Desarrollo de un Sistema Predictivo del Tonelaje de Caña por Hectárea

Trabajo de graduación en modalidad de Trabajo Profesional presentado
por
Juan Angel Carrera Soto
para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computacion y Tecnologías de la Información

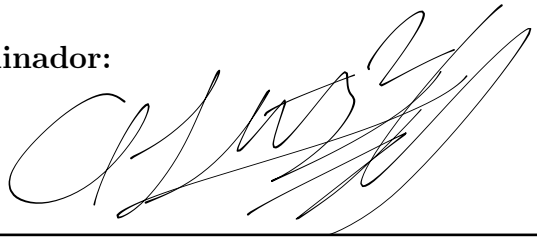
Guatemala
2024

Vo. Bo.:

(f) _____

Msc. MBA. Ing. Luis Alberto Suriano Saravia

Tribunal Examinador:



(f) _____

Msc. MBA. Ing. Luis Alberto Suriano Saravia

(f) _____

MBIA Ing. Carlos Jorge Valdéz Bautista

(f) _____

Ing. Eddy Omar Castro Jáuregui

Fecha de aprobación: Guatemala, 03 de diciembre de 2024.

Prefacio

La motivación fundamental detrás de este proyecto de investigación surge del deseo de llevar las herramientas de Inteligencia Artificial (IA) a áreas de alto potencial dentro de Guatemala, con un enfoque particular en el sector agrícola. La agricultura, siendo un pilar esencial de la economía guatemalteca, presenta una oportunidad única para la aplicación de tecnologías avanzadas que pueden revolucionar las prácticas tradicionales, mejorar la eficiencia y aumentar la productividad.

Este trabajo busca demostrar cómo la implementación de sistemas predictivos basados en IA puede transformar la gestión de cultivos, específicamente en la producción de caña de azúcar. Al combinar datos climatológicos, información satelital y registros históricos de cosechas, aspiramos a proporcionar a los agricultores y a la industria azucarera herramientas poderosas para la toma de decisiones informadas y la optimización de recursos.

La elección de enfocarnos en la caña de azúcar no es casual. Este cultivo no solo es de gran importancia económica para Guatemala, sino que también enfrenta desafíos significativos en términos de manejo y predicción de rendimientos. A través de este proyecto, esperamos contribuir al desarrollo sostenible del sector agrícola guatemalteco, promoviendo la adopción de tecnologías innovadoras que puedan mejorar la competitividad y la resiliencia de los agricultores frente a los cambios climáticos y económicos.

Quisiera expresar mi más profundo agradecimiento al Ing. Alberto Suriano por su invaluable apoyo durante todos estos años, tanto en su papel como profesor como en su rol de asesor en este proyecto. Su guía, conocimiento y constante aliento han sido fundamentales en cada etapa de esta investigación y en mi formación académica.

Asimismo, extiendo un agradecimiento especial al Ingenio Pantaleon, cuya colaboración ha sido crucial para el éxito de este sistema. Su disposición para compartir datos, conocimientos y recursos ha permitido la creación de un dataset robusto y representativo, sin el cual este proyecto no habría sido posible. La apertura y el compromiso del Ingenio Pantaleon con la innovación y la investigación son ejemplares y han enriquecido enormemente este trabajo.

Índice

Lista de figuras

Lista de cuadros

Resumen

En el sector agrícola guatemalteco, la predicción precisa del rendimiento de los cultivos de caña de azúcar representa un desafío significativo debido a la complejidad de las variables involucradas. Este proyecto abordó esta problemática mediante el desarrollo de un sistema predictivo del Tonelaje de Caña por Hectárea (TCH) utilizando técnicas avanzadas de aprendizaje automático. El trabajo se estructuró en tres fases principales: primero, la integración de múltiples fuentes de datos, incluyendo registros climatológicos del Instituto de Cambio Climático, índices vegetativos derivados de imágenes satelitales de NAX Solutions, y datos históricos de cosecha del Ingenio Pantaleon; segundo, el desarrollo y optimización de modelos predictivos mediante Azure AutoML, implementando diferentes arquitecturas de aprendizaje automático; y tercero, el despliegue de una infraestructura MLOps completa con una API REST en Flask para la automatización del pipeline de datos y predicciones. Los resultados demostraron la efectividad del enfoque propuesto, con los modelos de ensamble basados en XGBoost y LightGBM alcanzando un R^2 de hasta 0.817 y un RMSE tan bajo como 8.638 toneladas por hectárea en las predicciones a corto plazo. En particular, el modelo demostró una capacidad robusta para predecir el TCH con diferentes horizontes temporales, desde 2 hasta 10 meses antes de la cosecha, manteniendo un R^2 superior a 0.77 incluso en las predicciones a largo plazo. La interpretabilidad del modelo, analizada mediante valores SHAP, reveló la importancia crítica de factores como el tipo de cosecha, la variedad de caña y los índices vegetativos en la determinación del rendimiento final. En conclusión, este proyecto no solo proporciona una herramienta práctica para la optimización de la producción de caña de azúcar, sino que también demuestra la viabilidad de integrar múltiples fuentes de datos y técnicas avanzadas de aprendizaje automático en un sistema operativo y escalable. El éxito en la implementación de una infraestructura MLOps completa subraya el potencial de la inteligencia artificial para transformar prácticas agrícolas tradicionales, ofreciendo soluciones que pueden mejorar significativamente la eficiencia y sostenibilidad de la industria azucarera.

CAPÍTULO 1

Introducción

En la era de la transformación digital, el sector agrícola enfrenta desafíos sin precedentes y oportunidades extraordinarias. La necesidad de optimizar los recursos, aumentar la productividad y garantizar la sostenibilidad ambiental ha impulsado la búsqueda de soluciones innovadoras. En este contexto, emerge este proyecto pionero destinado a revolucionar el monitoreo y gestión de cultivos de caña de azúcar mediante el uso de tecnologías avanzadas de inteligencia artificial.

La idea principal se centra en digitalizar un proceso que en los entornos actuales se realiza de forma manual, con la intención de detectar más tempranamente posibles áreas afectadas y proporcionar una mejor proyección sobre la cantidad de toneladas de azúcar que se pueden llegar a producir. Este proyecto incluye el desarrollo de un sistema predictivo del tonelaje de caña por hectárea (TCH) utilizando datos derivados de imágenes satelitales y datos climáticos.

En este proyecto nos enfocaremos en el analisis del inidice de madurez para poder ser modelado y predecido con Inteligencia Artificial. La primera etapa implica el analisis y el procesamiento de los datos de la zafra para poder relacionar los datos climaticos y indices de sensores hacia el rendimiento de cada uno de los terrenos. Se utilizaran datos historicos pára crear un conjunto de datos de entrenamiento y desplegar un modelo predictivo robusto.

Finalmente, se realizará una validación en campo durante la cosecha para evaluar la precisión de los modelos. El proyecto esta enfocado a poder dar un mayor contexto de cuales pueden ser las mejores decisiones para optimizar recursos y maximizar los rendimientos mientras se minimizan los costos de la empresa.

CAPÍTULO 2

Objetivos

2.1. Objetivo general

Desarrollar un sistema avanzado de monitoreo y gestión de cultivos de caña de azúcar que, mediante el uso de tecnologías de análisis de datos y modelado predictivo, mejore la evaluación de la madurez de los cultivos.

2.2. Objetivos específicos

- Analizar datos históricos climatológicos y datos derivados de imágenes satelitales para analizar su impacto en el rendimiento de los cultivos.
- Preprocesar datos para preparar el conjunto de datos para el entrenamiento de modelos predictivos.
- Identificar factores que influyen en el índice TCH mediante un análisis exploratorio de los datos.
- Entrenar al menos 2 modelos predictivos usando algoritmos de aprendizaje de máquina para estimar el impacto de múltiples variables en el índice de TCH.
- Validar el resultado de los modelos predictivos con datos recopilados en los cultivos de interés.

CAPÍTULO 3

Justificación

La gestión eficiente de las zafras de caña de azúcar enfrenta desafíos considerables, especialmente en un contexto marcado por la variabilidad climática y la necesidad de optimizar recursos. Este proyecto se centrará en el desarrollo de un sistema de análisis de datos y modelado predictivo para la predicción del Tonelaje de Caña por Hectárea (TCH) en cada zafra, integrando datos derivados de imágenes satelitales, información climatológica, datos de localización y detalles sobre el manejo de los ingenios.

La necesidad de mejorar la capacidad predictiva y la eficiencia en la gestión de las zafras de caña de azúcar es imperiosa. La predicción precisa del TCH es vital para la planificación estratégica y la asignación de recursos en la industria azucarera, lo que impacta directamente en su viabilidad económica y en la seguridad alimentaria (6).

El sistema que se desarrollará integrará datos específicos relacionados con la madurez de la caña, condiciones ambientales, datos derivados de imágenes satelitales, datos de localización y manejo de los ingenios. Los datos derivados de imágenes satelitales, proporcionados por plataformas avanzadas de análisis de datos geoespaciales, se consideran variables de importancia crítica para predecir el TCH (12). A través del análisis de estos datos, se buscará desarrollar un entendimiento más profundo de cómo las condiciones climáticas y la gestión del cultivo afectan el rendimiento (16).

Al predecir el TCH con mayor precisión, se facilitará la toma de decisiones informadas en tiempo real, lo que permitirá mejorar la capacidad de respuesta ante situaciones imprevistas y eventos climáticos extremos. Esto no solo minimizará las pérdidas económicas, sino que también contribuirá a la estabilidad y sostenibilidad a largo plazo de la industria azucarera. Este proyecto ofrecerá una herramienta innovadora y eficaz para la predicción del TCH en las zafras de caña de azúcar, basada en la integración de datos de diversas fuentes. Esta solución promoverá una gestión más eficiente y sostenible de las zafras, lo que repercutirá positivamente en la seguridad alimentaria, la viabilidad económica y la sostenibilidad ambiental de la industria azucarera.

CAPÍTULO 4

Marco teórico

4.1. Agricultura en Guatemala

De acuerdo con los documentos proporcionados, la agricultura juega un papel importante en la economía de Guatemala. Según los datos de la Encuesta Nacional de Empleo e Ingresos (ENEI) 2022, el 27.1 % de la población ocupada se dedica a actividades agrícolas, siendo uno de los sectores que más empleo genera junto con el comercio (10). En cuanto a su contribución al Producto Interno Bruto (PIB), las cifras más recientes del Banco de Guatemala indican que la agricultura, ganadería, silvicultura y pesca representaron el 10.6 % del PIB en el primer trimestre de 2024 (9). Esto la ubica como uno de los sectores más relevantes de la economía guatemalteca, aunque su participación ha disminuido ligeramente en los últimos años. Es importante destacar que la agricultura sigue siendo una fuente de empleo crucial, especialmente en las áreas rurales del país, donde ocupa a una proporción significativa de la fuerza laboral. Sin embargo, los ingresos en este sector tienden a ser más bajos en comparación con otras actividades económicas, lo que plantea desafíos en términos de desarrollo rural y reducción de la pobreza.

La agricultura en Guatemala es un pilar esencial no solo para la economía interna, sino también para su proyección internacional, particularmente a través de las exportaciones. En 2023, el azúcar, uno de los productos agrícolas más destacados del país, alcanzó exportaciones por un valor de USD $595,570,553. Este dato no solo resalta la importancia del azúcar en el comercio exterior, sino también cómo la agricultura guatemalteca contribuye al desarrollo económico, la generación de empleo y la balanza comercial del país. La influencia de la agricultura se extiende más allá de las fronteras, fortaleciendo la posición de Guatemala en el mercado global y subrayando su rol como exportador de productos agrícolas de alta demanda.(2)

Guatemala cuenta con una rica diversidad de zonas agroecológicas que van desde las tierras bajas tropicales hasta las zonas montañosas, lo que permite una amplia variedad de cultivos como café, cardamomo, maíz, y frijoles. Esta diversidad agroecológica es un recurso valioso para el país, pero también presenta desafíos significativos. La agricultura guatemalteca enfrenta problemas como la variabilidad climática, que incluye sequías prolongadas y lluvias intensas, la degradación del suelo, y la limitada disponibilidad de tecnología y financiamiento, especialmente para los pequeños agricultores. Estos desafíos son exacerbados por la falta de acceso a recursos tecnológicos y financieros, lo que limita la capacidad de los pequeños productores para adaptarse a las condiciones cambiantes y mejorar su productividad.

4.1.1. Tipos de cultivos en Guatemala

El sector agrario en los países en desarrollo se caracteriza por una marcada heterogeneidad, que refleja las diferentes capacidades productivas y económicas de sus actores. Este sector puede segmentarse en tres categorías principales, cada una con características distintas que influyen en la estructura y el desarrollo de la producción agrícola.

El primer segmento está liderado por grandes empresas y productores modernos. Este grupo se destaca por su dinamismo y elevado potencial económico, lo que les permite adoptar altos niveles de tecnología y alcanzar una mayor productividad. Estas grandes unidades de producción suelen estar orientadas al monocultivo, y su producción está destinada tanto a la exportación como al mercado interno. El uso de tecnologías avanzadas no solo mejora la eficiencia, sino que también incrementa la competitividad en mercados internacionales, contribuyendo significativamente a la economía del país y fomentando el crecimiento del sector agropecuario.

El segundo segmento lo conforman los pequeños y medianos productores, quienes, aunque cuentan con menores recursos, han comenzado a integrarse a nuevos mercados. Estos productores representan una parte vital del sector agrario, ya que buscan diversificar sus productos y acceder a mercados especializados. La integración de estos productores en la economía formal y en nuevos mercados es crucial para su desarrollo económico y social, ya que les permite aumentar sus ingresos y mejorar su calidad de vida.

Por último, el segmento de la agricultura familiar es el más numeroso y se compone principalmente de hogares rurales que dependen de pequeñas explotaciones para su subsistencia. Este segmento, que representa una parte considerable del sector agrícola en muchos países, enfrenta desafíos significativos debido a la limitada disponibilidad de activos y recursos. A pesar de estos desafíos, la agricultura familiar juega un papel esencial en la seguridad alimentaria y en la reducción de la pobreza en las áreas rurales. Según informes de la CEPAL y la FAO, este sector no solo es fundamental para el sustento de millones de familias, sino que también es clave para el desarrollo sostenible en el ámbito rural (15).

La estrategia de desarrollo en estos países subdesarrollados debe, por tanto, centrarse en impulsar el crecimiento agrícola de manera inclusiva. El fortalecimiento de la agricultura, especialmente a través de políticas que apoyen a los pequeños y medianos productores, así como a la agricultura familiar, es fundamental para mejorar la productividad y los ingresos en el campo. Además, la promoción de prácticas agrícolas sostenibles y el acceso a tecnologías modernas son vitales para mejorar la competitividad y asegurar la participación de estos segmentos en mercados nacionales e internacionales(1).

Estos esfuerzos no solo contribuirán a la reducción de la pobreza y el desempleo en las zonas rurales, sino que también impulsarán la producción alimentaria, mejorarán los ingresos por exportación y contribuirán al desarrollo económico general. Como señala la CEPAL, es esencial reconocer la importancia del sector rural y agrícola como motores de desarrollo económico, social y ambiental, lo que exige un compromiso renovado para transformar las estructuras agrarias y fomentar un crecimiento inclusivo y sostenible en la región (15)

4.2. Caña de azúcar en Guatemala

La caña de azúcar es uno de los cultivos comerciales más importantes de Guatemala. Durante la zafra 2022/2023, la producción de azúcar alcanzó alrededor de 3.0 millones de toneladas métricas, con una notable eficiencia en la cosecha, que cerró con un 62 % de mecanización, lo que marca un avance significativo en comparación con zafras anteriores. De esta producción, se exportaron cerca de 2.0 millones de toneladas métricas, lo que refleja la importancia de este cultivo no solo para la

economía local sino también para el comercio internacional. Este cultivo es vital para la economía del país, generando alrededor de 65,000 empleos directos y 350,000 indirectos, y contribuyendo significativamente al PIB de las exportaciones agrícolas(7).

4.2.1. Tipos de caña de azúcar

En Guatemala, la caña de azúcar es un cultivo de gran relevancia económica y, por tanto, se han desarrollado y adoptado diversas variedades para maximizar su producción y adaptabilidad a las condiciones locales. Entre las principales variedades cultivadas en el país se encuentran aquellas que han sido desarrolladas a través de programas de fitomejoramiento, como las variedades CG, desarrolladas por el Centro Guatemalteco de Investigación y Capacitación de la Caña de Azúcar (CENGICAÑA)(7). Estas variedades incluyen denominaciones como CG02-163, CG04-10295 y CGMex10-26315, cada una adaptada a diferentes condiciones agroecológicas del país.

Las abreviaturas que acompañan los nombres de las variedades de caña tienen su origen en los países o instituciones que las desarrollaron. Por ejemplo, «CG» se refiere a CENGICAÑA, indicando variedades desarrolladas en Guatemala. Otras siglas como «CP» corresponden a variedades provenientes del Canal Point en Florida, Estados Unidos, y «CL» a variedades desarrolladas en el estado de Louisiana, también en Estados Unidos. Estas siglas ayudan a identificar el origen y, en algunos casos, las características de resistencia y productividad de las variedades.

Las variedades de caña se seleccionan no solo por su capacidad de producir altos rendimientos de azúcar, sino también por su resistencia a enfermedades, su adaptabilidad a diferentes tipos de suelo y clima, y su capacidad para soportar las prácticas de manejo utilizadas en la agroindustria guatemalteca. Por ejemplo, las variedades como CG02-163 han demostrado ser altamente productivas en términos de toneladas de azúcar por hectárea, superando en algunos casos a variedades tradicionales como la CP72-2086 (7).

4.2.2. Etapas de crecimiento de la caña de azúcar

El crecimiento de la caña de azúcar en Guatemala se divide en varias etapas cruciales, cada una con características y requerimientos específicos. Estas etapas incluyen la siembra, la emergencia, el macollamiento, la elongación y la maduración, que se desarrollan a lo largo de aproximadamente 12 meses, dependiendo de las condiciones climáticas y el manejo del cultivo.

Siembra: La siembra marca el inicio del ciclo de cultivo, donde las semillas o esquejes de caña se plantan en los surcos preparados. Esta etapa requiere una cuidadosa preparación del suelo y una selección adecuada de variedades, teniendo en cuenta la zona agroecológica y el propósito del cultivo. La siembra se realiza manualmente o mediante maquinaria especializada, asegurando una distribución uniforme de los esquejes para un crecimiento homogéneo.

Emergencia: Tras la siembra, los brotes comienzan a emerger del suelo, en un proceso que puede tomar varias semanas dependiendo de la temperatura del suelo y la humedad disponible. La emergencia es una etapa crítica, ya que define la densidad final de las plantas por unidad de área. Un manejo adecuado del riego y el control de malezas es esencial para asegurar un buen establecimiento del cultivo.

Macollamiento: Durante el macollamiento, las plantas de caña desarrollan múltiples brotes secundarios o macollos a partir de la base de los tallos principales(6). Esta etapa es fundamental para definir el número de tallos productivos que se cosecharán al final del ciclo. Factores como la disponibilidad de nutrientes y agua, así como las prácticas de manejo, influyen directamente en la cantidad y calidad de los macollos.

Elongación: La elongación es la fase en la que los tallos de la caña crecen en altura y diámetro. Es durante esta etapa que la caña acumula la mayor parte de su biomasa, y la eficiencia en el uso del agua y nutrientes es crucial para maximizar el crecimiento. Los tallos se alargan considerablemente, y se comienzan a formar los entrenudos, que son los segmentos entre las hojas.

Maduración: La maduración es la última etapa del ciclo de crecimiento, donde la caña alcanza su máximo contenido de sacarosa. Durante esta fase, el manejo del riego y la aplicación de madurantes químicos se utilizan para inducir la maduración uniforme y mejorar la concentración de azúcar en los tallos. La correcta sincronización de la cosecha es vital para obtener el máximo rendimiento de azúcar por hectárea(6).

4.2.3. Fitomejoramiento

El fitomejoramiento en la caña de azúcar es una estrategia clave en Guatemala para mejorar la rentabilidad y sostenibilidad de la agroindustria azucarera. El proceso de fitomejoramiento, tal como lo implementa CENGICAÑA, se enfoca en la creación de variabilidad genética, la selección y desarrollo de genotipos superiores, y la liberación de variedades para su producción comercial. Este programa busca generar nuevas variedades que no solo sean altamente productivas en términos de toneladas de azúcar por hectárea, sino que también sean resistentes a enfermedades y adaptables a las diferentes condiciones climáticas y de suelo que prevalecen en las zonas cañeras del país.

La creación de variabilidad genética es fundamental para el éxito del programa de fitomejoramiento. Esto se logra mediante el cruzamiento de variedades seleccionadas que presentan características deseables, como alta productividad, resistencia a enfermedades y buena adaptabilidad. Durante el periodo 2022-2023, se llevaron a cabo más de 600 cruzamientos, que dieron lugar a miles de nuevas plántulas que luego fueron evaluadas y seleccionadas en varios estados de desarrollo. El proceso de selección es riguroso y se realiza en diferentes estaciones experimentales para garantizar que las variedades seleccionadas puedan adaptarse a diversos ambientes agroecológicos.

El desarrollo y selección de nuevas variedades pasan por varias etapas antes de su liberación para la producción comercial. Estas etapas incluyen pruebas de campo para evaluar la productividad de azúcar, la resistencia a enfermedades, y las características agronómicas como el crecimiento y la calidad del tallo. Solo aquellas variedades que demuestran ser superiores en todas estas áreas son liberadas y adoptadas por los ingenios azucareros en Guatemala. Variedades como CG02-163 y CG11-07922 han sido el resultado de este exhaustivo proceso de fitomejoramiento, contribuyendo significativamente al aumento de la productividad azucarera en el país(7).

4.2.4. Índices de la caña de azúcar

Los índices de la caña de azúcar y los indicadores de cosecha son herramientas esenciales para la gestión y optimización del cultivo en Guatemala. Estos índices proporcionan información valiosa sobre el estado del cultivo y permiten a los productores tomar decisiones informadas para maximizar la producción y minimizar las pérdidas.

Entre los índices más utilizados está el Índice de Vegetación de Diferencia Normalizada (NDVI, por sus siglas en inglés), que se emplea para monitorear la salud y el vigor del cultivo a lo largo de su ciclo de crecimiento. El NDVI se calcula a partir de imágenes satelitales y permite identificar áreas del campo que pueden necesitar atención especial, como ajustes en el riego o la aplicación de fertilizantes. Otro índice importante es el Índice de Humedad del Suelo (NDWI), que se utiliza para evaluar la disponibilidad de agua en el suelo y planificar el riego de manera más eficiente.

Los indicadores de cosecha, por otro lado, incluyen métricas como el rendimiento en toneladas de caña por hectárea (TCH) y el rendimiento de azúcar por hectárea (TAH). Estos indicadores se miden

durante la cosecha y son cruciales para evaluar la eficiencia del cultivo y la calidad del producto final. Por ejemplo, la variedad CG02-163 ha mostrado un rendimiento superior en términos de TAH, lo que la hace especialmente valiosa para los productores.

El seguimiento de estos índices y indicadores permite a los ingenios y a los agricultores ajustar sus prácticas de manejo en tiempo real, optimizando el uso de recursos como el agua y los fertilizantes, y asegurando una cosecha de alta calidad y cantidad. Además, el uso de tecnologías como el monitoreo satelital y las plataformas digitales para la gestión del riego ha permitido mejorar significativamente la eficiencia y sostenibilidad de la producción azucarera en Guatemala(7).

4.3. Tecnologías aplicadas en el monitoreo y gestión de cultivos

4.3.1. Agricultura 5.0

La Agricultura 5.0 representa la siguiente etapa en la evolución del sector agrícola, superando las innovaciones de la Agricultura 4.0 al incorporar tecnologías aún más avanzadas y un enfoque estratégico que busca optimizar la eficiencia energética y reducir costos en la producción agrícola. Este concepto integra inteligencia artificial, robótica, biotecnología avanzada y sistemas energéticos sostenibles, con el objetivo de transformar la agricultura en una industria más inteligente, adaptable y resiliente frente a los desafíos globales contemporáneos.

En el marco de la Agricultura 5.0, los avances tecnológicos no solo se centran en la precisión y automatización, sino también en la integración de sistemas energéticos que optimicen el uso de recursos. Esto incluye el uso de energías renovables y la implementación de sistemas de gestión energética inteligentes que permiten a las explotaciones agrícolas reducir significativamente su huella de carbono mientras mantienen o incluso mejoran su productividad(16).

La Agricultura 5.0 también enfatiza el desarrollo de tecnologías de vanguardia, como drones equipados con inteligencia artificial para el monitoreo de cultivos, robots agrícolas capaces de realizar tareas con una precisión milimétrica, y sistemas biotecnológicos que permiten el desarrollo de cultivos más resistentes y de mayor rendimiento. Estas tecnologías no solo aumentan la eficiencia de las operaciones agrícolas, sino que también contribuyen a la sostenibilidad y a la adaptación al cambio climático(16).

Un aspecto crucial de la Agricultura 5.0 es su enfoque en la gestión estratégica de los recursos y la reducción de costos operativos. La integración de sistemas de big data y análisis predictivo permite a los agricultores optimizar cada aspecto de su producción, desde la siembra hasta la cosecha, garantizando que cada decisión se base en datos precisos y en tiempo real. Esta capacidad de toma de decisiones informada y rápida es fundamental para enfrentar los desafíos de un mercado global cada vez más competitivo y cambiante.

Con la Agricultura 5.0, el sector agrícola no solo se moderniza tecnológicamente, sino que también adopta un enfoque más holístico y sostenible, que tiene en cuenta la necesidad de equilibrar la productividad con la conservación de los recursos naturales y la mitigación del impacto ambiental. Este enfoque promete no solo mejorar la eficiencia y reducir los costos, sino también garantizar que la agricultura pueda seguir alimentando a una población mundial en crecimiento de manera sostenible y responsable.

4.3.2. Imágenes satelitales y SIG

Las imágenes satelitales han emergido como una herramienta esencial en la agricultura moderna, revolucionando la forma en que se monitorean y gestionan los cultivos a gran escala. Desde la década de 1970, con el lanzamiento de las primeras misiones Landsat por parte de NASA, las observaciones satelitales han permitido un seguimiento global de los sistemas agrícolas, proporcionando datos críticos para evaluar una variedad de parámetros geofísicos y biofísicos, como la precipitación, la temperatura, la evapotranspiración, la humedad del suelo y la salud de la vegetación (13).Estos datos permiten a los agricultores y a las agencias gubernamentales tomar decisiones informadas sobre la gestión de los cultivos, el manejo de recursos hídricos y la mitigación de riesgos asociados con eventos climáticos adversos.

El uso de sistemas de información geográfica (SIG) junto con la teledetección satelital ha ampliado aún más el potencial de las imágenes satelitales en la agricultura. SIG permite integrar y analizar la dimensión espacial de los datos agrícolas, facilitando la estimación del rendimiento de los cultivos, la evaluación de la fertilidad del suelo, el monitoreo de los patrones de cultivo y la detección y manejo de plagas y enfermedades (12). En particular, la tecnología SIG ha demostrado ser valiosa para la agricultura de precisión, un enfoque que busca optimizar el uso de insumos como fertilizantes y agua, minimizando el impacto ambiental y mejorando la sostenibilidad de los sistemas agrícolas.

Además, la integración de datos satelitales en políticas agrícolas y de seguridad alimentaria se ha vuelto cada vez más crucial, especialmente en el contexto del cambio climático. Las imágenes satelitales permiten una evaluación precisa de la productividad agrícola, lo que es vital para la estabilidad del mercado y la planificación de la ayuda humanitaria en regiones afectadas por sequías o desastres naturales(13). No obstante, aunque las aplicaciones de SIG en la agricultura han ganado popularidad en la última década, persisten desafíos significativos, especialmente en regiones de bajos ingresos, donde la falta de infraestructura y capacitación limita la adopción de estas tecnologías(12).

4.3.3. Datos climáticos y estaciones meteorológicas

El papel de las estaciones meteorológicas en la agricultura ha ganado una relevancia crítica en el contexto de la adaptación al cambio climático. Estas estaciones proporcionan datos precisos y en tiempo real sobre las condiciones climáticas locales, como la temperatura, la precipitación, la humedad del suelo y la velocidad del viento, que son esenciales para tomar decisiones informadas en la gestión agrícola. En particular, las estaciones meteorológicas del Instituto de Ciencias del Clima (ICC) juegan un papel fundamental al suministrar información que permite a los agricultores ajustar sus prácticas en función de las variaciones climáticas, optimizando así el uso de recursos y mejorando la resiliencia de los cultivos ante fenómenos extremos.

El uso de datos climáticos generados por estas estaciones es crucial para la implementación de estrategias de agricultura inteligente frente al clima, que se centran en adaptar las técnicas agrícolas para enfrentar los desafíos impuestos por el cambio climático. Por ejemplo, la monitorización precisa de la humedad del suelo y la temperatura permite a los agricultores determinar los momentos óptimos para la siembra, riego y cosecha, minimizando el riesgo de pérdida de cultivos debido a condiciones climáticas adversas.

Además, estos datos no solo son útiles a nivel operativo, sino que también juegan un papel esencial en la planificación a largo plazo. Las estaciones meteorológicas permiten la recopilación de datos históricos que pueden ser analizados para identificar patrones y tendencias climáticas, lo que a su vez facilita la predicción de futuros escenarios climáticos y la preparación de estrategias agrícolas más robustas. En el contexto del cultivo de caña de azúcar, esta capacidad para predecir y gestionar el impacto del clima es particularmente relevante, ya que este cultivo es altamente sensible a las variaciones climáticas, y cualquier desajuste en la gestión del clima puede resultar en pérdidas significativas de productividad.

4.3.4. Inteligencia Artificial en la agricultura

La integración de la inteligencia artificial (IA) en la agricultura está marcando un punto de inflexión en la manera en que se gestionan y optimizan los sistemas agrícolas. Las aplicaciones de IA, como el aprendizaje automático y las redes neuronales profundas, han demostrado ser herramientas poderosas en la clasificación de cultivos, la detección temprana de enfermedades y plagas, y la optimización de recursos como el agua y los fertilizantes. Estas tecnologías permiten a los agricultores tomar decisiones más precisas y en tiempo real, lo que se traduce en una mayor eficiencia y sostenibilidad de las prácticas agrícolas(4). La robótica y los vehículos aéreos no tripulados (UAVs), equipados con IA y sensores avanzados, están siendo cada vez más utilizados para el monitoreo continuo de los cultivos, permitiendo una intervención rápida y eficaz ante cualquier irregularidad detectada. Este monitoreo constante mejora no solo la productividad, sino también la resiliencia de los sistemas agrícolas frente a los desafíos climáticos y ambientales.

A pesar de estos avances, la adopción de tecnologías de IA en la agricultura no está exenta de desafíos. Para pequeños y medianos agricultores, los costos iniciales de implementación y la necesidad de formación especializada siguen siendo barreras significativas. Aunque los costos de hardware y software han disminuido, aún pueden resultar prohibitivos, limitando el acceso a estas innovaciones. Además, se requiere una mayor intervención de políticas públicas para desarrollar tecnologías más accesibles y proporcionar la capacitación necesaria para que todos los agricultores, independientemente de su tamaño o ubicación, puedan beneficiarse de estas herramientas(4).

La capacidad de la IA para manejar y analizar grandes volúmenes de datos es crucial en la agricultura moderna. Este análisis detallado y en tiempo real permite prever rendimientos de cultivos con una precisión sin precedentes, lo que a su vez mejora la planificación y la gestión agrícola, reduciendo los riesgos asociados con factores impredecibles como el clima y las enfermedades(4). La implementación de IA en la agricultura no solo mejora la productividad y eficiencia, sino que también es fundamental para la gestión sostenible de los recursos naturales y la adaptación al cambio climático, factores críticos en el desarrollo de sistemas avanzados de monitoreo y gestión de cultivos como el propuesto en el proyecto de caña de azúcar.

El avance de la inteligencia artificial (IA) en la agricultura no solo se ha centrado en la optimización de las prácticas agrícolas tradicionales, sino que también ha dado lugar a la aparición de nuevos modelos de negocio que están transformando el sector agrícola de manera significativa. Según el estudio .ªrtificial intelligence and new business models in agriculture: a structured literature review and future research agenda", la implementación de IA en la agricultura ha facilitado la creación de enfoques de gestión más eficientes y sostenibles, impulsando la digitalización de la cadena de suministro agrícola y fomentando la adopción de prácticas de agricultura de precisión(5).

Estos nuevos modelos de negocio están profundamente interrelacionados con conceptos como la agricultura 4.0 y la agricultura 5.0, donde la IA no solo se emplea para mejorar la producción, sino también para integrar toda la cadena de valor agrícola, desde la producción hasta la distribución. Esto implica una mayor coordinación y optimización en todos los niveles, utilizando tecnologías avanzadas para predecir la demanda, gestionar recursos de manera más eficiente y reducir el desperdicio. Además, la digitalización y el uso de plataformas basadas en IA permiten a los agricultores acceder a mercados más amplios y mejorar su competitividad global, lo que es crucial en un entorno económico cada vez más globalizado(5).La literatura sugiere que, aunque la IA presenta un potencial enorme, su éxito depende en gran medida de la adaptación de los agricultores y de la estructura del mercado agrícola. La creación de modelos de negocio innovadores que aprovechen al máximo las capacidades de la IA es esencial para garantizar que estas tecnologías se integren de manera efectiva en el sector. Sin embargo, también se requiere un marco regulatorio adecuado que fomente la adopción de estas tecnologías, garantizando al mismo tiempo la equidad y sostenibilidad en su implementación. Este enfoque complementa la visión de cómo la IA puede no solo optimizar los procesos agrícolas existentes, sino también transformar la agricultura en una industria más conectada, eficiente y resiliente frente a los desafíos del futuro.

4.4. Aprendizaje por máquina

4.4.1. Aprendizaje supervisado

El aprendizaje automático, conocido como machine learning, es una subdisciplina clave dentro del campo de la inteligencia artificial. Se centra en el desarrollo de algoritmos y modelos que permiten a las máquinas aprender de los datos y mejorar su desempeño en tareas específicas sin la necesidad de ser explícitamente programadas para cada tarea. A diferencia de la inteligencia artificial simbólica, que se basa en la codificación de reglas predefinidas, el aprendizaje automático se fundamenta en la idea de que los sistemas pueden identificar patrones en los datos y utilizar estos patrones para hacer predicciones o tomar decisiones.

El aprendizaje supervisado es una de las principales categorías del aprendizaje automático. En este enfoque, el modelo se entrena con un conjunto de datos etiquetados, donde cada ejemplo de entrada está asociado con una salida deseada. El objetivo es aprender una función que pueda mapear nuevas entradas a las salidas correctas. Este tipo de aprendizaje es particularmente útil para tareas de clasificación y regresión.

La historia del aprendizaje automático es una evolución natural de los intentos iniciales de emular la inteligencia humana. Con el tiempo, los investigadores se dieron cuenta de que para crear sistemas verdaderamente inteligentes, las máquinas necesitaban la capacidad de aprender de su entorno y adaptarse a nuevos datos sin intervención humana constante. Este cambio de paradigma ha sido impulsado por avances en la capacidad de procesamiento de datos, el desarrollo de algoritmos más sofisticados, y la disponibilidad de grandes volúmenes de datos.

El aprendizaje automático se ha convertido en una columna vertebral esencial de la inteligencia artificial moderna, ofreciendo un enfoque flexible y poderoso para el desarrollo de sistemas capaces de realizar tareas complejas y adaptarse a nuevas circunstancias. Ha demostrado su capacidad para transformar la forma en que se resuelven problemas en una variedad de dominios, subrayando su importancia creciente en el avance de la IA (11).

Árboles de decisión

Uno de los enfoques más prominentes dentro del aprendizaje automático supervisado es el uso de árboles de decisión. Los árboles de decisión son modelos predictivos que dividen repetidamente un conjunto de datos en subconjuntos más pequeños basados en características específicas, de tal manera que cada nodo en el árbol representa una decisión basada en un atributo particular. Este proceso de partición continua se realiza hasta que los datos se hayan clasificado de manera efectiva o se alcance un criterio de parada predefinido(11).

El concepto de árboles de decisión se ha vuelto particularmente popular debido a su simplicidad y su capacidad para manejar tanto problemas de clasificación como de regresión. Estos modelos son intuitivos y fáciles de interpretar, lo que los hace valiosos en aplicaciones donde la interpretabilidad del modelo es crucial, como en el ámbito médico o financiero.

Además, los árboles de decisión forman la base de técnicas más avanzadas en aprendizaje automático, como los bosques aleatorios (random forests) y los métodos de boosting, que combinan múltiples árboles para mejorar la precisión y la robustez del modelo. Estas técnicas de conjunto han demostrado ser altamente efectivas en una amplia gama de aplicaciones, desde la predicción de riesgos financieros hasta el diagnóstico médico automatizado.

La versatilidad y la interpretabilidad de los árboles de decisión los han convertido en una herramienta fundamental en el toolkit del aprendizaje automático, proporcionando una base sólida para el análisis de datos y la toma de decisiones basada en modelos en diversos campos (11).

XGBoost

XGBoost (Extreme Gradient Boosting) es una implementación optimizada de árboles de decisión en gradiente boosting que ha demostrado ser altamente eficiente y flexible para problemas de regresión y clasificación. Su popularidad radica en su capacidad para manejar grandes volúmenes de datos, mejorar la precisión mediante regularización y reducir el riesgo de sobreajuste, convirtiéndolo en una de las herramientas más poderosas en el campo del aprendizaje automático supervisado. (8)

Funcionamiento XGBoost construye un conjunto de árboles de decisión de forma secuencial. Cada nuevo árbol se entrena para corregir los errores cometidos por los árboles anteriores. El proceso se puede resumir en los siguientes pasos:

Inicialización: Se crea un primer árbol simple. Cálculo de residuos: Se calculan los errores (residuos) del modelo actual. Construcción de nuevo árbol: Se entrena un nuevo árbol para predecir estos residuos. Adición al modelo: El nuevo árbol se añade al modelo con un peso determinado. Iteración: Se repiten los pasos 2-4 hasta alcanzar un criterio de parada.

XGBoost incorpora varias mejoras sobre el gradient boosting tradicional, incluyendo:

Regularización para prevenir el sobreajuste. Manejo eficiente de valores faltantes. Paralelización y optimización de hardware para mayor velocidad.

Hiperparámetros principales XGBoost ofrece una amplia gama de hiperparámetros para ajustar el rendimiento del modelo. Algunos de los más importantes son:

n_estimators: Número de árboles en el modelo (número de rondas de boosting). **max_depth**: Profundidad máxima de cada árbol. **learning_rate**: Tasa de aprendizaje, controla cuánto contribuye cada árbol al modelo final. **subsample**: Fracción de muestras utilizadas para entrenar cada árbol. **colsample_bytree**: Fracción de características utilizadas en cada árbol. **gamma**: Parámetro de regularización que controla la poda del árbol. **min_child_weight**: Suma mínima de peso de instancias necesaria en un hijo. **alpha**: Término de regularización L1 en los pesos. **lambda**: Término de regularización L2 en los pesos. **scale_pos_weight**: Control del balance de clases para problemas no balanceados. **early_stopping_rounds**: Número de rondas sin mejora antes de detener el entrenamiento. **objective**: Función objetivo a optimizar (por ejemplo, 'binary:logistic' para clasificación binaria). **eval_metric**: Métrica de evaluación utilizada durante el entrenamiento.

La optimización de estos hiperparámetros es crucial para obtener el mejor rendimiento de XGBoost. Técnicas como la validación cruzada y la búsqueda en cuadrícula o aleatoria se utilizan comúnmente para encontrar la mejor combinación de hiperparámetros para un problema específico.

LGBMRegressor

LightGBM (Light Gradient Boosting Machine) es un framework de gradient boosting desarrollado por Microsoft, y LGBMRegressor es su implementación para problemas de regresión. LightGBM se destaca por su eficiencia y velocidad, especialmente en conjuntos de datos grandes, gracias a sus innovadoras técnicas de construcción de árboles.

Funcionamiento LGBMRegressor utiliza una técnica llamada GOSS (Gradient-based One-Side Sampling) para la construcción de árboles. Los pasos principales son:

Inicialización: Se crea un modelo inicial simple. Cálculo de gradientes: Se calculan los gradientes para cada instancia. Muestreo: Se seleccionan instancias basadas en la magnitud de sus gradientes.

Construcción del árbol: Se construye un árbol utilizando las instancias seleccionadas. Actualización del modelo: El nuevo árbol se añade al modelo. Iteración: Se repiten los pasos 2-5 hasta cumplir el criterio de parada.

LightGBM introduce varias mejoras sobre los algoritmos de gradient boosting tradicionales:

Crecimiento de árboles por hojas (leaf-wise) en lugar de por niveles (level-wise). Técnica de binning de características para reducir el tiempo de entrenamiento. Soporte para características categóricas sin necesidad de codificación one-hot.

Hiperparámetros principales LGBMRegressor ofrece una amplia gama de hiperparámetros para ajustar el rendimiento del modelo. Algunos de los más importantes son:

n_estimators: Número de árboles en el modelo (iteraciones de boosting). **learning_rate**: Tasa de aprendizaje, controla el impacto de cada árbol en el modelo final. **max_depth**: Profundidad máxima de los árboles. **num_leaves**: Número máximo de hojas en cada árbol. **min_child_samples**: Número mínimo de muestras en cada nodo hoja. **subsample**: Fracción de muestras utilizadas para entrenar cada árbol. **colsample_bytree**: Fracción de características utilizadas en cada árbol. **reg_alpha**: Término de regularización L1. **reg_lambda**: Término de regularización L2. **min_split_gain**: Ganancia mínima para realizar una división en el árbol. **boosting_type**: Tipo de boosting ('gbdt' por defecto, también 'dart', 'goss', 'rf'). **feature_fraction**: Fracción de características a seleccionar aleatoriamente en cada iteración. **bagging_fraction**: Fracción de datos a utilizar para cada iteración. **bagging_freq**: Frecuencia para realizar bagging. **early_stopping_rounds**: Número de rondas sin mejora antes de detener el entrenamiento. **metric**: Métrica de evaluación para la validación.

La optimización de estos hiperparámetros es crucial para obtener el mejor rendimiento de LGBMRegressor. Técnicas como la validación cruzada, la búsqueda en cuadrícula o la optimización bayesiana se utilizan comúnmente para encontrar la mejor combinación de hiperparámetros para un problema específico

VotingRegressor

El VotingRegressor es un meta-estimador que combina las predicciones de múltiples modelos base para producir una predicción final. Este enfoque de conjunto (ensemble) puede mejorar la robustez y la precisión de las predicciones al aprovechar las fortalezas de diferentes modelos. El VotingRegressor puede operar en dos modos principales: hard voting y soft voting.

Hard Voting En el modo de hard voting para regresión, también conocido como "predicción promedio", el VotingRegressor simplemente calcula la media aritmética de las predicciones de todos los modelos base. Este método es simple y puede ser efectivo cuando los modelos base tienen un rendimiento similar. La predicción final para una instancia (x) se calcula como:

$$\hat{y}(x) = \frac{1}{n} \sum_{i=1}^{n} \hat{y}_i(x)$$

donde $\hat{y}_i(x)$ es la predicción del i-ésimo modelo base y n es el número total de modelos base.

Soft Voting El soft voting en regresión, también conocido como "predicción ponderada", asigna pesos diferentes a las predicciones de cada modelo base. Estos pesos pueden ser definidos por el

usuario o calculados automáticamente basándose en el rendimiento de los modelos. La predicción final para una instancia (x) en soft voting se calcula como: $\hat{y}(x) = \sum_{i=1}^{n} w_i \hat{y}_i(x)$

donde w_i es el peso asignado al i-ésimo modelo base, y $\sum_{i=1}^{n} w_i = 1$.

Cálculo de pesos en Soft Voting En el modo soft, los pesos pueden ser determinados de varias maneras:

Pesos definidos por el usuario: El usuario puede asignar pesos manualmente basándose en su conocimiento del problema o el rendimiento esperado de cada modelo.

Pesos basados en el rendimiento: Los pesos pueden calcularse automáticamente basándose en el rendimiento de cada modelo en un conjunto de validación. Algunos métodos comunes incluyen:

a) **Inverso del error**: Los pesos se calculan como el inverso del error de cada modelo. Por ejemplo, usando el Error Cuadrático Medio (MSE):

$$w_i = \frac{1/MSE_i}{\sum_{j=1}^{n} 1/MSE_j}$$

b) **Basados en el coeficiente de determinación ($\mathbf{R^2}$)**: Los pesos se calculan proporcionalmente al R^2 de cada modelo:

$$w_i = \frac{R_i^2}{\sum_{j=1}^{n} R_j^2}$$

Optimización de pesos: En algunos casos, los pesos pueden ser optimizados directamente minimizando una función de pérdida en el conjunto de validación. Esto puede hacerse usando técnicas de optimización como el descenso de gradiente.

4.4.2. Aprendizaje profundo

Perceptrón y Redes Neuronales Feedforward

El perceptrón es el bloque de construcción fundamental de las redes neuronales artificiales. Un perceptrón simple consiste en múltiples entradas $(x_1, x_2, \ldots, x_n)$, cada una con un peso asociado $(w_1, w_2, \ldots, w_n)$, y un sesgo (b) (14). Estas entradas se combinan de manera lineal para producir un valor ponderado que luego se pasa a través de una función de activación. El objetivo del perceptrón es tomar una decisión basada en la combinación de las entradas ponderadas.

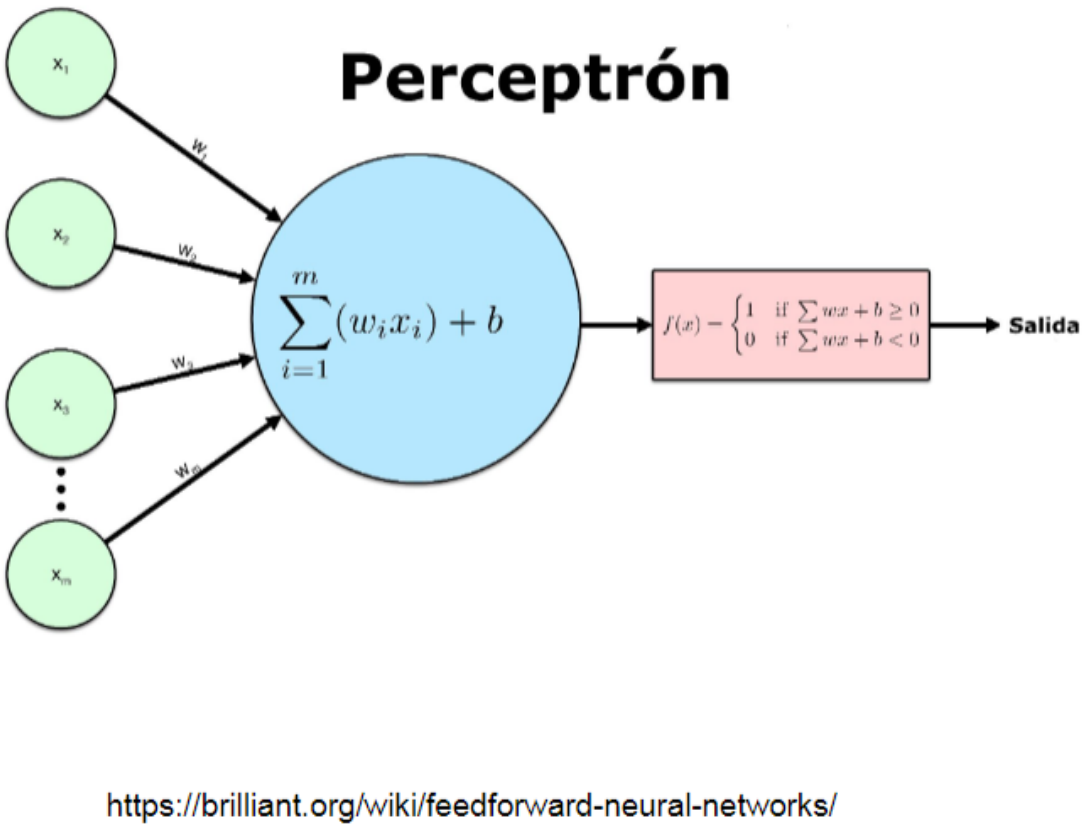


Figura 4.1: Diagrama de un perceptrón

El perceptrón básico puede aprender a clasificar datos linealmente separables mediante el ajuste iterativo de los pesos. Sin embargo, para problemas más complejos, se requiere una estructura más avanzada, como las redes neuronales feedforward.

Las redes neuronales feedforward están compuestas por múltiples capas de perceptrones, también conocidas como neuronas. Estas redes constan de una capa de entrada, una o más capas ocultas y una capa de salida. Cada neurona en una capa recibe entradas de las neuronas de la capa anterior, aplica pesos (w_{ij}) a esas entradas y, después de pasar por una función de activación, transmite el resultado a las neuronas de la siguiente capa.

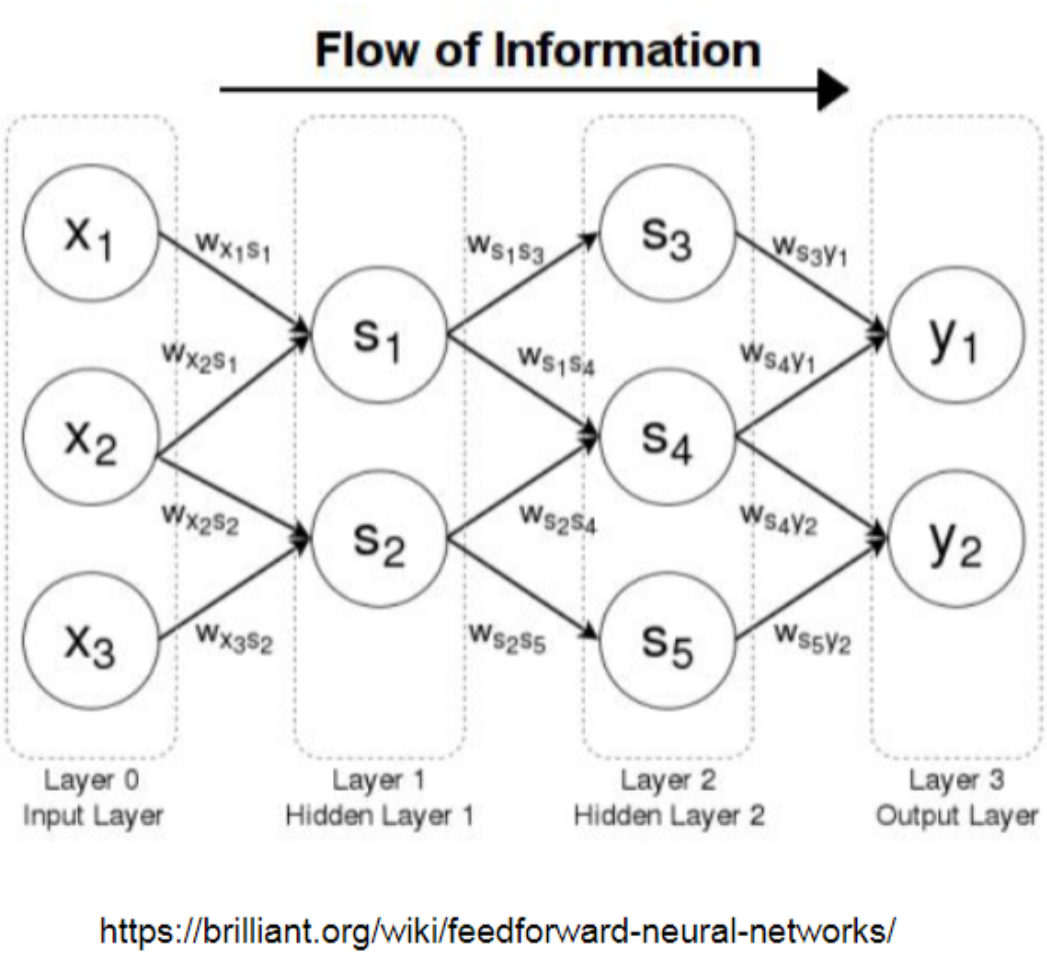


Figura 4.2: Diagrama de una red neuronal FeedFoward

En una red feedforward típica, la información se mueve en una única dirección, desde la capa de entrada, a través de las capas ocultas, hasta la capa de salida, sin que haya retroalimentación (es decir, conexiones de vuelta). Este tipo de arquitectura es esencial para tareas de clasificación y regresión, donde el objetivo es aproximar funciones complejas a partir de datos de entrada(14).

Deep Learning

El Deep Learning (DL), o aprendizaje profundo, es una subdisciplina del Machine Learning que ha revolucionado el campo de la inteligencia artificial al permitir que las máquinas aprendan representaciones jerárquicas de datos a través de múltiples capas de procesamiento no lineal. A diferencia de los métodos tradicionales de aprendizaje automático, que suelen depender de características manualmente diseñadas, el DL emplea redes neuronales profundas, las cuales son capaces de aprender automáticamente características de alto nivel a partir de datos brutos (3). Estas redes neuronales, también conocidas como Deep Neural Networks (DNNs), están compuestas por múltiples capas de nodos o neuronas, donde cada capa recibe entradas de la capa anterior, las procesa a través de funciones de activación no lineales, y produce una salida que se convierte en la entrada de la siguiente capa.

El concepto fundamental detrás del DL se basa en la arquitectura de las redes neuronales artificiales, inspiradas en el funcionamiento del cerebro humano, donde las neuronas biológicas se activan en respuesta a ciertos estímulos y transmiten señales a otras neuronas. En las redes neuronales artificiales, una neurona recibe varias entradas, las pondera, las combina y pasa el resultado a través de una función de activación, generando una salida que puede ser transmitida a otras neuronas en la red (3). A medida que la red se entrena con grandes volúmenes de datos, las conexiones entre las neuronas se ajustan, optimizando el modelo para realizar tareas específicas como clasificación, regresión, detección de objetos, entre otras.

La capacidad del DL para manejar y procesar grandes cantidades de datos ha permitido avances significativos en diversas áreas como la visión por computadora, el procesamiento del lenguaje natural y la bioinformática, entre otros. Además, el desarrollo de técnicas como la retropropagación, que facilita el ajuste de los pesos en la red, y el aumento de la capacidad computacional disponible a través de unidades de procesamiento gráfico (GPUs), han sido cruciales para el éxito de las redes neuronales profundas en aplicaciones del mundo real. El poder de generalización de estas redes es notable, permitiéndoles no solo aprender patrones complejos en los datos, sino también aplicar ese conocimiento aprendido a situaciones nuevas y no vistas anteriormente (3).

La regularización es una técnica crucial en la construcción de redes neuronales profundas, ya que previene el sobreajuste (overfitting) y mejora la capacidad de generalización de la red. Los métodos de regularización L1 y L2 añaden términos de penalización en la función de pérdida para controlar la magnitud de los pesos. L1 favorece soluciones esparsas, donde muchos pesos son cero, mientras que L2 distribuye los valores de los pesos de manera más uniforme.

Otra técnica de regularización es el dropout, que durante el entrenamiento omite aleatoriamente algunas neuronas, forzando a la red a ser más robusta y reduciendo la coadaptación entre neuronas. De esta manera, se mejora la capacidad de generalización y se mitiga el riesgo de sobreajuste.

4.4.3. Preprocesamiento de datos

El preprocesamiento de datos es una etapa crucial en cualquier proyecto de aprendizaje automático. Implica la transformación de los datos brutos en un formato más adecuado para el modelado. Dos técnicas comunes de preprocesamiento son la estandarización (StandardScaler) y la normalización (Normalizer).

StandardScaler

El StandardScaler, también conocido como estandarización Z-score, es una técnica de preprocesamiento que transforma los datos de manera que tengan una media de 0 y una desviación estándar de 1. Esta transformación se aplica independientemente a cada característica.

Funcionamiento Para cada característica (x), el StandardScaler realiza la siguiente transformación:

$$z = \frac{x - \mu}{\sigma}$$

Donde:

- z es el valor estandarizado
- x es el valor original

- μ es la media de la característica
- σ es la desviación estándar de la característica

Ventajas

- Útil cuando las características tienen diferentes escalas o unidades.
- Ayuda a que algoritmos como los basados en gradiente converjan más rápidamente.
- Necesario para muchos algoritmos que asumen que los datos están centrados alrededor de cero y tienen varianza unitaria.

Consideraciones

- No cambia la forma de la distribución original.
- Sensible a valores atípicos.
- No escala los datos a un rango específico, lo que puede ser necesario para algunos algoritmos.

Normalizer

El Normalizer escala cada muestra (fila) de manera independiente para tener una norma unitaria (generalmente la norma L2). Este proceso es también conocido como normalización de vectores.

Funcionamiento Para cada muestra $\mathbf{x} = (x_1, \ldots, x_n)$, el Normalizer realiza la siguiente transformación:

$$\mathbf{x}_{\text{norm}} = \frac{\mathbf{x}}{|\mathbf{x}|_2} = \frac{\mathbf{x}}{\sqrt{\sum_{i=1}^{n} x_i^2}}$$

Donde $|\mathbf{x}|_2$ es la norma L2 del vector $\mathbf{x}$.

Ventajas

- Útil cuando solo la dirección (y no la magnitud) de los vectores es importante.
- Comúnmente utilizado en procesamiento de texto y análisis de documentos.
- Puede ayudar en la comparación de muestras con diferentes escalas.

Consideraciones

- No preserva las relaciones de magnitud entre las características dentro de una muestra.
- Puede no ser apropiado cuando las magnitudes absolutas son importantes.
- Puede amplificar el ruido en vectores con valores pequeños.

MaxAbsScaler

MaxAbsScaler es una técnica de escalado que transforma cada característica dividiendo por el valor máximo absoluto de esa característica. Esto escala los datos al rango [-1, 1].

Funcionamiento

Funcionamiento Para cada característica x, MaxAbsScaler realiza la siguiente transformación:

$$x_{\text{scaled}} = \frac{x}{\text{máx}(|x|)}$$

Donde máx($|x|$) es el valor máximo absoluto de la característica x en el conjunto de datos.

Ventajas

- Preserva el cero en los datos sparse (dispersos).
- No desplaza/centra los datos, lo que puede ser útil para datos sparse.
- Útil cuando los datos ya están centrados en cero o cuando se quiere preservar la estructura sparse.

Consideraciones

- No reduce la importancia de los outliers tanto como otras técnicas de escalado.
- Puede no ser adecuado cuando las características tienen diferentes escalas de magnitud.

OneHotEncoder

OneHotEncoder es una técnica utilizada para convertir variables categóricas en una forma que pueda ser proporcionada a algoritmos de ML que esperan datos numéricos.

Funcionamiento Para cada categoría en una variable categórica, OneHotEncoder crea una nueva columna binaria (0 o 1) que indica la presencia o ausencia de esa categoría. Por ejemplo, si tenemos una variable "Color" con categorías "Rojo", "Verde", "Azul", se transformaría en:

Color_Rojo	Color_Verde	Color_Azul
1	0	0
0	1	0
0	0	1

Ventajas

- Elimina cualquier orden implícito en las categorías originales.
- Permite que los algoritmos traten cada categoría de manera independiente.
- Necesario para muchos algoritmos que no pueden manejar directamente variables categóricas.

Consideraciones

- Puede aumentar significativamente la dimensionalidad de los datos, especialmente con variables categóricas de alta cardinalidad.
- Puede llevar a la "maldición de la dimensionalidad"si se usa indiscriminadamente.
- No es adecuado para variables ordinales donde el orden es importante.

OrdinalEncoder

OrdinalEncoder se utiliza para codificar variables categóricas como valores enteros ordenados. Es especialmente útil cuando existe un orden natural en las categorías.

Funcionamiento OrdinalEncoder asigna un valor entero único a cada categoría. Por defecto, los valores se asignan según el orden alfabético de las categorías, pero se puede especificar un orden personalizado. Por ejemplo, para una variable "Tamaño" con categorías "Pequeño", "Mediano", "Grande", podría codificarse como:

Tamaño	Tamaño_Codificado
Pequeño	0
Mediano	1
Grande	2

Ventajas

- Mantiene el orden de las categorías, lo cual es importante para variables ordinales.
- No aumenta la dimensionalidad de los datos.
- Útil para algoritmos que pueden manejar variables ordinales directamente.

Consideraciones

- Introduce una relación ordinal que puede no existir en los datos originales.
- Puede llevar a interpretaciones incorrectas si se usa con variables nominales (sin orden intrínseco).
- Los algoritmos pueden interpretar erróneamente la distancia entre categorías como significativa.

4.4.4. SHAP (SHapley Additive exPlanations)

SHAP (SHapley Additive exPlanations) es un método basado en la teoría de juegos para explicar las predicciones de cualquier modelo de aprendizaje automático. Esta técnica asigna a cada característica un valor de importancia para una predicción particular, basándose en los conceptos de los valores Shapley de la teoría de juegos cooperativos.

Fundamentos teóricos

Los valores SHAP se basan en tres conceptos principales:

- **Valores Shapley**: Provienen de la teoría de juegos y representan la contribución marginal promedio de una característica a través de todas las posibles combinaciones de características.
- **Explicaciones locales**: SHAP puede explicar predicciones individuales, mostrando cómo cada característica contribuye a alejar la predicción del valor base (predicción promedio del modelo).
- **Propiedades aditivas**: La suma de los valores SHAP de todas las características explica la diferencia entre la predicción actual y la predicción base.

Cálculo de valores SHAP

El valor SHAP para una característica i se calcula como:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_x(S \cup \{i\}) - f_x(S)]$$

Donde:

- F es el conjunto de todas las características
- S es un subconjunto de características
- $f_x(S)$ es la predicción del modelo para el conjunto de características S

Tipos de visualizaciones SHAP

SHAP ofrece varias visualizaciones útiles para la interpretación del modelo:

- **Summary Plot**: Muestra la importancia global de las características y su impacto (positivo o negativo) en las predicciones.
- **Dependence Plot**: Visualiza cómo una característica específica afecta las predicciones del modelo, considerando las interacciones con otras características.
- **Force Plot**: Muestra cómo cada característica contribuye a empujar la predicción desde el valor base hasta el valor final.
- **Waterfall Plot**: Desglosa una predicción individual, mostrando cómo cada característica contribuye al valor final.

Ventajas y limitaciones

Ventajas:

- Proporciona explicaciones consistentes y teóricamente fundamentadas
- Puede aplicarse a cualquier modelo de machine learning

- Ofrece tanto interpretabilidad local como global
- Considera las interacciones entre características

Limitaciones:

- Alto costo computacional para modelos complejos
- Puede ser difícil de interpretar cuando hay muchas características
- Las explicaciones pueden ser sensibles a la selección del conjunto de datos base

Interpretación de resultados SHAP

Los gráficos generados por SHAP proporcionaron insights valiosos sobre el modelo:

- **Importancia global de características:** El gráfico de resumen SHAP mostró qué características tenían el mayor impacto en las predicciones del modelo a nivel global.
- **Impacto de características individuales:** Los gráficos de dependencia revelaron cómo cambios en valores específicos de una característica afectaban las predicciones del modelo.
- **Interacciones entre características:** SHAP también permitió identificar interacciones complejas entre diferentes características y su impacto conjunto en las predicciones.

4.5. Azure Machine Learning y Automated Machine Learning (AutoML)

Azure Machine Learning se ha consolidado como una plataforma líder en la nube para el desarrollo, entrenamiento y despliegue de modelos de aprendizaje automático. Una de sus características más innovadoras es el *Automated Machine Learning* (AutoML), una tecnología que automatiza el proceso de selección, entrenamiento y ajuste de modelos de *machine learning*. Su principal objetivo es democratizar el acceso a técnicas avanzadas de aprendizaje automático, optimizando el rendimiento de los modelos sin requerir una amplia experiencia en ciencia de datos.

4.5.1. Funcionamiento de AutoML en Azure

El proceso de AutoML en Azure sigue un enfoque metódico y sistemático para encontrar el modelo más óptimo. A continuación se describen los pasos principales:

1. **Preprocesamiento de datos**: AutoML realiza un exhaustivo preprocesamiento de los datos. Esto incluye imputación de valores faltantes, codificación de variables categóricas y normalización de características. Estos pasos son cruciales para garantizar que los datos estén en un formato adecuado para el modelado.
2. **Exploración de algoritmos**: Una vez procesados los datos, AutoML explora una amplia gama de algoritmos, que incluyen desde la regresión lineal hasta técnicas avanzadas como los bosques aleatorios, *gradient boosting* (LightGBM, XGBoost) e incluso redes neuronales. Esto asegura que se consideren múltiples enfoques para el problema.

3. **Ajuste de hiperparámetros**: AutoML ajusta los hiperparámetros de cada modelo utilizando técnicas como la optimización bayesiana. Este ajuste, que tradicionalmente es una tarea laboriosa, se realiza de forma automatizada y eficiente.

4. **Validación y ensamblaje**: Para garantizar la robustez y la generalización de los modelos, se utilizan técnicas de validación cruzada. Además, AutoML explora combinaciones de modelos a través de técnicas de ensamblaje para mejorar el rendimiento.

5. **Selección de modelos**: Finalmente, AutoML selecciona el modelo (o conjunto de modelos) que ofrezca el mejor desempeño según métricas como el *Error Cuadrático Medio* (MSE) o el *Coeficiente de Determinación* (R^2).

4.5.2. Ventajas y limitaciones de AutoML

Entre las ventajas más destacadas de AutoML se encuentran:

- **Eficiencia**: Automatiza tareas que suelen ser laboriosas y que requieren mucho tiempo, ahorrando recursos.
- **Exploración exhaustiva**: Evalúa una amplia gama de algoritmos y configuraciones, lo que difícilmente podría lograrse manualmente.
- **Optimización avanzada de hiperparámetros**: Simplifica uno de los aspectos más complejos del aprendizaje automático.
- **Escalabilidad**: Puede manejar grandes volúmenes de datos y ejecutar múltiples experimentos en paralelo.

Sin embargo, también presenta algunas limitaciones:

- **Naturaleza de çaja negra"**: Puede ser difícil entender cómo AutoML toma decisiones durante el proceso de selección y ajuste de modelos, lo que puede afectar la interpretabilidad.
- **Sobreadaptación**: Existe el riesgo de sobreajuste si no se supervisa adecuadamente el proceso.
- **Costos computacionales**: Los experimentos exhaustivos pueden ser computacionalmente intensivos, incrementando los costos.

4.6. MLOps y herramientas de implementación

4.6.1. Machine Learning Operations (MLOps)

MLOps representa la aplicación de los principios de DevOps al ciclo de vida del Machine Learning, facilitando la automatización y monitoreo de todos los pasos del desarrollo de modelos de ML: desde la integración y pruebas hasta el despliegue y la gestión de infraestructura.

Componentes principales de MLOps

- **Integración Continua (CI)**: Automatiza la integración de código y componentes de ML.
- **Entrega Continua (CD)**: Automatiza el proceso de despliegue de modelos.

- **Control de Versiones**: Mantiene un registro de cambios en código, datos y modelos.
- **Monitoreo**: Supervisa el rendimiento y la salud de los modelos en producción.
- **Reproducibilidad**: Garantiza que los experimentos y resultados sean reproducibles.

4.6.2. Flask

Flask es un framework web ligero y flexible para Python, diseñado para crear aplicaciones web y APIs de manera rápida y eficiente.

Características principales

- **Minimalista**: Proporciona los componentes básicos necesarios, permitiendo extensiones según se requiera.
- **Enrutamiento URL**: Facilita la definición de endpoints y el manejo de solicitudes HTTP.
- **Compatible con REST**: Ideal para crear APIs RESTful.
- **Integración con ML**: Se integra fácilmente con modelos de machine learning para servir predicciones.

Ventajas en MLOps

- Facilita la creación de APIs para servir modelos de ML.
- Permite el manejo eficiente de solicitudes y respuestas.
- Ofrece flexibilidad para implementar lógica personalizada.
- Proporciona herramientas para el manejo de errores y logging.

4.6.3. Docker

Docker es una plataforma de containerización que permite empaquetar aplicaciones y sus dependencias en contenedores estandarizados.

Conceptos fundamentales

- **Contenedores**: Unidades estandarizadas de software que empaquetan código y dependencias.
- **Imágenes**: Plantillas inmutables para crear contenedores.
- **Dockerfile**: Script que define cómo construir una imagen.
- **Registry**: Repositorio para almacenar y distribuir imágenes.

Beneficios en MLOps

- **Portabilidad**: Los contenedores pueden ejecutarse en cualquier entorno que soporte Docker.
- **Reproducibilidad**: Garantiza que el entorno de ejecución sea idéntico en desarrollo y producción.
- **Aislamiento**: Cada contenedor opera de manera independiente, evitando conflictos.
- **Escalabilidad**: Facilita el despliegue y la gestión de múltiples instancias.

4.6.4. Integración en MLOps

La combinación de Flask y Docker en un pipeline de MLOps proporciona una solución robusta para el despliegue de modelos de machine learning:

- **Desarrollo**: Los modelos se desarrollan y prueban localmente.
- **Empaquetado**: El modelo y la API Flask se empaquetan en un contenedor Docker.
- **Despliegue**: El contenedor se despliega en el entorno de producción.
- **Monitoreo**: Se supervisa el rendimiento y la salud del modelo.

4.6.5. Beneficios del enfoque MLOps

Este enfoque de CI/CD para MLOps ofrece varios beneficios clave:

- **Automatización**: Reduce la intervención manual y los errores asociados.
- **Escalabilidad**: Puede manejar volúmenes crecientes de datos y solicitudes.
- **Consistencia**: Asegura que todos los usuarios accedan a los mismos datos actualizados.
- **Flexibilidad**: Permite fácil integración con diversas aplicaciones y sistemas.
- **Monitoreo**: Facilita el seguimiento del rendimiento del modelo y la calidad de las predicciones.

4.7. Análisis exploratorio de datos

El análisis exploratorio de datos (EDA, por sus siglas en inglés) es una aproximación al análisis de conjuntos de datos que resume sus características principales, a menudo con métodos visuales y estadísticos. Este proceso es fundamental para comprender la estructura y características de los datos antes de aplicar técnicas de modelado más complejas.

4.7.1. Identificación de tipos de variables

La clasificación adecuada de las variables es el primer paso crucial en el análisis exploratorio de datos, ya que determina los métodos estadísticos apropiados para su análisis.

Variables cuantitativas

Las variables cuantitativas son aquellas que pueden medirse numéricamente y se dividen en dos categorías principales:

- **Variables continuas:** Pueden tomar cualquier valor dentro de un rango específico. Por ejemplo:
  - Rendimiento en toneladas por hectárea
  - Temperatura
  - Precipitación
  - Índices de vegetación (NDVI)
- **Variables discretas:** Solo pueden tomar valores enteros específicos. Por ejemplo:
  - Número de cortes de caña
  - Conteo de días de cultivo
  - Edad del cultivo en días

Variables cualitativas

Las variables cualitativas o categóricas describen características o cualidades y se clasifican en:

- **Variables nominales:** No tienen un orden natural. Por ejemplo:
  - Variedad de caña
  - Sistema de riego
  - Tipo de suelo
- **Variables ordinales:** Presentan un orden natural. Por ejemplo:
  - Etapas de crecimiento
  - Categorías de rendimiento (bajo, medio, alto)
  - Meses del año

4.7.2. Análisis de normalidad

El análisis de normalidad es crucial para determinar si los datos siguen una distribución normal, lo cual es un supuesto importante para muchas técnicas estadísticas.

Métodos de evaluación de normalidad

- **Métodos gráficos:**
  - *Histogramas:* Proporcionan una visualización de la forma de la distribución.
  - *Gráficos Q-Q:* Comparan los cuantiles de los datos con los cuantiles teóricos de una distribución normal.
  - *Box plots:* Identifican la simetría y los valores atípicos.

- **Pruebas estadísticas:**
  - *Prueba de Lilliefors:* Una adaptación de la prueba Kolmogorov-Smirnov.
  - *Prueba de Shapiro-Wilk:* Especialmente efectiva para muestras pequeñas.
  - *Test de Anderson-Darling:* Enfatiza las colas de la distribución.

4.7.3. Análisis de distribuciones categóricas

Para variables cualitativas, el análisis se centra en la frecuencia y proporción de las diferentes categorías.

- **Técnicas de visualización:**
  - Gráficos de barras
  - Diagramas de sectores
  - Diagramas de Pareto
- **Medidas descriptivas:**
  - Frecuencias absolutas y relativas
  - Modas
  - Razones y proporciones

4.7.4. Análisis de correlación

El análisis de correlación examina la relación entre variables y es fundamental para identificar patrones y dependencias en los datos.

- **Métodos de correlación:**
  - *Correlación de Pearson:* Para relaciones lineales entre variables continuas.
  - *Correlación de Spearman:* Para relaciones monótonas, especialmente útil con variables ordinales.
  - *Correlación de Kendall:* Alternativa robusta para datos no paramétricos.
- **Visualizaciones:**
  - Matrices de correlación
  - Diagramas de dispersión
  - Heat maps

CAPÍTULO 5

Alcance

El presente proyecto se enfoca en el desarrollo de un sistema predictivo del tonelaje de caña por hectárea (TCH) para el Ingenio Pantaleon, utilizando datos históricos que abarcan el período 2019-2024. El sistema integra tres fuentes principales de datos:

- Datos climatológicos proporcionados por las estaciones meteorológicas del Instituto de Cambio Climático (ICC)
- Índices vegetativos derivados de imágenes satelitales procesadas por NAX Solutions
- Registros históricos de cosecha y manejo del Ingenio Pantaleon

El sistema desarrollado tiene la capacidad de generar predicciones del TCH con diferentes horizontes temporales (2, 4, 6, 8 y 10 meses antes de la cosecha), permitiendo una planificación más efectiva de las operaciones de campo. Sin embargo, es importante señalar que la precisión de las predicciones puede variar según el horizonte temporal y está sujeta a la calidad y disponibilidad de los datos de entrada. El proyecto incluye el desarrollo de una API REST implementada en Flask, que permite la automatización del pipeline de datos y la integración con los sistemas existentes del ingenio. Esta API facilita:

- La actualización automática mensual de datos climatológicos e índices vegetativos
- El reentrenamiento programado de los modelos predictivos
- El acceso a las predicciones a través de endpoints REST estandarizados
- La generación de visualizaciones y reportes de rendimiento

Es importante destacar que el sistema está diseñado específicamente para las condiciones y características particulares de los cultivos del Ingenio Pantaleon, por lo que su aplicabilidad a otros ingenios o regiones podría requerir ajustes significativos. Además, la precisión del sistema depende de la continuidad en el suministro de datos de las fuentes mencionadas y del mantenimiento adecuado de la infraestructura de MLOps implementada. El alcance incluye la implementación de un ciclo completo de MLOps, pero se limita a la infraestructura y recursos disponibles en Azure, donde se realiza el entrenamiento y despliegue de los modelos. La interpretabilidad de los modelos se proporciona a través de análisis SHAP, permitiendo entender la importancia relativa de las diferentes variables en las predicciones del TCH.

CAPÍTULO 6

Metodología

6.1. Creación del conjunto de datos

El conjunto de datos para este estudio se compone de tres fuentes principales, abarcando un período histórico de 5 años, desde 2019 hasta 2024. Estas fuentes proporcionan una visión integral del proceso de cultivo de caña de azúcar, incluyendo datos de manejo de zafra, información climatológica e índices vegetales derivados de imágenes satelitales.

6.1.1. Recolección de datos

Datos de manejo de zafra

Se estableció una colaboración estrecha con el Ingenio Pantaleon para acceder a sus registros detallados de mediciones de zafra. Esta colaboración implicó una serie de reuniones presenciales con los analistas de datos del ingenio, quienes proporcionaron una explicación exhaustiva del proceso de recolección de caña de azúcar. De particular importancia fue la obtención de los datos de pesaje de cada terreno al final de cada zafra, lo cual proporciona la variable crucial de Toneladas de Caña por Hectárea (TCH).

Datos climatológicos

Para obtener un registro histórico completo de las condiciones climatológicas, se desarrolló un web scraper especializado para extraer datos de la página web del Instituto de Cambio Climático (ICC). Este proceso automatizado permitió la recopilación de información diaria de las estaciones meteorológicas relevantes, cubriendo el período de estudio de 5 años. Los datos extraídos incluyen variables como temperatura, precipitación, humedad y velocidad del viento, entre otros.

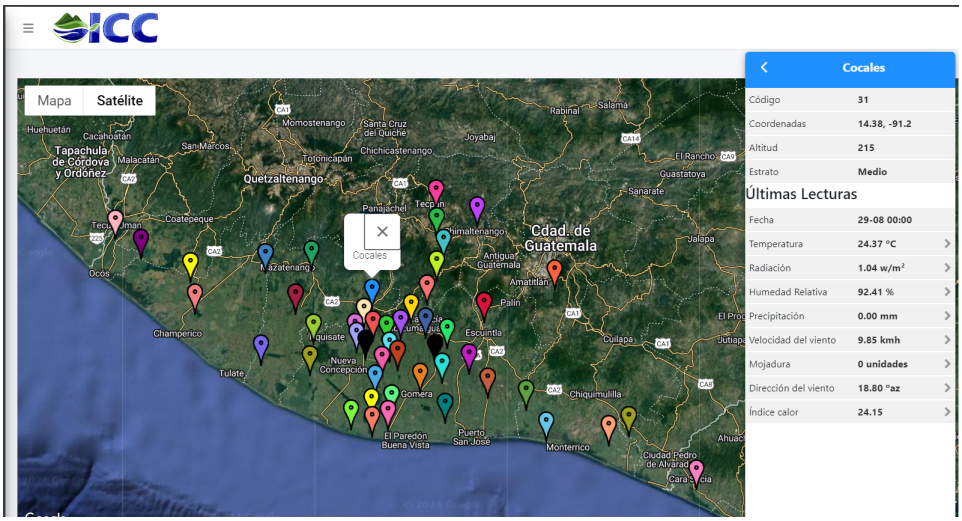


Figura 6.1: Pagina del Instituto de Cambio Climático

Índices vegetales de imágenes satelitales

Gracias a la colaboración con el Ingenio Pantaleon, se obtuvo acceso a la plataforma NAX Solutions, un servicio avanzado de análisis de imágenes satelitales. Utilizando la API proporcionada por NAX Solutions, se desarrolló un sistema automatizado para extraer datos de índices vegetales cada 5 días. Estos índices, que incluyen métricas como el NDVI (Índice de vegetación de diferencia Normalizada), LAI (Índice de área foliar) y otros indicadores de salud vegetal, proporcionan una visión detallada del desarrollo y estado de los cultivos de caña de azúcar a lo largo del tiempo. La integración de estos tres conjuntos de datos - manejo de zafra, climatología e índices vegetales satelitales - proporciona una base de datos rica y multidimensional. Esta combinación permite un análisis comprehensivo de los factores que influyen en el rendimiento de la caña de azúcar, facilitando el desarrollo de modelos predictivos más precisos y robustos

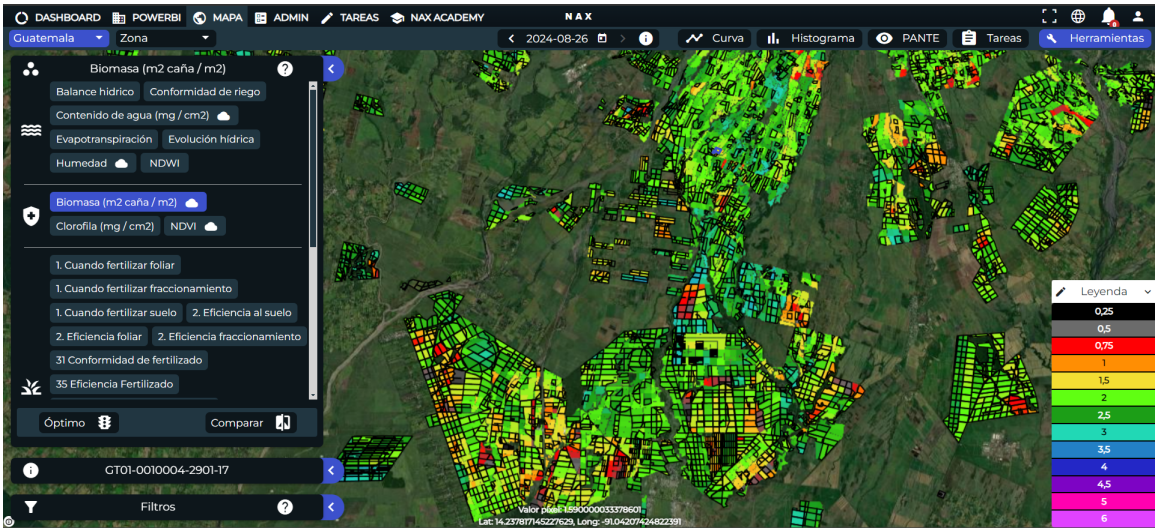


Figura 6.2: Página de NAX Solutions

6.2. Análisis exploratorio de datos

El conjunto de datos está compuesto por 3 bases de datos diferentes. El conjunto de datos principal es el conjunto de datos de las zafra este conjunto se obtiene al final de cada zafra, esta compuesto por 13659 registros y 28 variables, entre las que destacan el tipo de muestra, la zafra, el terreno, la fecha de corte y el TCH de los cultivos. Las variables cuantitativas como el área y el rendimiento varían entre valores mínimos de 0 y 74.92 respectivamente, hasta máximos de 8,071,809 para el área y 177.18 para el rendimiento.

Variable	**Mínimo**	**Media**	**Máximo**
fecha	2019-11-06	2021-10-03	2024-04-13
fi	2018-11-19	2020-10-21	2023-05-21
ff	2019-11-17	2021-10-14	2024-04-25
area	0.1	7484.23	8071809
TCH	50.27	101.35	160.60
edad proyectada	218	358.3	502
edad actual	208	346.6	491
días corte	10	11.69	14
semana corte	1	14.82	30

Tabla 6.1: Resumen estadístico del conjunto de datos de zafra.

El conjunto de datos de los índices tiene 596,751 filas y 12 columnas. Este conjunto incluye variables como el NDVI, el porcentaje de agua, el índice de área foliar (LAI), y la producción ponderada. Este conjunto de datos obtiene un registro nuevo cada 5 dias ya que es dependiente de las imagenes satelitales disponibles. Para unirlo al conjunto de datos de la zafra se utiliza el ID del terreno que esta definido con GIS. A continuación se presenta un resumen estadístico de las principales variables cuantitativas.

Variable	**Mínimo**	**Media**	**Máximo**
FECHA	2019-06-04	2022-02-05	2024-05-23
NDVI_POND	0.000	0.611	0.922
AGUA_POND	0.000	0.035	0.293
LAI_POND	-0.45	1.83	85.17
PRODUCCION_POND	-0.912	0.537	8.665
ANOMALIAS_POND	-6.79	7.15	746.20
HUMEDAD_POND	0.00	76.15	90.11

Tabla 6.2: Resumen estadístico del conjunto de datos de índices de la cosecha.

El conjunto de datos del clima tiene 112,264 filas y 32 columnas. Las principales variables incluyen la evapotranspiración (ETP), la radiación solar, la temperatura, la humedad relativa y la velocidad del viento. Este conjunto de datos se optuvo de las estaciones meteorologiacas del ICC y se unieron al dataset de la zafra verificando cual de las estaciones estaba mas cercana al terreno. Los datos se obtienen de manera diaria de 21 estaciones diferentes. A continuación se presenta un resumen estadístico de las variables cuantitativas más relevantes.

Variable	Mínimo	Media	Máximo
ETP	-4.967	5.374	15.550
Radiación (MJ/m^2)	0.00	18.75	79.98
Amplitud Térmica	0.00	11.28	71.50
Temperatura	-40.00	26.28	34.28
Temperatura Mínima	-40.00	21.44	31.20
Temperatura Máxima	-40.00	32.74	60.00
Humedad Relativa	0.00	84.52	100.00
Precipitación (mm)	0.00	4.861	380.20
Velocidad Viento (m/s)	0.00	4.947	151.18
Presión Atmosférica (hPa)	0.00	989.7	1132.0
Dirección del Viento (grados)	0.0	175.8	1043.1

Tabla 6.3: Resumen estadístico del conjunto de datos del clima.

6.2.1. Limpieza de valores nulos por Dataset

Descripción

Se realizó una limpieza de datos para eliminar variables con un porcentaje considerable de valores nulos. Solo se mantuvieron aquellas variables con un número suficiente de datos válidos, mientras que las que presentaron una gran cantidad de valores faltantes fueron eliminadas del análisis. A continuación, se presenta un resumen de las variables eliminadas y el porcentaje de valores nulos en cada uno de los conjuntos de datos (clima e índices de la caña).

Las siguientes variables presentaron un porcentaje significativo de valores nulos y, por lo tanto, fueron eliminadas las que tuvieran mas del 25 % de faltantes:

Clima

Variable	% de Valores nulos
presion atmosferica máxima	95.399238
presion atmosferica mínima	95.399238
presion atmosferica	95.399238
mojadura	0.457849
temperatura	0.112235
humedad relativa máxima	0.112235
humedad relativa mínima	0.112235
humedad relativa	0.112235
velocidad viento	0.005345
velocidad viento máxima	0.005345
dirección viento	0.004454
velocidad viento mínima	0.004454

Tabla 6.4: Procentaje de nulos en conjunto de datos de clima

Índices de la caña

Variable	% de Valores nulos
ESTATUS_COSECHA	78.006907
MADURACION_POND	48.804610
NITROGENADO_POND	44.073491
ANOMALIAS_POND	26.007330
LAI_POND	12.446732
HUMEDAD_POND	11.029894
PRODUCCION_PON	3.380304
NDVI_POND	1.386005

Tabla 6.5: Procentaje de nulos en conjunto de datos de indices

Datos de la cosecha

Variable	% de Valores nulos
fecha_mad	25.238991
semana_corte	13.586752

Tabla 6.6: Procentaje de nulos en conjunto de datos de cosecha

Tras realizar el análisis de valores nulos, se tomaron decisiones sobre las variables que se eliminaron de cada conjunto de datos. En el conjunto de datos de clima, se eliminaron las variables relacionadas con la presión atmosférica debido a su alto porcentaje de valores nulos (más del 95 %). Además, aunque variables como la temperatura y la humedad relativa presentaron un porcentaje bajo de valores faltantes (0.11 %), también fueron eliminadas para garantizar la consistencia en el análisis.

En el conjunto de datos de Índices de la caña, se eliminaron variables clave como ESTATUS_COSECHA y MADURACION_POND debido a la gran cantidad de datos faltantes (78 % y 48 %, respectivamente). Otras variables, como NITROGENADO_POND y ANOMALIAS_POND, también fueron eliminadas al superar el umbral del 25 % de valores nulos. Sin embargo, se mantuvieron variables como LAI_POND y NDVI_POND, que presentaron un porcentaje manejable de datos faltantes.

Finalmente, en el conjunto de datos de índices de la cosecha, se eliminó la variable fecha_mad debido a que su porcentaje de valores nulos (25.24 %) superó el límite establecido, mientras que semana_corte se mantuvo ya que presentaba un 13.59 % de valores nulos, por debajo del umbral del 25 %.

Estas decisiones se tomaron para garantizar que los análisis posteriores se realicen con variables que contengan datos suficientes y sean representativas del comportamiento de los fenómenos observados.

6.2.2. Identificación de variables

- **Descripción:** Se realizó una clasificación de las variables en los diferentes conjuntos de datos utilizados (cosecha, índices de la caña, clima) para entender los datos y encontrar una forma de agruparlos para el modelo.

A continuación, se detallan los tipos de variables de cada uno de los tres conjuntos de datos utilizados.

Datos de cosecha

- **Variables cuantitativas**
  - **Continuas:** Área del terreno, rendimiento.
  - **Discretas:** Edad proyectada, edad actual de la zafra, número de cortes, días de corte, semanas de corte.
- **Variables cualitativas**
  - **Nominales:** Tipo de muestra, sistema de riego, tipo de cosecha, estrato, región, variedad de caña.
  - **Ordinales:** Año de la zafra, Fecha de inicio y fin, mes de muestra, mes de corte.

Datos de Índices de la caña

- **Variables cuantitativas**
  - **Continuas:** Índice NDVI, índice de contenido de agua, índice de área foliar, índice de producción, índice de anomalías, índice de nitrógeno, índice de maduración, índice de humedad.
- **Variables cualitativas**
  - **Nominales:** ID de campo, estado de la cosecha.
  - **Ordinales:** Fecha de muestra.

Datos del clima

- **Variables cuantitativas**
  - **Continuas:** Evapotranspiración, radiación solar, amplitud térmica, temperatura, temperatura mínima y máxima, humedad relativa, precipitación, velocidad del viento, presión atmosférica, dirección del viento.
- **Variables cualitativas**
  - **Nominales:** Cuadrante, estrato, región, estación meteorológica.
  - **Ordinales:** Año, mes, día de la muestra.

6.2.3. Análisis de normalidad por dataset

- **Descripción:** La evaluación de la normalidad es fundamental para determinar la aplicabilidad de métodos estadísticos paramétricos y para identificar posibles transformaciones necesarias en los datos antes del modelado.
- **Pruebas realizadas:**
  - **Histograma:** Se utilizó como primera herramienta visual para examinar la forma de la distribución de los datos, permitiendo identificar rápidamente patrones como asimetría, multimodalidad y la presencia de valores atípicos. Esta visualización proporciona una comprensión inicial de cómo se distribuyen los valores en cada variable.

- **Q-Q Plot:** Se implementó para comparar cuantitativamente la distribución de nuestros datos contra una distribución normal teórica. Esta herramienta es especialmente útil para detectar desviaciones de la normalidad en los extremos de la distribución y para identificar patrones específicos de no normalidad, como colas pesadas o asimetría.
- **Prueba de Lilliefors:** Se aplicó como prueba estadística formal para cuantificar la evidencia contra la hipótesis de normalidad. Esta prueba fue seleccionada por ser una modificación de la prueba Kolmogorov-Smirnov que no requiere especificar los parámetros de la distribución normal teórica, haciéndola más apropiada para nuestro caso donde estos parámetros deben estimarse a partir de los datos.
- **Transformaciones:** En casos donde se detectó una desviación significativa de la normalidad, se exploraron transformaciones matemáticas (logarítmica y raíz cuadrada) para aproximar los datos a una distribución normal. La selección de estas transformaciones específicas se basó en su efectividad conocida para diferentes tipos de asimetrías y su interpretabilidad en el contexto agrícola.

- **Justificación del enfoque:** Este conjunto completo de pruebas fue diseñado para proporcionar tanto evidencia visual como estadística sobre la normalidad de los datos, permitiendo una evaluación robusta y múltiples perspectivas sobre la distribución de cada variable. La combinación de métodos visuales y estadísticos formales permite una evaluación más confiable y completa de la normalidad, fundamental para las decisiones posteriores sobre el modelado y análisis de los datos.

6.2.4. Análisis de datos cualitativos

Histograma de Frecuencias: Se utilizaron histogramas para visualizar la distribución de las principales categorías en las variables cualitativas. Estas visualizaciones son fundamentales para identificar las características predominantes y las frecuencias relativas de las categorías dentro del conjunto de datos.

6.3. Procesamiento de datos

El proceso de unificación de los tres conjuntos de datos (cosecha, clima e índices) se realizó mediante un enfoque de agregación temporal y espacial. Este proceso fue crucial para alinear los datos con diferentes frecuencias temporales y referencias espaciales en un único conjunto de datos coherente para el análisis posterior.

6.3.1. Alineación temporal

Para cada registro en el conjunto de datos de cosecha (anual), se definieron dos fechas clave:

- **Fecha de inicio (fi):** La fecha de inicio del ciclo de cultivo.
- **Fecha de predicción:** Calculada como 180 días (aproximadamente 6 meses) antes de la fecha de fin del ciclo (ff).

Utilizando estas fechas, se filtró la información climática y de índices relevante para cada ciclo de cultivo.

6.3.2. Agregación de datos climáticos

Los datos climáticos diarios se procesaron de la siguiente manera:

- Se filtraron los datos correspondientes al período entre la fecha de inicio y la fecha de predicción.
- Se calcularon estadísticas para tres períodos: Q1 (primeros 2 meses), Q2 (primeros 4 meses) y el período completo de 6 meses.
- Para cada variable climática, se computaron las siguientes métricas:
  - Media, suma y desviación estándar para Q1, Q2 y 6 meses.
  - Integral (usando el método del trapecio) para Q1, Q2 y 6 meses.

6.3.3. Procesamiento de índices

Los datos de índices, con frecuencia semanal, se trataron de manera similar:

- Se filtraron los datos relevantes para cada ciclo de cultivo.
- Se calcularon estadísticas para Q1, Q2 y el período completo de 6 meses.
- Para cada índice, se computaron:
  - Media, suma y desviación estándar para Q1, Q2 y 6 meses.
  - Integral para Q1, Q2 y 6 meses.

6.3.4. Unificación de datos

El proceso de unificación combinó:

- Datos originales de la cosecha (rendimiento, TCH, área, etc.).
- Estadísticas agregadas de datos climáticos.
- Estadísticas agregadas de índices.

La alineación espacial se logró utilizando el identificador de cuadrante para los datos climáticos y el identificador de terreno (ABS_IDCOMP) para los índices.

6.3.5. Conjunto de datos final

El conjunto de datos resultante incluye:

- Variables originales de la cosecha (rendimiento, TCH, área, etc.).
- Variables categóricas (estación, variedad, sistema de riego, tipo de cosecha, región, estrato, etc.).
- Estadísticas agregadas de variables climáticas e índices para diferentes períodos temporales.

Este proceso de unificación permite analizar cómo las condiciones climáticas y los índices de vegetación a lo largo del ciclo de cultivo influyen en el rendimiento final de la cosecha, proporcionando una base sólida para el análisis predictivo y la modelización.

6.3.6. Normalización de datos

Después de la unificación de los datos, se aplicó un proceso de normalización para estandarizar ciertas variables:

- Se identificaron las columnas que contenían sumas ('_sum') e integrales ('_integral').
- Se utilizó MinMaxScaler para normalizar estos valores en un rango de 0 a 1.
- La normalización se aplicó in-place, actualizando los valores en el conjunto de datos final.

Este paso es crucial para asegurar que todas las variables estén en una escala comparable, lo que es particularmente importante para muchos algoritmos de aprendizaje automático.

6.3.7. Codificación de variables categóricas

Para preparar las variables categóricas para el análisis de aprendizaje automático, se aplicaron diferentes técnicas de codificación:

1. **One-Hot Encoding:** Se aplicó a variables nominales con pocas categorías:
   - Variables: 'prod_mad', 'sist_riego', 'tipo_cosecha', 'PRODUCTO_ACTUAL'
   - Crea nuevas columnas binarias para cada categoría.
2. **Ordinal Encoding:** Se utilizó para variables ordinales:
   - Variables: 'region', 'estrato'
   - Asigna valores numéricos manteniendo el orden de las categorías.
3. **Label Encoding:** Se aplicó a variables nominales con muchas categorías:
   - Variables: 'estacion', 'variedad', 'cuadrante'
   - Asigna un valor numérico único a cada categoría.

Después de la codificación, se realizaron los siguientes pasos adicionales:

- Se combinaron los datos codificados con las variables numéricas originales.
- Se manejaron los valores faltantes en columnas numéricas utilizando SimpleImputer con la estrategia de media.
- Se eliminó la columna 'rendimiento' del conjunto de datos codificado.

El conjunto de datos resultante se guardó como un nuevo archivo CSV, listo para su uso en tareas de modelado predictivo.

Esta etapa de procesamiento asegura que todas las variables, tanto numéricas como categóricas, estén en un formato adecuado para su uso en algoritmos de aprendizaje automático, manteniendo la integridad de la información original mientras se optimiza para el análisis computacional.

6.4. Entrenamiento de modelos

6.4.1. Configuración del entorno

- **Framework utilizado:** PyTorch con soporte CUDA.
- **Configuración global:** Se estableció un ambiente global con una semilla para asegurar la reproducibilidad de los resultados. Para las librerias de torch, pandas, random, numpy y cuda.
  - Sistema operativo: Windows 11 X86_64
  - Procesador: Intel(R) Core(TM) i9-12900H de 12ava generación
  - GPU: Nvidia GeForce RTX 3070 Ti con 16GB de VRAM
  - Memoria RAM: 32GB
  - Almacenamiento: 200 GB de espacio en disco duro
- **Especificaciones de nube:** Para el desarrollo de los modelos de aprendizaje automático, se utilizó Azure con las siguientes configuraciones:
  - Plataforma: Azure Machine Learning Studio
  - Tipo de máquina virtual: CPU Dedicado
  - Tamaño de máquina: Standard_DS3_v2
  - Recursos: 4 núcleos, 14 GB RAM, 28 GB de almacenamiento
  - Modo de ejecución: Sin Servidor

6.4.2. Preparación del conjunto de datos

Regresión:

- **Variable objetivo:** Tonelaje de caña por hectárea (TCH).
- **División temporal de datos:**
  - **Conjunto de entrenamiento:** Todas las zafras excepto la zafra '23-24'.
  - **Conjunto de drueba:** La zafra '23-24' se utilizó como dataset de test.
- **Preprocesamiento:**
  - Se eliminaron las columnas 'ABS_IDCOMP' y 'ZAFRA' de los conjuntos de entrenamiento y prueba, ya que estas variables sólo se utilizarán para identificar los terrenos después del entrenamiento o para hacer predicciones futuras.
  - La variable objetivo (TCH) se separó de las características en ambos conjuntos.

Clasificación:

- **Variable objetivo:** Grupos de tonelaje de caña por hectárea (TCH_grupo), creados a partir del TCH.
- **Creación de grupos de TCH:**
  - Se crearon intervalos de TCH utilizando el siguiente proceso:
    - Se definieron los límites mínimos y máximos del TCH.

- Se crearon bins de 2 unidades de TCH, desde el valor mínimo hasta el máximo.
- Se asignaron etiquetas a cada grupo en el formato 'i-i+2', donde 'i' representa el valor mínimo del grupo.
- La variable resultante se denominó 'TCH_grupo'.

- **División temporal de datos:**
  - **Conjunto de entrenamiento:** Todas las zafras excepto la zafra '23-24'.
  - **Conjunto de prueba:** La zafra '23-24' se utilizó como dataset de test.
- **Preprocesamiento:**
  - Se eliminaron las columnas 'ABS_IDCOMP' y 'ZAFRA' de los conjuntos de entrenamiento y prueba.
  - La variable de grupos de TCH ('TCH_grupo') se separó de las características en ambos conjuntos.

6.4.3. Modelos entrenados

Modelos de redes neuronales

Se implementaron y entrenaron dos modelos de redes neuronales Fully Connected para las tareas de regresión y clasificación. A continuación, se detalla la arquitectura utilizada para ambos modelos.

Modelo de regresión:

El modelo utilizado para predecir el tonelaje de caña por hectárea (TCH) es una red neuronal completamente conectada con 16 capas lineales. La arquitectura final se muestra en la Figura 7.22.

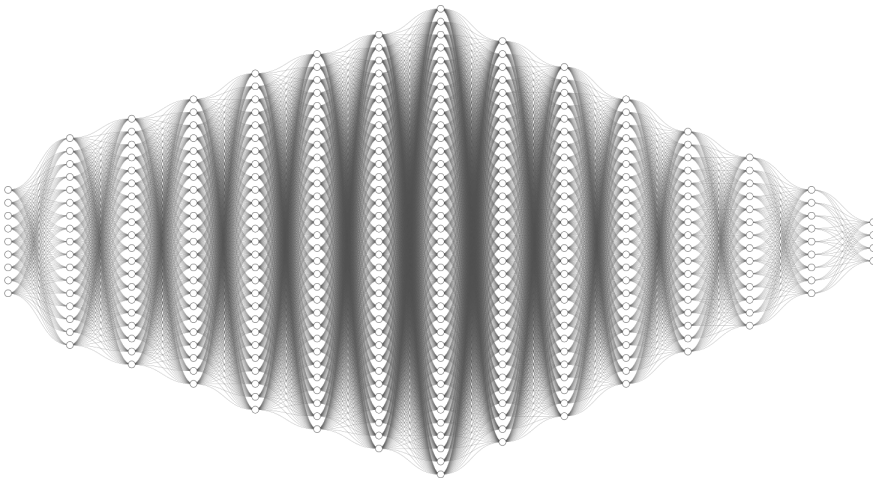


Figura 6.3: Arquitectura de la red neuronal para regresión

El modelo presenta las siguientes características:

- **Capas ascendentes y descendentes:** El número total de capas es 16, con una arquitectura de .ªscenso-descenso.en la que el tamaño de las capas aumenta progresivamente hasta un pico y luego disminuye hacia la capa de salida.
- **Activaciones:** Se utilizaron diferentes funciones de activación: `Sigmoid`, `LeakyReLU` y `ReLU`, dependiendo de la posición de la capa en la red.
- **Capa de salida:** Una única neurona sin función de activación para la predicción continua de la variable objetivo (TCH).

Técnicas de regularización

Para mejorar la generalización del modelo y prevenir el sobreajuste, se aplicaron las siguientes técnicas de regularización:

- **Dropout:** Se aplicó un *dropout* del 25 % cada dos capas ocultas, previniendo la coadaptación de las neuronas y mejorando la robustez del modelo.
- **Regularización L1 y L2:** Se implementaron penalizaciones L1 y L2 en los pesos del modelo, controladas mediante los hiperparámetros `l1_lambda` y `l2_lambda`, para reducir la magnitud de los pesos y evitar la complejidad excesiva.
- **Early Stopping:** El entrenamiento se detiene cuando el rendimiento en el conjunto de validación deja de mejorar, evitando así el sobreajuste.
- **Learning Rate Scheduler:** Se implementó un programador de tasa de aprendizaje que reduce la tasa en un factor de 0.1 cada 30 épocas, afinando el modelo en las etapas finales del entrenamiento.

Modelo de clasificación:

Para la tarea de clasificación, se implementó una red con la misma arquitectura de 16 capas, pero con un ajuste en la capa de salida para predecir las clases de tonelaje de caña por hectárea. La arquitectura se muestra en la Figura 7.22.

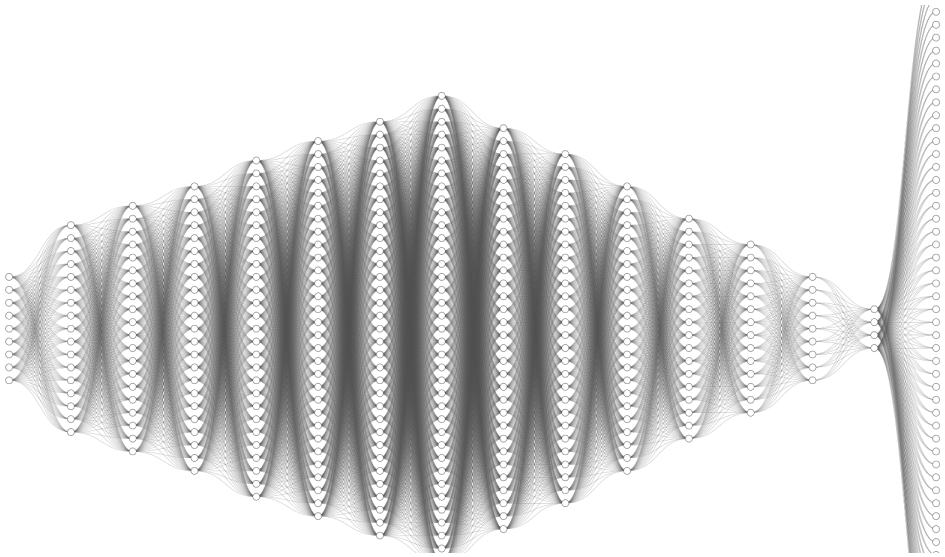


Figura 6.4: Arquitectura de la red neuronal para clasificación

- **Capas ascendentes y descendentes:** El número total de capas es 16, con una arquitectura de .ªscenso-descenso.en la que el tamaño de las capas aumenta progresivamente hasta un pico y luego disminuye hacia la capa de salida.
- **Activaciones:** Se utilizaron diferentes funciones de activación: `LeakyReLU` y `ReLU`, dependiendo de la posición de la capa en la red.
- **Capa de salida:** 56 neuronas, una representando cada grupo del TCH creado, sin función de activación para la predicción continua de la variable objetivo (TCH_grupo).

Técnicas de regularización

Para mejorar la generalización del modelo y prevenir el sobreajuste, se aplicaron las siguientes técnicas de regularización:

- **Dropout:** Se aplicó un *dropout* del 25 % cada dos capas ocultas, previniendo la coadaptación de las neuronas y mejorando la robustez del modelo.
- **Regularización L1 y L2:** Se implementaron penalizaciones L1 y L2 en los pesos del modelo, controladas mediante los hiperparámetros `l1_lambda` y `l2_lambda`, para reducir la magnitud de los pesos y evitar la complejidad excesiva.
- **Early Stopping:** El entrenamiento se detiene cuando el rendimiento en el conjunto de validación deja de mejorar, evitando así el sobreajuste.
- **Learning Rate Scheduler:** Se implementó un programador de tasa de aprendizaje que reduce la tasa en un factor de 0.1 cada 30 épocas, afinando el modelo en las etapas finales del entrenamiento.

Búsqueda de hiperparámetros (GridSearch)

Para optimizar ambos modelos, se llevó a cabo una búsqueda en cuadrícula (GridSearch) sobre diferentes hiperparámetros. A continuación se muestran los detalles de los parámetros evaluados para cada modelo:

Modelo	Hiperparámetros evaluados
Clasificación	`learning_rate`: [0.01, 0.001] `l1_lambda`: [0.0, 0.001] `l2_lambda`: [0.0, 0.001] `batch_size`: [2048] `num_epochs`: [200] `optimizer_type`: [Adam, SGD] `N`: [16]
Regresión	`learning_rate`: [0.1, 0.01, 0.001] `l1_lambda`: [0.0, 0.001] `l2_lambda`: [0.0, 0.001] `batch_size`: [1024] `num_epochs`: [200] `optimizer_type`: [Adam, SGD] `criterion`: [MSE, MAE, SmoothL1Loss] `N`: [16]

Tabla 6.7: Parámetros de GridSearch para los modelos de clasificación y regresión

Donde la N es la cantidad de capas ocultas que tendria la Red Neuronal.

6.4.4. Implementación de AutoML en Azure para Regresión

Para implementar AutoML en Azure para nuestro problema de regresión de tonelaje de caña por hectárea (TCH), seguimos los siguientes pasos:

1. **Configuración del experimento:**

```
from azureml.core import Workspace, Experiment
from azureml.train.automl import AutoMLConfig
# Configurar el espacio de trabajo
ws = Workspace.from_config()

# Definir el experimento
experiment = Experiment(ws, "TCH_Regression_AutoML")

# Configurar AutoML
automl_config = AutoMLConfig(
    task='regression',
    primary_metric='normalized_root_mean_squared_error',
    training_data=train_data,
    label_column_name='TCH',
    n_cross_validations=5,
    max_concurrent_iterations=4,
    max_cores_per_iteration=-1,
    iterations=100,
    experiment_timeout_minutes=60
)
```

2. **Ejecución del experimento:**

```
remote_run = experiment.submit(automl_config)
```

3. **Monitoreo y evaluación:**

```
from azureml.widgets import RunDetails
RunDetails(remote_run).show()

best_run, fitted_model = remote_run.get_output()
print(best_run.get_metrics())
```

Ventajas del uso de AutoML en Azure

La utilización de AutoML en Azure para nuestro problema de regresión ofrece varias ventajas significativas:

- **Eficiencia:** Automatiza el proceso de selección y ajuste de modelos, ahorrando tiempo y recursos.
- **Exploración exhaustiva:** Evalúa una amplia gama de algoritmos y configuraciones que podrían ser pasados por alto en un enfoque manual.

- **Objetividad:** Reduce el sesgo en la selección de modelos al basarse en métricas de rendimiento cuantificables.
- **Escalabilidad:** Aprovecha los recursos en la nube para realizar experimentos a gran escala.
- **Interpretabilidad:** Proporciona herramientas para entender la importancia de las características y el funcionamiento del modelo seleccionado.

Limitaciones y consideraciones

A pesar de sus numerosas ventajas, es importante reconocer algunas limitaciones del enfoque AutoML:

- **Caja negra:** Puede ser difícil entender completamente el proceso de decisión detrás de la selección del modelo.
- **Consumo de recursos:** Los experimentos de AutoML pueden ser computacionalmente intensivos y costosos en términos de tiempo y recursos en la nube.
- **Dependencia de datos:** La calidad de los resultados sigue dependiendo en gran medida de la calidad y representatividad de los datos de entrada.

6.4.5. Explicabilidad del modelo con SHAP

Tras obtener el mejor modelo a través de AutoML en Azure, se implementó SHAP (SHapley Additive exPlanations) para proporcionar una interpretación detallada de las predicciones del modelo. SHAP es una técnica de explicabilidad de modelos basada en la teoría de juegos que asigna a cada característica un valor de importancia para una predicción particular.

Implementación de SHAP

Para aplicar SHAP al modelo seleccionado por AutoML, se siguieron estos pasos:

1. **Reentrenamiento del modelo:** Aunque AutoML proporcionó el mejor modelo, se reentrenó localmente para obtener acceso completo a los datos de SHAP. En este caso, se utilizó un ensamble de dos modelos XGBoost con diferentes configuraciones de preprocesamiento.
2. **Creación del explainer:** Se creó un objeto explainer de SHAP específico para el modelo XGBoost:

   ```
   explainer = shap.Explainer(pipeline_0.named_steps['model'])
   ```

3. **Cálculo de valores SHAP:** Se calcularon los valores SHAP para el conjunto de prueba:

   ```
   shap_values = explainer(X_test_scaled)
   ```

4. **Visualización de resultados:** Se generaron varios gráficos para interpretar la importancia y el impacto de las características:
   - Gráfico de resumen SHAP
   - Gráfico de barras de importancia de características

- Gráficos de dependencia para características específicas

La combinación de AutoML de Azure para la selección y optimización del modelo, junto con SHAP para la explicabilidad, resultó en un enfoque robusto y transparente para nuestro problema de regresión de tonelaje de caña por hectárea (TCH).

6.5. Ciclo CI/CD para MLOps

En esta sección, se describe el flujo de trabajo automatizado para el procesamiento de datos, predicciones y almacenamiento que se ejecuta mensualmente como parte del pipeline de MLOps.

6.5.1. Visión general del flujo de trabajo

El ciclo CI/CD para MLOps está diseñado para manejar eficientemente la ingesta de datos mensuales, realizar predicciones utilizando modelos pre-entrenados, y almacenar los resultados de manera accesible tanto para análisis internos como para aplicaciones externas. El flujo de trabajo se compone de los siguientes elementos principales:

- Fuentes de datos: ICC CLIMA DATA y NAX INDEX DATA
- Procesamiento y predicción: Azure ML
- Almacenamiento de datos: Azure Blob Storage y CosmosDB
- Despliegue: Contenedor Docker
- Acceso a datos: API Web con Flask

6.5.2. Componentes del sistema

Fuentes de datos

Los datos mensuales provienen de dos fuentes principales:

- **ICC CLIMA DATA**: Proporciona datos climáticos relevantes para las predicciones.
- **NAX INDEX DATA**: Ofrece índices específicos del sector que influyen en el modelo predictivo.

Azure Blob Storage

Se utiliza Azure Blob Storage como repositorio intermedio para los datos de entrenamiento. Este componente permite un almacenamiento escalable y seguro de grandes volúmenes de datos no estructurados.

Azure ML

Azure ML constituye el núcleo del proceso de predicción. En este componente:

- Se cargan los datos desde Azure Blob Storage.
- Se aplican los modelos pre-entrenados a los nuevos datos.
- Se generan las predicciones mensuales.

Almacenamiento de resultados

Las predicciones generadas se almacenan en dos ubicaciones:

- **Azure Blob Storage**: Los resultados se guardan en formato GeoJSON, compatible con ArcGIS Web, facilitando su uso en aplicaciones de mapeo y análisis espacial.
- **CosmosDB**: Una base de datos NoSQL en Azure que almacena los resultados de manera estructurada, permitiendo consultas rápidas y eficientes.

Contenedor Docker

Se implementa un contenedor Docker para encapsular la API web desarrollada en Flask. Esto asegura:

- Consistencia en el entorno de ejecución.
- Facilidad de despliegue y escalabilidad.
- Aislamiento de la aplicación y sus dependencias.

API Web con Flask

La API web se desarrolla utilizando el framework Flask de Python y se despliega a través del contenedor Docker, sirviendo como punto de acceso unificado a las predicciones. Esta API permite:

- El acceso a los equipos de negocio a los datos de predicción para análisis y toma de decisiones.
- La integración de las predicciones en las interfaces y funcionalidades de las aplicaciones web del megaproyecto.
- El procesamiento eficiente de solicitudes HTTP mediante las capacidades de Flask.
- La implementación de endpoints RESTful para diferentes funcionalidades del sistema.

6.5.3. Flujo de trabajo mensual

Ingesta de datos: Se cargan los nuevos datos de ICC CLIMA DATA y NAX INDEX DATA en Azure Blob Storage.

Procesamiento y predicción: Azure ML accede a los datos, aplica los modelos pre-entrenados y genera nuevas predicciones.

Almacenamiento de resultados:

- Se guardan los resultados en formato GeoJSON en Azure Blob Storage.
- Se almacenan las predicciones estructuradas en CosmosDB.

Actualización de la API: Se actualiza el contenedor Docker con la API web de Flask para reflejar los nuevos datos.

Despliegue: Se despliega la API actualizada, permitiendo el acceso inmediato a las nuevas predicciones.

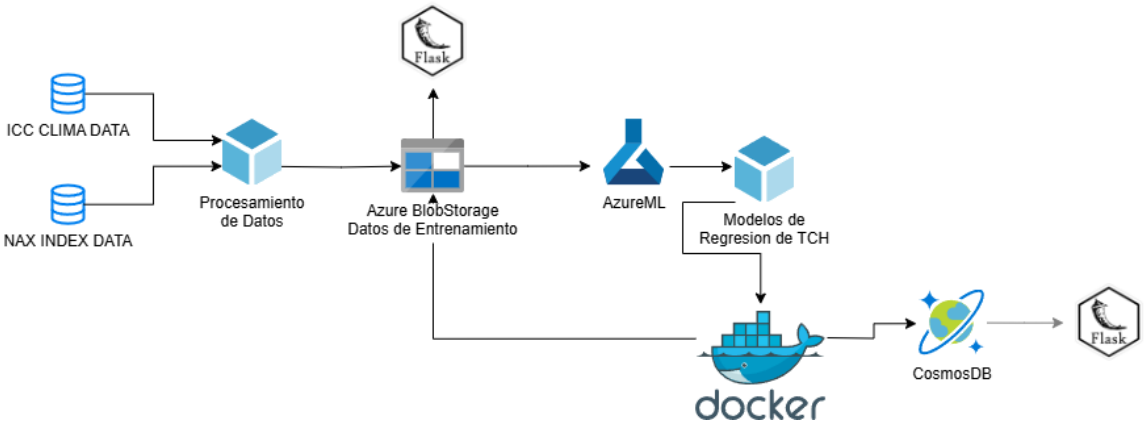


Figura 6.5: Ciclo de CICD

El ciclo CI/CD para MLOps representa un enfoque moderno y eficiente para el manejo de predicciones a gran escala. La combinación de las capacidades de Azure, Docker y una arquitectura API-first basada en Flask ha permitido la creación de un sistema robusto que puede evolucionar con las necesidades cambiantes del negocio y del megaproyecto.

CAPÍTULO 7

Resultados

7.1. Resultados de análisis exploratorio

7.1.1. Resultados de análisis de normalidad por Dataset

Datos de cosecha

- **Variables evaluadas:**
  - Edad proyectada (discreta):
    - El histograma muestra que la distribución de la variable tiene una forma ligeramente asimétrica hacia la izquierda, aunque cercana a una distribución normal.
    - El Q-Q plot revela pequeñas desviaciones en los extremos, indicando que hay cierta desviación de la normalidad en los valores más bajos y altos.
    - Tras aplicar la transformación logarítmica, el ajuste a la normalidad mejora considerablemente, con el Q-Q plot mostrando mayor alineación a la línea diagonal.
    - La transformación de raíz cuadrada también mejora la simetría, aunque no tan efectivamente como la logarítmica.
    - Conclusión: La variable se comporta relativamente bien en términos de normalidad, siendo la transformación logarítmica la más efectiva para ajustar la distribución.

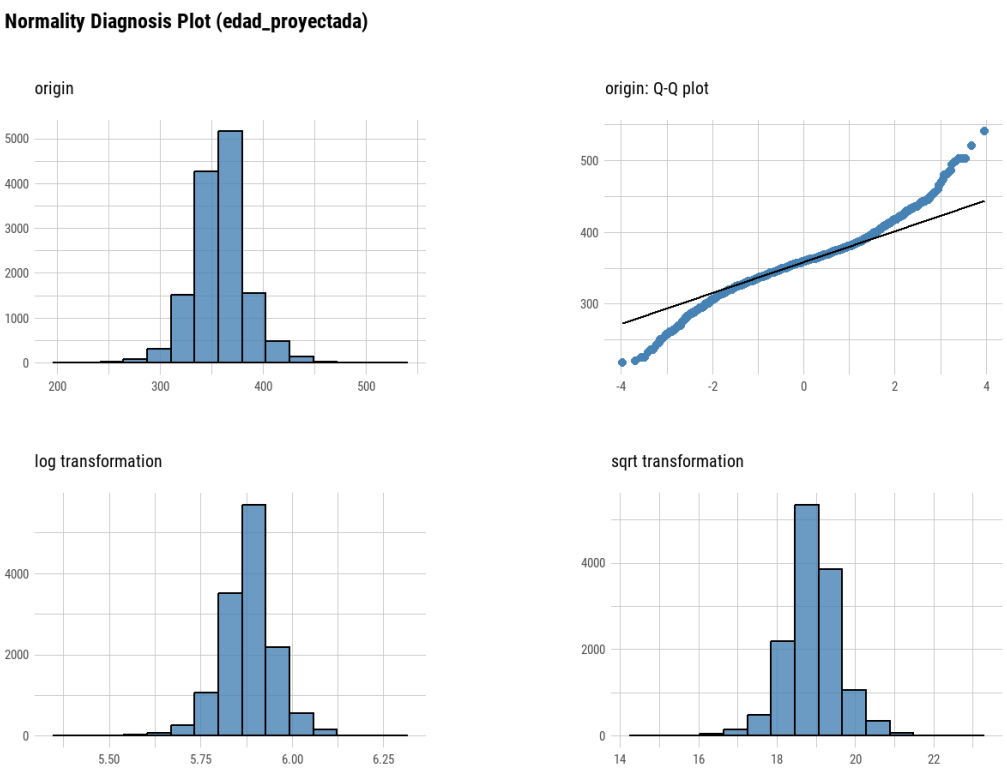


Figura 7.1: Análisis de normalidad de edad_proyectada

- TCH (Toneladas de caña por hectárea):
  - El histograma muestra una fuerte asimetría positiva, con una concentración significativa de valores bajos y una cola extendida hacia valores mayores, lo que indica una distribución no normal.
  - El Q-Q plot confirma esta observación, con grandes desviaciones en los extremos superiores.
  - Al aplicar la transformación logarítmica, la distribución mejora significativamente, mostrando una mayor simetría y un ajuste mucho más cercano a la normalidad.
  - La transformación de raíz cuadrada también mejora la distribución, aunque en menor medida que la transformación logarítmica.
  - Conclusión: La variable tiene una fuerte asimetría en su forma original, y la transformación logarítmica es la opción más adecuada para acercar la variable a una distribución normal.

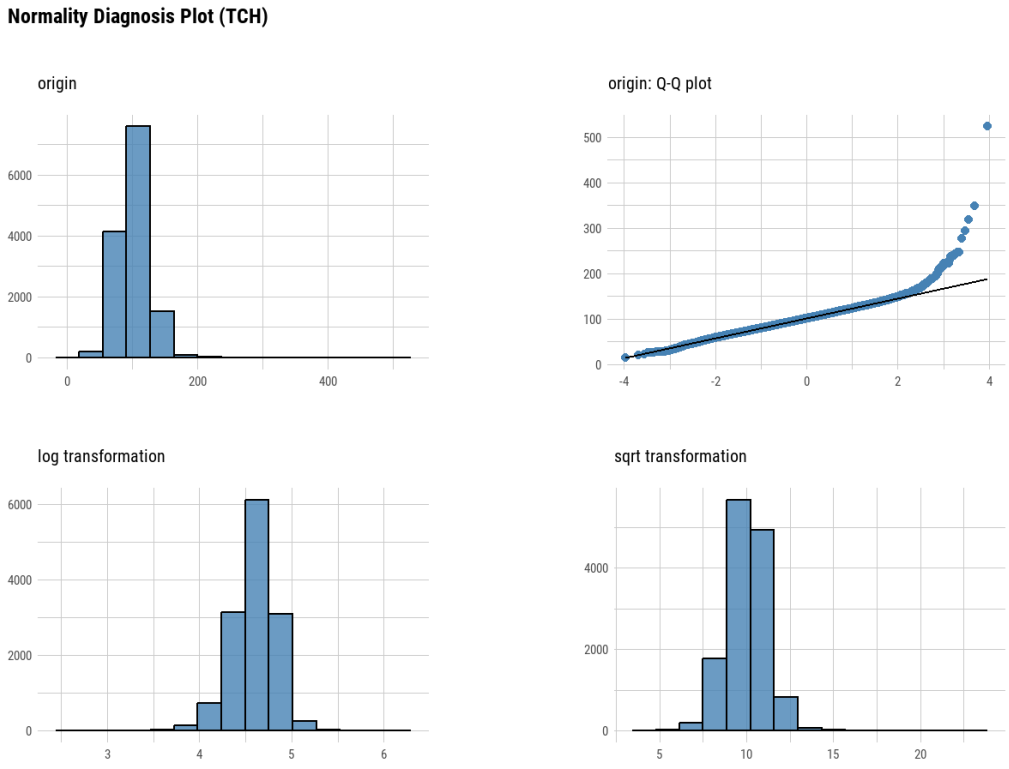


Figura 7.2: Análisis de normalidad TCH

Datos de clima

- **Variables evaluadas:**
  - Radiación (MJ/m^2):
    - El histograma muestra una fuerte asimetría positiva, con una concentración significativa de valores bajos y una cola extendida hacia valores mayores, lo que indica una distribución no normal.
    - El Q-Q plot confirma esta observación, con grandes desviaciones en los extremos superiores.
    - Al aplicar la transformación logarítmica, la distribución mejora significativamente, mostrando una mayor simetría y un ajuste mucho más cercano a la normalidad.
    - La transformación de raíz cuadrada también mejora la distribución, aunque en menor medida que la transformación logarítmica.
    - Conclusión: La variable tiene una fuerte asimetría en su forma original, y la transformación logarítmica es la opción más adecuada para acercar la variable a una distribución normal.

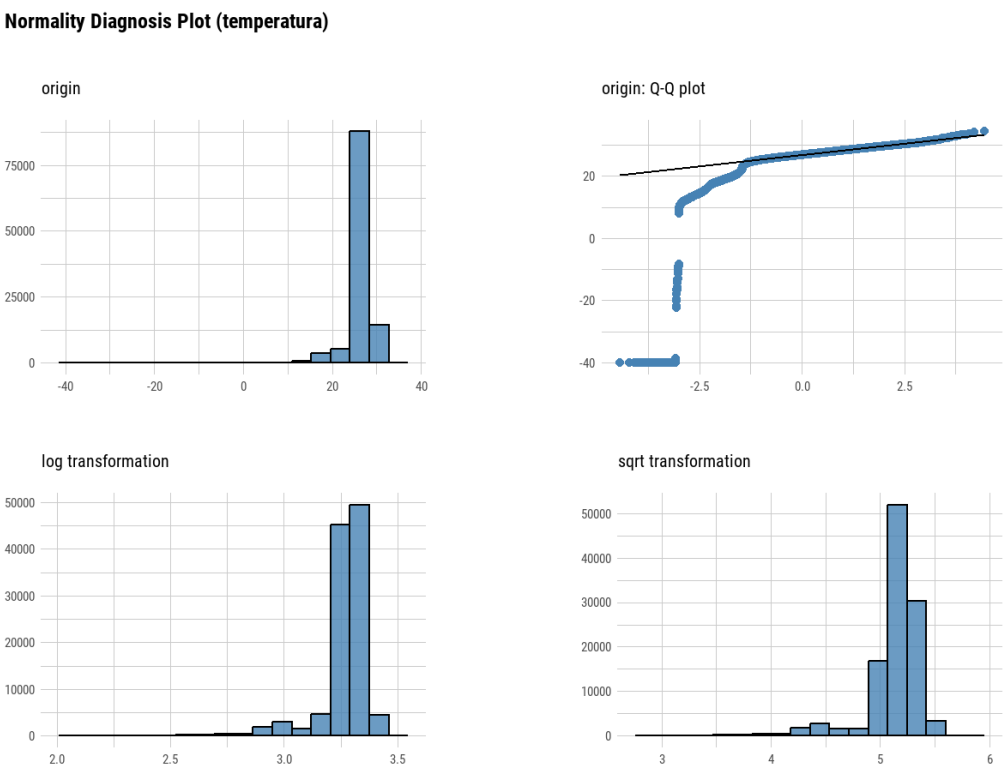


Figura 7.3: Análisis de normalidad de radiación (MJ/m^2)

- Temperatura máxima:
  - El histograma muestra una fuerte asimetría negativa, con una concentración significativa de valores alrededor de los 20 a 30 grados, pero con una cola hacia valores negativos.
  - El Q-Q plot revela grandes desviaciones en los extremos, particularmente en los valores negativos.
  - Tras aplicar la transformación logarítmica, la distribución mejora en la parte positiva, pero los valores negativos no se ven afectados, lo que mantiene cierta asimetría.
  - La transformación de raíz cuadrada también mejora la distribución para los valores positivos, aunque los valores negativos siguen generando problemas en la distribución.
  - Conclusión: La variable presenta un comportamiento atípico con valores negativos que afectan la normalización, aunque en la parte positiva, la transformación de raíz cuadrada es la más efectiva.

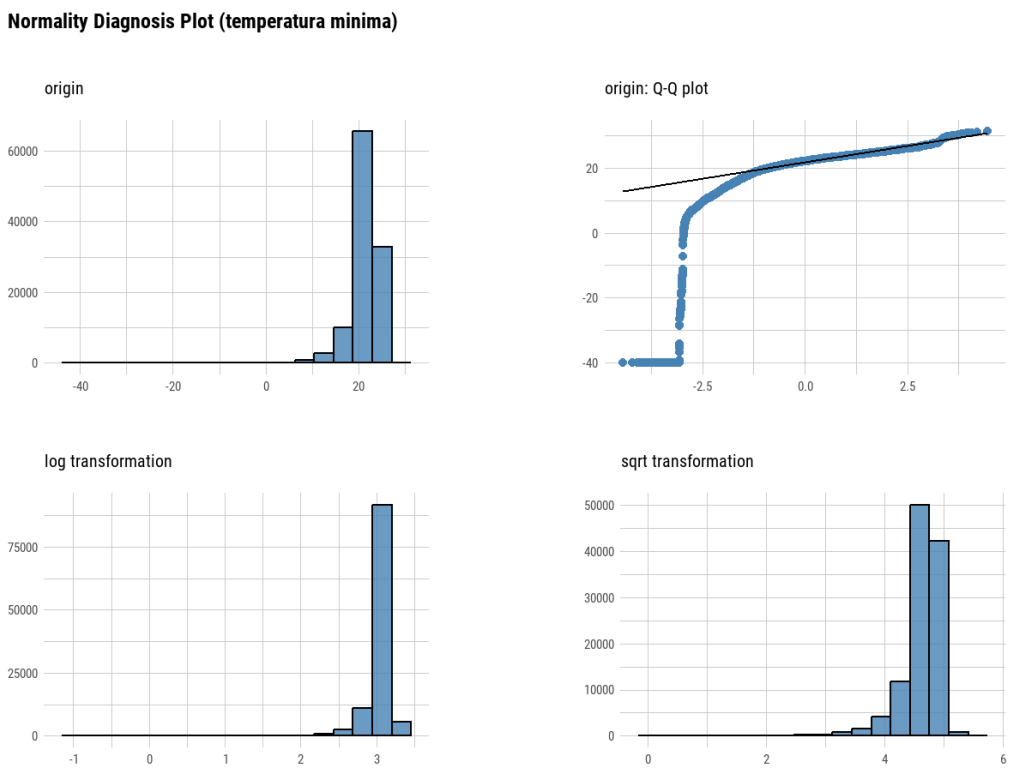


Figura 7.4: Análisis de normalidad de temperatura máxima

Datos de índices

- **Variables evaluadas:**
  - NDVI (POND):
    - El histograma muestra una fuerte asimetría positiva, con una concentración significativa de valores alrededor de 0.75 y una cola extendida hacia valores más bajos.
    - El Q-Q plot confirma la desviación de la normalidad, con grandes diferencias en los extremos inferiores.
    - Tras aplicar la transformación logarítmica, la distribución no mejora significativamente y el Q-Q plot sigue mostrando grandes desviaciones.
    - La transformación de raíz cuadrada logra mejorar la distribución, acercándola a la normalidad, pero aún se observan ligeras desviaciones en los extremos.
    - Conclusión: La transformación de raíz cuadrada es la más adecuada para normalizar la distribución de NDVI.

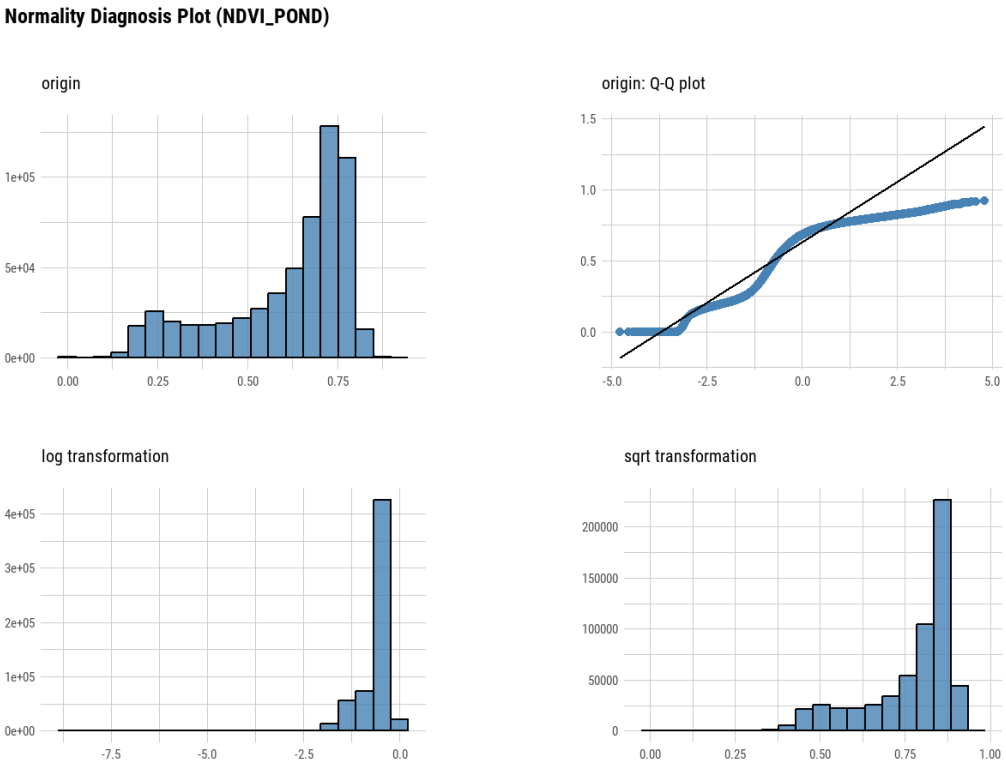


Figura 7.5: Análisis de normalidad de NDVI (POND)

- Humedad (POND):
  - El histograma muestra una ligera asimetría positiva, con una concentración significativa de valores entre 70 y 75.
  - El Q-Q plot revela pequeñas desviaciones en los extremos, lo que indica que hay ligeras desviaciones de la normalidad.
  - Tras aplicar la transformación logarítmica, la distribución mejora, aunque los extremos aún muestran desviaciones menores.
  - La transformación de raíz cuadrada mejora la simetría de manera similar a la logarítmica.
  - Conclusión: La variable tiene un comportamiento bastante cercano a la normalidad, y ambas transformaciones, logarítmica y de raíz cuadrada, son efectivas para normalizar la distribución.

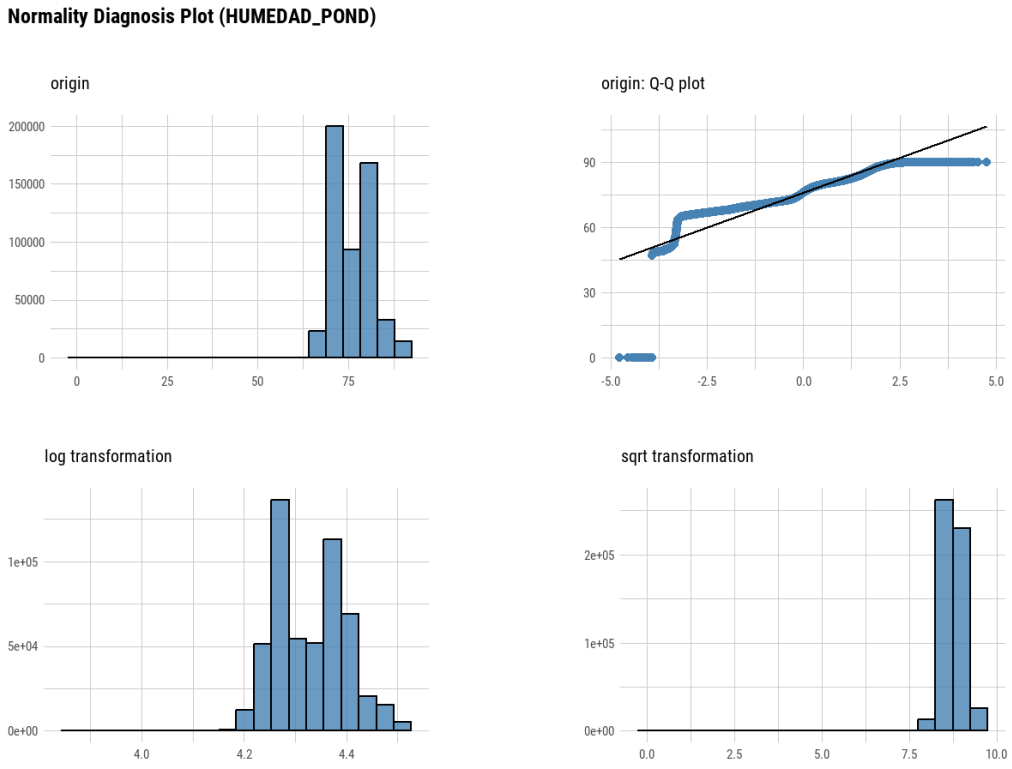


Figura 7.6: Análisis de normalidad de humedad (POND)

7.1.2. Resultados de análisis de datos cualitativos

- **Variables cualitativas evaluadas:**

  - **Zafra:** El gráfico muestra la distribución de la frecuencia de datos a lo largo de distintas zafras (19-20, 20-21, 21-22, 22-23, 23-24), siendo la zafra 19-20 la más representativa, con una disminución en las zafras más recientes.
  - **Fecha:** La variable fecha presenta una agrupación de frecuencias a lo largo de varios años. La frecuencia de eventos varía según el periodo, con picos visibles en años clave como 2020 y 2021.
  - **Producto aplicado (prod_mad):** Se observa que los productos más frecuentes son la madurez natural y timepaxe, seguidos por el uso de glifosato. Los productos biosimulantes y protecantes presentan frecuencias menores en comparación.
  - **Estación:** Las estaciones con mayor frecuencia en el análisis incluyen Teculuttan, Cengicana, y El Basamo, con otras estaciones como Puyumatán y Concepción con menor representación en los datos.
  - **Variedad:** Las variedades CG01-187 y CP72-1210 dominan significativamente en la base de datos, representando la mayor frecuencia de ocurrencia. Otras variedades como CG03-148 y CP88-1762 tienen menor participación.
  - **Sistema de riego (sist_riego):** La mayoría de las áreas están bajo el sistema de aspersion, seguido por áreas que no cuentan con riego (sin riego), con una menor proporción en sistemas como pivot y gravedad.

- **Tipo de cosecha:** En términos de cosecha, se observa una distribución relativamente balanceada entre la cosecha mecánica y la cosecha manual, con una ligera preponderancia de la cosecha mecánica.

- **Interpretación:** Las variables cualitativas analizadas revelan tendencias importantes en el manejo de las zafras y las características agronómicas predominantes. Las categorías dominantes en productos aplicados y variedades resaltan prácticas clave dentro de la producción. Las diferencias en frecuencia entre estaciones y tipos de cosecha también destacan la variabilidad regional y operacional.

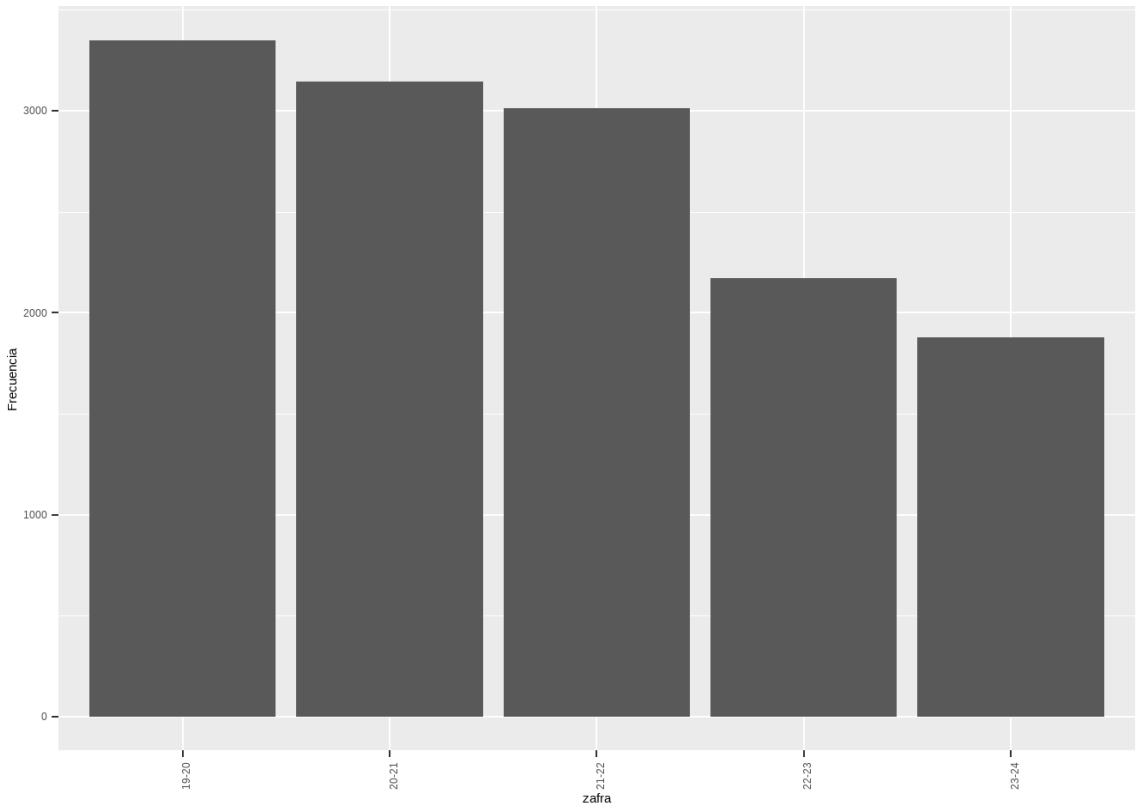


Figura 7.7: Histograma de frecuencias de zafras

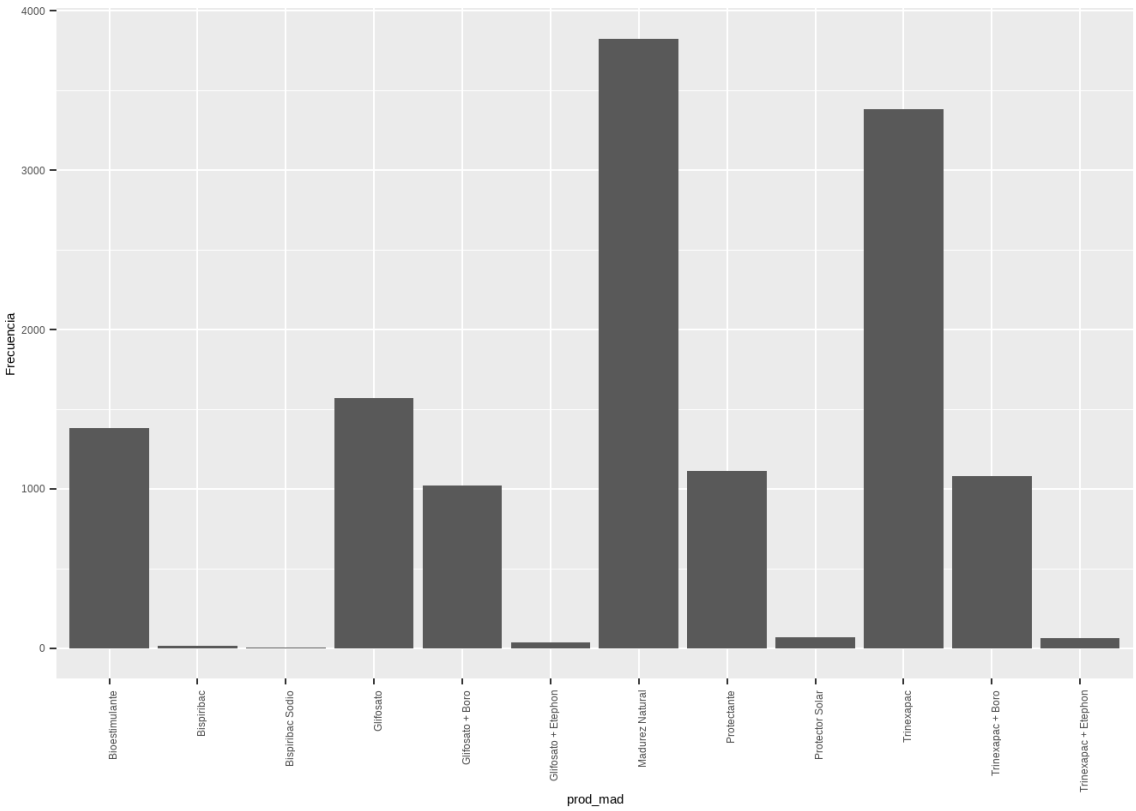


Figura 7.8: Histograma de frecuencias de productos aplicados

7.1.3. Análisis de correlación por Dataset

Descripción

Se realizó una evaluación de la correlación entre las variables cuantitativas en los diferentes conjuntos de datos utilizados (clima, índices de la caña, y datos de cosecha). Este análisis permite identificar relaciones lineales fuertes que puedan impactar la multicolinealidad en los modelos predictivos. Para todas las variables con una correlación superior a 0.9, se aplicó un proceso de limpieza de datos, eliminando aquellas altamente correlacionadas.

Clima

- **Resultados:**
  - **Radiación** presenta una alta correlación con **radiación promedio** (0.93), lo que sugiere que ambas variables aportan información redundante.
  - **Humedad relativa** y **humedad relativa máxima** tienen una correlación significativa (0.84), indicando que puede no ser necesario incluir ambas variables en el modelo.
  - La **temperatura mínima** y la **temperatura** muestran una alta correlación (0.91), lo que también sugiere redundancia.

  Todas estas correlaciones mayores a 0.9 fueron eliminadas en el proceso de limpieza para evitar problemas de multicolinealidad en los modelos. La matriz de correlación (Figura 7.9) muestra estas relaciones, con las áreas en color más intenso indicando las correlaciones más fuertes.

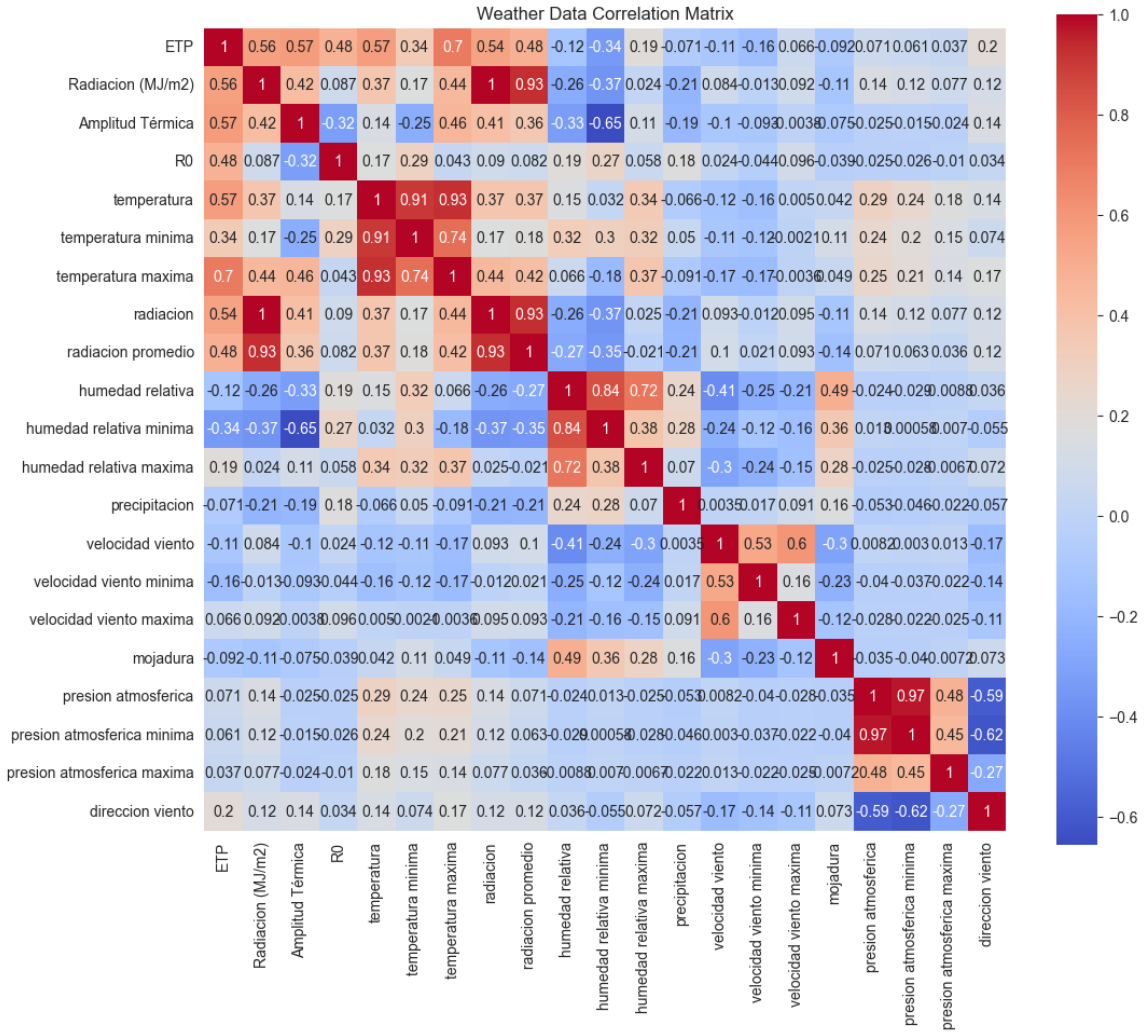


Figura 7.9: Matriz de correlación de las variables de clima

Índices de la caña

- **Resultados:**
  - Se observó una alta correlación entre **NDVI** y **AGUA_POND** (0.90), lo que indica una redundancia considerable.
  - La **maduración** y **AGUA_POND** también presentan una fuerte correlación (0.84), sugiriendo que la maduración podría depender fuertemente de la disponibilidad de agua.

  Tras el proceso de limpieza, se eliminaron las variables altamente correlacionadas (>0.9). La Figura 7.10 presenta el corplot de estas variables, donde se puede visualizar estas correlaciones significativas.

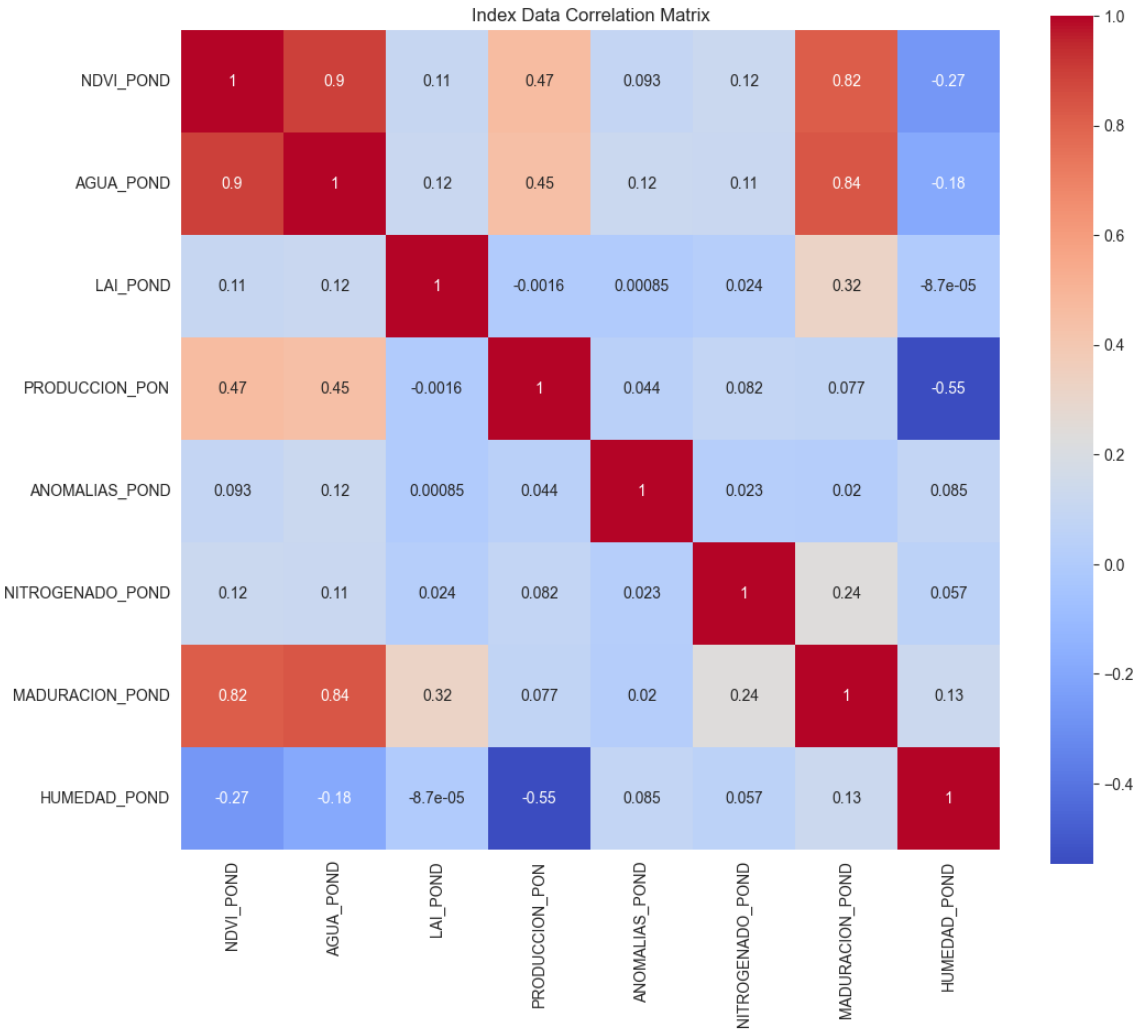


Figura 7.10: Matriz de correlación de las variables de índices de caña

Cosecha

- **Resultados:**
  - No se observan correlaciones mayores a 0.9 en este conjunto de datos, lo que sugiere que las variables son más independientes entre sí.
  - Sin embargo, variables como **edad proyectada** y **edad actual** tienen una correlación perfecta (1.0), lo que era de esperarse debido a su naturaleza dependiente.

  Dado que no se encontraron otras correlaciones mayores a 0.9, no se aplicó ningún proceso de eliminación para este conjunto. La Figura 7.11 muestra la matriz de correlación.

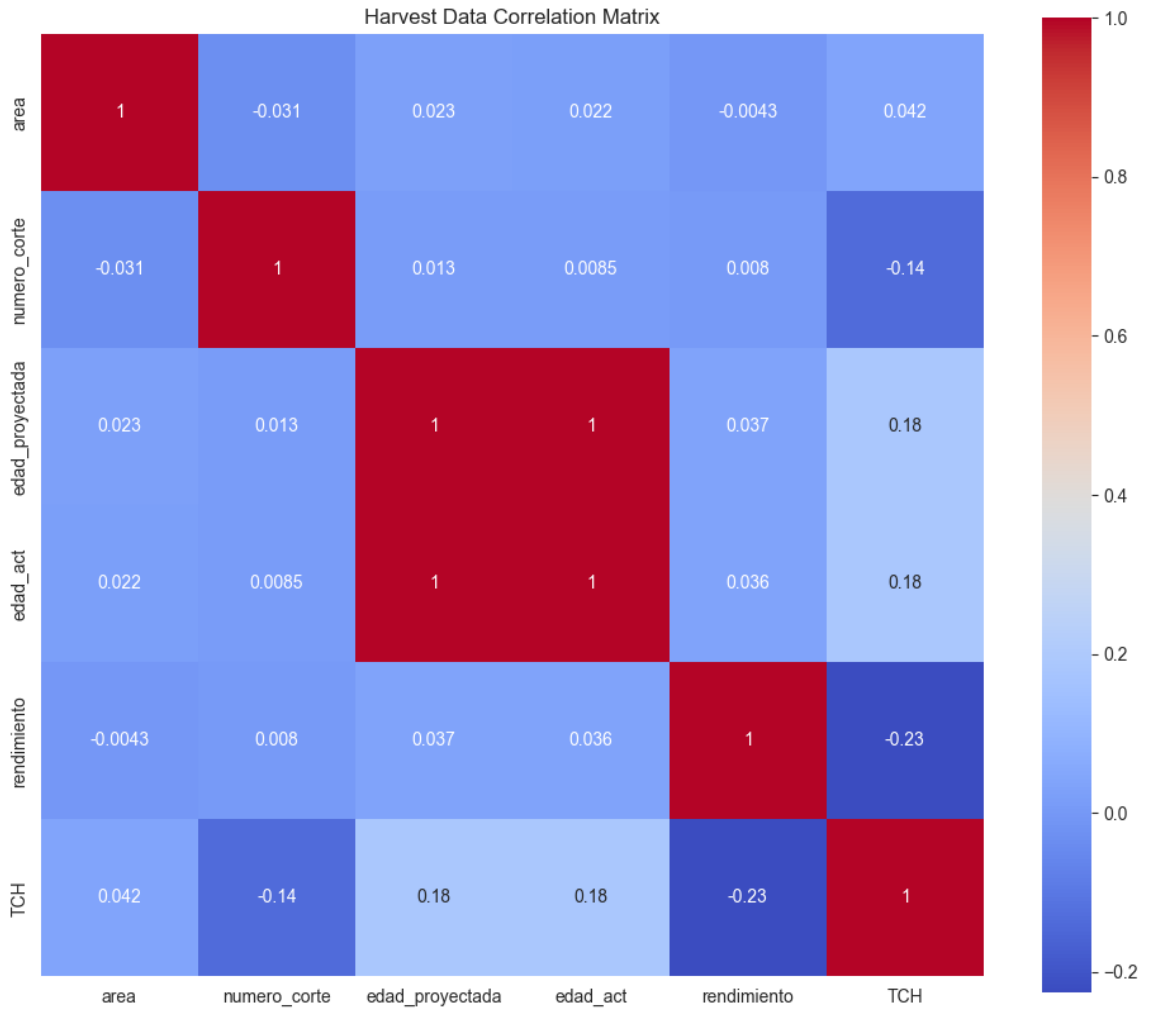


Figura 7.11: Matriz de correlación de las variables de cosecha

7.2. Resultados de modelos Deep Learning

7.2.1. Modelo de regresión

Descripción del modelo

El modelo implementado es una red neuronal profunda fully connected con 16 capas lineales, diseñada para la tarea de regresión para predecir el tonelaje de caña por hectárea (TCH). La arquitectura del modelo sigue un patrón de äscenso-descenso", donde el tamaño de las capas aumenta progresivamente hasta un pico y luego disminuye hacia la capa de salida.

Hiperparámetros optimizados

Se realizó una búsqueda en cuadrícula (Grid Search) para optimizar los siguientes hiperparámetros:

Hiperparámetro	Valores evaluados	Valor final seleccionado
Learning Rate	0.1, 0.01, 0.001	0.001
L1 Lambda	0.0, 0.001	0.0
L2 Lambda	0.0, 0.001	0.001
Batch Size	1024	1024
Número de Épocas	200	200
Optimizador	Adam, SGD	SGD
Función de Pérdida	MSE, MAE, SmoothL1Loss	MSE

Tabla 7.1: Hiperparámetros, valores evaluados y valor final seleccionado durante el entrenamiento

Rendimiento del modelo

Tras la optimización de hiperparámetros, el modelo alcanzó el siguiente rendimiento en el conjunto de validación:

Métrica	Valor
RMSE	23.0635
R^2	-0.0000

Tabla 7.2: Métricas de rendimiento del modelo de regresión DL

Predicciones vs valores reales

La siguiente figura muestra la comparación entre los valores predichos por el modelo y los valores reales de TCH:

Figura 7.12: Comparación entre valores predichos (azul) y reales (rojo) para el modelo de regresión DeepLearning

Evolución del entrenamiento

La siguiente figura muestra la evolución de la pérdida durante el entrenamiento tanto para el conjunto de entrenamiento como para el de validación:

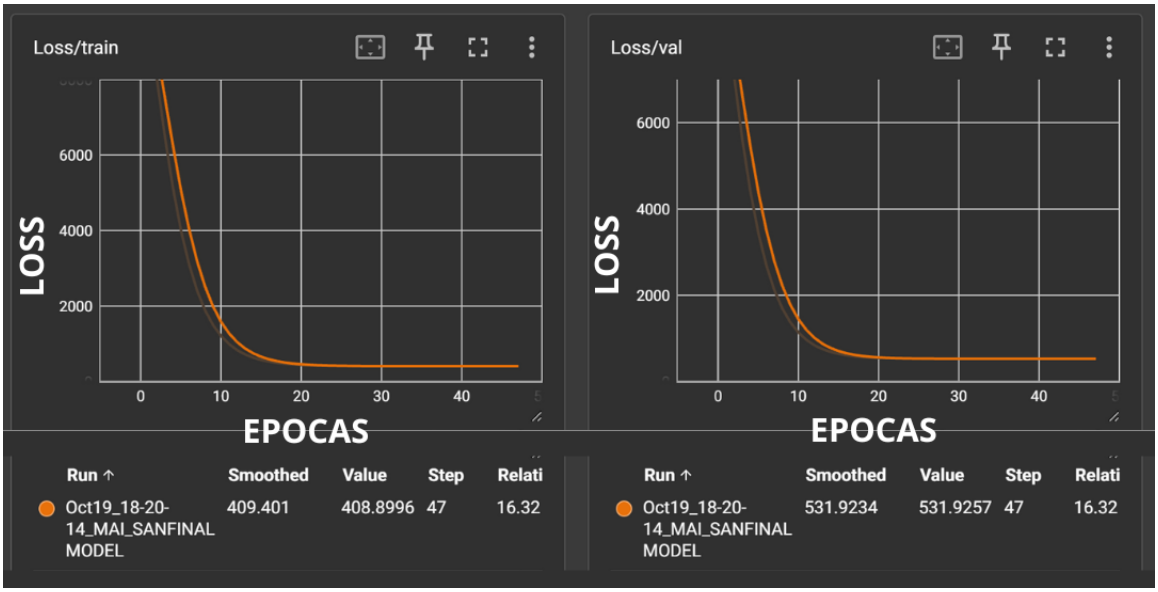


Figura 7.13: Comparación entre valores predichos y reales para el modelo de regresión DeepLearning

Es importante notar que, a pesar de las técnicas de regularización implementadas y la optimización de hiperparámetros, el modelo muestra signos de sobreajuste, como se evidencia en la divergencia entre las curvas de pérdida de entrenamiento y validación. Los resultados detallados de todas las combinaciones evaluadas en el Grid Search se encuentran en el Anexo A.

7.2.2. Modelo de clasificación

Descripción del modelo

El modelo implementado es una red neuronal profunda *fully connected* con 16 capas lineales, diseñada para la tarea de clasificación multiclase del tonelaje de caña por hectárea (TCH) en grupos predefinidos. La arquitectura sigue un patrón similar al modelo de regresión, con un "ascenso-descenso" en el tamaño de las capas.

Hiperparámetros optimizados

Se realizó una búsqueda en cuadrícula (*Grid Search*) para optimizar los siguientes hiperparámetros:

Hiperparámetro	Valores evaluados	Valor final seleccionado
Learning Rate	0.01, 0.001	0.001
L1 Lambda	0.0, 0.001	0.0
L2 Lambda	0.0, 0.001	0.0
Batch Size	2048	2048
Número de Épocas	200	200
Optimizador	Adam, SGD	Adam
Número de Capas (N)	16	16

Tabla 7.3: Hiperparámetros evaluados y valor final seleccionado en el Grid Search para el modelo de clasificación

Rendimiento del modelo

Tras la optimización de hiperparámetros, el modelo alcanzó el siguiente rendimiento en el conjunto de prueba:

Métrica	Valor
Test Loss	3.8602
Test Accuracy	0.0211

Tabla 7.4: Métricas de rendimiento del modelo de clasificación DL

Matriz de confusión

La siguiente figura muestra la matriz de confusión del modelo en el conjunto de prueba:

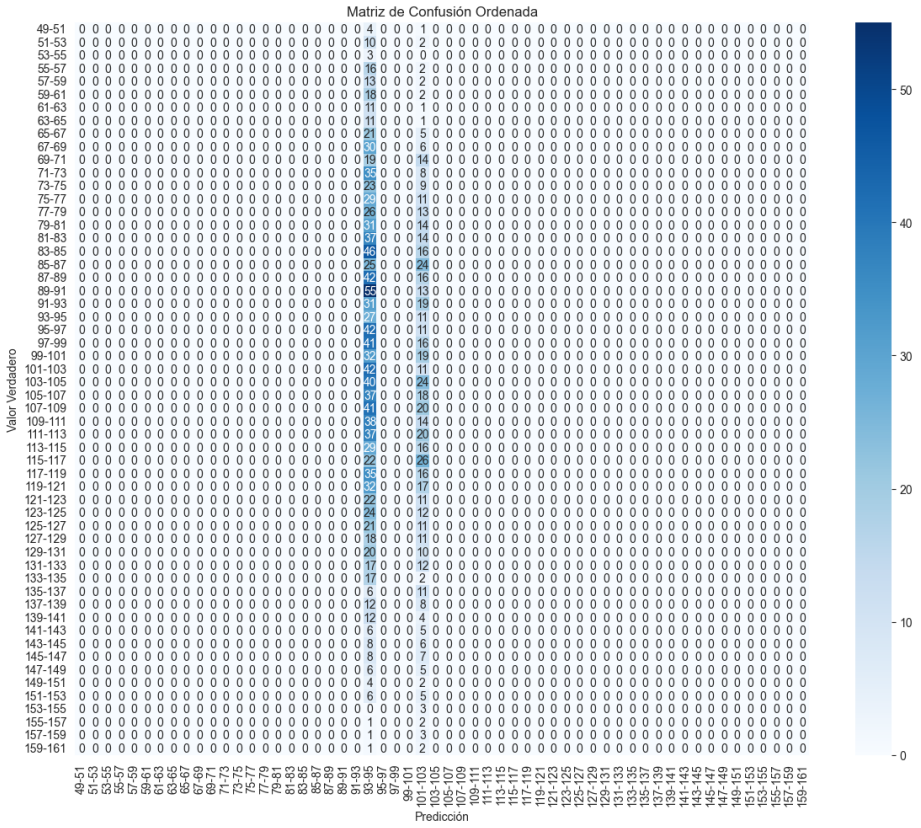


Figura 7.14: Matriz de confusión del modelo de clasificación DeepLearning

Como se puede observar en la matriz de confusión, el modelo muestra una tendencia a clasificar la mayoría de las muestras en solo dos grupos, lo que indica un rendimiento deficiente en la tarea de clasificación multiclase.

Evolución del Loss durante el entrenamiento

La siguiente figura muestra la evolución del loss tanto para el conjunto de entrenamiento como para el de validación durante el proceso de entrenamiento:

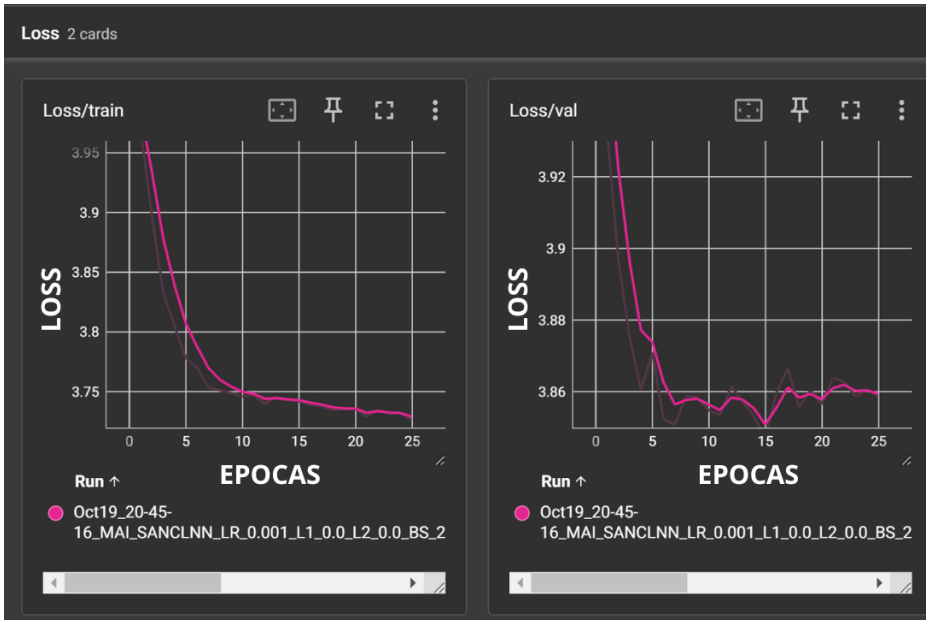


Figura 7.15: Evolución del loss durante el entrenamiento del modelo de clasificación DeepLearning

Como se puede observar en las gráficas de loss, tanto el loss de entrenamiento como el de validación disminuyen rápidamente en las primeras épocas, pero luego se estabilizan en valores relativamente altos (alrededor de 3.75 para el entrenamiento y 3.86 para la validación). Esta estabilización temprana del loss, junto con la pequeña brecha entre el loss de entrenamiento y validación, sugiere que el modelo no está sobreajustándose significativamente. Sin embargo, el alto valor del loss final indica que el modelo no ha logrado aprender una representación efectiva de las clases, lo que se refleja en su baja precisión. La falta de mejora sustancial después de las primeras épocas también sugiere que el modelo puede haber alcanzado rápidamente su capacidad máxima para esta tarea, posiblemente debido a la complejidad del problema de clasificación multiclase con clases desequilibradas.

7.3. Modelos de Azure

7.3.1. Modelo de Regresión a 6 Meses con pre-procesamiento

Este modelo de regresión se entrenó utilizando AutoML de Azure con datos pre-procesados. El objetivo es predecir el tonelaje de caña por hectárea (TCH) con 6 meses de anticipación.

Preprocesamiento de datos

Antes de entrenar el modelo en Azure AutoML, se realizó un extenso preprocesamiento de los datos. Este proceso incluyó varias etapas de codificación para variables categóricas y manejo de valores faltantes. A continuación, se detallan los pasos principales:

Codificación de variables categóricas Se utilizaron tres métodos de codificación para las variables categóricas:

1. **One-Hot Encoding**: Aplicado a variables nominales con pocas categorías como `prod_mad`,

`sist_riego`, `tipo_cosecha`, y `PRODUCTO_ACTUAL`. Este método crea nuevas columnas binarias para cada categoría.

2. **Ordinal Encoding**: Utilizado para variables ordinales como `region` y `estrato`. Este método asigna valores numéricos manteniendo el orden de las categorías.
3. **Label Encoding**: Aplicado a variables nominales con muchas categorías como `estacion`, `variedad`, y `cuadrante`. Este método asigna un valor numérico único a cada categoría.

Para cada método de codificación, se guardaron diccionarios de mapeo para facilitar la interpretación posterior de los resultados.

Manejo de valores faltantes Se utilizó `SimpleImputer` con la estrategia de media para manejar los valores faltantes en las columnas numéricas.

Eliminación de multicolinealidad Se calculó una matriz de correlación y se eliminaron las columnas con una correlación superior a 0.90 para reducir la multicolinealidad en el conjunto de datos.

Preparación de conjuntos de entrenamiento y prueba Los datos se dividieron en conjuntos de entrenamiento y prueba basados en la zafra:

- **Conjunto de entrenamiento**: Todas las zafras excepto '23-24'
- **Conjunto de prueba**: Zafra '23-24'

Descripción del modelo y transformaciones de datos

Tras el preprocesamiento inicial, se aplicaron transformaciones adicionales y se implementaron dos modelos diferentes de `XGBoost` para la predicción del TCH. El proceso fue el siguiente:

Transformaciones de datos

- **Imputación**: Se utilizó `SimpleImputer` con estrategia de media para manejar valores faltantes en las características numéricas.
- **Escalado**: Se aplicó `MaxAbsScaler` para normalizar las características en uno de los modelos, mientras que en el otro se usó `TruncatedSVD` para la reducción de dimensionalidad.

Modelos implementados

En este estudio, se implementaron dos modelos de `XGBoost` con diferentes configuraciones, que luego se combinaron en un modelo de ensamble mediante `VotingRegressor`. A continuación, se detalla cada modelo y sus resultados.

Modelo 1: `XGBoost` con `MaxAbsScaler`

Este modelo aplicó un escalado de características utilizando `MaxAbsScaler` antes de entrenar el modelo `XGBoost`. Los parámetros utilizados fueron:

- `learning_rate = 0.3`
- `max_depth = 6`
- `n_estimators = 100`

SHAP del modelo

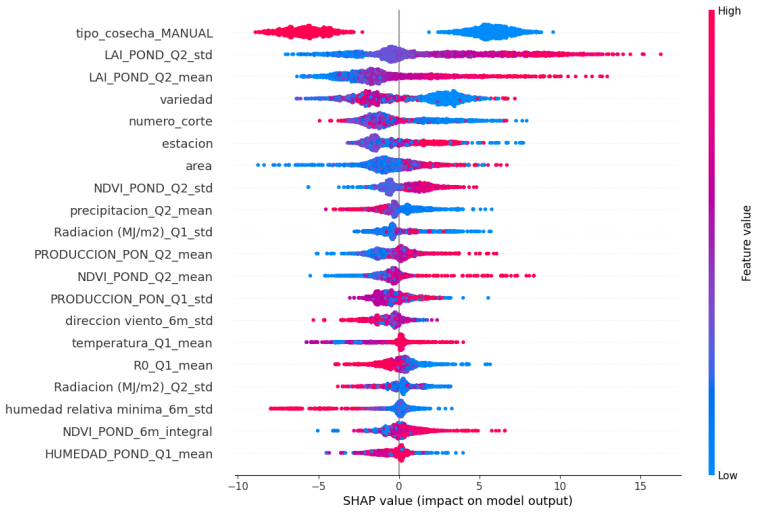


Figura 7.16: Análisis SHAP para el modelo `MaxAbsScaler`

Modelo 2: `XGBoost` con `TruncatedSVD`

Este modelo implementó una reducción de dimensionalidad utilizando `TruncatedSVD`, reteniendo las 14 componentes principales, antes de entrenar el modelo `XGBoost`. Los parámetros del modelo fueron:

- `learning_rate = 0.2`
- `max_depth = 10`
- `n_estimators = 400`
- `colsample_bytree = 0.8`
- `subsample = 0.6`
- `gamma = 10`
- `reg_lambda = 0.625`

SHAP del Modelo

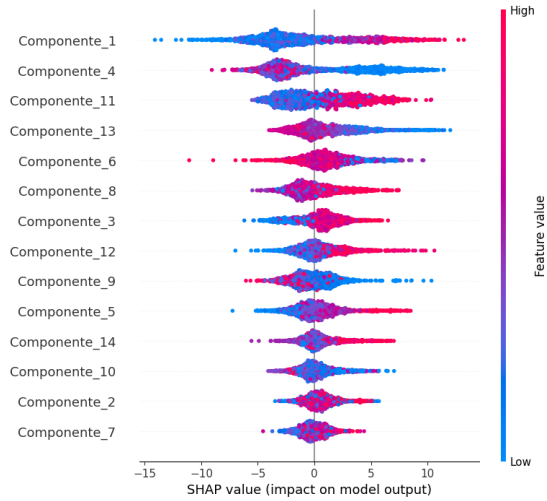


Figura 7.17: Análisis SHAP para el modelo `TruncatedSVD`

Modelo de ensamble: VotingRegressor

Para combinar las fortalezas de los dos modelos XGBoost descritos anteriormente (uno con MaxAbsScaler y otro con TruncatedSVD), se implementó un modelo de ensamble utilizando VotingRegressor. Este enfoque permite aprovechar las predicciones de ambos modelos para obtener una estimación más robusta y precisa del Tonelaje de Caña por Hectárea (TCH).

Resultados del modelo de ensamble El modelo de ensamble VotingRegressor logró los siguientes resultados en el conjunto de prueba:

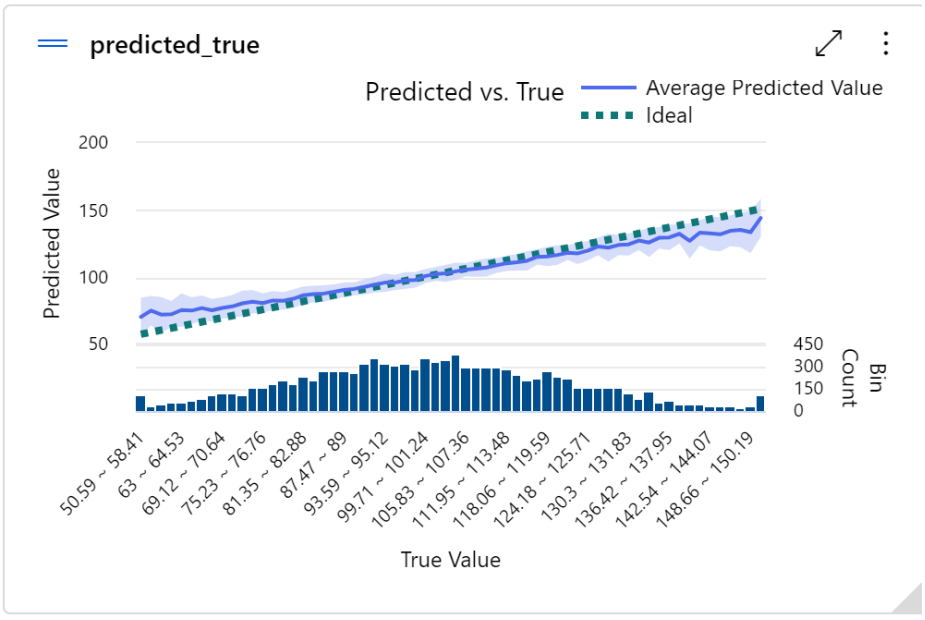


Figura 7.18: Métricas de rendimiento y gráfica de predicciones vs valores reales para el modelo `VotingEmsamble`

- R^2 Score: 0.8174278
- Root Mean Squared Error (RMSE): 8.638410

7.3.2. Modelo de Regresión a 2 meses

Este modelo de regresión se entrenó utilizando AutoML de Azure con datos sin preprocesamiento adicional, más allá del procesamiento estándar descrito en la Sección 6.3. El objetivo es predecir el tonelaje de caña por hectárea (TCH) con 2 meses de anticipación.

Preparación de datos

Para este modelo, se utilizó el conjunto de datos procesado según lo descrito en la Sección 6.3. Los pasos clave incluyeron:

- **División temporal de datos**:
  - Conjunto de entrenamiento: Todas las zafras anteriores a '23-24'.
  - Conjunto de prueba: La zafra '23-24'.
- **Selección de características**:
  - Se mantuvieron todas las variables procesadas, incluyendo las características climáticas, índices de vegetación y variables categóricas codificadas.
  - Se eliminaron las columnas 'ABS_IDCOMP', 'ZAFRA', 'fecha' y 'rendimiento' por no ser relevantes para la predicción.

Descripción del modelo y transformaciones de datos

Tras el preprocesamiento inicial, se aplicaron transformaciones adicionales y se implementaron tres modelos diferentes de `XGBoost` para la predicción del TCH. El proceso fue el siguiente:

Transformaciones de datos

- **Imputación**: Se utilizó `SimpleImputer` con estrategia de media para manejar valores faltantes en las características numéricas.
- **Escalado**: Se aplicaron diferentes técnicas de escalado para cada modelo:
  - `StandardScaler` para los modelos 0 y 1.
  - `MaxAbsScaler` para el modelo 2.
- **Codificación**: Se utilizó `OneHotEncoder` para variables categóricas nominales y `OrdinalEncoder` para variables ordinales.

Modelos implementados

En este estudio, se implementaron tres modelos de `XGBoost` con diferentes configuraciones, que luego se combinaron en un modelo de ensamble mediante `VotingRegressor`. A continuación, se detalla cada modelo y sus resultados.

Modelo 0: `XGBoost` con `StandardScaler`

Este modelo aplicó un escalado estándar de características antes de entrenar el modelo `XGBoost`. Los parámetros utilizados fueron:

- `learning_rate = 0.3`
- `max_depth = 7`
- `n_estimators = 800`
- `colsample_bytree = 0.5`
- `subsample = 0.5`
- `reg_alpha = 2.291666666666667`
- `reg_lambda = 0.8333333333333334`

SHAP del modelo

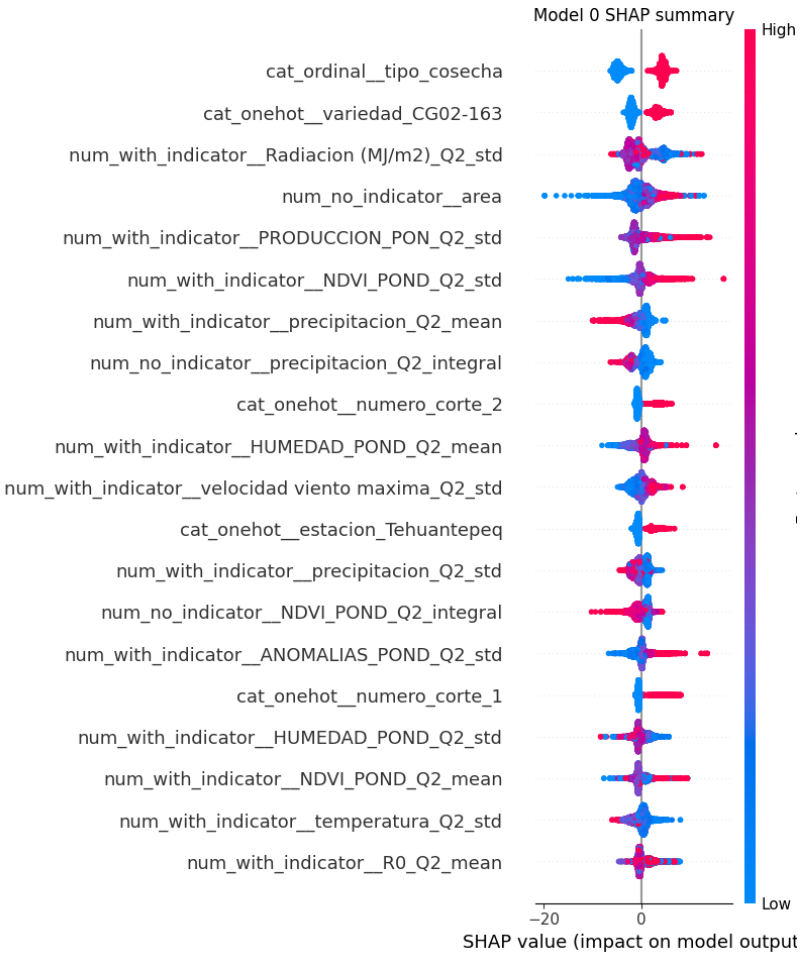


Figura 7.19: Análisis SHAP para el modelo `XGBoost con StandardScaler`

Modelo 1: `XGBoost` con `StandardScaler`

Este modelo también utilizó `StandardScaler` pero con una configuración diferente de `XGBoost`:

- `learning_rate = 0.5`
- `max_depth = 8`
- `n_estimators = 50`
- `colsample_bytree = 0.9`
- `subsample = 1`
- `reg_alpha = 1.3541666666666667`
- `reg_lambda = 1.6666666666666667`
- `gamma = 0.01`

SHAP del modelo

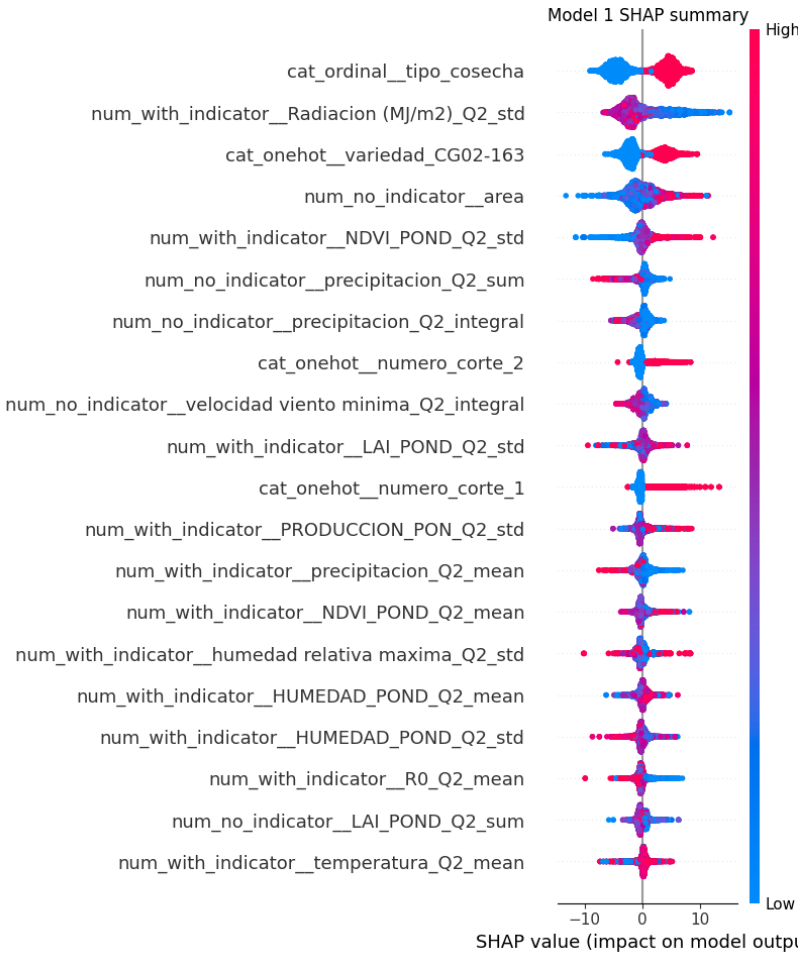


Figura 7.20: Análisis SHAP para el modelo `XGBoost con StandardScaler`

Modelo 2: `XGBoost` con `MaxAbsScaler`

Este modelo aplicó un escalado de características utilizando `MaxAbsScaler` antes de entrenar el modelo `XGBoost`. Los parámetros utilizados fueron:

- `learning_rate = 0.3`
- `max_depth = 6`
- `n_estimators = 100`

SHAP del modelo

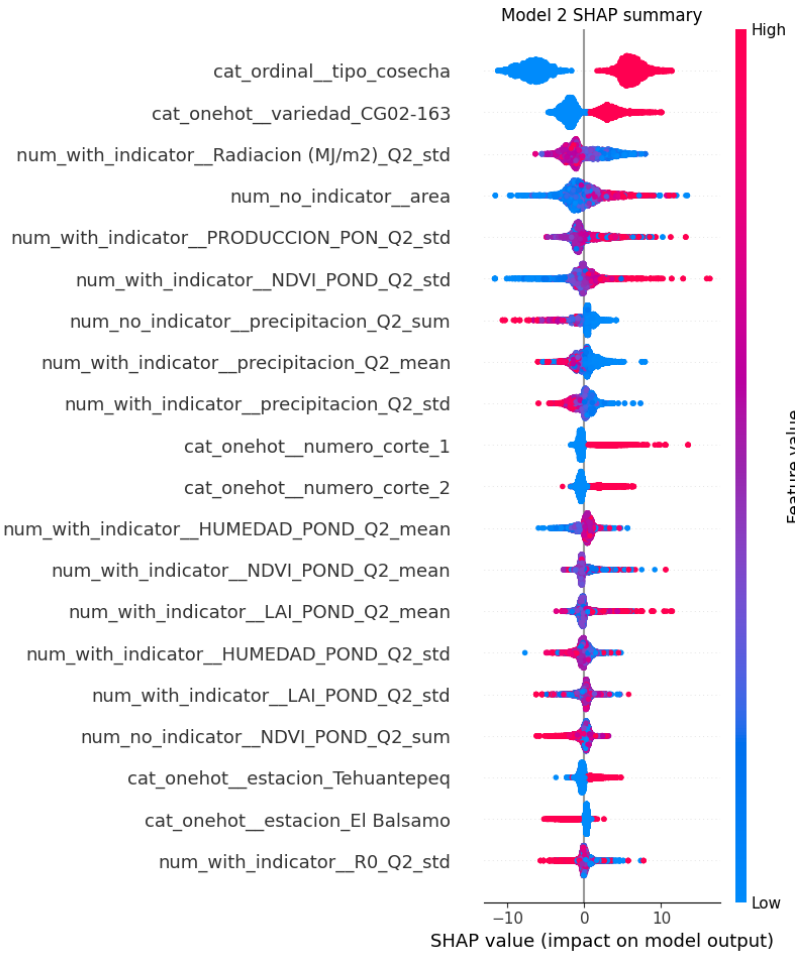


Figura 7.21: Análisis SHAP para el modelo `XGBoost con MaxAbsScaler`

Modelo de ensamble: `VotingRegressor`

Los tres modelos `XGBoost` se combinaron utilizando `VotingRegressor` con los siguientes pesos:

- Modelo 0: 0.5714285714285714
- Modelo 1: 0.2857142857142857
- Modelo 2: 0.14285714285714285

Resultados del modelo

El modelo de ensamble `VotingRegressor` logró los siguientes resultados en el conjunto de prueba:

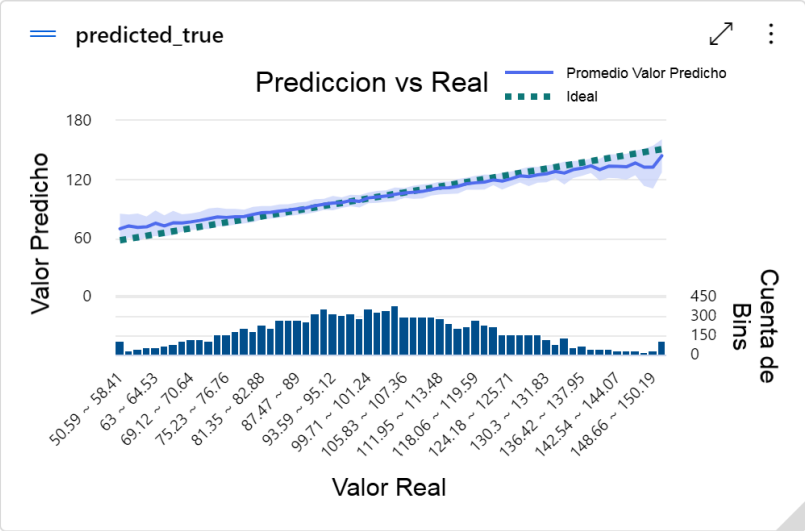


Figura 7.22: Métricas de rendimiento y gráfica de predicciones vs valores reales para el modelo `VotingEmsamble 2 Meses`

- R^2 Score: 0.8174278
- Root Mean Squared Error (RMSE): 8.638410

Estos resultados indican un buen rendimiento del modelo, explicando aproximadamente el 81.74 % de la variabilidad en el TCH y con un error promedio de predicción de alrededor de 8.64 toneladas por hectárea.

7.3.3. Modelo de regresión a 4 meses

Este modelo de regresión se entrenó utilizando un enfoque de ensamble con dos modelos XGBoost, con el objetivo de predecir el tonelaje de caña por hectárea (TCH) con 4 meses de anticipación.

Preparación de datos

Para este modelo, se utilizó el conjunto de datos procesado según lo descrito en la Sección 6.3.

Transformaciones de datos

- **Imputación**: Se utilizó `SimpleImputer` con estrategia de media para manejar valores faltantes en las características numéricas.
- **Codificación**: Se utilizó `OneHotEncoder` para variables categóricas nominales y `OrdinalEncoder` para variables ordinales.
- **Escalado**: Se aplicó `MaxAbsScaler` para el segundo modelo.

Modelos implementados

Se implementaron dos modelos de `XGBoost` con diferentes configuraciones, que luego se combinaron en un modelo de ensamble mediante `VotingRegressor`. A continuación, se detalla cada modelo:

Modelo 0: `XGBoost` sin escalado adicional

Este modelo no aplicó un escalado adicional antes de entrenar el modelo `XGBoost`. Los parámetros utilizados fueron:

- `learning_rate = 0.5`
- `max_depth = 8`
- `n_estimators = 50`
- `colsample_bytree = 0.9`
- `gamma = 0.01`
- `reg_alpha = 1.3541666666666667`
- `reg_lambda = 1.6666666666666667`

SHAP del modelo

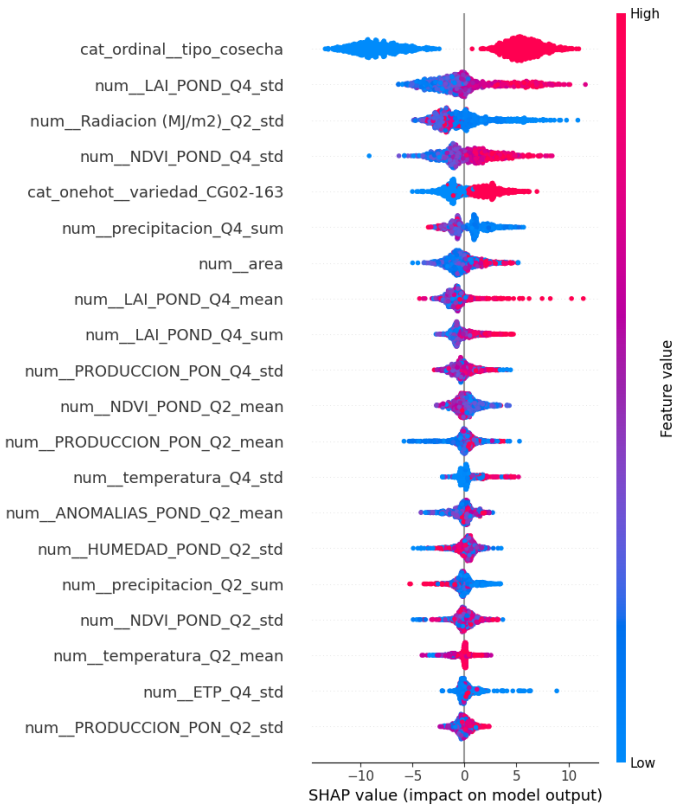


Figura 7.23: Análisis SHAP para el modelo `XGBoost sin escalado adicional`

Modelo 1: `XGBoost` con `MaxAbsScaler`

Este modelo aplicó un escalado de características utilizando `MaxAbsScaler` antes de entrenar el modelo `XGBoost`. Los parámetros utilizados fueron:

- `learning_rate = 0.3`
- `max_depth = 6`
- `n_estimators = 100`
- `colsample_bytree = 1`
- `gamma = 0`
- `reg_alpha = 0`
- `reg_lambda = 1`

SHAP del modelo

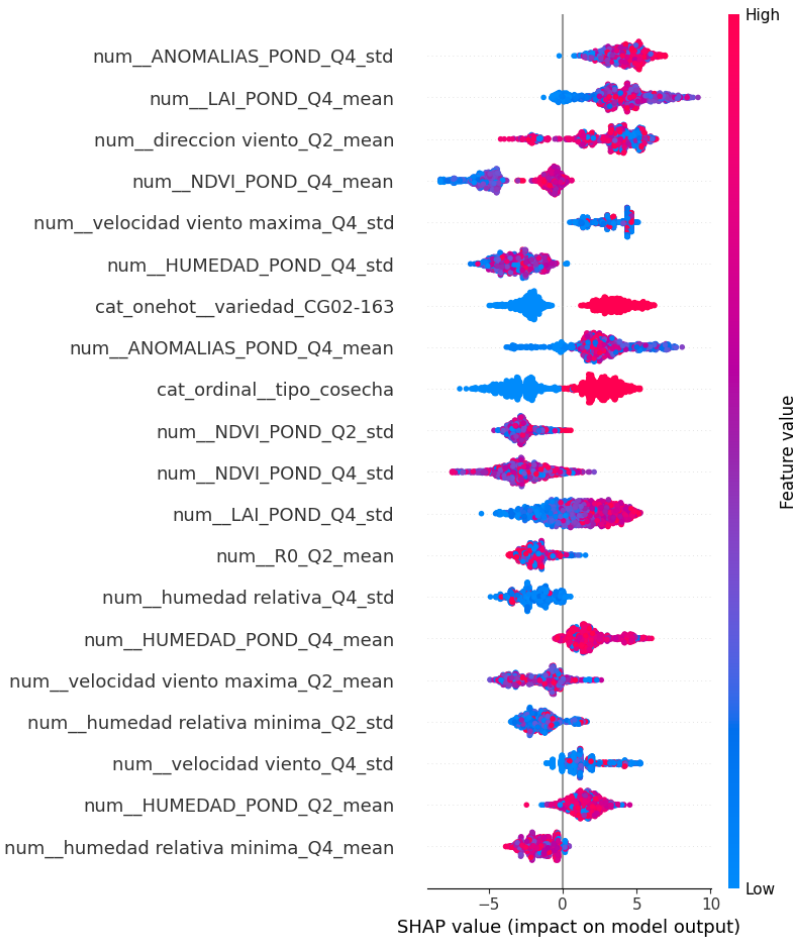


Figura 7.24: Análisis SHAP para el modelo `XGBoost con MaxAbsScaler`

Modelo de ensamble: `VotingRegressor`

Los dos modelos `XGBoost` se combinaron utilizando `VotingRegressor` con los siguientes pesos:

- Modelo 0: 0.5714285714285714
- Modelo 1: 0.42857142857142855

Resultados del modelo

El modelo de ensamble `VotingRegressor` logró los siguientes resultados en el conjunto de prueba:

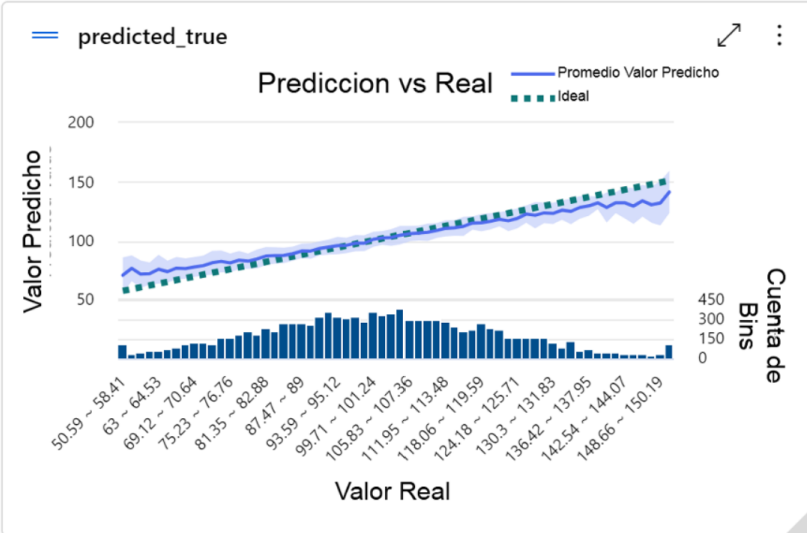


Figura 7.25: Métricas de rendimiento y gráfica de predicciones vs valores reales para el modelo `VotingEmsamble 4 Meses`

- R^2 Score: 0.8174278
- Root Mean Squared Error (RMSE): 8.638410

Estos resultados indican un buen rendimiento del modelo, explicando aproximadamente el 81.74 % de la variabilidad en el TCH y con un error promedio de predicción de alrededor de 8.64 toneladas por hectárea.

7.3.4. Modelo de regresión a 6 meses

Este modelo de regresión se entrenó utilizando un enfoque de ensamble con cuatro modelos diferentes, combinando LGBMRegressor y XGBRegressor, con el objetivo de predecir el tonelaje de caña por hectárea (TCH) con 6 meses de anticipación.

Preparación de datos

Para este modelo, se utilizó el conjunto de datos procesado según lo descrito en la Sección 6.3

Transformaciones de datos

- **Preprocesador común**: Aplicado a todos los modelos, incluyendo imputación y codificación one-hot para variables categóricas.
- **Escalado específico**: Cada modelo utilizó un escalador diferente antes del entrenamiento.

Modelos implementados

Modelo 0: `LGBMRegressor` con `StandardScaler`

Este modelo aplicó `StandardScaler` antes de entrenar el modelo `LGBMRegressor`. Algunos parámetros clave fueron:

- `learning_rate = 0.1789484210526316`
- `max_depth = 8`
- `n_estimators = 600`
- `num_leaves = 255`
- `colsample_bytree = 0.4`
- `subsample = 0.6`

SHAP del modelo

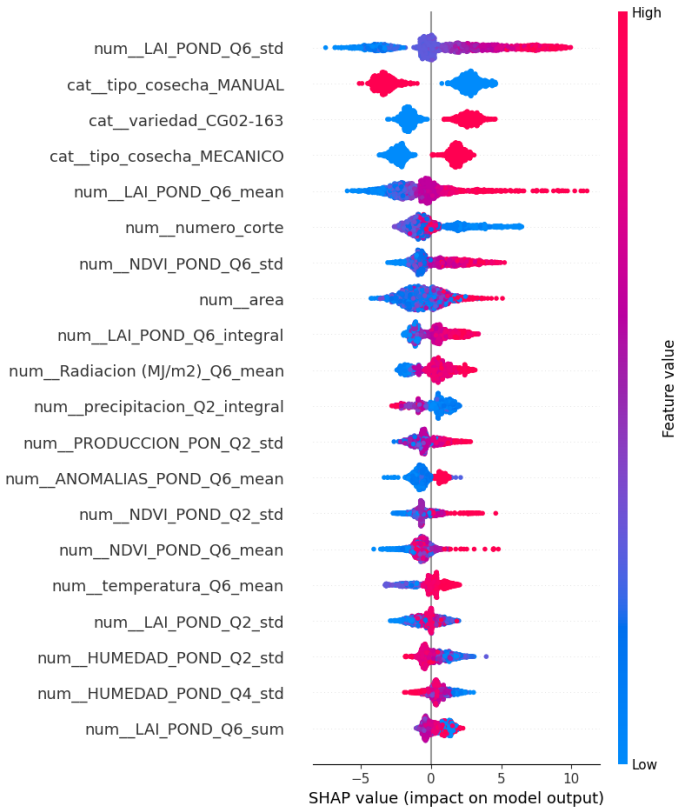


Figura 7.26: Análisis SHAP para el modelo LGBMRegressor con StandardScaler

Modelo 1: `LGBMRegressor` **con** `Normalizer`

Este modelo aplicó `Normalizer` antes de entrenar otro `LGBMRegressor`. Parámetros clave:

- `learning_rate = 0.042113157894736845`
- `max_depth = 9`
- `n_estimators = 800`
- `num_leaves = 63`
- `colsample_bytree = 0.2`
- `subsample = 0.75`

SHAP del modelo

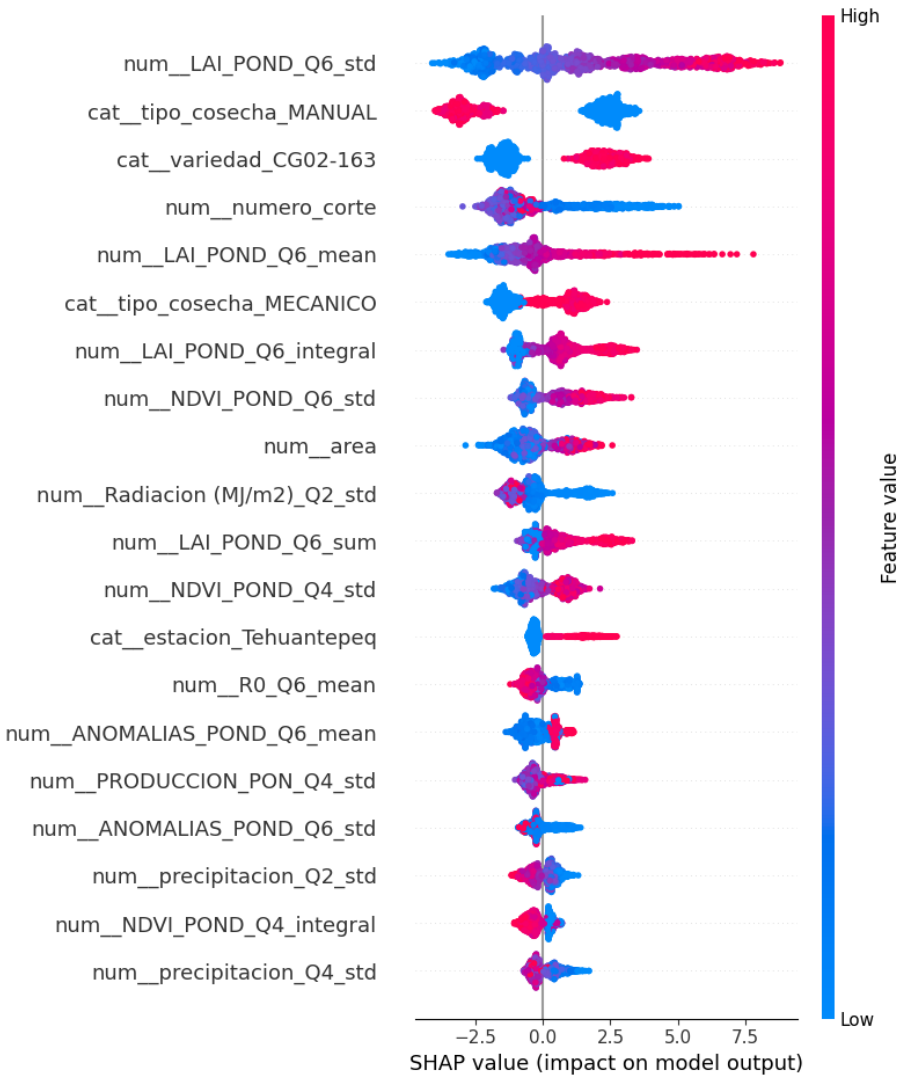


Figura 7.27: Análisis SHAP para el modelo LGBMRegressor con Normalizer

Modelo 2: `LGBMRegressor` **con** `StandardScaler`

Similar al Modelo 0, pero con una configuración diferente:

- `learning_rate = 0.16842263157894738`
- `max_depth = 9`
- `n_estimators = 100`
- `num_leaves = 255`
- `colsample_bytree = 0.6`
- `subsample = 0.9`

SHAP del modelo

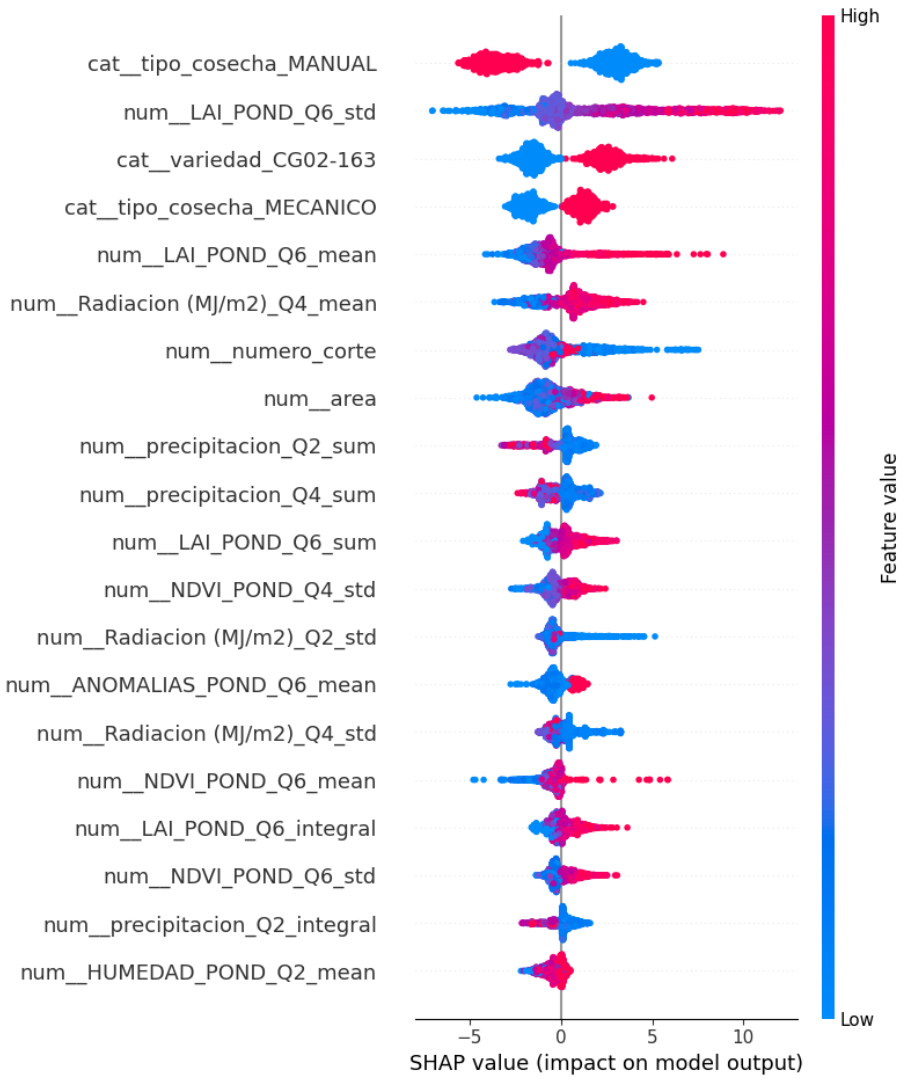


Figura 7.28: Análisis SHAP para el segundo modelo LGBMRegressor con StandardScaler

Modelo 3: `XGBRegressor` con `MaxAbsScaler`

Este modelo utilizó `XGBRegressor` con `MaxAbsScaler`:

- Parámetros por defecto de XGBRegressor
- `verbosity = 0`
- `n_jobs = -1`
- `random_state = 42`

SHAP del modelo

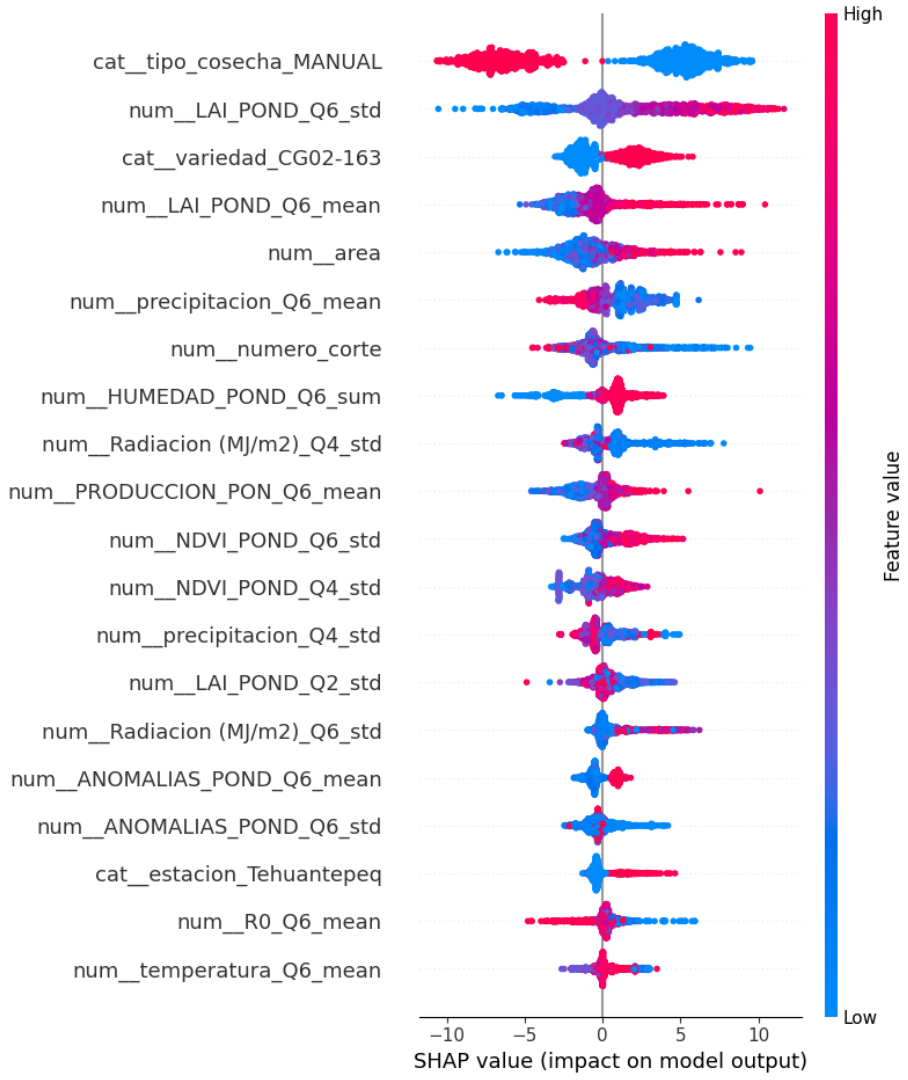


Figura 7.29: Análisis SHAP para el modelo XGBRegressor con MaxAbsScaler

Modelo de ensamble: `VotingRegressor`

Los cuatro modelos se combinaron utilizando `VotingRegressor` con los siguientes pesos:

- Modelo 0: 0.46153846153846156
- Modelo 1: 0.23076923076923078
- Modelo 2: 0.07692307692307693
- Modelo 3: 0.23076923076923078

Resultados del modelo

El modelo de ensamble `VotingRegressor` logró los siguientes resultados en el conjunto de prueba:

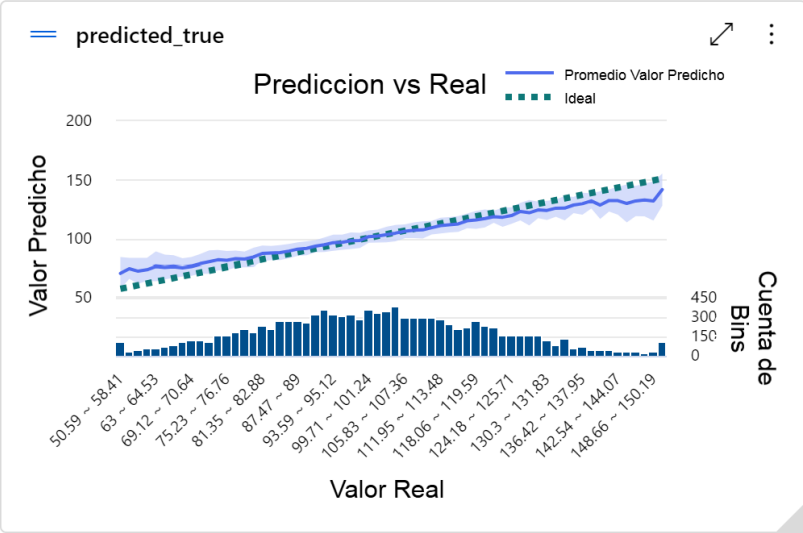


Figura 7.30: Métricas de rendimiento y gráfica de predicciones vs valores reales para el modelo `VotingEmsamble 6 Meses`

- R^2 Score: 0.8056124
- Root Mean Squared Error (RMSE): 8.915656

Estos resultados indican un buen rendimiento del modelo, explicando aproximadamente el 80.56 % de la variabilidad en el TCH y con un error promedio de predicción de alrededor de 8.92 toneladas por hectárea.

7.3.5. Modelo de regresión a 8 meses

Este modelo de regresión se entrenó utilizando un enfoque de ensamble con tres modelos XGBoost, con el objetivo de predecir el tonelaje de caña por hectárea (TCH) con 8 meses de anticipación.

Preparación de datos

Para este modelo, se utilizó el conjunto de datos procesado según lo descrito en la Sección 6.3. Los pasos clave incluyeron:

- **División temporal de datos**:
  - Conjunto de entrenamiento: Todas las zafras anteriores a '23-24'.
  - Conjunto de prueba: La zafra '23-24'.
- **Identificación de tipos de columnas**:
  - Columnas numéricas: Identificadas automáticamente (tipos int64 y float64).
  - Columnas categóricas: Identificadas automáticamente (tipos object y category).
- **Preprocesamiento**:
  - Datos numéricos: Imputación de valores faltantes con la media.
  - Datos categóricos: Imputación de valores faltantes con la moda y codificación one-hot.

Descripción del modelo y transformaciones de datos

Se implementaron tres modelos XGBoost con diferentes configuraciones, que luego se combinaron en un modelo de ensamble. El proceso fue el siguiente:

Transformaciones de datos

- **Preprocesador común**: Aplicado a todos los modelos, incluyendo imputación y codificación one-hot para variables categóricas.
- **Escalado específico**: Cada modelo utilizó un escalador diferente antes del entrenamiento.

Modelos implementados

Se implementaron tres modelos XGBoost con diferentes configuraciones, que luego se combinaron en un modelo de ensamble mediante `VotingRegressor`. A continuación, se detalla cada modelo:

Modelo 0: `XGBoost` con `StandardScaler`

Este modelo aplicó `StandardScaler` antes de entrenar el modelo `XGBoost`. Los parámetros utilizados fueron:

- `colsample_bytree = 0.6`
- `learning_rate = 0.3`
- `max_depth = 8`
- `max_leaves = 3`
- `n_estimators = 100`
- `reg_alpha = 1.25`
- `reg_lambda = 0.10416666666666667`
- `subsample = 0.7`

SHAP del modelo

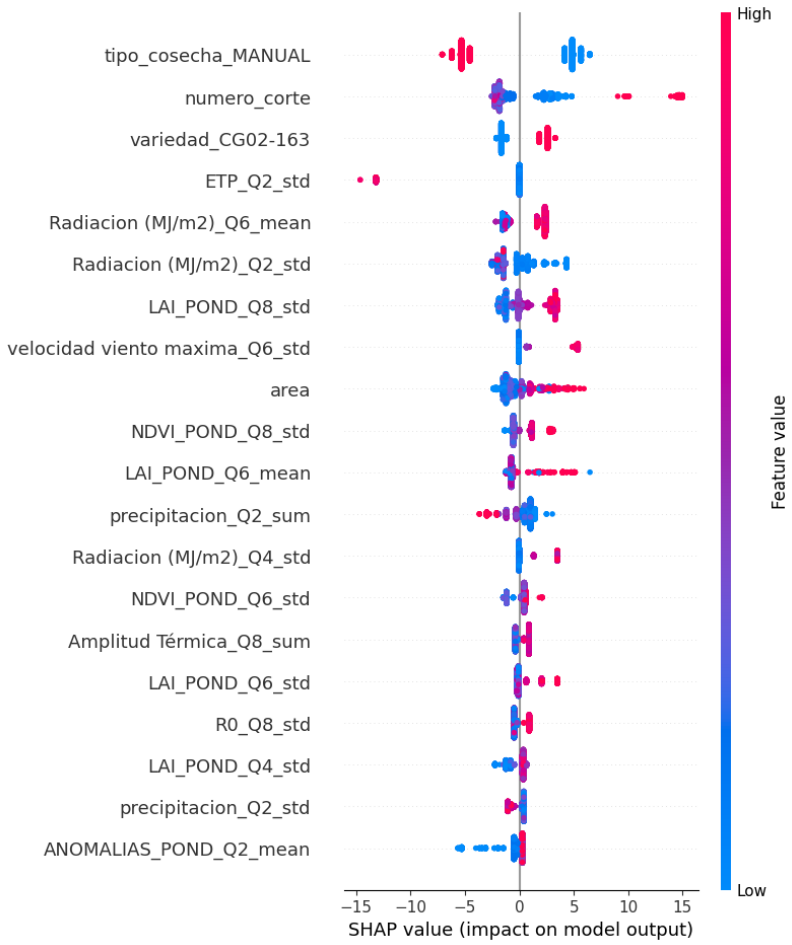


Figura 7.31: Análisis SHAP para el modelo XGBoost con StandardScaler

Modelo 1: `XGBoost` con `StandardScaler`

Este modelo también utilizó `StandardScaler` pero con una configuración diferente de `XGBoost`:

- `colsample_bytree = 0.9`
- `learning_rate = 0.5`
- `max_depth = 8`
- `n_estimators = 50`
- `reg_alpha = 1.3541666666666667`
- `reg_lambda = 1.6666666666666667`
- `subsample = 1`
- `gamma = 0.01`

SHAP del Modelo

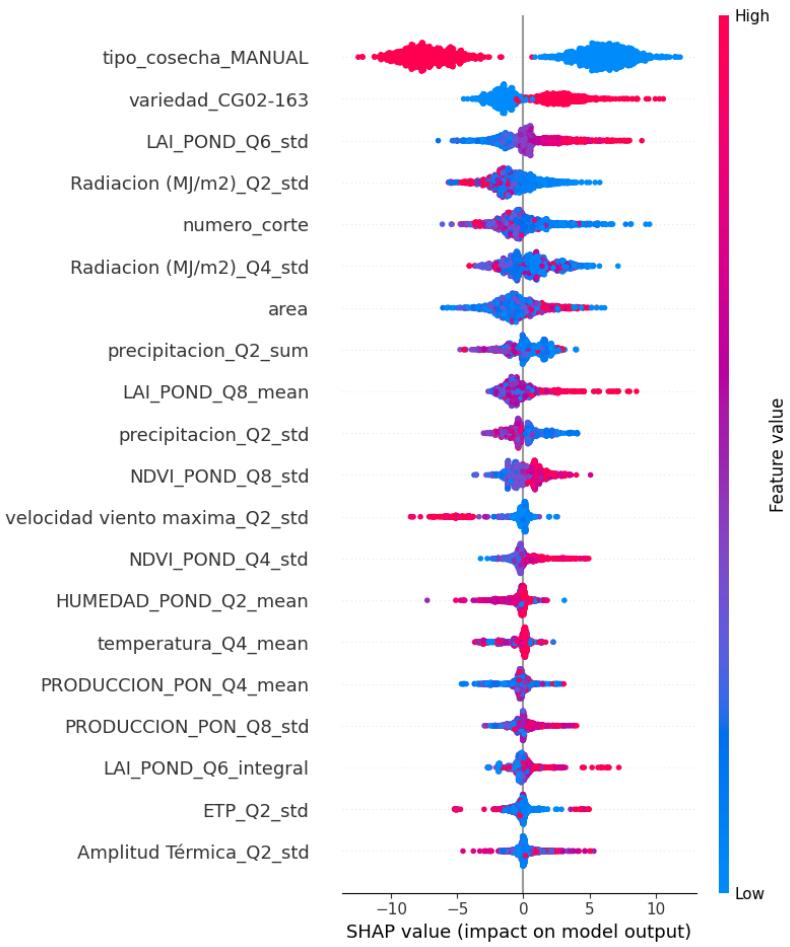


Figura 7.32: Análisis SHAP para el modelo XGBoost con StandardScaler y configuración personalizada

Modelo 2: XGBoost con MaxAbsScaler Este modelo aplicó `MaxAbsScaler` antes de entrenar el modelo `XGBoost`:

- `colsample_bytree = 1`
- `learning_rate = 0.3`
- `max_depth = 6`
- `n_estimators = 100`
- `reg_alpha = 0`
- `reg_lambda = 1`
- `subsample = 1`

SHAP del modelo

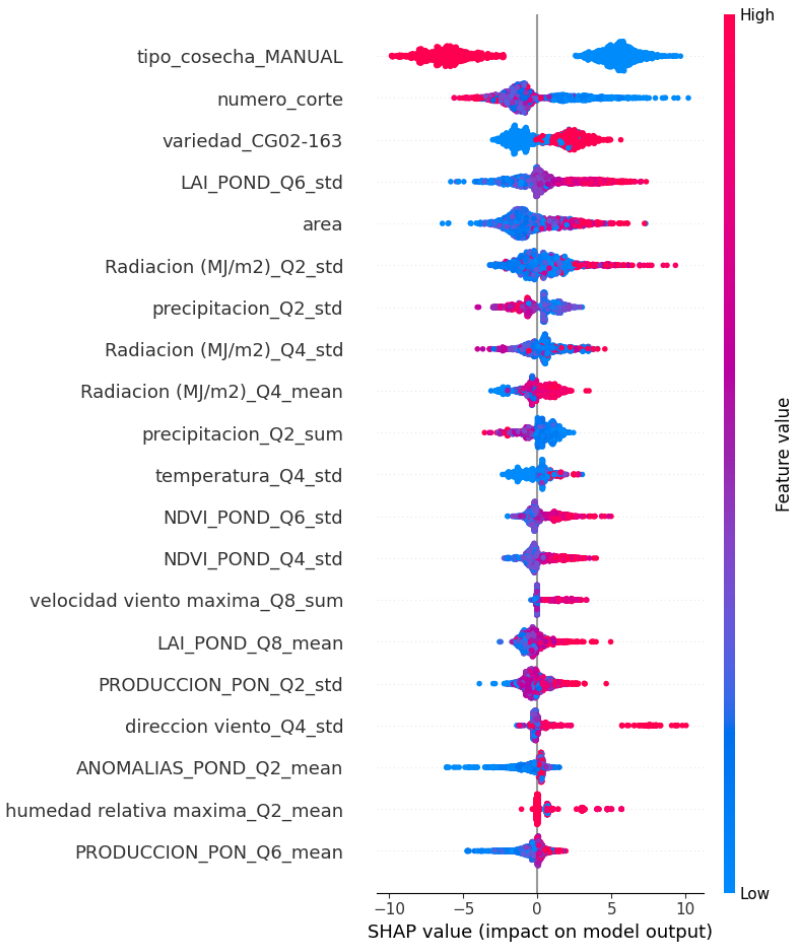


Figura 7.33: Análisis SHAP para el modelo XGBoost con MaxAbsScaler

Modelo de ensamble: `VotingRegressor`

Los tres modelos `XGBoost` se combinaron utilizando `VotingRegressor` con los siguientes pesos:

- Modelo 0: 0.4666666666666667
- Modelo 1: 0.3333333333333333
- Modelo 2: 0.2

Resultados del modelo

El modelo de ensamble `VotingRegressor` logró los siguientes resultados en el conjunto de prueba:

- R^2 Score: 0.8062667
- Root Mean Squared Error (RMSE): 8.897884

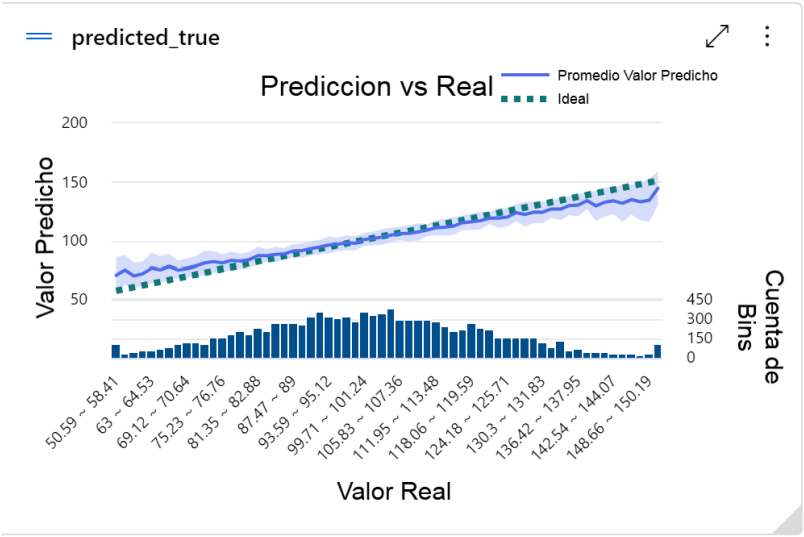


Figura 7.34: Métricas de rendimiento y gráfica de predicciones vs valores reales para el modelo `VotingEmsamble 8 Meses`

Estos resultados indican un buen rendimiento del modelo, explicando aproximadamente el 80.63 % de la variabilidad en el TCH y con un error promedio de predicción de alrededor de 8.90 toneladas por hectárea.

7.3.6. Modelo de Regresión a 10 Meses

Este modelo de regresión se entrenó utilizando un enfoque de ensamble con cinco modelos diferentes, combinando XGBoost y LightGBM, con el objetivo de predecir el tonelaje de caña por hectárea (TCH) con 10 meses de anticipación.

Preparación de datos

Para este modelo, se utilizó el conjunto de datos procesado según lo descrito en la Sección 6.3.

Descripción del modelo y transformaciones de datos

Se implementaron cinco modelos con diferentes configuraciones, que luego se combinaron en un modelo de ensamble. El proceso fue el siguiente:

Transformaciones de datos

- **Feature union**: Se utilizó FeatureUnion para combinar diferentes transformaciones para grupos de columnas específicos.
- **Escalado específico**: Cada modelo utilizó un escalador diferente antes del entrenamiento.

Modelos implementados

Se implementaron cinco modelos con diferentes configuraciones, que luego se combinaron en un modelo de ensamble mediante `VotingRegressor`. A continuación, se detalla cada modelo:

Modelo 0: `XGBoost` con `StandardScaler`

Este modelo aplicó `StandardScaler` antes de entrenar el modelo `XGBoost`. Algunos parámetros clave fueron:

- `colsample_bytree = 0.6`
- `learning_rate = 0.4`
- `max_depth = 10`
- `n_estimators = 50`
- `subsample = 0.8`

SHAP del modelo

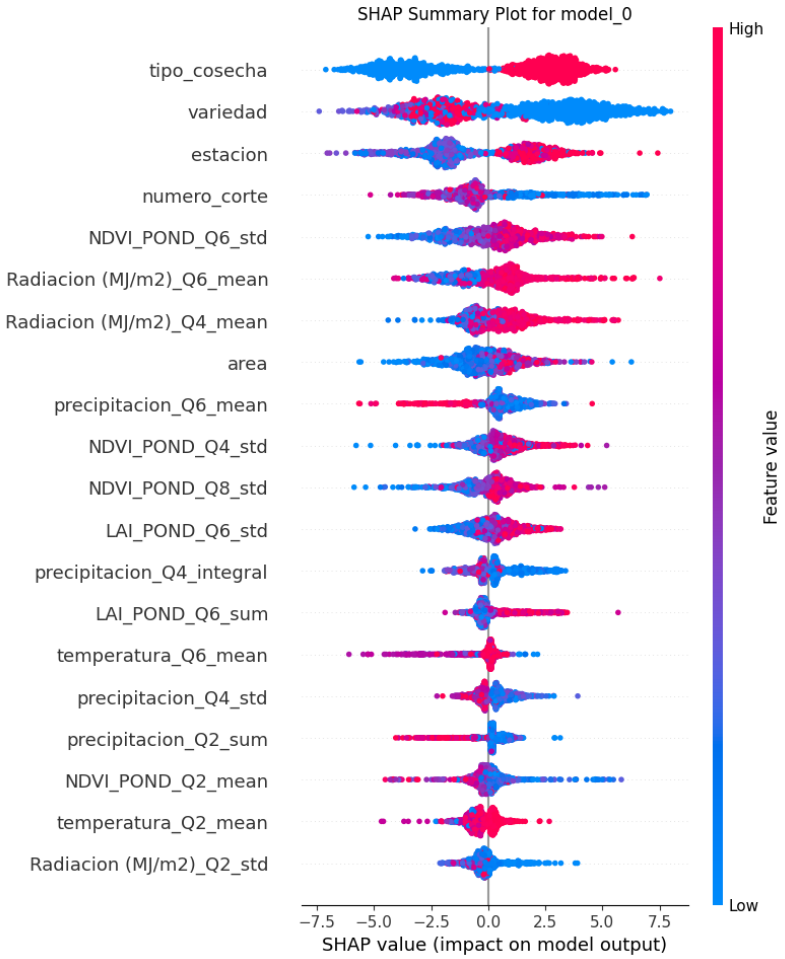


Figura 7.35: Análisis SHAP para el modelo XGBoost con StandardScaler

Modelo 1: `XGBoost` con `StandardScaler`

Este modelo también utilizó `StandardScaler` pero con una configuración diferente de `XGBoost`:

- `colsample_bytree = 0.9`
- `learning_rate = 0.5`
- `max_depth = 8`
- `n_estimators = 50`
- `reg_alpha = 1.3541666666666667`
- `reg_lambda = 1.6666666666666667`

SHAP del modelo

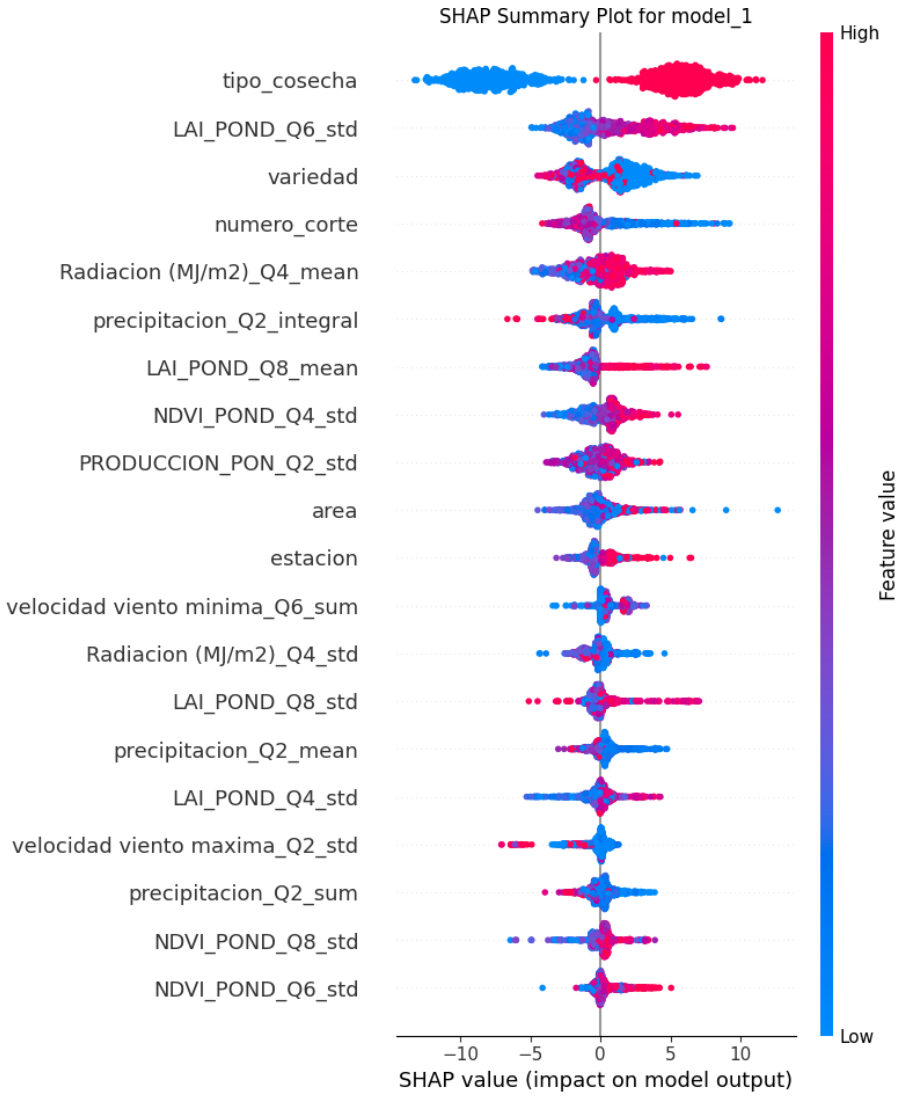


Figura 7.36: Análisis SHAP para el segundo modelo XGBoost con StandardScaler

Modelo 2: `LightGBM` **con** `StandardScaler`

Este modelo utilizó `LGBMRegressor` con `StandardScaler`:

- `colsample_bytree = 0.6`
- `learning_rate = 0.16842263157894738`
- `max_depth = 9`
- `n_estimators = 100`
- `num_leaves = 255`
- `subsample = 0.9`

SHAP del modelo

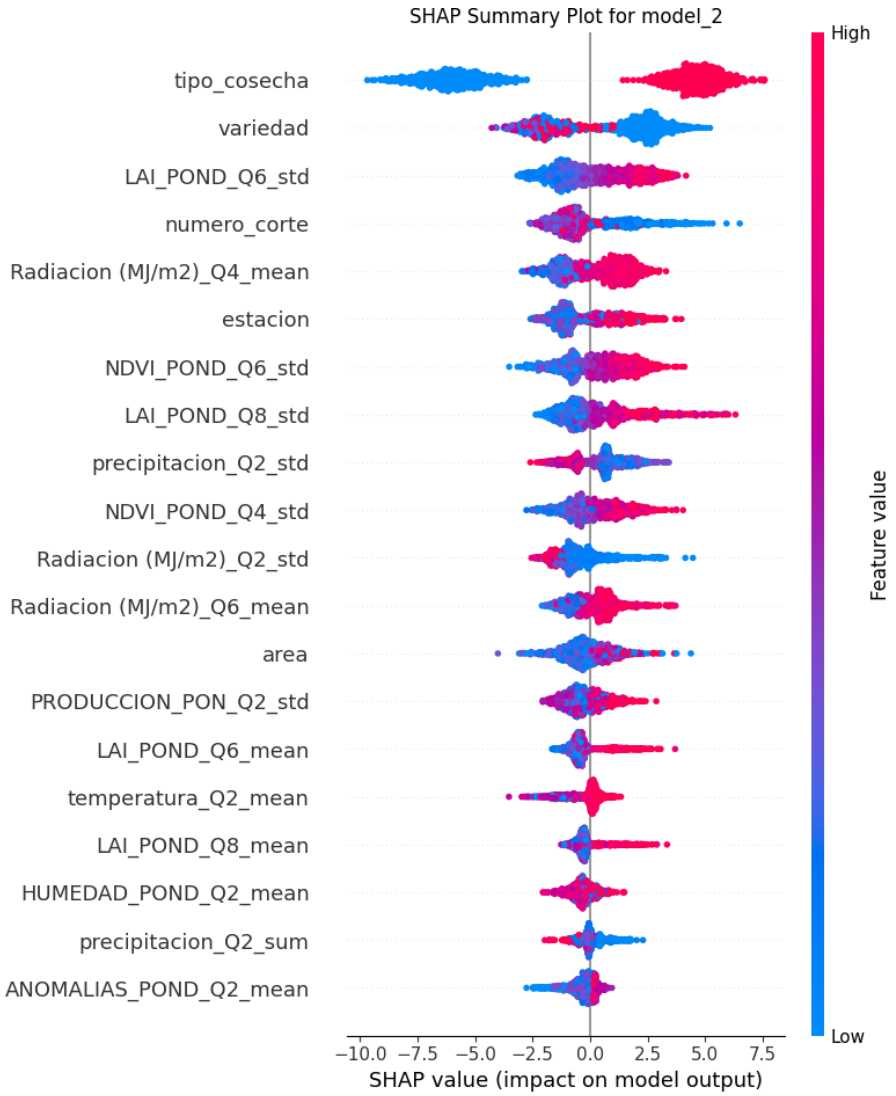


Figura 7.37: Análisis SHAP para el modelo LightGBM con StandardScaler

Modelo 3: `XGBoost` con `MaxAbsScaler`

Este modelo aplicó `MaxAbsScaler` antes de entrenar el modelo `XGBoost`:

- `max_depth = 6`
- `n_estimators = 100`

SHAP del modelo

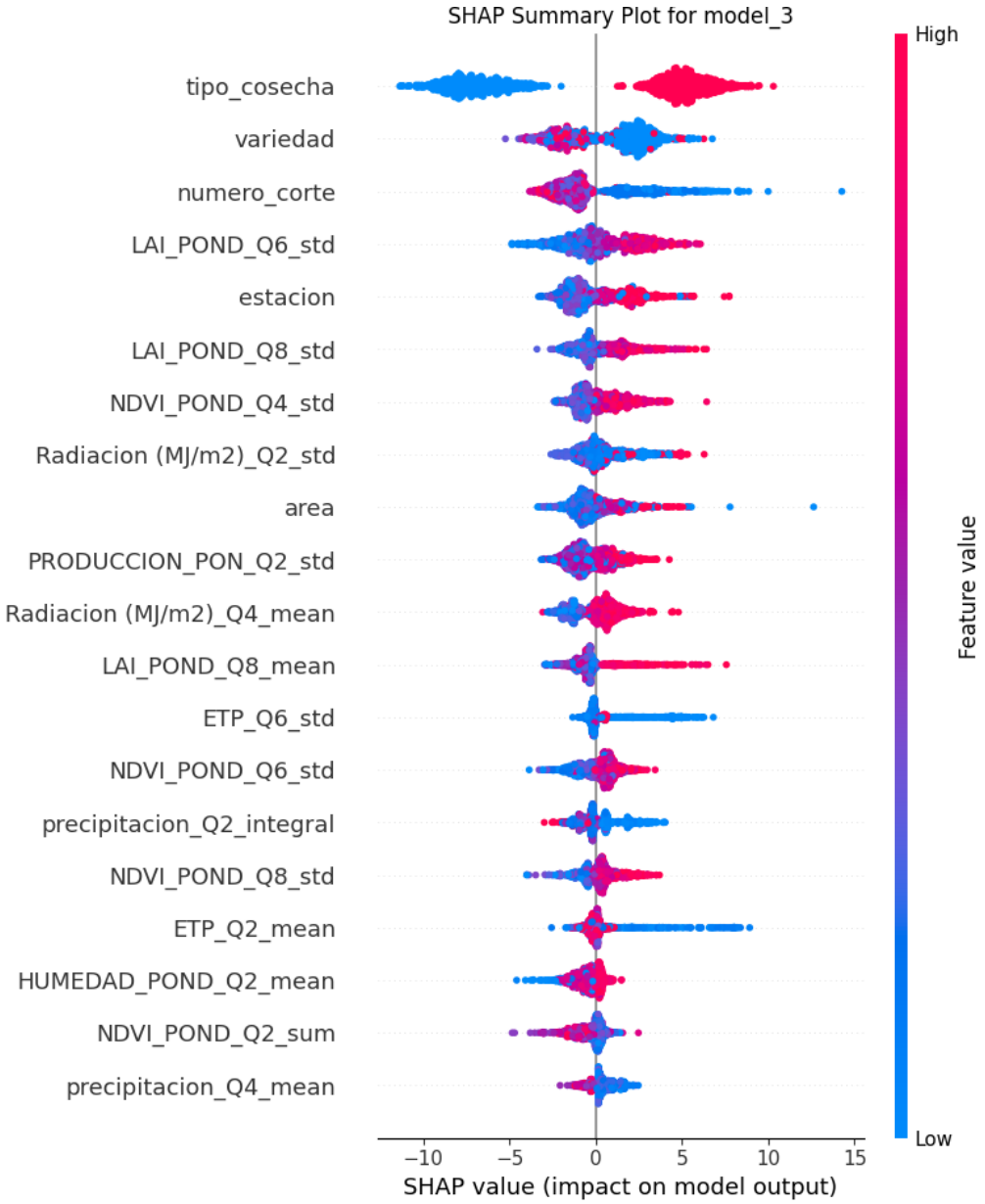


Figura 7.38: Análisis SHAP para el modelo XGBoost con MaxAbsScaler

Modelo 4: `XGBoost` con `StandardScaler`

Este modelo aplicó `StandardScaler` antes de entrenar otro modelo `XGBoost`:

- `colsample_bytree = 0.5`
- `max_depth = 9`
- `n_estimators = 50`
- `reg_alpha = 0.3125`
- `reg_lambda = 1.5625`
- `subsample = 0.5`

SHAP del modelo

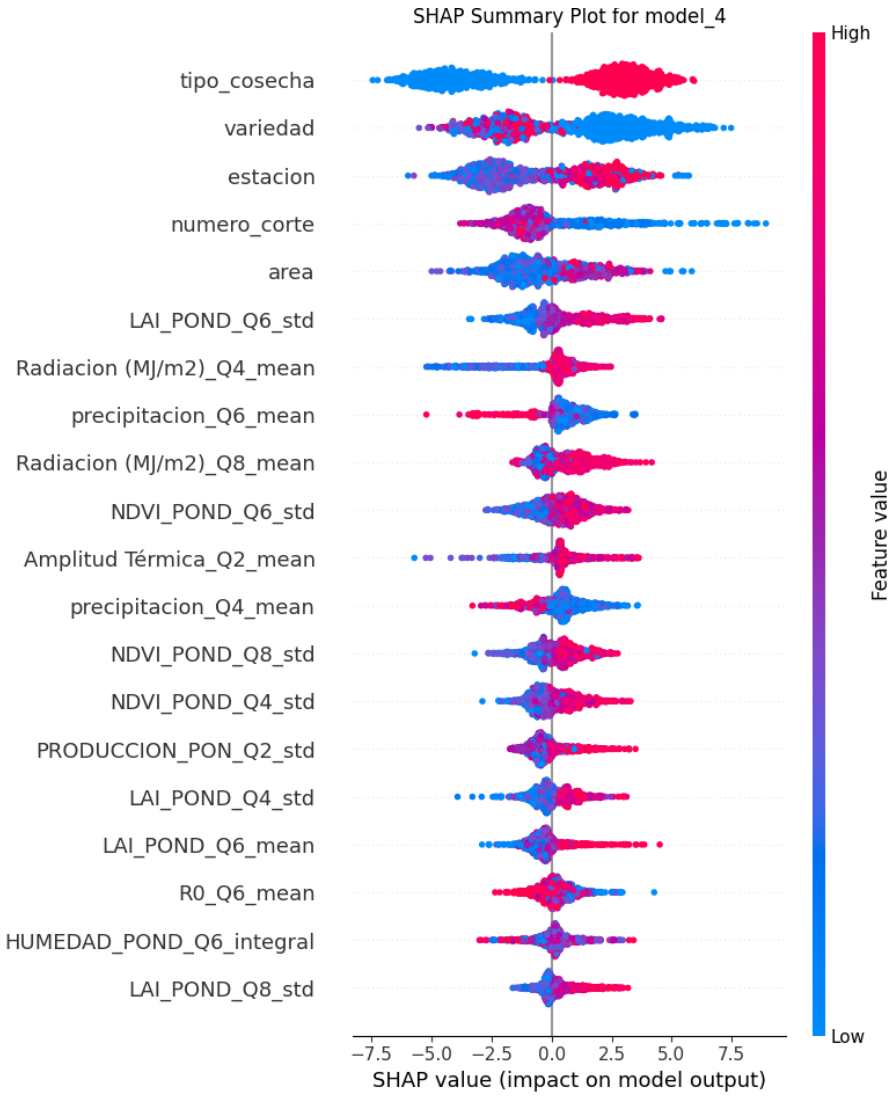


Figura 7.39: Análisis SHAP para el segundo modelo XGBoost con StandardScaler

Modelo de ensamble: `VotingRegressor`

Los cinco modelos se combinaron utilizando `VotingRegressor` con los siguientes pesos:

- Modelo 0: 0.4
- Modelo 1: 0.3
- Modelo 2: 0.1
- Modelo 3: 0.1
- Modelo 4: 0.1

Resultados del modelo

El modelo de ensamble `VotingRegressor` logró los siguientes resultados en el conjunto de prueba:

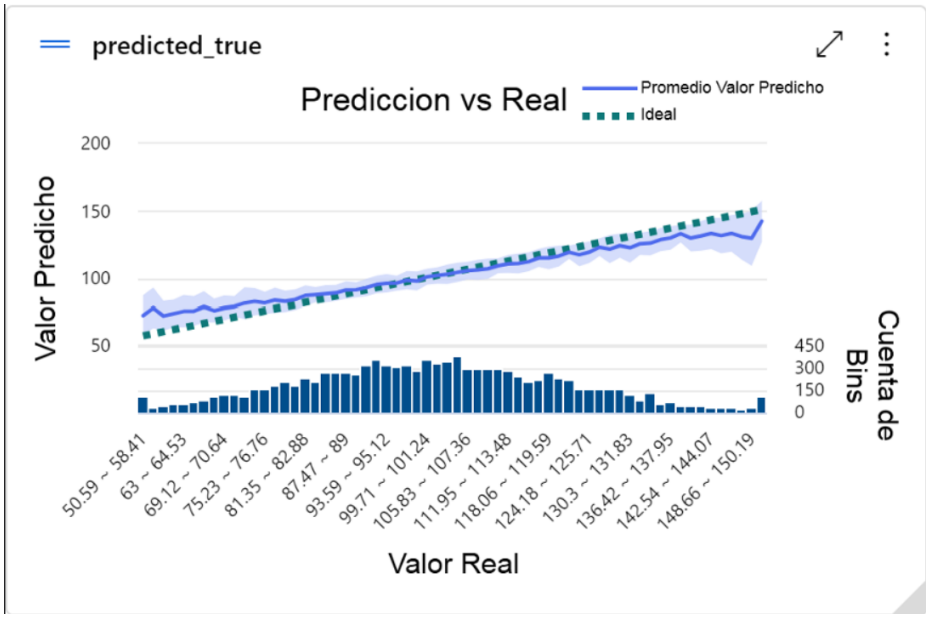


Figura 7.40: Métricas de rendimiento y gráfica de predicciones vs valores reales para el modelo `VotingEmsamble 10 Meses`

- R^2 Score: 0.7786154
- Root Mean Squared Error (RMSE): 9.513731

Estos resultados indican un buen rendimiento del modelo, explicando aproximadamente el 77.86 % de la variabilidad en el TCH y con un error promedio de predicción de alrededor de 9.51 toneladas por hectárea.

7.3.7. Rendimiento de los modelos de Azure

En contraste, los modelos implementados en Azure, particularmente los ensambles basados en `XGBoost` y `LightGBM`, mostraron un rendimiento significativamente mejor. A continuación, se presenta una tabla comparativa de los resultados:

Modelo	R^2	RMSE
Regresión a 2 Meses	0.817428	8.638410
Regresión a 4 Meses	0.817428	8.638410
Regresión a 6 Meses	0.805612	8.915656
Regresión a 8 Meses	0.806267	8.897884
Regresión a 10 Meses	0.778615	9.513731

Tabla 7.5: Comparación de resultados de modelos de regresión en Azure

7.4. EndPoints API en FLASK

Endpoint de predicciones de modelos

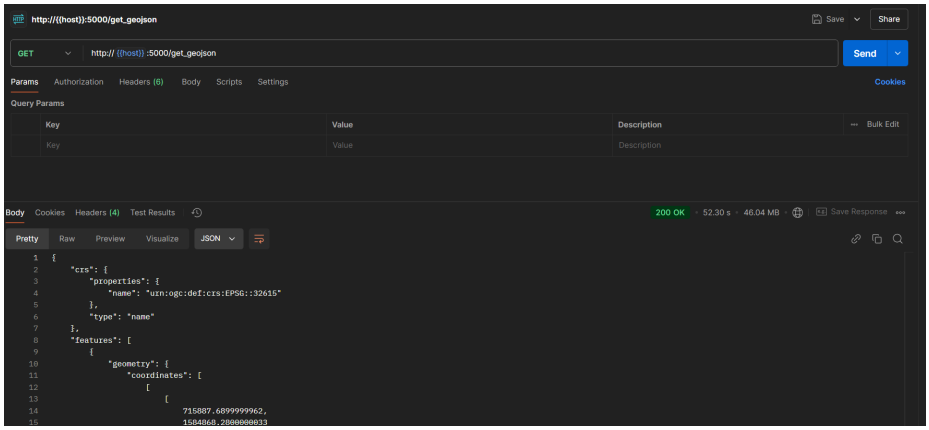


Figura 7.41: Route: get_geojson

Endpoint de estadísticas de modelos

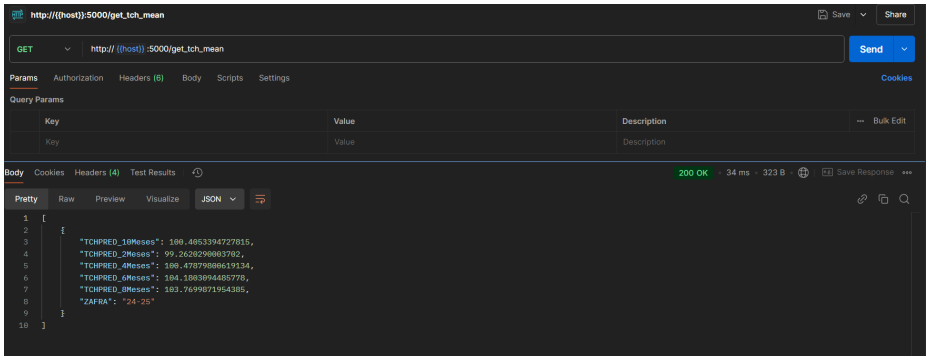


Figura 7.42: Route: get_tch_mean

CAPÍTULO 8

Análisis de resultados

Los modelos de Deep Learning implementados mostraron un claro caso de sobreajuste, con un R^2 de -0.0000 y un RMSE de 23.0635 para el modelo de regresión, a pesar de implementar múltiples técnicas de regularización. Este fenómeno puede atribuirse a varios factores críticos. Lo primero es la arquitectura de la red neuronal, aunque se probó con múltiples capas y un patrón de .ªscenso-descenso", resultó excesivamente compleja para la naturaleza de los datos procesados. Esto pudo ser producto de la relación no lineal de los datos de la caña de azúcar, que requieren un enfoque más profundo del que se puede obtener de una red neuronal profunda generalista. La variabilidad inherente en los datos agrícolas, influenciada por los múltiples factores ambientales y de manejo, además del procesamiento para reducir la dimensionalidad de los datos a predicción anual, creó patrones demasiado complejos para ser captados efectivamente por la arquitectura propuesta.

Dentro de las técnicas de regularización de los modelos de aprendizaje profundo, el dropout de 0.25 no logró prevenir efectivamente la coadaptación de las neuronas. El fallo del dropout y la arquitectura simplificada demuestra que el problema no era simplemente de complejidad excesiva, sino de un desajuste fundamental entre la estructura del modelo y la naturaleza de los datos agrícolas. Las relaciones temporales y las dependencias complejas entre variables requieren un enfoque más sofisticado que el dropout uniforme y una arquitectura feed-forward simple no pueden proporcionar. De la misma forma, las otras técnicas de regularización L1 y L2 no lograron controlar la complejidad del modelo. El early stopping tampoco pudo compensar la tendencia del modelo a memorizar los patrones, esto debido a que la caída del loss era muy temprana y el modelo no aprendía a generalizar con los datos, como se observa en la Figura 7.12.

La falta de interpretabilidad de los modelos de Deep Learning presentó un obstáculo significativo, ya que los intentos de aplicar análisis SHAP resultaron computacionalmente inviables. Esta limitación impidió entender cómo el modelo estaba procesando las diferentes características, información que hubiera sido valiosa para mejorar su capacidad de generalización con los datos.

Por otro lado, los modelos generados en una plataforma de AutoML, en este caso Azure, se entrenaron con otro enfoque que les permitió obtener mejores resultados, como se observa en la Tabla 7.5. Los modelos lograron un buen ajuste y consiguieron generalizar sobre los datos, lo cual puede atribuirse a diversos factores, principalmente a la concatenación de modelos mediante VotingEnsemble, el uso de modelos de Gradient Boosting para trabajar relaciones no lineales y el empleo

de submuestreos con técnicas de regularización dentro de Azure.

El VotingEnsemble de Azure destaca por su capacidad de integrar múltiples métodos con pesos adaptativos y diferentes tipos de modelos, manteniendo un equilibrio entre complejidad e interpretabilidad. En contraste, el Boosting, a pesar de su potencial, presenta una mayor tendencia al sobreajuste debido a dos aspectos fundamentales de su diseño: su naturaleza secuencial, donde cada modelo subsiguiente se concentra en corregir los errores de sus predecesores, lo que puede resultar en una memorización excesiva del ruido en los datos y una sobreespecialización en casos atípicos; y su sistema de ponderación de instancias que, al aumentar el peso de los casos mal clasificados, puede dar demasiada importancia a outliers o datos ruidosos, comprometiendo la capacidad de generalización del modelo. Por estas características, aunque el Boosting podría ofrecer una precisión ligeramente superior en situaciones específicas, el VotingEnsemble demostró ser una solución más balanceada y práctica para el contexto agrícola en cuestión, proporcionando un mejor equilibrio entre precisión, interpretabilidad y robustez en las predicciones.

Los modelos de Gradient Boosting, como XGBoost y LightGBM utilizados en este proyecto, demostraron ser particularmente efectivos para trabajar con datos agrícolas debido a su capacidad inherente para capturar relaciones no lineales a través de su estructura de árboles de decisión secuenciales. Su eficacia radica en que cada árbol nuevo se construye para corregir los errores residuales del conjunto anterior, permitiendo modelar interacciones complejas entre variables, como la relación no lineal entre el clima, los índices de vegetación y el rendimiento del cultivo. Esta capacidad se potencia mediante la optimización del gradiente de la función de pérdida, que permite al modelo ajustarse progresivamente a patrones sutiles en los datos sin asumir una forma funcional específica de la relación entre variables. Además, estos modelos manejan naturalmente diferentes tipos de variables (numéricas y categóricas) y son robustos ante valores atípicos y datos faltantes, características comunes en datos agrícolas. Como se evidencia en los resultados del proyecto, donde se alcanzaron R^2 superiores a 0.80, los modelos de Gradient Boosting pudieron capturar efectivamente las complejas interacciones entre factores como la radiación solar, la precipitación, los índices vegetativos y el rendimiento final de la caña de azúcar, sin requerir transformaciones explícitas de las variables para modelar estas relaciones no lineales.

El AutoML de Azure implementó varias técnicas estratégicas para controlar el sobreajuste en los modelos de Gradient Boosting, marcando una diferencia significativa con los modelos de Deep Learning que presentaron overfitting. La clave del éxito radicó en la combinación metódica de varios enfoques: primero, aplicó diferentes técnicas de regularización en los modelos XGBoost y LightGBM, ajustando automáticamente parámetros como reg_alpha y reg_lambda para penalizar la complejidad del modelo; segundo, implementó submuestreo de características (colsample_bytree) y de instancias (subsample), como se observa en las configuraciones de los modelos donde estos valores oscilaban entre 0.5 y 0.9, reduciendo la dependencia de subconjuntos específicos de datos. Adicionalmente, AutoML optimizó la profundidad de los árboles (max_depth entre 6 y 10) y el número de estimadores, encontrando un balance óptimo entre la capacidad de modelado y la generalización. En contraste, los modelos de Deep Learning, con sus 16 capas, sufrieron de sobreajuste debido a su excesiva capacidad de memorización y la falta de estas restricciones automatizadas, resultando en un R^2 negativo comparado con el 0.80 consistente de los modelos de Gradient Boosting. Esta diferencia demuestra cómo las técnicas de control de complejidad implementadas por AutoML fueron cruciales para mantener la capacidad predictiva sin sacrificar la generalización.

El análisis de las características mediante SHAP nos deja bastante claro cuáles son los factores que se deben tener en consideración para mejorar las toneladas de azúcar por hectárea. La característica principal en la mayoría de los modelos es el tipo de cosecha, que se divide en cosecha manual con machete o mecánica con tractor. La relación que nos muestran los modelos indica que la cosecha mecánica es más eficiente que la manual, lo cual podría deberse a que las áreas cosechadas con maquinaria tienen menos defectos en el suelo y son más fáciles de trabajar que aquellas que requieren intervención manual. Otra variable presente como característica principal en muchos modelos es la cantidad de cortes que tiene un terreno, ya que después de 2 cortes podemos observar que empieza

a generar una regresión negativa para el TCH. Esto podría deberse a que la tierra empieza a perder nutrientes importantes por el crecimiento de la caña, lo que sugeriría la necesidad de implementar rotación de cultivos para prevenirlo.

Dentro del analisis de shap surgieron las variables de la variedad de la caña, el cual el que tuvo un mejor resultado fue el CG02-163, este proviene del Centro Guatemalteco de Investivacion, CEGICAÑA, desarrollada en 2002. El porque esta variedad tuvo un mejor desempeño que las otras se puede deber a que fue adaptado a las condiciones climaticas de Guatemala que son muy dispersas. Estas variables en especifico tienen resistencia a enfermedades comonues por lo que le agrega un plus en cuanto remuneracion y ahorro en pesticidas a la hora de plantarlo. Adicionalmente existe una varible llamada LAI, Leaf Area Index por sus siglas en ingles o indice de area foliar. Este indice nos indica la cantidad de area foliar por unidad de suelo en la superficie, esto puede explicar el porque tambien esta presente con tatnta importancia en los modelos ya que nos relata cuanto esta creciendo cada planta dentro de un area y nos pone una metrica para poder controlar el riego que es algo muy critico de controlar ya que el TCH puede ser muy perjudicado con condiciones muy humedas.

La implementación de una API RESTful desarrollada en Flask constituyó un elemento transformador que elevó este proyecto más allá de un simple modelo predictivo, convirtiéndolo en un sistema integral y operativo. Esta API actúa como la columna vertebral del sistema, facilitando la automatización completa del pipeline de datos desde las fuentes del ICC y NAX Solutions, el procesamiento de datos climáticos e índices vegetales, y la actualización periódica del conjunto de datos de entrenamiento. Además, proporciona una gestión robusta de los modelos, permitiendo su reentrenamiento programado, versionamiento y despliegue automatizado a la nube mediante integración continua. Como interfaz de servicio, la API expone endpoints REST que facilitan el consumo de predicciones y permiten una integración fluida con los sistemas existentes del ingenio. Esta arquitectura no solo garantiza el mantenimiento y la escalabilidad del sistema, sino que también facilita su integración efectiva en el flujo de trabajo operativo, transformando un proyecto de ciencia de datos en una herramienta empresarial funcional y accesible. Este enfoque arquitectónico fue fundamental para asegurar que las predicciones de TCH pudieran ser utilizadas de manera efectiva en la toma de decisiones operativas del ingenio, cumpliendo así con el objetivo de crear un sistema no solo preciso, sino también práctico y utilizable en el contexto real de la industria azucarera.

CAPÍTULO 9

Conclusiones

- El desarrollo de un sistema avanzado de monitoreo y gestión de cultivos de caña de azúcar mediante tecnologías de análisis de datos y modelado predictivo se logró con éxito. El sistema demostró una capacidad significativamente mejorada para evaluar la madurez de los cultivos y predecir el rendimiento, superando las limitaciones de los métodos tradicionales y ofreciendo una herramienta valiosa para la industria azucarera.
- El análisis exhaustivo de datos históricos climatológicos y de imágenes satelitales resultó fundamental para comprender los factores que influyen en el rendimiento de los cultivos. Este enfoque permitió identificar patrones y relaciones complejas que no eran evidentes mediante métodos convencionales, sentando las bases para un modelado predictivo más preciso.
- El preprocesamiento de los datos fue crucial para el éxito del proyecto. La limpieza, normalización y transformación de los datos no solo mejoró la calidad de la información utilizada, sino que también permitió la integración efectiva de diversas fuentes de datos historicos. Este paso, aunque a menudo subestimado, demostró ser esencial para el desarrollo de modelos.
- El análisis exploratorio de datos reveló *insights* valiosos sobre los factores que influyen en el índice de Toneladas de Caña por Hectárea (TCH). Se identificaron variables clave como la precipitación, la radiación solar y ciertos índices vegetativos derivados de imágenes satelitales, que mostraron correlaciones significativas con el rendimiento de los cultivos. Este conocimiento no solo informó el desarrollo de los modelos predictivos, sino que también proporcionó información accionable para los agricultores y gestores de cultivos.
- El entrenamiento de múltiples modelos predictivos utilizando algoritmos de aprendizaje automático demostró ser una estrategia efectiva. La comparación entre diferentes enfoques, como las redes neuronales profundas y los modelos basados en árboles de decisión, permitió identificar el algoritmo más adecuado para esta tarea específica. Los modelos de ensamble, particularmente aquellos basados en `XGBoost` y `LightGBM`, mostraron un rendimiento superior, logrando un R^2 de hasta 0.817 y un RMSE tan bajo como 8.638 toneladas por hectárea en las predicciones a corto plazo.
- La validación de los modelos predictivos con datos reales de los cultivos fue un paso crítico que demostró la aplicabilidad práctica del sistema desarrollado. Esta fase no solo confirmó la precisión de las predicciones en condiciones reales, sino que también proporcionó retroalimentación valiosa para el refinamiento continuo de los modelos. La capacidad del sistema para adaptarse y mejorar con nuevos datos subraya su potencial como una herramienta dinámica para la gestión de cultivos.

CAPÍTULO 10

Recomendaciones

- Se recomienda explorar plataformas alternativas de MLOps como Kubeflow y MLflow para la gestión del ciclo de vida de los modelos. Estas plataformas de código abierto ofrecen capacidades robustas para el versionamiento de modelos, seguimiento de experimentos y automatización de pipelines, lo que podría complementar o reemplazar la dependencia actual de Azure. Además, proporcionan mayor flexibilidad y control sobre la infraestructura, permitiendo implementaciones tanto en la nube como on-premise.
- Se recomienda implementar técnicas adicionales de interpretabilidad de modelos más allá de SHAP. Esto incluye el uso de LIME (Local Interpretable Model-agnostic Explanations) para explicaciones locales, Partial Dependence Plots (PDP) para entender relaciones entre variables, y Accumulated Local Effects (ALE) plots para visualizar el impacto de características correlacionadas. Estas técnicas proporcionarían una comprensión más completa del comportamiento del modelo desde diferentes perspectivas.
- Se recomienda explorar frameworks de AutoML alternativos como H2O.ai AutoML y AutoGluon. Estas plataformas ofrecen enfoques diferentes para la optimización automática de modelos y podrían proporcionar insights complementarios. H2O.ai, en particular, destaca por su capacidad de manejar grandes volúmenes de datos y su enfoque en la interpretabilidad, mientras que AutoGluon sobresale en el apilamiento de modelos y la optimización de hiperparámetros.
- Se recomienda implementar un sistema de monitoreo de sesgos en los modelos utilizando herramientas como Aequitas o AI Fairness 360. Estas herramientas permitirían identificar y cuantificar sesgos potenciales en las predicciones del modelo, especialmente en relación con diferentes regiones geográficas o tipos de cultivo, asegurando que el modelo no favorezca inadvertidamente ciertas condiciones sobre otras.
- Se recomienda desarrollar un sistema de alertas tempranas basado en los valores SHAP y otras métricas de interpretabilidad. Este sistema podría identificar automáticamente cuando las predicciones del modelo se desvían significativamente de los patrones históricos o cuando ciertas variables comienzan a tener un impacto inusualmente alto en las predicciones, permitiendo una intervención proactiva.
- Se recomienda implementar técnicas de cuantificación de incertidumbre como Bootstrapping o Dropout Bayesiano. Esto proporcionaría intervalos de confianza para las predicciones del modelo, permitiendo una toma de decisiones más informada basada no solo en el valor predicho sino también en la incertidumbre asociada a cada predicción.

CAPÍTULO 11

Bibliografía

[1] Arias Segura, Joaquín, Adrián Rodríguez y Luiz Carlos Beduschi Filho: *Perspectivas de la agricultura y del desarrollo rural en las Américas Una mirada hacia América Latina y el Caribe*. repositorio.iica.int, Septiembre 2021. https://repositorio.iica.int/handle/11324/1850, visitado el 2024-05-17.

[2] Banguat: *Exportaciones (FOB) realizadas | Banco de Guatemala*, 2024. https://banguat.gob.gt/page/exportaciones-fob-realizadas-0, visitado el 2024-08-12.

[3] Bonner, Anne: *The Complete Beginner's Guide to Deep Learning: Artificial Neural Networks*, Junio 2019. https://towardsdatascience.com/simply-deep-learning-an-effortless-introduction-45591a1c4abb.

[4] Cavalcante, Rosana, Silva: *Artificial Intelligence in Agriculture: Benefits, Challenges, and Trends*. Applied sciences, 13:7405–7405, Junio 2023.

[5] Cavazza, Alberto, Francesca Dal Mas, Paola Paoloni y Martina Manzo: *Artificial intelligence and new business models in agriculture: a structured literature review and future research agenda*. British Food Journal, 125:436–461, Julio 2023.

[6] CENGICAÑA, Centro Guatemalteco de Investigación y Capacitación de la Caña de Azúcar : *El cultivo de la caña de azúcar en Guatemala*. Artemis Edinter, 2014. https://dialnet.unirioja.es/descarga/libro/572719.pdf.

[7] CENGICAÑA, Centro Guatemalteco de Investigación y Capacitación de la Caña de Azúcar : *Informe anual 2022-2023.*, 2024.

[8] Chen, Tianqi y Carlos Guestrin: *XGBoost: a Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, página 785–794, 2016.

[9] Guatemala, Banco de: *Producto Interno Bruto Trimestral*, Junio 2024. https://banguat.gob.gt/sites/default/files/banguat/cuentasnac/PIB2013/PDF_graficas_y_cuadros_estadisticos.pdf, visitado el 2024-08-12.

[10] Guatemala, Instituto Nacional de Estadistica de: *Principales resultados de la encuesta nacional de empleo e ingresos 2022*, Febrero 2023. https://www.ine.gob.gt/sistema/uploads/2023/03/23/202303232142069Odm3oxU9mTY58hkborwzylm7MJop05q.pdf.

[11] Gupta, Prashant: *Decision Trees in Machine Learning*, Mayo 2017. https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052.

[12] Mathenge, Mwehe, Ben G. J. S. Sonneveld y Jacqueline E. W. Broerse: *Application of GIS in Agriculture in Promoting Evidence-Informed Decision Making for Improving Agriculture Sustainability: A Systematic Review*. Sustainability, 14:9974, Agosto 2022.

[13] McCartney, S., A. Mehta, E. Podest y C. Hain: *ARSET - Satellite Remote Sensing for Agricultural Applications | NASA Applied Sciences*, Abril 2020. http://appliedsciences.nasa.gov/get-involved/training/english/arset-satellite-remote-sensing-agricultural-applications, visitado el 2024-08-12.

[14] McGonagle, John, José Alonso García y Saruque Mollick: *Feedforward Neural Networks | Brilliant Math Science Wiki*, 2019. https://brilliant.org/wiki/feedforward-neural-networks/.

[15] Molina, Elda y Ernesto Victorero: *La agricultura en países subdesarrollados. Particularidades de su financiamiento*, 2015. http://biblioteca.clacso.edu.ar/Cuba/ciei-uh/20150908010537/Financiamientoagricultura.pdf.

[16] Ragazou, Konstantina, Alexandros Garefalakis, Eleni Zafeiriou y Ioannis Passas: *Agriculture 5.0: A New Strategic Management Mode for a Cut Cost and an Energy Efficient Agriculture Sector*. Energies, 15:3113, Abril 2022.

CAPÍTULO 12

Anexo A: GridSearch de modelos de redes neuronales

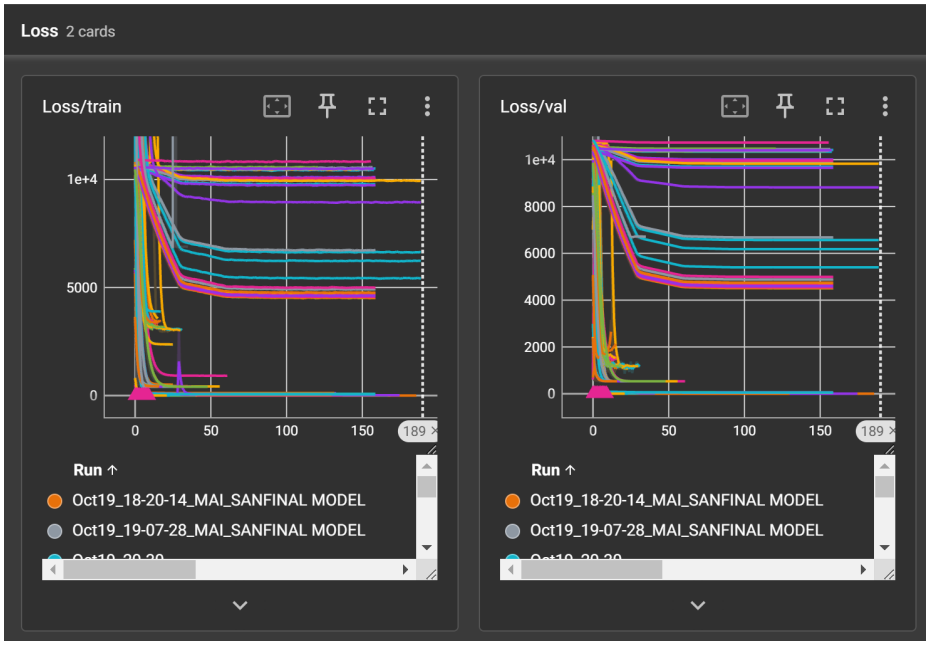


Figura 12.1: `Loss` de los modelos de redes neuronales

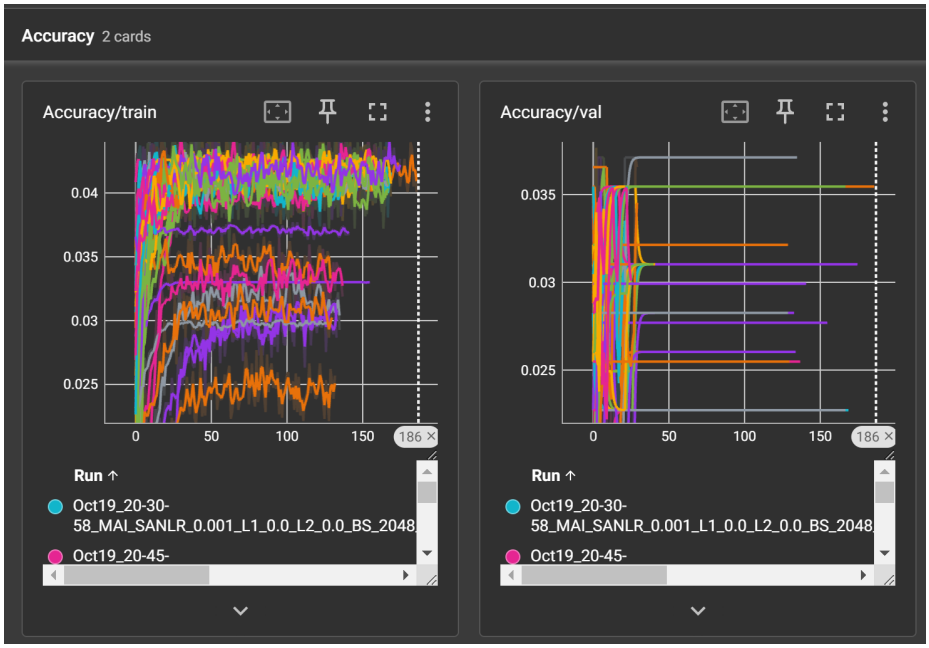


Figura 12.2: `Accuracy` del modelo de clasificación de redes neuronales

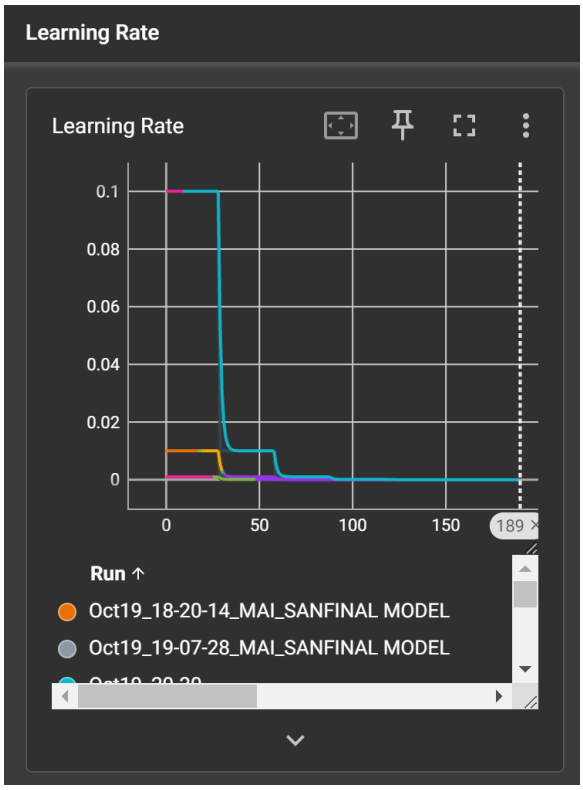


Figura 12.3: `LearninRate` de los modelos de redes neuronales

CAPÍTULO 13

Anexo B: Componentes de TruncatedSVD

Variable	**Componente 1**
área	0.999748
dirección viento_Q1_mean	0.000039
humedad relativa maxima_Q1_mean	0.000023
humedad relativa_Q2_mean	0.000020
humedad relativa_Q1_mean	0.000020

Tabla 13.1: Top 5 Variables para Componente 1

Variable	**Componente 2**
dirección viento_Q1_mean	0.169227
humedad relativa maxima_Q1_mean	0.090188
humedad relativa_Q2_mean	0.077030
humedad relativa_Q1_mean	0.076040
HUMEDAD_POND_Q1_mean	0.075092

Tabla 13.2: Top 5 Variables para Componente 2

Variable	**Componente 3**
dirección viento_Q1_mean	0.153416
humedad relativa minima_Q2_mean	0.066615
humedad relativa minima_Q1_mean	0.066485
humedad relativa_Q2_mean	0.060137
humedad relativa maxima_Q1_mean	0.057427

Tabla 13.3: Top 5 Variables para Componente 3

Variable	Componente 4
variedad	0.298652
dirección viento_Q1_std	0.094521
dirección viento_Q2_std	0.081240
dirección viento_6m_std	0.070855
dirección viento_Q1_mean	0.057378

Tabla 13.4: Top 5 Variables para Componente 4

Variable	Componente 5
dirección viento_Q1_std	0.155529
dirección viento_Q2_std	0.148681
variedad	0.145503
dirección viento_6m_std	0.132696
dirección viento_Q1_mean	0.106650

Tabla 13.5: Top 5 Variables para Componente 5

Variable	Componente 6
humedad relativa mínima_Q1_mean	0.084174
ANOMALÍAS_POND_Q1_std	0.081764
Humedad_POND_Q1_mean	0.080733
precipitación_Q1_mean	0.068424
humedad relativa mínima_Q2_mean	0.065242

Tabla 13.6: Top 5 Variables para Componente 6

Variable	Componente 7
ANOMALÍAS_POND_Q1_std	0.295095
humedad relativa mínima_Q1_mean	0.064460
humedad relativa mínima_Q2_mean	0.063179
precipitación_Q1_mean	0.045615
Humedad_POND_Q2_mean	0.041996

Tabla 13.7: Top 5 Variables para Componente 7

Variable	Componente 8
dirección viento_6m_std	0.155080
dirección viento_Q1_std	0.134858
estación	0.066540
cuadrante	0.059309
temperatura_Q1_mean	0.043775

Tabla 13.8: Top 5 Variables para Componente 8

Variable	Componente 9
velocidad viento maxima_Q1_mean	0.065167
humedad relativa minima_Q2_std	0.057437
humedad relativa minima_Q1_std	0.051181
velocidad viento maxima_Q1_std	0.050682
dirección viento_Q1_std	0.049281

Tabla 13.9: Top 5 Variables para Componente 9

Variable	Componente 10
cuadrante	0.114887
velocidad viento máxima_6m_std	0.095501
velocidad viento máxima_Q2_std	0.086461
velocidad viento máxima_Q1_std	0.063886
temperatura_Q1_mean	0.046626

Tabla 13.10: Top 5 Variables para Componente 10

Variable	Componente 11
estación	0.221132
velocidad viento maxima_Q1_mean	0.092602
dirección viento_Q1_std	0.083712
dirección viento_6m_std	0.065165
temperatura_Q1_mean	0.041232

Tabla 13.11: Top 5 Variables para Componente 11

Variable	Componente 12
humedad relativa mínima_Q1_mean	0.076054
humedad relativa mínima_Q2_mean	0.062583
dirección viento_Q2_std	0.059238
humedad relativa_Q2_mean	0.054870
velocidad viento máxima_6m_std	0.043845

Tabla 13.12: Top 5 Variables para Componente 12

Variable	Componente 13
precipitación_Q2_mean	0.108835
estación	0.088008
precipitación_Q1_mean	0.083695
Humedad_POND_Q2_mean	0.064695
humedad relativa minima_Q1_mean	0.059562

Tabla 13.13: Top 5 Variables para Componente 13

Variable	**Componente 14**
dirección viento_Q2_std	0.145398
velocidad viento maxima_Q1_mean	0.082044
dirección viento_6m_std	0.080794
dirección viento_Q1_std	0.071249
velocidad viento maxima_6m_std	0.065310

Tabla 13.14: Top 5 Variables para Componente 14

CAPÍTULO 14

Anexo C: Variables de SHAP

Este anexo presenta una descripción detallada de todas las variables utilizadas en el modelo predictivo de tonelaje de caña por hectárea (TCH). Las variables están organizadas por categorías y se incluyen detalles sobre su codificación y significado.

14.1. Índices de vegetación

14.1.1. NDVI (Índice de vegetación de diferencia normalizada)

El NDVI es un indicador de la salud y vigor de la vegetación. Se calcula a partir de las mediciones de la reflectancia de las bandas espectrales R (rojo) e IR (infrarrojo cercano).

- **NDVI_POND_Q2_mean**: Media del NDVI ponderado en el segundo trimestre.
- **NDVI_POND_Q4_std**: Desviación estándar del NDVI ponderado en el cuarto trimestre.
- **NDVI_POND_Q6_std**: Desviación estándar del NDVI ponderado en el sexto trimestre.
- **NDVI_POND_Q8_std**: Desviación estándar del NDVI ponderado en el octavo trimestre.

14.1.2. LAI (Índice de área foliar)

El LAI es una medida adimensional del área foliar por unidad de superficie de suelo.

- **LAI_POND_Q6_integral**: Integral del LAI ponderado en el sexto trimestre.
- **LAI_POND_Q6_mean**: Media del LAI ponderado en el sexto trimestre.
- **LAI_POND_Q6_std**: Desviación estándar del LAI ponderado en el sexto trimestre.

- **LAI_POND_Q8_mean**: Media del LAI ponderado en el octavo trimestre.
- **LAI_POND_Q8_std**: Desviación estándar del LAI ponderado en el octavo trimestre.

14.2. Variables meteorológicas

14.2.1. Temperatura y radiación

- **Amplitud_Térmica_Q2_std**: Desviación estándar de la amplitud térmica en el segundo trimestre.
- **Amplitud_Térmica_Q8_sum**: Suma de la amplitud térmica en el octavo trimestre.
- **Radiacion_Q2_std**: Desviación estándar de la radiación solar (MJ/m^2) en el segundo trimestre.
- **Radiacion_Q4_mean**: Media de la radiación solar en el cuarto trimestre.
- **Radiacion_Q6_mean**: Media de la radiación solar en el sexto trimestre.

14.2.2. Precipitación

- **precipitacion_Q2_integral**: Integral de la precipitación en el segundo trimestre.
- **precipitacion_Q2_mean**: Media de la precipitación en el segundo trimestre.
- **precipitacion_Q4_std**: Desviación estándar de la precipitación en el cuarto trimestre.
- **precipitacion_Q6_mean**: Media de la precipitación en el sexto trimestre.

14.3. Variables categóricas

14.3.1. Ubicación y zonificación

Tabla 14.1: Codificación de cuadrantes

Cuadrante	Código
CENTRO ALTO	0
CENTRO BAJO	1
CENTRO ESTE ALTO	2
CENTRO ESTE BAJO	3
CENTRO ESTE LITORAL	4
CENTRO LITORAL	5
CENTRO MEDIO	6
CENTRO OESTE BAJO	7
CENTRO OESTE LITORAL	8
CENTRO OESTE MEDIO	9
ESTE LITORAL	10
OESTE BAJO	11
OESTE LITORAL	12

Tabla 14.2: Codificación de estratos

Estrato	Código
ZONA ALTA	0
ZONA BAJA	1
ZONA LITORAL	2
ZONA MEDIA	3

14.3.2. Características del cultivo

Tabla 14.3: Codificación de sistema de riego

Sistema	Código
Aspersión	0
Goteo	1
Gravedad	2
Pivote	3
Sin riego	4

Tabla 14.4: Codificación de tipo de cosecha

Tipo	Código
MANUAL	0
MECANICO	1

14.4. Variables de producción

- **PRODUCCION_PON_Q2_std**: Desviación estándar de la producción ponderada en el segundo trimestre.
- **PRODUCCION_PON_Q4_mean**: Media de la producción ponderada en el cuarto trimestre.
- **PRODUCCION_PON_Q6_mean**: Media de la producción ponderada en el sexto trimestre.
- **PRODUCCION_PON_Q8_std**: Desviación estándar de la producción ponderada en el octavo trimestre.
- **area**: Área del cultivo en hectáreas.

14.5. Notas sobre la nomenclatura

- Los sufijos _Q2, _Q4, _Q6 y _Q8 indican el trimestre al que corresponden las mediciones.
- Los sufijos _mean, _std, _sum e _integral indican el tipo de agregación estadística:
  - _mean: Promedio del período.
  - _std: Desviación estándar.
  - _sum: Suma total.
  - _integral: Integral del período.