UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE ECONOMÍA Y EMPRESA
CARRERA DE NEGOCIOS INTERNACIONALES

TEMA:

Implementación de un algoritmo de clasificación para la asignación del sector destino en importaciones del mercado ecuatoriano.

AUTOR (ES):
Campoverde Cedeño, Snaycer Rodrigo
Miranda Yunda, Alisson Melina

Trabajo de titulación previo a la obtención del título de
LICENCIADOS EN NEGOCIOS INTERNACIONALES

TUTOR:
Mgs. Carrera Buri, Félix Miguel

Guayaquil, Ecuador
12 de febrero del 2026

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE ECONOMÍA Y EMPRESA

CARRERA DE NEGOCIOS INTERNACIONALES

CERTIFICACIÓN

Certificamos que el presente trabajo de titulación fue realizado en su totalidad por **Campoverde Cedeño, Snaycer Rodrigo y Miranda Yunda, Alisson Melina** como requerimiento para la obtención del título de **Licenciados en Negocios Internacionales.**

TUTOR

f._____

Mgs. Carrera Buri, Félix Miguel

DIRECTOR DE LA CARRERA

f._____

Mgs. Hurtado Cevallos, Gabriela Elizabeth

Guayaquil, a los 12 días del mes de febrero del año 2026

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE ECONOMÍA Y EMPRESA
CARRERA DE NEGOCIOS INTERNACIONALES

DECLARACIÓN DE RESPONSABILIDAD

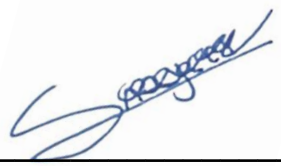
Yo, **Campoverde Cedeño, Snaycer Rodrigo y**
Miranda Yunda, Alisson Melina

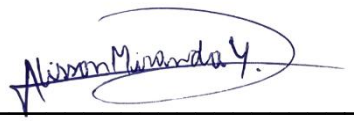
DECLARAMOS QUE:

El Trabajo de Titulación, **Implementación de un algoritmo de clasificación para la asignación del sector destino en importaciones del mercado ecuatoriano**, previo a la obtención del título de **Licenciados en Negocios Internacionales**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría. En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 12 días del mes de febrero del año 2026

AUTORES:





Campoverde Cedeño, Snaycer Rodrigo **Miranda Yunda, Alisson Melina**

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE ECONOMÍA Y EMPRESA
CARRERA DE NEGOCIOS INTERNACIONALES

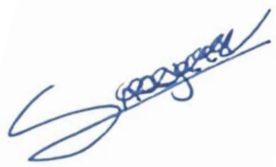
AUTORIZACIÓN

Yo, **Campoverde Cedeño, Snaycer Rodrigo y**
Miranda Yunda, Alisson Melina

Autorizo a la Universidad Católica de Santiago de Guayaquil a la publicación en la biblioteca de la institución del Trabajo de Titulación, **Implementación de un algoritmo de clasificación para la asignación del sector destino en importaciones del mercado ecuatoriano.**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 12 días del mes de febrero del año 2026

AUTORES:



Campoverde Cedeño, Snaycer Rodrigo **Miranda Yunda, Alisson Melina**

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE ECONOMÍA Y EMPRESA

CARRERA DE NEGOCIOS INTERNACIONALES

REPORTE DE COMPILATIO

CERTIFICADO DE ANÁLISIS
magister

Tesis Campoverde Snaycer - Miranda Alisson - Implementacion de un algortimo de clasificacion para la asignacion del sector destino en importaciones del ecuador

2% Textos sospechosos

2% Similitudes
0 % similitudes entre comillas
< 1 % entre las fuentes mencionadas
7% Idiomas no reconocidos (ignorado)
15% Textos potencialmente generados por la IA (ignorado)

Nombre del documento: Tesis Campoverde Snaycer - Miranda Alisson - Implementacion de un algortimo de clasificacion para la asignacion del sector destino en importaciones del ecuador.pdf
ID del documento: f51e27e8edfa12c328195a4069d8dcfc79953e75
Tamaño del documento original: 4,02 MB

Depositante: Felix Miguel Carrera Buri
Fecha de depósito: 15/2/2026
Tipo de carga: interface
fecha de fin de análisis: 15/2/2026

Número de palabras: 27.253
Número de caracteres: 201.896

Ubicación de las similitudes en el documento:

Fuentes principales detectadas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	revistasbolivianas.umsa.bo | Revista Varianza - Árbol de Decisión en Aprendizaj... http://revistasbolivianas.umsa.bo/scielo.php?script=sci_arttext&pid=S9876-67892022000100... 3 fuentes similares	< 1%		Palabras idénticas: < 1% (273 palabras)
2	localhost | Evaluación a 5 años, de los efectos terapéuticos del metrotexato pare... http://localhost:8080/xmlui/bitstream/3317/9365/3/T-UCSG-PRE-MED-634.pdf.txt 14 fuentes similares	< 1%		Palabras idénticas: < 1% (115 palabras)
3	repositorio.ucsg.edu.ec | Actividades básicas de la vida diaria de los adultos may... http://repositorio.ucsg.edu.ec/bitstream/3317/2211S/1/T-UCSG-PRE-MED-ENF-895.pdf 11 fuentes similares	< 1%		Palabras idénticas: < 1% (102 palabras)
4	Eduardo Alvarado, Grace Moreira 50 % avance barberan.docx | Eduar... #65757d Viene de de mi grupo 4 fuentes similares	< 1%		Palabras idénticas: < 1% (62 palabras)
5	repositorio.ucsg.edu.ec | Estudio del impacto que genera el marketing emocion... http://repositorio.ucsg.edu.ec/bitstream/3317/18315/1/T-UCSG-PRE-CEAE-CNI-8.pdf 2 fuentes similares	< 1%		Palabras idénticas: < 1% (51 palabras)

f._____

Mgs. Carrera Buri, Félix Miguel
TUTOR

AGRADECIMIENTO – ALISSON MELINA MIRANDA YUNDA

Quiero comenzar agradeciendo a Dios por darme la fortaleza, sabiduría y perseverancia para culminar esta etapa universitaria llena de desafíos y aprendizaje. Por ser parte fundamental en mi vida, guiándome en cada paso de mi formación académica y desarrollo personal.

Agradecer a mis padres quienes me han dado su amor y apoyo incondicional a lo largo de mi vida, especialmente por su sacrificio para que me pueda formar como una profesional y darme la mejor herencia que es la educación. Gracias a ellos hoy sigo cumpliendo mis metas y espero se sientan orgullosos de mi como yo lo estoy de ser su hija. A mi hermano por motivarme a seguir adelante, aunque el camino este lleno de obstáculos y acompañarme sacándome una sonrisa cada vez que necesitaba.

Asimismo, agradezco a mi compañero de tesis con el que hemos trabajado de manera conjunta y comprometida en la elaboración de este proyecto, por estar dispuesto a resolver cualquier inquietud, por escuchar cada opinión con respeto y por su constancia en las largas noches de trabajo.

Finalmente, a mi tutor Félix, el cual siempre nos ha brindado su ayuda para resolver cada dificultad presentada y a todos los profesores que compartieron su conocimiento a lo largo de la carrera universitaria.

DEDICATORIA – ALISSON MELINA MIRANDA YUNDA

Me gustaría dedicarles este logro a quienes más amo y son pilares fundamentales en mi vida, a Dios, a mis padres, a mi hermano, a mis abuelitos que me ven y cuidan desde el cielo, y a mi abuelita que ha sido mi fiel compañera.

AGRADECIMIENTO – SNAYCER RODRIGO CAMPOVERDE CEDEÑO

Quiero comenzar agradeciendo a toda mi familia, a mi mamá Maira y mi papá Ivan, ya que sin ellos yo no estaría aquí, gracias siempre por su apoyo por siempre creer en mí, así como en mis hermanos, gracias por su apoyo a lo largo de toda mi vida estudiantil, así como en las cosas pequeñas que me han ayudado a convertirme en la persona que soy, para que yo pueda cumplir mis metas.

También, mencionar a mis hermanos Ivan y Scarleth que siempre han sido un apoyo emocional para mí. Gracias Ivan por ser mi hermano mayor y servir como un faro de luz para mí y Scarleth, tus hermanos menores, para demostrarnos el camino que debemos seguir y ser nuestra guía en la vida. Gracias Scarleth por ser mi hermana menor, la menor de la casa, la princesa de la familia, por siempre alegrarnos a mí y tu familia.

Agradecerle a mi compañera de tesis, Alisson, por todas las largas horas que invertimos para poder realizar esta tesis y así cumplir nuestra meta de graduarnos.

Agradecer a mis amigos por estar ahí siempre para divertirnos cuando se necesitaba. A Geanella la cual ha sido una gran amiga a lo largo de toda mi vida, y así espero que sigamos teniendo muchos años más de amistad.

Agradecerles a mis mascotas Catalina, Cataleya (+), Sasha, Noa y Oliver a quienes quiero mucho y siempre espero que estén conmigo.

Agradecerles a mis abuelos de parte de mamá Francisco y Blanca por quererme mucho y por ser unos grandes familiares para mí. A mis abuelos de parte de papá Segundo (+) y Mercedes (+) les mando un agradecimiento infinito hasta el cielo, por cuidarme desde arriba y espero siempre recuerden que los quiero mucho.

DEDICATORIA – SNAYCER RODRIGO CAMPOVERDE CEDEÑO

Me gustaría dedicarles este logro a todas las personas que siempre estuvieron ahí para mí, que siempre me motivaron para seguir adelante y poder triunfar como persona. A mi familia que siempre estuvieron, están, y estarán presentes en cada momento importante de mi vida. Esto es para ustedes, los amo mucho.

INDICE

Índice de Ilustraciones

Resumen

La presente investigación tuvo como objetivo desarrollar y evaluar un modelo de clasificación basado en técnicas de aprendizaje automático para la asignación del sector destino de las importaciones del mercado ecuatoriano. El estudio se enmarca dentro de un enfoque cuantitativo de tipo categórico, al centrarse en la clasificación de registros en categorías específicas (Acuícola, Pecuario, Químico y Veterinario) a partir del análisis de datos históricos provenientes de la plataforma COBUS y de información interna empresarial.

Metodológicamente, se realizó una revisión de literatura sobre los principales enfoques de machine learning aplicables a problemas de clasificación, destacando los métodos basados en árboles. Posteriormente, se llevó a cabo la preparación y depuración de la base de datos, garantizando su calidad, coherencia y consistencia mediante procesos de limpieza, eliminación de variables irrelevantes y estandarización de formatos.

El modelo principal implementado fue Random Forest, complementado con un modelo de Árbol de Decisión para fines comparativos. Se dividió el conjunto de datos en 75 % para entrenamiento y 25 % para prueba, utilizando estratificación para mantener la proporción de clases. La evaluación del desempeño se realizó mediante métricas de exactitud, reporte de clasificación, matriz de confusión y curva ROC multiclase, lo que permitió medir la capacidad discriminativa del modelo.

Los resultados evidenciaron que el modelo Random Forest presentó mayor estabilidad y precisión en la clasificación del sector destino en comparación con el Árbol de Decisión individual. Asimismo, el análisis de importancia de variables permitió identificar los factores más influyentes en el proceso de clasificación, mientras que la detección de registros con baja probabilidad aportó información relevante para el control de calidad de los datos.

Palabras Clave: Aprendizaje automático, Clasificación, Random Forest, Importaciones, Sector destino, Comercio exterior, Datos históricos, Árbol de Decisión, Evaluación, Predicción.

Abstract

This research aimed to develop and evaluate a classification model based on machine learning techniques for assigning the destination sector of imports in the Ecuadorian market. The study follows a quantitative categorical approach, focusing on classifying records into specific categories (Aquaculture, Livestock, Chemical, and Veterinary) through the analysis of historical data obtained from the COBUS platform and internal company records.

Methodologically, a literature review was conducted on the main machine learning approaches applicable to classification problems, emphasizing supervised models and, particularly, tree-based methods. Subsequently, data preparation and cleaning processes were carried out to ensure quality, coherence, and consistency through data filtering, removal of irrelevant variables, and standardization procedures.

The main implemented model was Random Forest, complemented by a Decision Tree model for comparative purposes. The dataset was divided into 75% for training and 25% for testing, using stratification to maintain class proportions. Model performance was evaluated through accuracy metrics, classification reports, confusion matrices, and multiclass ROC curves to measure discriminative capacity.

The results demonstrated that the Random Forest model achieved greater stability and predictive accuracy compared to the individual Decision Tree model. Additionally, feature importance analysis identified the most influential variables in the classification process, while the detection of low-probability records contributed to data quality assessment.

Keywords: Machine Learning, Classification, Random Forest, Imports, Destination Sector, Foreign Trade, Historical Data, Decision Tree, Evaluation, Prediction.

Résumé

La présente recherche avait pour objectif de développer et d'évaluer un modèle de classification basé sur des techniques d'apprentissage automatique pour l'attribution du secteur de destination des importations sur le marché équatorien. L'étude s'inscrit dans une approche quantitative de type catégoriel, visant à classer les enregistrements en catégories spécifiques (Aquacole, Élevage, Chimique et Vétérinaire) à partir de l'analyse de données historiques issues de la plateforme COBUS et de données internes de l'entreprise.

Sur le plan méthodologique, une revue de la littérature a été réalisée concernant les principales approches de l'apprentissage automatique applicables aux problèmes de classification, en mettant l'accent sur les modèles supervisés et, en particulier, les méthodes basées sur les arbres de décision. Par la suite, des processus de préparation et de nettoyage des données ont été effectués afin de garantir leur qualité, leur cohérence et leur fiabilité, notamment par la suppression des variables non pertinentes et la standardisation des formats.

Le modèle principal mis en œuvre a été le Random Forest, complété par un modèle d'Arbre de Décision à des fins comparatives. L'ensemble de données a été divisé en 75 % pour l'entraînement et 25 % pour le test, en utilisant une stratification afin de maintenir la proportion des classes. L'évaluation des performances a été réalisée au moyen de mesures d'exactitude, de rapports de classification, de matrices de confusion et de courbes ROC multiclasses.

Les résultats ont montré que le modèle Random Forest présente une plus grande stabilité et une meilleure précision prédictive que le modèle d'Arbre de Décision individuel. De plus, l'analyse de l'importance des variables a permis d'identifier les facteurs les plus influents dans le processus de classification, tandis que la détection d'enregistrements à faible probabilité a contribué au contrôle de la qualité des données.

Mots-clés: Apprentissage automatique, Classification, Forêt aléatoire, Importations, Secteur de destination, Commerce extérieur, Données historiques, Arbre de décision, Évaluation, Prédiction.

1 Introducción

A lo largo del tiempo, el comercio exterior ha sido descrito como uno de los mecanismos más potentes para expandir la frontera productiva de los países. A través de las importaciones, las economías acceden a bienes de capital, insumos intermedios, componentes tecnológicos y conocimientos incorporados que muchas veces no están disponibles de manera local o no lo están con la calidad, el costo o la escala requeridos para el nicho de mercado deseado.

Por su parte, las exportaciones permiten capturar economías de escala, diversificar riesgos de demanda y convertir ventajas relativas en ingresos y empleo. Por lo que, en conjunto estas dos corrientes del comercio han comenzado a impulsar la productividad sistémica, integrando cadenas globales de valor y condicionan la competitividad de las empresas.

Debido a esto, en la última década, el comercio internacional ha comenzado a desempeñar uno de los papeles más importantes dentro del país, ya que es fundamental en el desarrollo económico del Ecuador, siendo las importaciones un componente clave para el abastecimiento del mercado interno, la diversificación productiva y la competitividad empresarial, en la medida en que una parte importante de las compras externas corresponde a materias primas, insumos y bienes de capital, los cuales fortalecen la cadena de producción interna (Ministerio de Producción, Comercio Exterior, Inversiones y Pesca, 2023, p. 59). Y a través de las importaciones, el país accede a bienes de capital, materias primas, insumos intermedios y productos de consumo que permiten sostener y dinamizar diversas actividades económicas.

Más allá de las cifras, el comercio exterior es un conjunto complejo de procesos, donde pueden llegar a intervenir diferentes agentes de este sector tales como aduanas, aforadores, agentes aduaneros, agentes regulatorios, autoridades sanitarias, navieras, bancos, aseguradoras, múltiples compradores y vendedores, donde todos ellos deben de coordinar toda la información

posible y en algunos casos en tiempos de respuestas aún más cortos que antes debido a la continua expansión del comercio.

Por lo que, la logística se convierte en uno de los puntos más importante dentro de esta área, ya que, al gestionar toda la documentación autorizada con eficiencia tales como Bill of loading, facturas comerciales, cartas portes, seguros, certificado de origen, certificado de libre venta, permisos sanitarios, etc. Se puede lograr llegar a reducir costos dentro de cada una de las operaciones, lo que generaría una mayor rentabilidad dentro de cada una de las mismas, así como también reduciría la incertidumbre logística de las mismas y los ciclos de abastecimiento.

Asimismo, el comercio no es el único campo que ha crecido, la digitalización en Ecuador se ha acelerado y eso ya está cambiando la forma en que se compra, se vende y se gestiona la logística. Por ejemplo, a inicios de 2024 había 15,29 millones de usuarios de Internet en el país (83,6% de la población), lo que muestra una base digital cada vez más amplia para transacciones y trámites en línea. Además, Ecuador registró 97 suscripciones móviles por cada 100 habitantes (2022), un indicador de alta masificación del acceso móvil (clave porque gran parte del comercio digital se mueve desde el celular).

Debido a esto, la modernización de todos los procesos aduaneros y los sistemas de gestión de riesgos ya no son solo lujos si no necesidades. Siendo estas condiciones operativas para que el comercio funcione con previsibilidad y para que las empresas, en especial las pequeñas y medianas, participen en mercados más exigentes

En este contexto, el machine learning (ML) se ha consolidado como uno de los habilitadores tecnológico clave para lograr reducir costos de transacción, mejorar la trazabilidad, optimizar la logística y fortalecer la integridad de los procesos aduaneros y financieros que se asocian al comercio internacional.

Los choques de los últimos años tales como congestión portuaria, disrupciones en transporte, cambios de demanda y oferta, evidenciaron que la supervivencia del comercio no depende sólo de una infraestructura física, sino también de la capacidad de leer datos en tiempo real y tomar decisiones basadas en evidencia. Esto implica el navegar en procesos guiados por reglas generales y revisiones masivas, apoyados en información histórica y señales operativas.

Al trabajar con criterios de riesgo y posibles escenarios que pueden suceder, las cargas legítimas fluyen de forma más rápida y los controles están donde dan más valor. Al mismo tiempo, el sector privado mejora la planificación de las compras, del inventario y del financiamiento porque puede anticipar los cuellos de botella y los plazos mucho más fáciles.

La adopción del machine learning en el ámbito empresarial se ha convertido en un nuevo punto de partida para mejorar pronósticos y priorizar riesgos en las cadenas de suministro, ya que la analítica basada en IA puede reducir errores de forecast entre 20% y 50% y traducirse en mejoras operativas relevantes (McKinsey & Company, 2022. Cuando las diferentes organizaciones combinan datos operativos de calidad con capacidades analíticas y rediseño de procesos, los modelos de machine learning ayudan a elevar la calidad del servicio y ganar trazabilidad en decisiones clave, pero ese valor solo se captura plenamente si se acompaña con cambios organizacionales y actualizaciones de procesos (Alicke et al., 2021).

Estas mejoras no dependen de un único algoritmo aplicado, sino de una arquitectura de datos, que sean confiables y de donde se puedan aprender patrones que permitan aprender y escalar con evidencia.

Y dentro de todos estos usos posibles del machine learning, nos encontramos con ese campo en el que se basa nuestro proyecto de investigación, el sector destino de las importaciones.

En este campo, la correcta asignación del sector destino de los bienes importados, la cual podemos entender como la identificación del ámbito económico en el que serán finalmente utilizados, es importante para generar estadísticas que tengan sentido, orientar políticas públicas, fortalecer el control sanitario y facilitar decisiones empresariales acerca de abastecimiento, inventarios y cumplimiento regulatorio.

No obstante, la asignación del sector destino continúa, en gran parte, siendo un proceso manual y muy lineal. En la práctica, estas dependen de interpretaciones muy específicas de códigos arancelarios, de la lectura de descripciones de despacho no estandarizadas y del conocimiento adquirido de analistas vinculados a instituciones y empresas.

Esta realidad produce dos posibles escenarios que se pueden presentar: (i) inconsistencia e incongruencia de criterios entre periodos y entre organizaciones, y (ii) una limitada trazabilidad sobre las razones que explican la clasificación final.

Los resultados pueden llegar a ser muy perjudiciales: se debilita la comparabilidad temporal de los indicadores del sector; la importancia de las medidas de fomento y control se vuelve complicada; y las empresas adquieren un factor de duda en lo que respecta a la asignación de sus artículos, con costos de reembolso y riesgo de las consecuencias de los incumplimientos no intencionados.

En el contexto ecuatoriano es posible contar con una opción para construir un sistema con el enfoque de inteligencia del negocio que permita automatizar, documentar y auditar la asignación del sector destino que eleve la calidad de la información con métricas verificables y procedimientos reproducibles.

Dicho esto, la asignación correcta del sector destino de estos productos es un reto constante de las instituciones públicas, de las empresas privadas y de los analistas del comercio exterior, dada la complejidad y la variedad de las transacciones comerciales.

2 Problemática

En la actualidad, el contexto empresarial presenta alta volatilidad y ciclos competitivos cada vez más cortos, los distintos individuos y organizaciones serán condicionadas por la innovación tecnológica y la gestión inteligente de la información, sin duda, factores cruciales para mantener el desempeño organizacional. En ese contexto, las organizaciones dependen de su capacidad para localizar la forma en que se pueden detectar las tendencias en las preferencias de los clientes, dimensionar cuales son los públicos adecuados a los que dirigimos la oferta que realizamos y comprender cual es la apertura para introducir mejoras o desarrollar nuevos usos para las ofertas de productos o servicios que realizamos.

Esta orientación hacia el análisis de datos, como herramienta de respaldo y ayuda para poder identificar tendencias, segmentar mercados y anticipar cambios, se ha consolidado como una práctica extendida en la gestión moderna (Díaz et al., 2021).

La naturaleza de esta realidad se concentra en cadenas logísticas y cadenas regulatorias con diferentes actores dentro del mercado ecuatoriano (aduanas, ventanillas únicas, agencias sanitarias, compañías navieras, agentes de carga, bancos y empresas importadoras; agentes que deben conectar información en poco tiempo con compromisos cortos y exigentes. Dentro de este campo en continuo movimiento, la asignación de un "sector final" implica una gota de conflicto. No se trata de un etiquetado de menor importancia, ya que esta variable condiciona muchos puntos como la lectura del resto de la información del sector; el riesgo; la coordinación entre las organizaciones que intervienen y la posibilidad de obtener permisos y certificaciones; y también la planificación empresarial de compras e inventarios por especie o línea productiva.

Una de las primeras fuentes de la problemática es la heterogeneidad y ambigüedad textual de las descripciones de despacho. En la práctica, los ítems se describen con redacciones libres que combinan nombres comerciales, principios activos, concentraciones, formatos de

presentación, acrónimos (feed grade, premix), anglicismos o traducciones parciales. Esta diversidad lingüística dificulta la interpretación correcta de los productos, sobre todo cuando la partida arancelaria no discrimina con suficiente confianza el destino final del producto. Algunas pequeñas variaciones en el texto, como por ejemplo, incluir o no la mención a una especie o proceso productivo, pueden inclinar la balanza hacia sectores distintos, en especial en mercancías multiuso.

En segundo lugar, el carácter de múltiples finalidades de muchos insumos agropecuarios que son importados al Ecuador que aumentan las zonas grises en donde pequeñas diferencias en el texto (por ejemplo, colocar o no la especie o el proceso de producción) hacen que la asignación se dirija a un sector diferente.

Tercero, la dependencia de las múltiples opiniones que suceden bajo la presión operativa en picos altos de demanda, donde por lo general, la última palabra se apoya en reglas generales que varían entre diferentes periodos de tiempo, generando inestabilidad.

Según el Banco Mundial, "incorporar nuevos indicadores claves de rendimiento nos ayuda a medir la velocidad del comercio internacional" (World Bank, 2023, p.16), de modo que si la variable de salida, en este caso el sector destino de un producto importado, toda la logística pierde precisión, llegando a ocasionar posibles riesgos al no tener la capacidad de prevenirlos. Asimismo, la OMA (Organización Mundial de Aduanas) resalta mucho la importancia de tener un control total de los datos de logística, para así poder concentrar toda la mano de obra en cargas de alto riesgo y acceder a una mayor facilidad en la liberación de las cargas que tienen bajo riesgo: (WTO, 2021, p.69).

Las consecuencias que se derivan de esto cumplen su significado en diferentes perspectivas. En la estadística, si existe un choque de criterios disminuye la calidad de las series y se vuelve más complicado el estudiar y anticipar las tendencias sobre las demandas actuales

y futuras de insumos. Además, impide el cálculo de factores a tomar en cuenta dentro de las cadenas más vulnerables como lo son las cadenas avícola y porcina. En la operativa genera reprocesos, en la regulación debilita la selectividad y la coordinación, y así en el resto de los campos que se encuentran estrechamente relacionados.

En resumen, la asignación del sector destino se torna estructural dentro de este ámbito como consecuencia de la repetición y vagancia continua textual de los productos, siendo alguno de sus efectos la pérdida de calidad estadística, reprocesos, tiempos adicionales y problemas en registros de los productos. Por esto, cada vez se exige una mayor medición de rendimiento mediante la trata de datos recolectados por las empresas.

3 Justificación

La presente investigación está enfocada en el desarrollo de un modelo de clasificación de la variable llamada sector destino a través del machine learning, donde la asignación del sector destino permitirá una mejor toma de decisiones dentro del campo de las importaciones, así como también una reducción de reprocesos innecesarios que generan costos de más y un fortalecimiento de la trazabilidad operativa pero que no reemplace el juicio experto.

Hay que destacar también la relevancia de la misma debido a la calidad de la información con la que operan instituciones públicas y empresas privadas en el comercio exterior ecuatoriano.

Hoy en día, el asignar un sector destino al importar suele hacerse a mano. Eso genera diferencias entre quienes deciden, ya que deja un rastro no muy claro y que tarda bastante. Por lo que, si se utiliza un enfoque que se base en el análisis de datos junto con modelos que aprenden por si solos de los datos proporcionados, se puede cambiar toda esa operación. Esto

mismo se puede cuantificar, repetir, y revisar, permitiendo identificar factores que tienen más peso sobre otros.

Según Bright et. Al (2025), los funcionarios públicos están asumiendo cada vez más la responsabilidad de experimentar con la IA en sus funciones laborales. Llegando a un punto en el que el 67% de países a nivel mundial se encuentran utilizando esta misma dentro del sector público para poder mejorar su eficiencia.

4 Alcance

El presente trabajo está dirigido a aquellos interesados en comprender y aplicar nuevas técnicas de inteligencia artificial y *machine learning* dentro del ámbito del comercio exterior ecuatoriano, específicamente en el proceso de clasificación del sector destino de las importaciones. Además, está orientado a quienes forman parte del sector empresarial y gubernamental, tales como economistas, analistas de comercio exterior, profesionales aduaneros, administradores, ingenieros en sistemas y estudiantes que se estén formando en las áreas de Economía, Comercio Internacional e Ingeniería Empresarial.

Dirigida a profesionales y técnicos del ámbito público hasta el privado, dicha investigación se convierte en un lugar para conocer en qué importan modelos de clasificación automatizados , cómo aplicarlos y cuando aplicar estas clasificaciones automatizadas para mejorar la gestión de la información y el análisis de información relacionada con las importaciones del país, y de esta manera, conseguir resultados más precisos, consistentes y basados en datos reales que les permiten un mejor proceso de toma de decisiones estratégicos.

A los analistas de comercio exterior, este trabajo les permitirá comprender cómo los algoritmos de *machine learning*, como *Random Forest*, pueden identificar patrones ocultos en

grandes volúmenes de información y mejorar la precisión en la asignación del sector destino de productos importados.

A los gestores empresariales y gerentes de logística, les servirá para evaluar el comportamiento de las importaciones por sectores económicos, facilitando la planificación de compras, la gestión de inventarios y la identificación de oportunidades de negocio.

A las entidades públicas, como el Servicio Nacional de Aduana del Ecuador (SENAE) o el Banco Central del Ecuador (BCE), esta investigación les permitirá modernizar sus procesos de clasificación y análisis estadístico, reduciendo el margen de error y optimizando los tiempos de procesamiento de datos.

Asimismo, a los investigadores y académicos, el estudio les ofrecerá un ejemplo práctico de cómo la ciencia de datos puede integrarse con la economía y el comercio internacional, aportando metodologías innovadoras que fortalezcan la toma de decisiones basadas en evidencia.

En general, este trabajo busca fomentar el uso de herramientas de inteligencia artificial en la gestión económica y comercial del Ecuador, promoviendo un enfoque tecnológico, eficiente y analítico en el tratamiento de grandes volúmenes de información provenientes del comercio exterior.

5 OBJETIVOS

5.1 Objetivo General

- Implementar un algoritmo de clasificación que permita automatizar y optimizar la asignación del sector destino en las importaciones del mercado ecuatoriano, mejorando la precisión y eficiencia del análisis de datos comerciales.

5.2 Objetivos Específicos

- Analizar los principales conceptos y enfoques de aprendizaje automático aplicables a la clasificación mediante una revisión de la literatura.
- Explicar la metodología del Random Forest basado en machine learning para la clasificación.
- Evaluar el modelo de clasificación, describiendo sus resultados mediante la discusión de hallazgos publicados en estudios paralelos.

6 Marco Teórico

6.1 Inteligencia artificial

La Inteligencia Artificial (IA) es un campo de la informática el cual tiene como objetivo el desarrollo de sistemas que sean capaces de la ejecución de tareas que requieren del esfuerzo propio e inteligencia, como aprender, razonar o tomar decisiones. Dentro de la IA, el aprendizaje automático (Machine Learning) se enfoca en algoritmos los cuales tiene como objetivo el reconocer y aprender patrones a partir de datos, con el fin de realizar tareas, en este caso de clasificación, reconocimiento o toma de decisiones sin estar programados de forma explícita para cada caso.

6.2 Aprendizaje Supervisado

Es un tipo de machine learning, donde cada ejemplo incluye una entrada x_i y una etiqueta y_i. El algoritmo aprende una función que mapea entradas a etiquetas, utilizando esas "respuestas correctas".

$$D = \{(x_i, y_i)\}_{i=1}^{n}$$

- D = es el dataset (conjunto de datos)
- x_i = es el vector de variables de entrada del ejemplo i
- y_i = es la etiqueta o clase verdadera del ejemplo i
- N = número total de observaciones

6.3 Variable objetivo

La variable objetivo (o label) es la clase que queremos clasificar. En el modelo usado en esta tesis, esta variable corresponde a una categórica (sector destino).

$$Y \in \{1, \cdots, K\}$$

En esta expresión matemática, se da a entender que la variable objetivo Y pertenece a cualquiera de las K clases.

$$Y \in \{0,1\}(caso\ binario)$$

En esta expresión matemática, se da a entender que la variable objetivo Y pertenece a una de las dos clases, planteando un resultado dicotómico de salida.

6.4 Variable predictora

Las variables predictoras son las características o vectores de entrada usados para predecir Y.

$$X = (X_1, X_2, \cdots, X_p)$$

$$x_i = (x_{i1}, \cdots, x_{ip})^T \in \mathbb{R}^p$$

Esa expresión significa que el espacio de entrada $\mathcal{X}$ (todas las observaciones posibles) es un subconjunto de $\mathbb{R}^p$. Es decir, es un vector con p valores, uno por cada variable.

6.5 Conjunto de entrenamiento

Es el subconjunto de datos utilizado para ajustar el modelo y que este "entrene", es decir que aprende de los patrones de los mismos

$$\mathcal{D}_{\text{train}} = (x_i, y_i) : i \in I_{\text{train}}$$

- I_{train}: observaciones usadas para entrenar el modelo.
- $I_{\text{train}} \cap I_{\text{test}} =$ ninguna observación está en entrenamiento y test a la vez.

- $I_{\text{train}} \cup I_{\text{test}} = \{1, \dots, n\}$ significa que si se unen ambos conjuntos se tiene el dataset completo.

6.6 Conjunto de prueba

Es el subconjunto de datos el cual se usa únicamente para evaluar desempeño (no se entrena con él) y estimar la capacidad de generalización.

$$\mathcal{D}_{\text{test}} = \{(x_i, y_i) : i \in I_{\text{test}}\}$$

- I_{test}: observaciones usadas para probar el modelo.
- $I_{\text{train}} \cap I_{\text{test}} =$ ninguna observación esta en entrenamiento y test a la vez.
- $I_{\text{train}} \cup I_{\text{test}} = \{1, \dots, n\}$ (significa que si se unen ambos conjuntos se tiene el dataset completo.

6.7 Algoritmos de clasificación

Bishop (2006) plantea la clasificación como la construcción de modelos que, a partir de ejemplos etiquetados, estiman la probabilidad de pertenencia de cada observación a las distintas clases y permiten decidir la clase más probable. En resumen, a partir de un conjunto de ejemplos etiquetados $(\mathbf{x}_i, y_i)$, construye un modelo f capaz de asignar a cada nueva observación $\mathbf{x}$ una clase y perteneciente a un conjunto de categorías.

Se puede expresar como:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i)$$

- x_i: las variables predictoras de la observación i
- y_i: la verdadera clase de esa observación
- n: número total de observaciones de entrenamiento.
- $\mathcal{F}$: es la familia de clasificadores entre los cuales se va a buscar el modelo a usar.
- f: un posible clasificador que se encuentra dentro de esa familia $\mathcal{F}$.
- $L(f(x_i), y_i)$: es la función de pérdida, la cual mide cuánto se equivoca el modelo en esa observación.
- $\frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i)$: promedio de la pérdida en todos los datos de entrenamiento.
- $\arg\min_{f\in\mathcal{F}}$: se refiere a que toma el f que minimiza la pérdida dentro de la familia $\mathcal{F}$.

6.8 Árbol de decisión

De acuerdo con Utgoff (1998), un árbol de decisión es una representación gráfica que organiza, en forma de árbol, el procedimiento para clasificar o evaluar un ítem a partir de una secuencia de pruebas sobre sus atributos.

Figura 1
Ejemplo de Árbol de Decisión

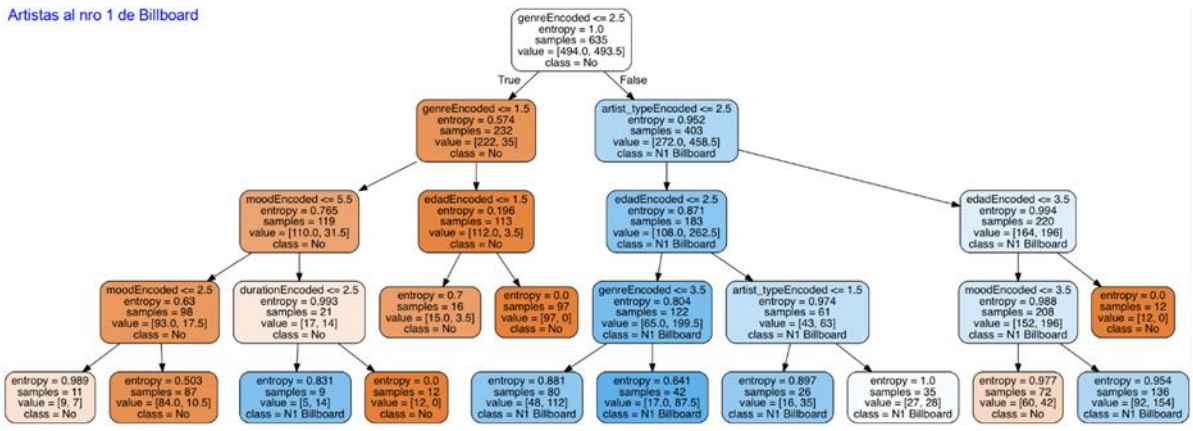


Ilustración 1 Diagrama que representa un árbol de decisión para clasificar datos de la revista Billboard*. Adaptado de* Árbol de decisión Billboard, (2018).

6.9 Funcionamiento

Un árbol de decisión define una partición del espacio de entrada $\mathcal{X}$ en M regiones disjuntas:

$$\mathcal{X} = \bigcup_{m=1}^{M} R_m \, , R_m \cap R_{m'} = \emptyset \text{ si } m \neq m'$$

donde:

- $\mathcal{X}$: espacio de todas las combinaciones posibles de características.
- R_m: región m-ésima del espacio (hoja del árbol).
- M: número total de hojas o nodos.
- $R_m \cap R_{m'} = \emptyset$ si $m \neq m'$: significa que ninguna observación pertenece a dos hojas a la vez, es decir, cada x cae en una sola hoja.

El árbol asigna una clase a cada región R_m. El modelo se puede escribir como:

$$f(x) = \sum_{m=1}^{M} c_m \ \mathbf{1}\{x \in R_m\}$$

donde:

- $f(x)$: clase asignada por el árbol para una observación con características x.
- c_m: clase asignada a la región R_m
- $\mathbf{1}\{x \in R_m\}$: vale 1 si x cae en la región R_m, 0 si es lo contrario.

6.10 Impureza

Se supone que un nodo t del árbol, contiene N_t observaciones. Si existen K clases posibles, se define:

$$p_{k|t} = \frac{N_{k,t}}{N_t}, k = 1, \dots, K$$

donde:

- N_t: número total de observaciones que llegan al nodo t.
- $N_{k,t}$: número de observaciones de la clase k en el nodo t.
- $p_{k|t}$: proporción de la clase k en el nodo t.

Esto se usa para poder calcular la impureza del nodo, el cual mide qué tan mezcladas están las clases dentro de ese nodo.

4.1. Índice Gini:

$$G(t) = 1 - \sum_{k=1}^{K} p_{k|t}^2$$

- $G(t)$: impureza Gini del nodo t.

Si en el nodo todas las observaciones pertenecen a una sola clase, se lo considera como un nodo puro, entonces $p_{k|t} = 1$ para una clase y 0 para el resto, lo que arrojaría un resultado de $G(t) = 0$. Un valor más alto nos indicaría que existe una mayor mezcla de clases.

6.11 Entropía:

$$H(t) = -\sum_{k=1}^{K} p_{k|t} \log p_{k|t}$$

- $H(t)$: entropía
- log: logaritmo

Es similar al índice Gini, ya que si $H(t)$ es 0, significa que el nodo contiene una sola clase.

6.12 Construcción del árbol

Se supone que tenemos un nodo padre t, el cual se divide en dos nodos hijo t_L (izquierdo) y t_R (derecho), con:

- N_t: número de observaciones en el nodo padre.
- N_L: número de observaciones en el nodo hijo izquierdo.
- N_R: número de observaciones en el nodo hijo derecho.

La reducción de impureza lograda por un split s se define como:

$$\Delta Q(s,t) = Q(t) - \left(\frac{N_L}{N_t} Q(t_L) + \frac{N_R}{N_t} Q(t_R)\right)$$

donde:

- $Q(t)$: impureza del nodo padre t (Gini o entropía).

- $Q(t_L)$, $Q(t_R)$: impureza de los nodos hijo izquierdo y derecho.
- $\frac{N_L}{N_t}$, $\frac{N_R}{N_t}$: proporción de observaciones que van a cada hijo.
- $\frac{N_L}{N_t}Q(t_L) + \frac{N_R}{N_t}Q(t_R)$: impureza esperada después del split.
- $\Delta Q(s,t)$: mejora neta de pureza que genera el split s en el nodo t.

Entonces:

$$\Delta Q(s,t) = \text{impureza antes del split} - \text{impureza promedio después del split}$$

Esto da a entender que es mejor que el valor sea más grande, ya que eso significa una mayor reducción de impureza.

6.13 Métodos ensemble, bootstrap y bagging en clasificación

6.13.1 Ensemble

Un método ensemble es aquel que combina varios clasificadores base para poder obtener un clasificador final más robusto. En términos coloquiales, se puede decir que es como tener un jurado de expertos en vez de uno solo. Supongamos que tenemos M clasificadores:

$$h_1(x), h_2(x), \dots, h_M(x)$$

y a cada uno se le asigna una clase a la observación x:

$h_m(x) \in \{1, \dots, K\}$.

La predicción del ensemble por voto mayoritario es:

$$\hat{C}_{ens}(x) = \arg\max_k \sum_{m=1}^{M} \mathbf{1}\{h_m(x) = k\}$$

donde:

- $\hat{C}_{ens}(x)$: clase predicha por el ensemble para la observación x.
- $\mathbf{1}\{h_m(x) = k\}$: vale 1 si el modelo base m predice la clase k, y 0 si no.
- La suma $\sum_{m=1}^{M} \mathbf{1}\{ h_m(x) = k\}$ cuenta cuántos modelos votan por la clase k.
- El operador $\arg\max_k$ selecciona la clase que recibe la mayor cantidad de votos.

6.13.2 Bootstrap

El Bootstrap se refiere a el como crear datos de entrenamiento variados a partir del mismo conjunto de datos original.

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$$

Una muestra bootstrap $\mathcal{D}_b$ se obtiene seleccionando n observaciones de $\mathcal{D}$, con reemplazo:

$$\mathcal{D}_b = \{(x_i, y_i): i \in I_b\}, | \ I_b \ | = n$$

donde:

- I_b: conjunto de las observaciones que se eligen para la muestra bootstrap número b.
- $| \ I_b \ |$: número de elementos de la muestra b, igual a n.

En cada muestra bootstrap, algunas observaciones pueden ser que se repitan, así como que otras no podrían aparecer; y estas últimas se denominan observaciones out-of-bag (OOB) respecto a esa muestra o árbol.

6.13.3 Bagging

Aquí se puede decir que se juntan la definición de los dos anteriores puntos, el ensemble y el bootstrapping, ya que el bagging usa bootstrap para entrenar múltiples clasificadores y luego combinarlos.

1. Se generan B muestras bootstrap $\mathcal{D}_1, \dots, \mathcal{D}_B$.
2. Se entrena un clasificador $h_b(\cdot)$ sobre cada muestra $\mathcal{D}_b$.

Para un nuevo dato x, es:

$$\hat{C}_{bag}(x) = \arg\max_k \sum_{b=1}^{B} \mathbf{1}\{ h_b(x) = k\}$$

donde:

- $\hat{C}_{bag}(x)$: clase elegida por el ensemble de bagging.
- $h_b(x)$: clase elegida por el clasificador entrenado con la muestra bootstrap $\mathcal{D}_b$.
- B: número total de clasificadores en el ensemble.
- $\arg\max_k$: se elige la clase con más votos.

En resumen, se puede decir que es un entrenamiento de muchos modelos sobre versiones bootstrap de los datos y luego se realiza un promedio o voto mayoritario de sus elecciones hechas por los mismos modelos.

$$H(p) = -\sum_{k=1}^{k} p_k \log_2 p_k$$

6.14 Random Forest

El algoritmo Random Forest (bosque aleatorio) es un método de ensemble que combina muchos árboles de decisión construidos sobre subconjuntos aleatorizados de los datos y de las variables. Fue formalizado por Leo Breiman (2001) como una familia de "predictores tipo árbol" que, al agregarse, mejoran la capacidad de generalización respecto a un solo árbol de decisión.

Por lo que se puede decir que, es un método ensemble que aplica bagging sobre árboles de decisión y añade una aleatoriedad adicional en la selección de variables en cada nodo.

6.15 Parámetros e Hiperparámetros del Random forest

Son los valores que asigna el analista de datos antes de entrenar y que controlan cómo se ajusta el modelo.

Algunos ejemplos de estos son:

- B = número de árboles
- m_{try} = número de variables candidatas en cada división
- max_depth = profundidad máxima de cada árbol
- $min\_samples\_split$ = número mínimo de ejemplos para dividir un nodo
- $min\_samples\_leaf$ = número mínimo de ejemplos de una hoja
- $bootstrap \in \{True, False\}$ = si se usan muestras Bootstrap o se entrena con conjunto completo

6.16 Construcción del bosque de clasificación

Dado un conjunto de entrenamiento $\mathcal{D}$:

1. Para $b = 1, \dots, B$:

- Se genera una muestra bootstrap $\mathcal{D}_b$.
- Se entrena un árbol de decisión de clasificación $h_b(\cdot)$sobre $\mathcal{D}_b$.
- En cada nodo del árbol, en lugar de considerar todas las p variables, se selecciona aleatoriamente un subconjunto de tamaño m y el mejor split se busca solo dentro de esas m variables.

Parámetros clave:

- B: número de árboles del bosque.
- p: número total de variables predictoras.
- m: número de variables candidatas a división en cada nodo.

6.17 Regla de clasificación en Random Forest

La elección de clase del Random Forest para una nueva observación x se obtiene por voto mayoritario de los B árboles:

$$\hat{C}_{RF}(x) = \arg\max_{k} \sum_{b=1}^{B} \mathbf{1}\{ h_b(x) = k\}$$

donde:

- $\hat{C}_{RF}(x)$: clase elegida por el Random Forest para la observación x.
- $h_b(x)$: clase elegida por el árbol b.

- $\mathbf{1}\{h_b(x) = k\}$: es 1 si el árbol b asigna la clase k a x.
- La suma cuenta cuántos árboles predicen cada clase k.

6.18 Error Out-of-Bag (OOB) en clasificación

Debido al uso de bootstrap, para cada observación existen árboles que no han visto ciertas observaciones en su entrenamiento.

$$\mathcal{B}_i = \{b \in \{1, \dots, B\} : (x_i, y_i) \notin \mathcal{D}_b\}$$

donde:

- $\mathcal{B}_i$: conjunto de árboles para los cuales la observación i es out-of-bag.
- $\mathcal{D}_b$: muestra bootstrap usada para entrenar el árbol b.

La clasificación OOB para la observación i se calcula votando solo con los árboles donde la observación fue OOB:

$$\hat{C}_{OOB}(x_i) = \arg\max_k \sum_{b \in \mathcal{B}_i} \mathbf{1}\{h_b(x_i) = k\}$$

El error global es:

$$\text{err}_{OOB} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{\hat{C}_{OOB}(x_i) \neq y_i\}$$

donde:

- $\hat{C}_{OOB}(x_i)$: clase elegida para la observación i usando solo árboles donde esa observación fue OOB.

- $1\{\hat{C}_{OOB}(x_i) \neq y_i\}$: vale 1 si la predicción OOB es incorrecta, 0 si es correcta.

6.19 Métricas de evaluación

6.19.1 Matriz de confusión

Es una tabla donde se puede observar las cantidades de clases reales contra las predichas.

	$Pred. Positiva$	$Pred. Negativa$
$Real\ Positiva$	TP	FN
$Real\ Negativa$	FP	TN

Donde:

$$TP = verdaderos\ positivos$$

$$TN = verdaderos\ negativos$$

$$FP = falsos\ positivos$$

$$FN = falsos\ negativos$$

6.19.2 F1-Score

Es la media armónica entre Precision y Recall; saca a relucir su utilidad cuando existe un desbalance de clases, porque penaliza cuando una de las dos es muy baja.

$$F1 = \frac{2 \times Precisión \times Recall}{Precisión + Recall}$$

6.19.3 Accuracy

Es la proporción de predicciones correctas sobre el total de ejemplos.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Donde:

$$TP = verdaderos\ positivos$$

$$TN = verdaderos\ negativos$$

$$FP = falsos\ positivos$$

$$FN = falsos\ negativos$$

6.19.4 Precision

Es la fracción de las predicciones positivas del modelo que son realmente positivas.

$$Precision = \frac{TP}{TP + FP}$$

Donde:

$$TP = verdaderos\ positivos$$

$$FP = falsos\ positivos$$

6.19.5 Recall

Nos señala la fracción de todos los positivos reales que fueron capturados por el modelo.

$$Recall = \frac{TP}{TP + FN}$$

Donde:

$$TP = verdaderos\ positivos$$

$$FN = falsos\ negativos$$

6.20 Curva Roc

Es un gráfico bidimensional donde en el eje X se representa la tasa de falsos positivos y en el eje Y la tasa de verdaderos positivos.

Figura 2
Curva ROC

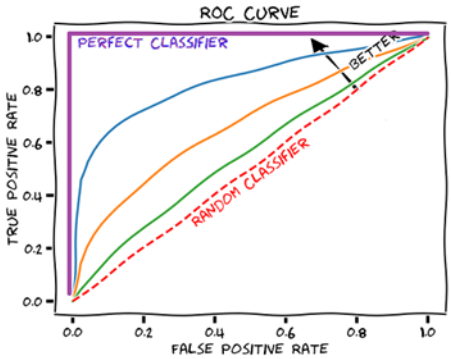


Ilustración 2: Gráfico que ilustra el cálculo y la interpretación de la curva ROC y el área bajo la curva (AUC) como medida de rendimiento de modelos. Adaptado de Measuring performance: AUC / AUROC, (2019).

decisión se expresa mediante un umbral τ:

$$\hat{y}(x;\tau) = \begin{cases} Y & \text{si } f(x) \geq \tau \\ N & \text{si } f(x) < \tau \end{cases}$$

donde:

- $f(x)$: probabilidad de la clase positiva para la observación x
- τ: umbral elegido
- Y: clase elegida positiva
- N: clase elegida negativa

Para cada valor de τ se obtiene:

$$\text{TPR}(\tau) = \frac{TP(\tau)}{P}, \text{FPR}(\tau) = \frac{FP(\tau)}{N}$$

$$TP = verdaderos\ positivos$$

$$FP = falsos\ positivos$$

La curva ROC es el conjunto de todos esos puntos:

$$\text{ROC} = \{(\text{FPR}(\tau), \text{TPR}(\tau)) : \tau \in \mathbb{R}\}$$

La diagonal entre (0,0) y (1,1) representa el comportamiento de un clasificador *m.* Un modelo útil debe situarse por encima de esa diagonal, acercándose al punto ideal (0,1), que correspondería a una clasificación perfecta (sin falsos positivos y sin falsos negativos).

6.20.1 Área bajo la curva ROC (AUC)

Para resumir el comportamiento de la curva se usa el área bajo la curva ROC, donde este valor se encuentra dentro de un intervalo entre 0 y 1:

AUC ≈ 0,5: azar

AUC = 1: muy buena capacidad de discriminar

6.20.2 Cálculo de la curva ROC

Para poder encontrar el área bajo la curva se hace de la siguiente manera:

1. Se tiene la curva ROC como una secuencia de puntos $(\text{FPR}_i, \text{TPR}_i)$ ordenados por FPR.
2. En lugar de contar solo rectángulos, se suman áreas de trapecios entre cada par de puntos consecutivos.
3. Al final se divide por el área total del cuadrado unidad para que el resultado quede escalado entre 0 y 1.

El área se aproxima con la regla del trapecio:

$$\text{AUC} \approx \sum_{i=1}^{n-1} \underbrace{(\text{FPR}_{i+1} - \text{FPR}_i)}_{\text{base del trapecio}} \cdot \underbrace{\frac{\text{TPR}_{i+1} + \text{TPR}_i}{2}}_{\text{altura media}}$$

- $\text{FPR}_{i+1} - \text{FPR}_i$: ancho del trapecio en el eje X.
- $\frac{\text{TPR}_{i+1} + \text{TPR}_i}{2}$: promedio de las dos alturas.

Una vez calculado esto, solo se comienza a recorrer la curva y acumulando el área de cada trapecio hasta obtener el AUC total.

7 Marco Conceptual

7.1 Inteligencia artificial

Con el pasar de los años el campo de la inteligencia artificial (IA) ha evolucionado y se ha convertido en un área interdisciplinaria que abarca desde la informática.

Según Rouhiainen (2018) describe a la inteligencia artificial como “la capacidad de las máquinas para usar algoritmos, aprender de los datos y emplear los conocimientos adquiridos al hacer elecciones de la misma manera que lo haría una persona” (p. 17).

La primera persona en hablar sobre IA fue el científico Alan Turing, quien en 1950 publicó su artículo llamado “Computing machinery and intelligence”, que decía que si una máquina puede comportarse como una persona implica tener inteligencia; de esto surgió la Prueba de Turing, y si un dispositivo lograba superar esa evaluación, era fundamental contar con:

- Reconocimiento del lenguaje natural.

- Razonamiento automatizado.

- Aprendizaje automático.

- Representación del conocimiento. (Russell y Norvig, 2010, p. 2)

7.2 Machine Learning (Aprendizaje automático)

Según la investigación de Jiménez Anderson & Diaz José (2021) mencionan que:

> El aprendizaje automático comúnmente abreviado como ML, es un tipo de inteligencia artificial (IA) que “aprende” o se adapta con el tiempo. En lugar de seguir reglas estáticas

codificadas en un programa, esta tecnología identifica patrones de entrada y contiene algoritmos que evolucionan con el tiempo. (p.114)

El Machine Learning se basa en tres conceptos importantes:

- Datos: La información que es procesada por el algoritmo
- Modelos: Representaciones matemáticas que intentan percibir los patrones o relaciones dentro de los datos.
- Algoritmos: Procedimientos que dirigen el aprendizaje y la mejora de los modelos (Alpaydin, 2010).

7.3 Aprendizaje Supervisado

Raschka y Mirjalili (2017) han mencionado lo siguiente:

> El objetivo principal del aprendizaje supervisado es aprender un modelo a partir de datos de entrenamiento etiquetados que nos permita hacer predicciones sobre datos desconocidos o futuros. Aquí, el término supervisado se refiere a un conjunto de muestras en las que las señales de salida deseadas (etiquetas) ya se conocen. (p.3)

En este enfoque, el modelo se entrena con un conjunto de datos etiquetados, que significa que los datos contienen ejemplos junto con sus respuestas correctas. El objetivo es que el modelo aprenda a predecir la respuesta correcta para nuevos datos no etiquetados. Los métodos de agrupamiento y estimación son casos de aprendizaje con supervisión (Murphy, 2012).

7.4 3.1 Variable objetivo y variable predictora

Alpaydin (2010) ha mencionado lo siguiente:

> Tanto la regresión como la clasificación son problemas de aprendizaje supervisado en los que hay una entrada, X, una salida, Y, y la tarea consiste en aprender la correspondencia entre la entrada y la salida. El enfoque en el aprendizaje automático es que asumimos un modelo definido hasta un conjunto de parámetros: $y=g(x|\theta)$ donde $g(\cdot)$ es el modelo y θ son sus parámetros. Y es un número en la regresión y es un código de clase (por ejemplo, 0/1) en el caso de la clasificación. $g(\cdot)$ es la función de regresión o, en la clasificación, es la función discriminante que separa las instancias de diferentes clases. (p. 9)

7.5 3.2 Conjunto de entrenamiento

Hastie, Tibshirani y Friedman (2009) afirman lo siguiente:

> Se tiene un conjunto de datos de entrenamiento, en el que se observan los resultados y las mediciones de las características de un conjunto de objetos (como personas). Utilizando estos datos, se construye un modelo de predicción, o aprendiz, que permitirá predecir el resultado para nuevos objetos desconocidos. (pp. 1-2)

7.6 3.3 Conjunto de validación

Alpaydin (2010) menciona lo siguiente:

> Se utiliza una parte para el entrenamiento (es decir, para ajustar una hipótesis), y la parte restante se denomina conjunto de validación y se utiliza para probar la capacidad de generalización. Es decir, dado un conjunto de posibles clases de hipótesis Hi, para cada una de ellas se ajusta la mejor $hi \in Hi$ en el conjunto de entrenamiento. (p. 40)

7.7 3.4 Conjunto de prueba

Alpaydin (2010) menciona lo siguiente:

> Se necesita un tercer conjunto, un conjunto de prueba, a veces también llamado conjunto de publicación, que contenga ejemplos no utilizados en el entrenamiento o la validación. Una analogía en la vida cotidiana es cuando se toma un curso: los ejemplos de problemas que el instructor resuelve en clase mientras enseña una materia forman el conjunto de entrenamiento; las preguntas del examen son el conjunto de validación; y los problemas que se resuelven en nuestra vida profesional posterior son el conjunto de prueba. (p. 40)

7.8 Algoritmos de Clasificación

Bird, Klein y Loper (2009) mencionaron que:

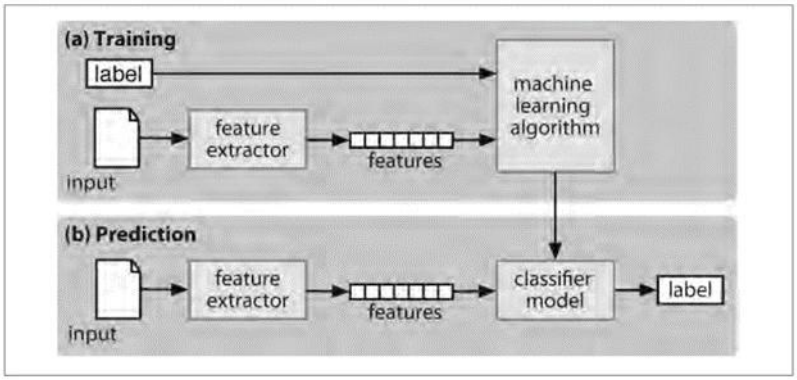
> La clasificación es la tarea de elegir la etiqueta de clase correcta para una entrada determinada. En las tareas básicas de clasificación, cada entrada se considera de forma aislada del resto de entradas, y el conjunto de etiquetas se define de antemano. Algunos ejemplos de tareas de clasificación son:
>
> - Decidir si un correo electrónico es spam o no.
> - Decidir de que tema trata un artículo de noticias.
> - Decidir si una determinada aparición de la palabra «banco» se utiliza para referirse a la orilla de un río, a una institución financiera, al acto de inclinarse hacia un lado o al acto de depositar algo en una institución financiera.

La clasificación, tiene variantes muy interesantes: la clasificación multicategoría, donde se pueden asignar varias etiquetas a cada instancia; y la clasificación de clase abierta, donde no

se conoce el conjunto de etiquetas de antemano; y la clasificación secuencial proporciona clasificaciones conjuntas de una serie de entradas.

Un clasificador se denomina supervisado si se construye a partir de corpus de entrenamiento que contienen la etiqueta correcta para cada entrada. (pp. 221-222)

Figura 3
Clasificación supervisada.



Nota. a) Durante el entrenamiento, se utiliza un extractor de características para convertir cada valor de entrada en un conjunto de características. Los pares de conjuntos de características y etiquetas se introducen en el algoritmo de aprendizaje automático para generar un modelo. (b) Durante la predicción, se utiliza el mismo extractor de características para convertir entradas no vistas en conjuntos de características. A continuación, estos conjuntos de características se introducen en el modelo, que genera etiquetas predichas. Adaptado de *Natural language processing with Python*, por Bird et al. (2009).

7.9 Teorema de Bayes

Han, Kamber y Pei (2012) han afirmado lo siguiente:

> El teorema de Bayes lleva el nombre de Thomas Bayes, un clérigo inglés inconformista que realizó los primeros trabajos sobre probabilidad y teoría de la decisión durante el siglo XVIII. Sea X una tupla de datos. En términos bayesianos, X se considera «evidencia». Como es habitual, se describe mediante mediciones realizadas en un conjunto de n atributos. Sea H una hipótesis tal como que la tupla de datos X pertenece a una clase C especificada. Para los problemas de clasificación, se quiere determinar P(H|X), la

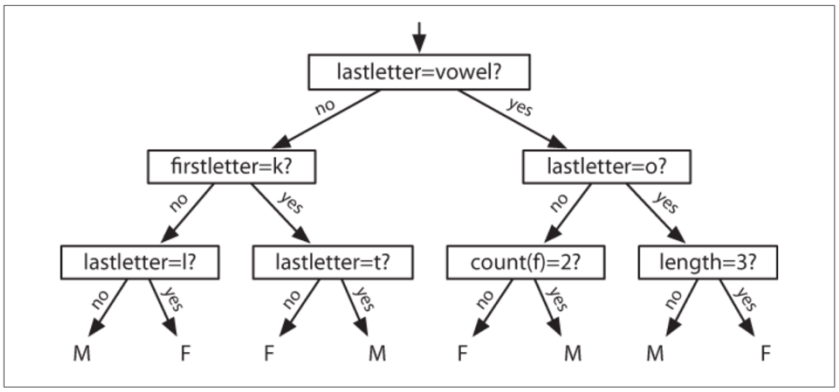
probabilidad de que la hipótesis H se cumpla dada la «evidencia» o tupla de datos observada X. En otras palabras, se busca la probabilidad de que la tupla X pertenezca a la clase C, dado que se conoce la descripción de los atributos de X. P(H|X) es la probabilidad a posteriori, o probabilidad posterior, de H condicionada a X. (pp. 350-351)

7.10 Árbol de decisión

Bird, et al. (2009) señalaron que:

> Un árbol de decisión es un diagrama de flujo sencillo que selecciona etiquetas para los valores de entrada. Este diagrama de flujo consta de nodos de decisión, que comprueban los valores de las características, y nodos hoja, que asignan etiquetas. Para elegir la etiqueta de un valor de entrada, se comienza en el nodo de decisión inicial del diagrama de flujo, conocido como nodo raíz. (p. 242)

Figura 4
Modelo de árbol de decisión.



Nota. Modelo de árbol de decisión para la tarea de género del nombre. Tener en cuenta que los diagramas de árbol se dibujan convencionalmente al revés, con la raíz en la parte superior y las hojas en la parte inferior. Adaptado de *Natural language processing with Python*, por Bird et al. (2009).

Otros autores afirman lo siguiente:

> Para un árbol de clasificación, se predice que cada observación pertenece a la clase más común de las observaciones de entrenamiento en la región a la que pertenece. Al interpretar los resultados de un árbol de clasificación, a menudo interesa no solo la predicción de clase correspondiente a una región de nodo terminal concreta, sino también las proporciones de clase entre las observaciones de entrenamiento que entran en esa región. (James, Witten, Hastie y Tibshirani, 2021, p. 335)

7.11 6.1 Estructura de un árbol de decisión

Rivero (2022) menciona que:

> Un árbol de decisión contiene los siguientes elementos:
>
> - **Nodo de decisión.** Aquel cuando un subnodo se divide en subnodos adicionales indicando la decisión que se tomará ante la disyuntiva.
> - **Nodo de probabilidad u hoja.** Presenta múltiples resultados inciertos. Describe las probabilidades de acierto de cada una de las opciones que se plantea ante una disyuntiva.
> - **Nodo terminal.** Indica el resultado definitivo como última opción en una ruta de decisión.
> - **División.** Proceso de división de un nodo en dos o más subnodos.
> - **Flecha.** Nexo que unen los nodos entre sí.
> - **Vector.** Representa la opción por la que se opta entre todas las posibilidades que define el nodo.
> - **Poda.** Esta se da cuando se reduce el tamaño de los árboles de decisión eliminando nodos opuestos a la división.

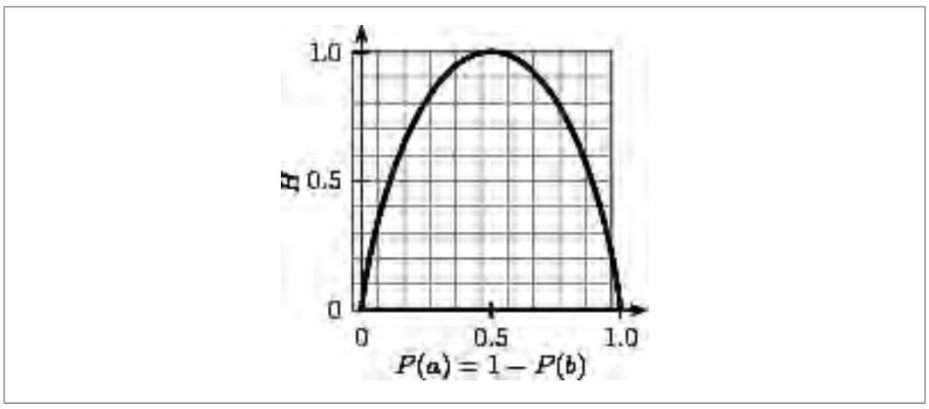
- **Rama o subárbol.** Una subsección del árbol de decisión se denomina rama o subárbol.
- **Etiqueta.** Permite la unión de nodos y flechas que denominan las acciones que se llevan a cabo. (p. 40)

7.12 6.2 Entropía

Rivero (2022) afirma lo siguiente:

> La entropía es una medida de la homogeneidad de los datos aleatorios presentes en un nodo de un árbol de clasificación. Si los datos son similares, el valor de entropía es pequeño o nulo.
>
> Se concluye que la entropía es cero si todos los elementos del grupo pertenecen a la misma clase, y la entropía es uno cuando la muestra contiene el mismo número de observaciones positivas y negativas. Si la muestra contiene un número desigual de observaciones positiva y negativas, la entropía está entre 0 y 1. (p. 42)

Figura 5
Entropía



Nota. La entropía de las etiquetas en la tarea de predicción del género de nombres, en función del porcentaje de nombres masculinos en un conjunto dado. Adaptado de *Natural language processing with Python*, por Bird et al. (2009).

7.13 6.3 Limitaciones de los árboles

Según Rivero (2022), “la incorporación de datos categóricos con múltiples niveles provoca que los resultados se inclinen a favor de los atributos con mayoría de niveles.

Los cálculos pueden volverse complejos al afrentarse con la falta de certezas y numerosos resultados relacionados (p. 41).

7.14 Sobreajuste

Alpaydin (2010) afirma lo siguiente:

> Si hay ruido, una hipótesis demasiado compleja puede aprender no solo la función subyacente, sino también el ruido en los datos y puede dar lugar a un mal ajuste, por ejemplo, al ajustar un polinomio de sexto orden a datos ruidosos muestreados a partir de un polinomio de tercer orden. Esto se denomina sobreajuste. En tal caso, disponer de más datos de entrenamiento ayuda, pero solo hasta cierto punto. Dado un conjunto de entrenamiento y H, se puede encontrar $h \in H$ que tenga el mínimo error de entrenamiento, pero si H no se elige bien, no importa qué $h \in H$ elijamos, no se tendrá una buena generalización. (p. 39)

7.15 Alta Variabilidad

Hastie, et al. (2009) mencionan que:

> Un problema importante de los árboles es su alta variabilidad. A menudo, un pequeño cambio en los datos puede dar lugar a una serie de divisiones muy diferentes, lo que hace que la interpretación sea algo precaria. La razón principal de esta inestabilidad es la naturaleza jerárquica del proceso: el efecto de un error en la división superior se propaga a todas las divisiones inferiores. (p. 312)

7.16 Ensemble

Géron (2019) indica que:

> Un grupo de predictores se denomina conjunto; por lo tanto, esta técnica se denomina aprendizaje conjunto, y un algoritmo de aprendizaje conjunto se denomina método conjunto.
>
> Como ejemplo de método conjunto, se puede entrenar un grupo de clasificadores de árboles de decisión, cada uno en un subconjunto aleatorio diferente del conjunto de entrenamiento. (p. 189)

7.17 Bootstrap

Alpaydin (2010) señala que:

> Para generar múltiples muestras a partir de una sola muestra, una alternativa a la validación cruzada es el bootstrap, que genera nuevas muestras extrayendo instancias de la muestra original con reemplazo. Las muestras del bootstrap pueden solaparse más que las muestras de la validación cruzada y, por lo tanto, sus estimaciones son más dependientes; pero se considera la mejor forma de realizar el remuestreo para conjuntos de datos muy pequeños. (p. 489)

7.18 Bagging

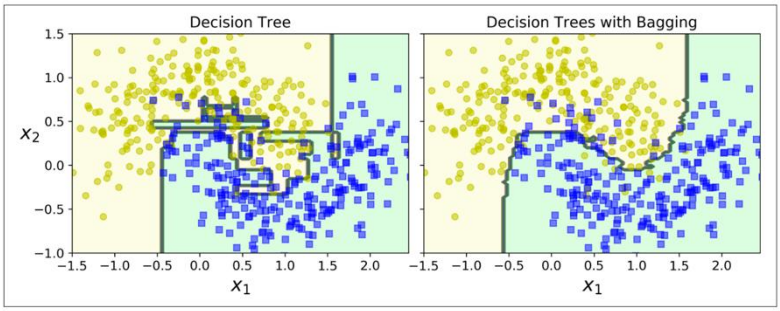
Alpaydin (2010) afirma lo siguiente:

> El bagging es un método de votación mediante el cual se diferencian los aprendices básicos entrenándolos con conjuntos de entrenamiento ligeramente diferentes. La generación de muestras ligeramente diferentes a partir de una muestra dada se realiza mediante bootstrap,

donde, dado un conjunto de entrenamiento X de tamaño N, se extrae N instancias aleatoriamente de X con reemplazo. Debido a que el muestreo se realiza con reemplazo, es posible que algunas instancias se extraigan más de una vez y que ciertas instancias no se extraigan en absoluto. (p. 430)

Figura 6

Un solo árbol de decisión (izquierda) versus un conjunto de 500 árboles (derecha)



Nota. Comparación entre un único árbol de decisión y un conjunto de árboles generados mediante bagging, mostrando la mejora en la estabilidad del modelo. Adaptado de *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.), por Gerón, (2019).

7.19 Poda del árbol

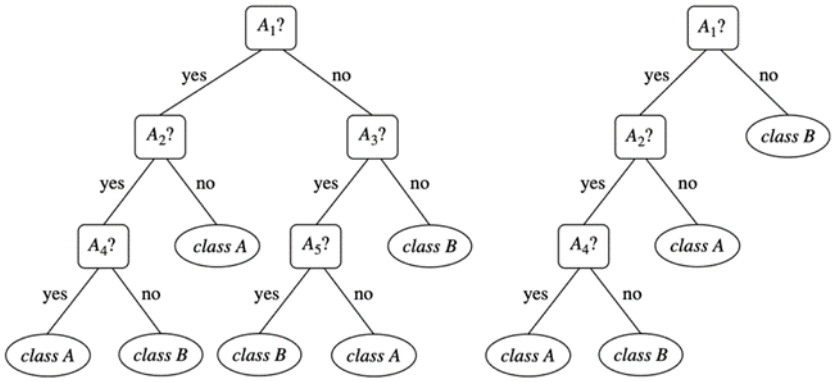
Murphy (2012) señala que:

Para evitar el sobreajuste, se puede detener el crecimiento del árbol si la disminución del error no es suficiente para justificar la complejidad adicional que supone añadir un subárbol extra. Sin embargo, esto tiende a ser demasiado miope.

Por lo tanto, el enfoque estándar consiste en hacer crecer un árbol «completo» y, a continuación, realizar una poda. Esto se puede hacer utilizando un esquema que pode las ramas que producen el menor aumento del error.

Para determinar hasta dónde recortar, se puede evaluar el error validado cruzado en cada uno de esos subárboles y, a continuación, elegir el árbol cuyo error CV esté dentro de 1 error estándar del mínimo. (p. 549)

Figura 7
Un árbol de decisiones sin podar y una versión podada del mismo.



Nota. Comparación un árbol de decisiones sin poda y su versión podada, mostrando la reducción de complejidad del modelo. Adaptado de *Data mining: Concepts and techniques* (3rd ed.), por Han et al. (2012).

7.20 Random Forest

Marsland (2014) ha mencionado lo siguiente:

> La idea es, en gran medida, que si un árbol es bueno, entonces muchos árboles (un bosque) deberían ser mejores, siempre que haya suficiente variedad entre ellos. Lo más interesante de un bosque aleatorio es la forma en que crea aleatoriedad a partir de un conjunto de datos estándar. (p. 275)

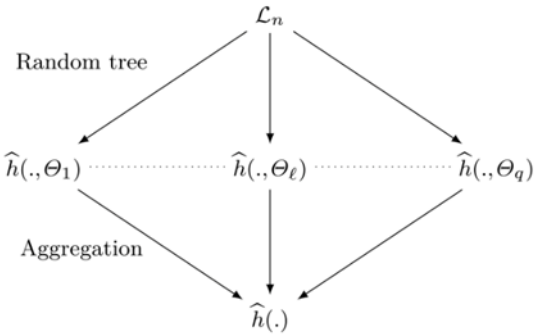
Otros autores mencionan que:

> La técnica conocida como bosques aleatorios (Breiman 2001a) intenta decorrelacionar los aprendices básicos mediante el aprendizaje de árboles basados en un subconjunto elegido aleatoriamente de variable de entrada, así como en un subconjunto elegido aleatoriamente de casos de datos. Estos modelos suelen tener una precisión predictiva muy buena (Caruana y Niculescu-Mizil 2006) y se han utilizado ampliamente en muchas aplicaciones (por ejemplo, para el reconocimiento de la postura corporal utilizando el popular sensor Kinect de Microsoft (Shotton et al. 2011)). (Murphy, 2012, p. 551)

Asimismo, James et al. (2021) afirman que:

> Los bosques aleatorios proporcionan una mejora con respecto a los árboles empaquetados mediante un pequeño ajuste que descorrelaciona los árboles. Al igual que en el empaquetado, se construye una serie de árboles de decisión sobre muestras de entrenamiento bootstrapped. Pero al construir estos árboles de decisión, cada vez que se considera una división en un árbol, se elige una muestra aleatoria de m predictores como candidatos a la división del conjunto completo de p predictores. La división solo puede utilizar uno de esos m predictores. (p. 343)

Figura 8
Esquema general de Bosques Aleatorios.



Nota. Esquema que ilustra la construcción de árboles aleatorios y su proceso de agregación para generar una predicción final. Adaptado de *Random Forests with R*, por Genuer, R., y Poggi, J.-M. (2020).

7.21 11.1. Hiperparámetros del Random Forest

7.21.1 Número de árboles (n_estimators)

Este parámetro indica la cantidad de árboles que forman el bosque. Un mayor número de árboles ayuda a reducir la variabilidad, ya que la predicción final es un promedio o votación entre varios modelos, en este caso de árboles. Esto genera resultados más estables, aunque incrementa el tiempo de procesamiento.

7.21.2 Número de variables por división (max_features / mtry)

Define cuántas variables se seleccionan de forma aleatoria en cada división del árbol. Esto introduce variedad entre los árboles, evitando que todos aprendan los mismos patrones. Mejorando así la capacidad predictiva general del modelo.

7.21.3 Bootstrap para generar cada árbol (bootstrap)

Consiste en crear muestras aleatorias a partir del conjunto de datos original. Cada árbol se entrena con una muestra diferente ayudando a disminuir el sobreajuste y mejorando la generalización del modelo.

7.21.4 Error out-of-bag (oob_score)

Las observaciones que no se utilizan en la muestra bootstrap sirven de validación. Con ellas se puede estimar el error sin la necesidad de crear un data set más derivado del data set original. Esto permite evaluar el desempeño aprovechando mejor la información disponible.

7.21.5 Profundidad máxima del árbol (max_depth)

Esto limita el crecimiento del árbol y evita que se vuelva demasiado difícil de entender. Al restringir la profundidad, el modelo reduce el riesgo de sobreajuste. De esta manera, se obtienen reglas de decisión más generales.

7.21.6 Mínimo de muestras para dividir un nodo (min_samples_split)

Establecer un número mínimo de datos por nodo evita divisiones basadas en muy pocas observaciones, lo que reduce el riesgo de estructuras demasiado detalladas.

7.21.7 Mínimo de muestras por hoja (min_samples_leaf)

Establece el número mínimo de observaciones que debe tener cada nodo. Así se evita que el árbol genere divisiones que puedan basarse en pocos datos.

7.21.8 Criterio de división (criterion)

Define la forma en la que el modelo evalúa la calidad de cada partición. Por lo general se usan medidas como la impureza de Gini o la entropía seleccionando la división que logra una mayor separación entre las clases.

7.21.9 Número máximo de nodos terminales (max_leaf_nodes)

Esto limita la cantidad total de hojas que puede tener el árbol controlando la complejidad del modelo, evitando un crecimiento excesivo que podría generar un sobreajuste.

7.21.10 Fracción de muestras por árbol (max_samples)

Definir qué porción de los datos se usa para cada árbol permite ajustar el grado de variabilidad entre los distintos modelos del bosque. (Hastie, et al. , 2009, cap. 15)

7.22 Matriz de confusión

Flach (2012) ha mencionado lo siguiente:

> El rendimiento de estos clasificadores se puede resumir mediante una tabla conocida como tabla de contingencia o matriz de confusión (izquierda). En esta tabla, cada fila se refiere a las clases reales registradas en el conjunto de prueba, y cada columna a las clases predichas por el clasificador. Así, por ejemplo, la primera fila indica que el conjunto de prueba contiene 50 positivos, 30 de los cuales se predijeron correctamente y 20 incorrectamente.

La última columna y la última fila proporcionan los marginales (es decir, las sumas de columnas y filas). Los marginales son importantes porque permiten evaluar la significación estadística. Por ejemplo, la tabla de contingencia (derecha) tiene los mismos marginales, pero el clasificador claramente hace una elección aleatoria en cuanto a qué predicciones son positivas y cuáles son negativas; como resultado, la distribución de positivos y negativos reales en cualquiera de las clases predichas es la misma que la distribución general (uniforme en este caso). (pp. 53-54)

Figura 9
Matriz de confusión.

	Predicted ⊕	*Predicted* ⊖	
Actual ⊕	30	20	50
Actual ⊖	10	40	50
	40	60	100

	⊕	⊖	
⊕	20	30	50
⊖	20	30	50
	40	60	100

Nota. (izquierda) Una tabla de contingencia de dos clases o matriz de confusión que representa el rendimiento del árbol de decisiones. Los números en la diagonal descendente indican las predicciones correctas, mientras que la diagonal ascendente se refiere a errores de predicción. (Derecha) Una tabla de contingencia con los mismos marginales, pero con filas y columnas independientes. Adaptado de *Machine learning: The art and science of algorithms that make sense of data*, por Flach, (2012).

7.23 Métricas de evaluación

7.23.1 Accuracy

Bird, et al. (2009) señalaron que:

> La métrica más simple que se puede utilizar para evaluar un clasificador, la precisión, mide el porcentaje de entradas en el conjunto de prueba que el clasificador ha etiquetado correctamente. Por ejemplo, un clasificador de género de nombres que predice el nombre correcto 60 veces en un conjunto de prueba que contiene 80 nombres tendría una precisión de 60/80 = 75 %. (p. 239)

7.23.2 Precision

Flach (2012) afirma que:

> La precisión es la contrapartida de la tasa de verdaderos positivos en el siguiente sentido: mientras que la tasa de verdaderos positivos es la proporción de positivos predichos entre los positivos reales, la precisión es la proporción de positivos reales entre los positivos predichos. (p. 57)

7.23.3 Recall

Marsland (2014) define a la recuperación como "la proporción del número de ejemplos positivos correctos entre los que se clasificaron como positivos, que es lo mismo que la sensibilidad" (p. 23).

7.23.4 F1-Score

Han, et al. (2012) han afirmado lo siguiente:

> Una forma alternativa de utilizar la precisión y la recuperación es combinarlas en una única medida. Este es el enfoque de la medida F (también conocida como puntuación F1 o puntuación F) y la medida Fβ. La medida F es la media armónica de la precisión y la recuperación. Da la misma importancia a la precisión y a la recuperación. (pp. 368-369)

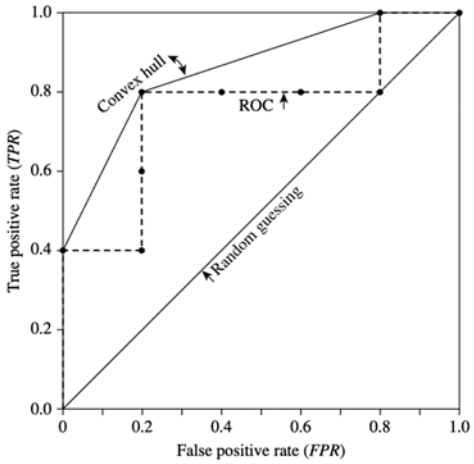
7.24 Curvas ROC

Han, et al. (2012) mencionan que:

> Las curvas características de funcionamiento del receptor son una herramienta visual útil para comparar dos modelos de clasificación. Las curvas ROC provienen de la teoría de detección de señales que se desarrolló durante la Segunda Guerra Mundial para el

análisis de imágenes de radar. Una curva ROC para un modelo determinado muestra la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR). (p. 374)

Figura 10
Curva ROC.



Nota. Representación de la curva ROC (Receiver Operating Characteristic), que muestra la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR), así como la línea de clasificación aleatoria. Adaptado de *Data mining: Concepts and techniques* (3rd ed.), por Han et al. (2012).

7.25 Cross-Validation

Bird, et al. (2009) indican que:

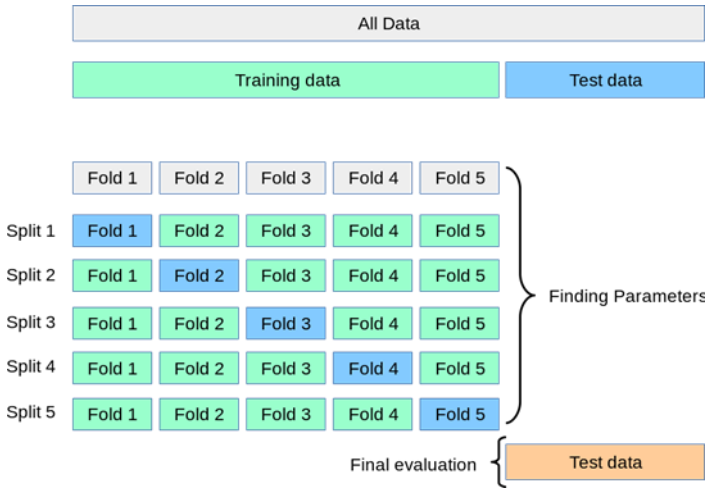
Para evaluar los modelos, se debe reservar una parte de los datos anotados para el conjunto de prueba. Como ya se ha mencionado, si el conjunto de prueba es demasiado pequeño, la evaluación puede no ser precisa. Sin embargo, aumentar el tamaño del conjunto de prueba suele implicar reducir el tamaño del conjunto de entrenamiento, lo que puede tener un impacto significativo en el rendimiento si se dispone de una cantidad limitada de datos anotados.

Una solución a este problema es realizar múltiples evaluaciones en diferentes conjuntos de prueba y luego combinar las puntuaciones de esas evaluaciones, una técnica conocida

como validación cruzada. En concreto, se subdivide el corpus original en N subconjuntos llamados pliegues. Para cada uno de estos pliegues, se entrena un modelo utilizando todos los datos excepto los datos de ese pliegue, y luego se prueba ese modelo en el pliegue. (p. 241)

Figura 11

Esquema del proceso de validación cruzada k-fold



Nota. Diagrama que ilustra la división del conjunto de datos en datos de entrenamiento y prueba, y la aplicación de validación cruzada k-fold. Adaptado de *Cross-validation* de scikit-learn (s. f.)

8 Marco Legal

El presente trabajo utiliza información proveniente de registros relacionados con las importaciones del mercado ecuatoriano, los cuales contienen datos de carácter comercial, operativo y administrativo asociados a los procesos de clasificación y asignación del sector destino. En función de ello, el desarrollo de la investigación se realiza conforme a los lineamientos establecidos por la Superintendencia de Protección de Datos Personales, como organismo competente en materia de protección y gobernanza de la información en el Ecuador.

Si bien el proyecto no implica el tratamiento de datos personales de personas naturales, dado que la información analizada corresponde a operaciones de comercio exterior, partidas arancelarias y variables asociadas a procesos productivos y comerciales, sí se trabaja con datos privados y sensibles desde el punto de vista empresarial, cuya divulgación o uso inadecuado podría afectar la confidencialidad y los intereses de las organizaciones involucradas. Por tal motivo, se considera necesario establecer un marco legal y ético que garantice un manejo responsable, seguro y transparente de la información utilizada.

En este contexto, se toma como principal referencia la Ley Orgánica de Protección de Datos Personales (LOPDP), promulgada en 2021, la cual constituye la normativa nacional aplicable en materia de protección de datos en el Ecuador. De acuerdo con lo establecido en el Artículo 2 de dicha ley, quedan excluidos de su ámbito de aplicación los datos anonimizados o aquellos que no permiten identificar directa o indirectamente a una persona natural. En concordancia con esta disposición, los datos empleados en la presente investigación han sido tratados de forma agregada y sin elementos que permitan la identificación de personas naturales.

a) Juridicidad: El tratamiento de la información proporcionada por la empresa se desarrollará conforme al marco normativo vigente, garantizando que el uso de los datos sea lícito y

coherente con los fines científicos y académicos de la investigación (Ley Orgánica de Protección de Datos Personales, 2021, art. 10).

b) Lealtad: Los datos facilitados serán gestionados de manera ética y responsable, asegurando una relación basada en la confianza con la empresa colaboradora y evitando prácticas que puedan generar interpretaciones indebidas o usos distintos a los previamente acordados (Ley Orgánica de Protección de Datos Personales, 2021, art. 10).

c) Transparencia: La entidad que provee la información contará con conocimiento claro y oportuno sobre los objetivos del tratamiento de los datos, las etapas del proceso de investigación y el alcance de su utilización durante el desarrollo del modelo de clasificación (Ley Orgánica de Protección de Datos Personales, 2021, art. 10).

d) Finalidad: La información recopilada será utilizada únicamente para el diseño, entrenamiento y validación del algoritmo de clasificación orientado a la asignación del sector destino en las importaciones del mercado ecuatoriano, sin ser empleada para fines ajenos a los objetivos del estudio (Ley Orgánica de Protección de Datos Personales, 2021, art. 10).

e) Pertinencia y minimización: Se limitará la recopilación de datos a aquellos estrictamente necesarios para el funcionamiento y desempeño del modelo, evitando la inclusión de variables irrelevantes o que no aporten valor analítico al proceso de clasificación (Ley Orgánica de Protección de Datos Personales, 2021, art. 10).

f) Proporcionalidad: El volumen y tipo de información tratada se ajustará de manera proporcional al alcance y necesidades del proyecto, procurando no procesar datos que excedan los requerimientos técnicos y metodológicos de la investigación (Ley Orgánica de Protección de Datos Personales, 2021, art. 10).

g) Confidencialidad: Los datos suministrados serán considerados de carácter reservado y se aplicarán medidas que impidan su acceso, uso o divulgación no autorizada, reconociendo su

valor estratégico y sensible para la empresa que los proporciona (Ley Orgánica de Protección de Datos Personales, 2021, art. 10).

h) Calidad y exactitud: La investigación se apoyará en información depurada, consistente y actualizada, dado que la calidad de los datos constituye un factor determinante para la fiabilidad de los resultados obtenidos mediante el modelo de clasificación (Ley Orgánica de Protección de Datos Personales, 2021, art. 10).

j) Seguridad de los datos: Se adoptarán medidas técnicas y organizativas adecuadas para reducir los riesgos asociados a la pérdida, alteración o acceso no autorizado de la información, especialmente en entornos digitales y durante el procesamiento computacional de los datos (Ley Orgánica de Protección de Datos Personales, 2021, art. 10).

k) Responsabilidad proactiva: Se documentarán de manera sistemática los procedimientos relacionados con el tratamiento de la información, con el propósito de garantizar trazabilidad, control y evidencia del cumplimiento de buenas prácticas en la gestión de los datos (Ley Orgánica de Protección de Datos Personales, 2021, art. 10).

l) Aplicación favorable al titular: Ante situaciones no contempladas expresamente durante el desarrollo del proyecto, se priorizarán decisiones orientadas a proteger los intereses y la información de la empresa colaboradora, adoptando un enfoque preventivo y responsable en el manejo de los datos (Ley Orgánica de Protección de Datos Personales, 2021, art. 10).

9 METODOLOGÍA

9.1 Introducción

En el presente capítulo se demuestra el segundo objetivo de presentar y desarrollar los métodos utilizados para la implementación de un modelo de clasificación orientado a la asignación del sector destino en las importaciones del mercado ecuatoriano**.** En este caso, primero se aplicó Árbol de decisión, con el fin de identificar patrones y relaciones entre las variables que caracterizan las operaciones de importación y el sector económico al que están destinadas. Para posteriormente proceder a aplicar el algoritmo Random Forest, con la finalidad de mejorar la capacidad predictiva del modelo como un conjunto de árboles de decisión que se combinan para conseguir disminuir el sobreajuste del modelo y mejorar la clasificación del sector destino. Se trató de un trabajo que se llevó a cabo mediante una investigación como enfoque cuantitativo, sobre la base de datos históricos de comercio exterior.

Cabe recalcar que no se utiliza ni población ni muestra sino la cantidad máxima de datos que se puedan recopilar porque los modelos de machine learning no se trabajan bajo un esquema poblacional, muestral o proporcional.

9.2 Consideraciones éticas y manejo de datos

Los datos fueron proporcionados por una empresa que autorizó su uso para fines académicos; sin embargo, solicitó no ser identificada con el fin de preservar cierta confidencialidad de los mismos. Por ello, se mantendrá el anonimato institucional y se aplicarán medidas de confidencialidad mediante codificación y resguardo de la base.

9.3 Python (Lenguaje de programación orientado a objetos)

Python es un lenguaje de programación ampliamente reconocido por su enfoque en la programación orientada a objetos, lo que permite estructurar el código mediante clases y

objetos que facilitan la organización, reutilización y mantenimiento de los programas. Esto posibilita la encapsulación, la herencia y el polimorfismo, características que contribuyen al desarrollo de aplicaciones más modulares, escalables y eficientes. Asimismo, Python mantiene una sintaxis clara y legible que simplifica la implementación de estructuras orientadas a objetos, reduciendo la complejidad del código y favoreciendo buenas prácticas de desarrollo. Gracias a su compatibilidad con múltiples sistemas operativos y su amplia disponibilidad de librerías especializadas, Python se posiciona como una herramienta idónea para el desarrollo de software estructurado, aplicaciones empresariales, aprendizaje automático y proyectos de análisis de datos que requieren una arquitectura organizada y mantenible.

9.4 Tipo de investigación

La presente investigación se clasifica como de tipo categórico debido a que la variable objetivo corresponde a una categoría nominal, específicamente el sector destino de las importaciones (Acuícola, Pecuario, Químico y Veterinario). Es así como, el estudio no busca predecir valores numéricos continuos, sino asignar cada observación a una clase determinada, lo que metodológicamente corresponde a un problema de clasificación multiclase dentro del aprendizaje automático.

9.5 Técnicas e instrumentos de recolección de datos

La recolección de datos de la presente investigación se pudo realizar a partir de información proveniente de la plataforma COBUS, así como de datos internos proporcionados por la empresa, los cuales permitieron acceder a registros históricos y operativos relacionados con las importaciones del mercado ecuatoriano. Estas fuentes proporcionaron información estructurada y confiable, necesaria para el análisis y la clasificación del sector destino de las importaciones.

Los datos extraídos comprendían registros históricos de importaciones, características del producto, valores declarados, países de origen y otras variables relacionadas con las operaciones de comercio exterior. El empleo de estos instrumentos de recolección de datos es el adecuado, ya que permite obtener información cuantitativa de una manera sistemática, simplificando el tratamiento de los datos y la puesta en práctica de modelos de clasificación influidos por técnicas de aprendizaje automático.

9.6 Métodos:

9.6.1 1. Data Clean

El Data Clean es el proceso mediante el cual se identifican, corrigen o eliminan inconsistencias, errores y valores incompletos presentes en los conjuntos de datos, constituyéndose como una etapa fundamental dentro del análisis de datos. Los datos por si solos contienen registros duplicados, valores faltantes, errores o incoherencias entre variables, los cuales pueden impactar de manera negativa la calidad de los resultados obtenidos.

La aplicación de técnicas de limpieza de datos permite mejorar la confiabilidad y precisión de la información utilizada, reduciendo sesgos y errores que podrían influir negativamente en los análisis estadísticos y en el desempeño de los modelos predictivos. Es así como, este proceso contribuye a la estandarización de formatos, la coherencia entre variables y la correcta representación de la información, facilitando el posterior procesamiento y análisis de los datos.

9.6.2 Descriptivos

Se realizará un análisis descriptivo para entender el comportamiento de las variables: medidas de tendencia central, dispersión, outliers, etc. Además, ayudará a detectar valores atípicos o inconsistencias que todavía podrían afectar al modelo.

Se usarán descriptivos como:

- Estadísticos (media, mediana, desviación, percentiles, min/max).
- Tablas de frecuencia para variables categóricas.
- Gráficos (histogramas, boxplots y conteos por categoría).

9.6.3 Preparación de variables

Como todo modelo necesita datos bien representados (limpios), se prepararán las variables. En la práctica esto incluye:

- Codificación de variables categóricas (por ejemplo: país de origen, proveedor, incoterm, partida, tipo de producto).
- Generar variables derivadas sólo si es posible que puedan aportar información a la variable a predecir (precio unitario, peso unitario, agrupaciones por familias de producto).
- Comprobar las variables que podrían hacer "fuga o traspaso de información" (cuando tenga la sensación o la certeza de que si una columna prácticamente lo que hace es revelar el sector).

9.6.4 División del dataset y validación

Para poder evaluar correctamente el rendimiento de los modelos propuestos, el dataset se dividirá en dos partes, siendo estas:

- Entrenamiento: para ajustar el modelo.
- Prueba: para evaluar qué tan bien generaliza.

9.7 Modelos por utilizar

9.7.1 Árbol de Decisión (Clasificación por perfiles)

Se utilizará un Árbol de Decisión como primer modelo de clasificación, por ser interpretable y útil para entender visualmente que variables dividen mejor los sectores. Sirviendo así, como modelo base y como comparación directa frente al modelo ensemble. Donde, se intentará controlar el sobreajuste.

9.7.2 Random Forest (Clasificación)

Se utilizará el modelo porque tiene la capacidad de incrementar la estabilidad y el rendimiento al combinar múltiples árboles. Se realizará el ajuste de hiperparámetros (por el número de árboles, profundidad, mínimo de muestras, número de variables que se tienen en cuenta en cada división) de esta manera se intentará encontrar el equilibrio entre el desempeño y la generalización.

9.7.3 Evaluación del modelo

Para comparar Árbol de Decisión y Random Forest se usarán métricas típicas de modelos de clasificación:

- Matriz de confusión (para ver sectores que más se confunden)
- Accuracy (rendimiento global)
- Precision, Recall y F1-score (especialmente útil si algunos sectores tienen pocos registros)

Además, se revisarán los errores más comunes (qué sectores se confunden entre sí) para justificar mejoras o limitaciones del modelo.

9.8 Herramienta de trabajo

El procesamiento de datos se realizará utilizando el lenguaje orientado a objetos Python, utilizando librerías estándar para manipulación de datos, visualización y entrenamiento de modelos (pandas, numpy, scikit-learn, matplotlib, etc.). El uso de estas herramientas permite replicabilidad del análisis y trazabilidad de resultados siendo útiles en la obtención de resultados y Machine Learning, como ya se explicó en puntos anteriores.

9.9 Librerías

Las librerías son códigos esenciales dentro del análisis de datos, ya que cada una de ellas nos proporciona diferentes funcionalidades, así como herramientas automatizadas que nos ayuda a poder analizar de una forma más practica los datos que necesitemos.

Las librerías utilizadas en nuestro proyecto son:

9.10 Pandas

Pandas es una librería de alto nivel para el análisis y manipulación de datos estructurados en Python, diseñada para trabajar con estructuras tabulares como Series y DataFrames. Su arquitectura permite realizar operaciones de limpieza, transformación, integración y análisis exploratorio de datos de manera eficiente. Internamente, Pandas está construida sobre NumPy, lo que le permite combinar la flexibilidad de estructuras etiquetadas con la eficiencia de los arreglos numéricos optimizados.

Desde una perspectiva metodológica, Pandas cumple un rol fundamental en la fase de Data Understanding y Data Preparation, permitiendo la lectura de archivos externos (como Excel), la identificación de tipos de datos, la detección de valores faltantes, la eliminación de

variables irrelevantes y la transformación de variables categóricas y numéricas. En investigaciones basadas en aprendizaje automático, esta librería constituye el núcleo del preprocesamiento de datos, etapa crítica para garantizar la calidad y coherencia del conjunto de entrenamiento.

9.11 NumPy

NumPy (Numerical Python) es una librería fundamental para el cálculo científico y numérico en Python. Su principal aporte es la implementación del objeto ndarray, una estructura de datos multidimensional optimizada para realizar operaciones matemáticas vectorizadas de manera eficiente. A diferencia de las listas tradicionales de Python, los arreglos de NumPy permiten ejecutar cálculos complejos con menor consumo de memoria y mayor velocidad computacional, gracias a su implementación en lenguaje C.

En el contexto de modelos de aprendizaje automático, NumPy proporciona la base matemática necesaria para operaciones matriciales, cálculos estadísticos, manipulación de índices y generación de estructuras numéricas utilizadas en métricas de evaluación, matrices de confusión y análisis probabilístico. Su uso es esencial en procesos que requieren eficiencia computacional, especialmente en algoritmos que operan sobre grandes volúmenes de datos o múltiples dimensiones.

9.12 Matplotlib

Matplotlib es una librería de visualización científica que permite la generación de gráficos bidimensionales de alta personalización. Su módulo pyplot ofrece una interfaz orientada a objetos para la creación de figuras, ejes y elementos gráficos, permitiendo un control detallado sobre la representación visual de los datos.

Desde el punto de vista metodológico, la visualización es una herramienta clave en la interpretación de resultados en modelos predictivos. En esta investigación, Matplotlib fue utilizada para representar la distribución de clases, matrices de confusión, curvas ROC y la importancia de variables del modelo Random Forest. Estas representaciones gráficas facilitan la comprensión del desempeño del modelo, la identificación de posibles desbalances y la evaluación visual de la capacidad discriminativa del clasificador. La visualización no solo cumple una función descriptiva, sino también analítica, ya que permite validar visualmente la calidad de los resultados obtenidos.

9.13 Seaborn

Seaborn es una librería de visualización estadística construida sobre Matplotlib, diseñada para simplificar la creación de gráficos estadísticos complejos con una sintaxis más intuitiva y una estética mejorada. Se integra directamente con estructuras de Pandas, lo que permite representar datos categóricos y numéricos de manera eficiente.

En el ámbito científico, Seaborn facilita el análisis exploratorio de datos mediante la visualización de distribuciones, correlaciones y patrones entre variables. Su utilización en la etapa exploratoria permite identificar tendencias preliminares, posibles outliers y relaciones relevantes antes de la construcción del modelo predictivo. Esta fase resulta esencial para comprender la estructura del conjunto de datos y fundamentar decisiones metodológicas posteriores.

9.14 Scikit-learn (sklearn)

Scikit-learn es una librería especializada en aprendizaje automático que implementa algoritmos supervisados y no supervisados bajo una arquitectura estandarizada y eficiente.

Proporciona herramientas para la división de datos, entrenamiento de modelos, validación, evaluación y selección de características. Su diseño modular permite aplicar algoritmos mediante una estructura uniforme basada en métodos como `fit()`, `predict()` y `predict_proba()`.

En esta investigación, Scikit-learn constituyó el componente central del modelado predictivo, permitiendo la implementación de algoritmos de clasificación como Árbol de Decisión y Random Forest. Asimismo, facilitó la evaluación del desempeño mediante métricas como exactitud, reporte de clasificación, matriz de confusión y curva ROC. Desde una perspectiva científica, Scikit-learn integra fundamentos estadísticos y computacionales que permiten construir modelos reproducibles, robustos y validados empíricamente, alineándose con estándares metodológicos en investigaciones de ciencia de datos y minería de datos.

9.15 Variables utilizadas

Ítem	Variable	Métrica	¿Qué significa?
1	DESCRIPCIÓN DEL DESPACHO	Categórica	Descripción de la importación
2	CAMARÓN	Binaria	Si aplica al segmento camarón
3	PORCINOS	Binaria	Si aplica al segmento porcinos
4	PERROS Y GATOS	Binaria	Si aplica al segmento mascotas (perros y gatos).
5	BOVINOS	Binaria	Si aplica al segmento bovinos.
6	POLLO	Binaria	Si aplica al segmento pollo
7	SECTOR DESTINO	Categórica	Sector asignado a la importación
8	ENTE REGULATORIO	Categórica	Entidad regulatoria aplicable
9	PAÍS DE ORIGEN	Categórica	Origen de la mercancía
10	VALOR FOB U$S	Numérica continua	Valor FOB
11	VALOR FLETE U$S	Numérica continua	Costo de flete
12	VALOR SEGURO U$S	Numérica continua	Costo de seguro
13	VALOR FACTURA U$S	Numérica continua	Valor total de la factura
14	VALOR KGS NETO	Numérica continua	Peso neto total (kg).
15	UNIDADES	Numérica discreta	Cantidad declarada
16	PRECIO UNITARIO U$S	Numérica continua	Valor unitario

Ítem	Variable	Métrica	¿Qué significa?
17	EMBARCADOR	Categórica	Empresa embarcadora/exportadora
18	TIPO DE CAPACIDAD	Categórica	Tipo de capacidad (aforo)

9.16 Resultados descriptivos

Figura 12

Gráfico de barras que muestra la frecuencia y distribución de los datos recolectados en los sectores destinos.

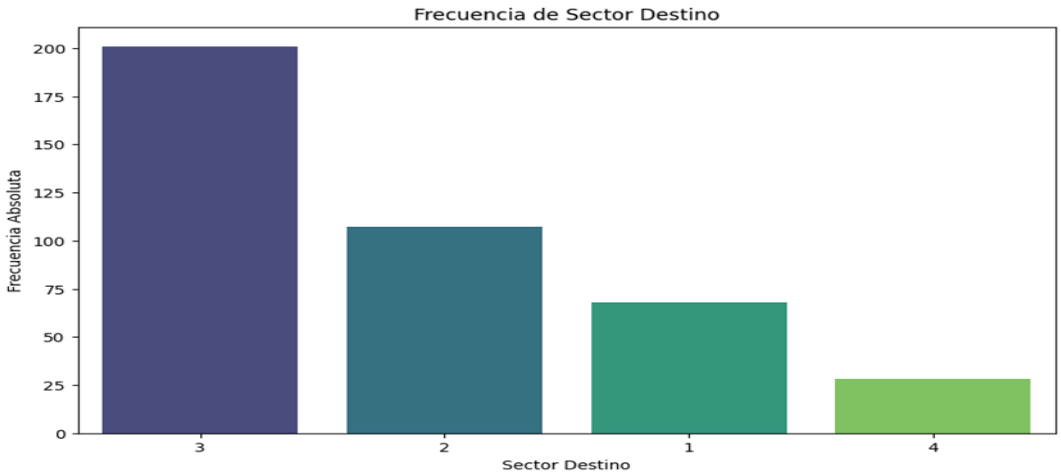
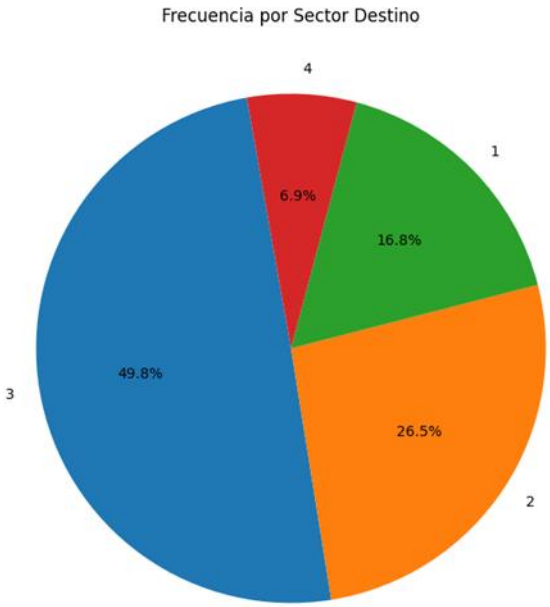


Figura 13

Diagrama de pastel de la frecuencia y distribución de los datos recolectados en los sectores destinos.



La Figura 12 presenta el cómo se distribuyen los datos recolectados a lo largo de las 4 variables de salida, que son los sectores destino. Los resultados evidencian una alta concentración en el sector 3, el cual corresponde al sector Químico, el cual representa el 49,8% del total de los registros analizados, posicionándose como el sector predominante.

En segundo lugar, tenemos el sector 2, el cual corresponde al sector Pecuario, el cual concentra el 26,5% de los datos, indicando un porcentaje relevante con respecto a los datos totales, aunque todavía considerablemente menor en comparación con el sector 3. El sector 1 representa el 16,8%, mostrando una cantidad apenas moderada en total. Y finalmente el sector 4 cuenta con la menor participación, con apenas el 6,9% del total, lo lo que podría sugerir una utilización limitada de este sector como destino de los despachos analizados.

Figura 14
Frecuencias de los sectores destino

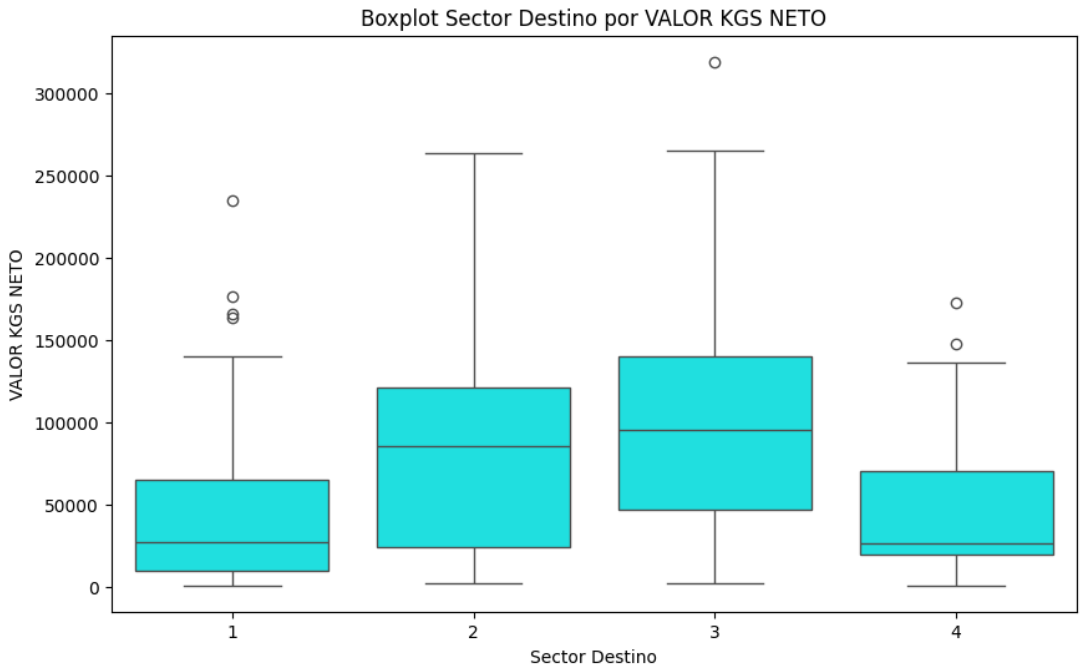
	Categoria	Frecuencia Absoluta	Frecuencia Relativa (%)	Frecuencia Acumulada	Frecuencia Relativa Acumulada (%)
0	3	201	49.75	201	49.75
1	2	107	26.49	308	76.24
2	1	68	16.83	376	93.07
3	4	28	6.93	404	100.00

En la presente tabla, mostramos la cantidad total de datos que se encuentran clasificados en cada uno de los sectores, dejando en evidencia que el sector más predominante es en efecto el sector 3, que corresponde al sector Químico.

9.16.1 Boxplot de Variables

Para comprender mejor el comportamiento de los despachos, se compararon el volumen exportado (kilogramos netos), el precio unitario y el valor FOB, considerando el sector destino: acuícola, pecuario, químico y veterinario, mediante Boxplots (diagramas de caja), los cuales permiten observar diferencias en los valores centrales, la dispersión y la presencia de datos extremos.

Figura 15
Boxplot Sector Destino por Valor KGS Neto



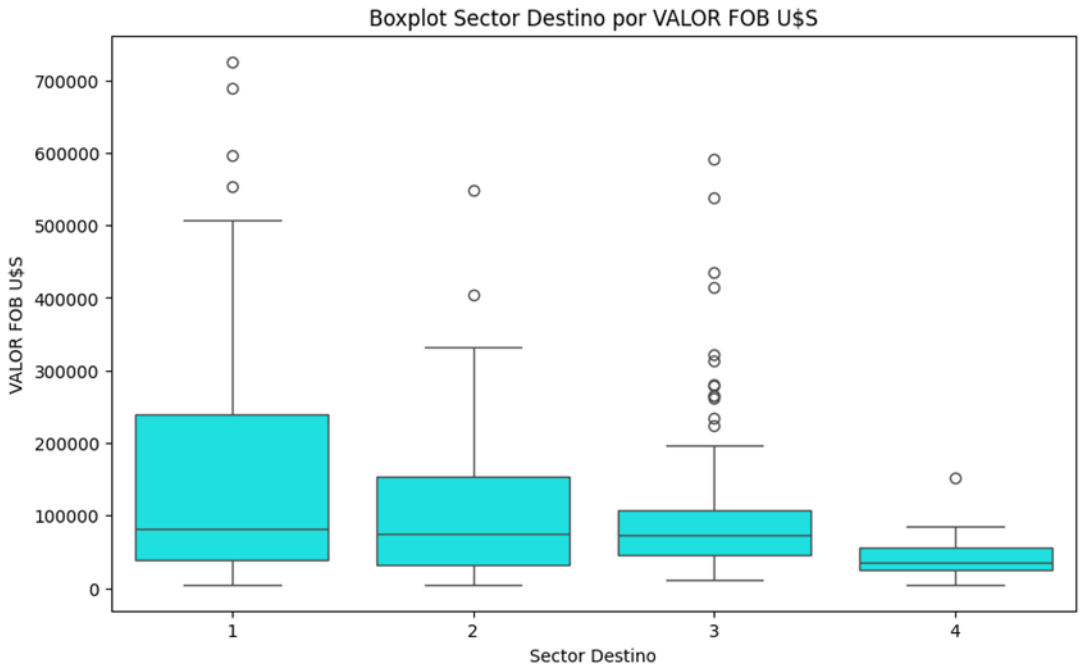
Este gráfico muestra diferencias claras entre los sectores. A nivel general, la base de datos presenta una media de 83.174 kg y una mediana cercana a 80.000 kg, lo que ya sugiere una distribución asimétrica, con despachos de gran volumen que influyen en el promedio.

También, se evidencia que el sector químico concentra los mayores volúmenes. Este sector presenta la mediana más alta, una amplia dispersión y varios valores extremos, lo que indica que, además de manejar grandes volúmenes de forma habitual, también registra despachos excepcionalmente grandes.

El sector pecuario muestra volúmenes intermedios, con una dispersión considerable, lo que refleja una estructura más pareja en relación con la cantidad de kilogramos despachados.

Por su parte, los sectores acuícola y veterinario se caracterizan por volúmenes más bajos. Sin embargo, en el sector acuícola se identifican algunos valores atípicos elevados, lo que confirma que, aunque no es lo más frecuente, existen despachos de gran volumen que elevan la media general observada en la base de datos.

Figura 16
Boxplot Sector Destino por Valor FOB



En relación con el valor FOB, el análisis muestra una media de USD 104.233 y una desviación estándar elevada, lo que evidencia una amplia variabilidad en los valores económicos de los despachos.

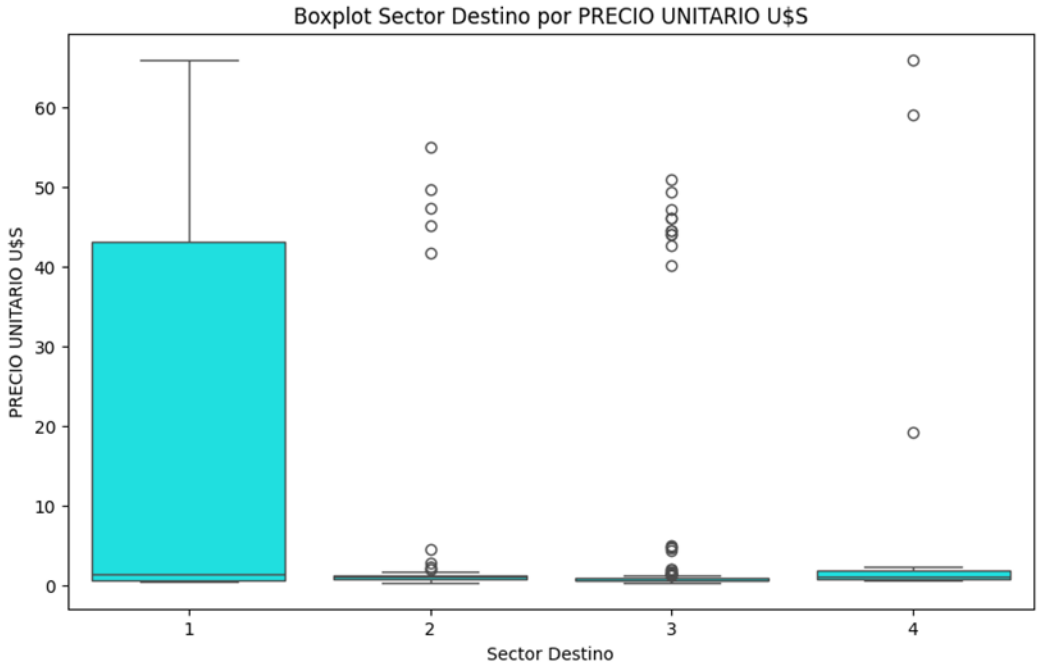
Los boxplots permiten identificar que el sector acuícola concentra los despachos de mayor valor económico. Este sector presenta una mediana superior y una dispersión amplia, lo que indica que combina operaciones de valor medio con despachos de alto valor FOB, representados por los valores atípicos observados.

Los sectores pecuario y químico presentan valores FOB intermedios. En particular, el sector químico muestra una mayor cantidad de valores extremos, lo que significa la presencia de cantidades de alto valor dentro de un conjunto más amplio de menor escala.

El sector veterinario presenta los valores FOB más bajos y una distribución más pareja, lo que sugiere que los despachos dirigidos a este sector tienden a ser de menor valor económico y con menor variabilidad.

Figura 17

Boxplot Sector Destino por Precio Unitario



En el boxplot del precio unitario podemos observar que, aunque la media del precio unitario es de USD 5,52, la mediana se sitúa en USD 0,83, evidencia de esa manera que cuenta con una fuerte distribución fuertemente asimétrica.

En todos los sectores se observa una alta concentración de precios bajos, acompañada de valores extremos elevados. El sector acuícola presenta la mayor dispersión y los valores máximos más altos, sugiriendo de esa manera una coexistencia de productos de bajo precio con otros de mayor valor agregado.

Los sectores pecuario y químico muestran precios unitarios generalmente bajos y bastante concentrados, a pesar de que cuenta con varios valores atípicos, lo que podría indicar que ciertos despachos específicos se realizan a precios significativamente superiores al resto.

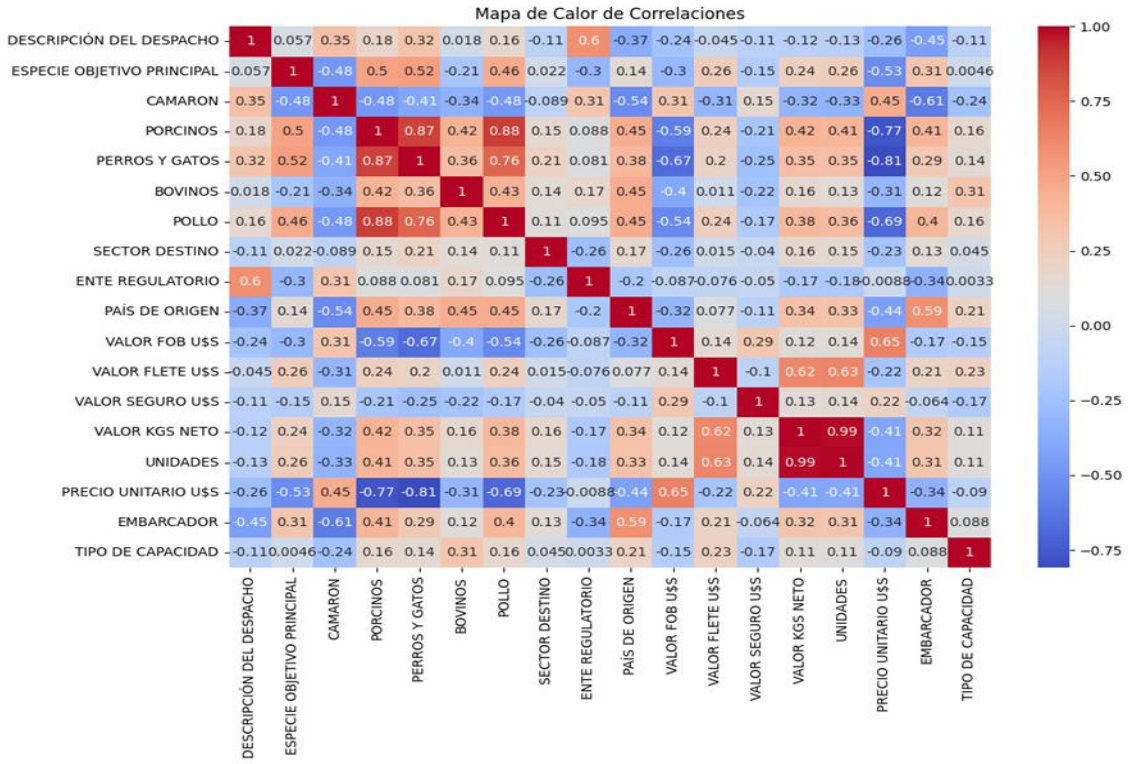
En sector veterinario podemos observar un comportamiento similar, aunque con una ligera tendencia a precios unitarios más altos en comparación con los sectores pecuario y químico, esto se evidencia en la posición de la mediana dentro del boxplot.

9.16.2 Mapa de calor de variables

Para saber las relaciones que existen entre las variables del estudio, se elaboró un mapa de calor para las correlaciones, el cual permite identificar asociaciones positivas y negativas entre las variables presentadas.

Cabe recalcar que este análisis es de carácter exploratorio, por lo que las correlaciones identificadas no implican causalidad, sino posibles patrones de asociación.

Figura 18
Mapa de Calor de Correlaciones



Una de las relaciones más claras es entre el valor de los kilogramos netos y el número de unidades, con una correlación muy alta y positiva ($r \approx 0{,}99$). Esta relación era predecible, ya que, por lógica a mayor volumen físico despachado, mayor es el número de unidades movilizadas.

También, el valor del flete presenta una correlación positiva moderada con los kilogramos netos y las unidades ($r \approx 0{,}62–0{,}63$), lo que podría indicar que los costos de transporte tienden a incrementarse conforme aumenta el volumen del despacho.

En cambio, el precio unitario muestra una correlación negativa moderada con los kilogramos netos y las unidades ($r \approx -0{,}41$), sugiriendo que los despachos de mayor volumen suelen estar asociados a precios unitarios más bajos.

Las variables que se asocian a (porcinos, perros y gatos, pollo y bovinos) muestran correlaciones positivas entre sí, reflejando coexistencia de varias especies dentro de un mismo despacho. En particular, se observa una fuerte relación entre porcinos y perros y gatos, así como entre porcinos y pollo, sugiriendo que estos productos suelen comercializarse conjuntamente.

El precio unitario presenta correlaciones negativas marcadas con porcinos ($r \approx -0{,}77$), perros y gatos ($r \approx -0{,}81$) y pollo ($r \approx -0{,}69$), indicando que los despachos asociados a estas especies tienden a manejar precios unitarios más bajos en comparación con otros productos. Sin embargo, la variable camarón muestra una correlación positiva con el precio unitario ($r \approx 0{,}45$), sugiriendo que los productos vinculados a esta especie presentan mayores precios por unidad.

El sector destino presenta correlaciones bajas con la mayoría de las variables económicas, indicando que no existe una relación lineal fuerte entre el sector y los valores económicos o de volumen.

El ente regulatorio y el país de origen muestran algunas correlaciones moderadas con variables económicas y operativas, sugiriendo que estos factores podrían influir en valor del despacho, aunque no de una forma fuerte.

Finalmente, el tipo de capacidad presenta correlaciones débiles con la mayoría de las variables, dando a entender que, si bien es un factor operativo relevante, no determina directamente el valor económico o el volumen del despacho.

9.17 Metodología aplicativa para la realización del Árbol de Decisión

El primer modelo aplicado fue el **Árbol de Decisión**, el cual pertenece a la familia de los modelos supervisados y se basa en una partición dicotómica del espacio de características.

Su objetivo principal es construir una estructura jerárquica de decisiones que permita asignar una observación a una clase específica a partir de reglas lógicas derivadas de los datos.

Sea el conjunto de entrenamiento:

$$D = \{(x_i, y_i)\}_{i=1}^{n}$$

donde:

- $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$representa el vector de variables explicativas,
- $y_i \in \{1,2, \dots, K\}$es la variable dependiente categórica,
- $n$es el número total de observaciones,
- Kes el número de clases.
- : conjunto total de datos utilizados para entrenar el modelo.
- x_i: vector de características (variables independientes) de la observación i.
- y_i: clase real asociada a la observación i.
- i: índice de la observación.
- n: número total de observaciones del conjunto de datos.

Esta expresión indica que cada observación está formada por variables explicativas y una clase conocida, logrando así el cometido principal del árbol de decisión el cual consiste en

determinar qué variable y qué punto de corte permiten separar mejor las observaciones, reduciendo la incertidumbre sobre la clase a la que pertenecen.

9.18 Impureza

Desde un punto de vista probabilístico, un nodo es considerado puro cuando la probabilidad de que una observación pertenezca a una sola clase es cercana a uno. Si un nodo contiene observaciones de varias clases, existe incertidumbre en si la asignación de la clase es la correcta.

Si tenemos un nodo t que contiene n_t observaciones. La probabilidad empírica de pertenecer a la clase k se define como:

$$p_k = \frac{n_k}{n_t}$$

p_k: probabilidad de que una observación pertenezca a la clase $k$dentro de un nodo.

n_k: número de observaciones de la clase $k$en el nodo.

n_t: número total de observaciones en el nodo t.

9.19 Medida de impureza

En los árboles de decisión, cada nodo representa un subconjunto de observaciones del conjunto de datos original. Para determinar la calidad de un nodo y decidir si es conveniente realizar una partición adicional, es necesario cuantificar el grado de heterogeneidad existente entre las clases que lo componen. Esta heterogeneidad se mide mediante una función de impureza, la cual refleja el nivel de incertidumbre asociado a la clasificación de una observación dentro del nodo.

Según esto, tenemos el Índice de Gini como medida de impureza, debido a su eficiencia computacional y a su amplia utilización en problemas de clasificación. Conceptualmente, el Índice de Gini se interpreta como la probabilidad de clasificar incorrectamente una observación seleccionada al azar, asumiendo que dicha observación es etiquetada de acuerdo con la distribución de clases presente en el nodo.

La probabilidad de cometer un error de clasificación en un nodo se expresa como:

$$P(\text{error}) = 1 - \sum_{k=1}^{K} p_k^2$$

A partir de esta expresión se obtiene directamente la formulación del Índice de Gini para un nodo t:

$$Gini(t) = 1 - \sum_{k=1}^{K} p_k^2$$

donde:

- $Gini(t)$: nivel de impureza del nodo t.
- K: número total de clases presentes en el problema de clasificación.
- p_k: probabilidad de que una observación pertenezca a la clase $k$dentro del nodo, calculada como la proporción de observaciones de dicha clase respecto al total del nodo.
- $\sum_{k=1}^{K} p_k^2$: probabilidad de clasificar correctamente una observación seleccionada al azar, dado el reparto de clases en el nodo.

El Índice de Gini toma valores en el intervalo $[0,1)$. Un valor cercano a cero indica que el nodo es altamente puro, es decir, que la mayoría de las observaciones pertenecen a una sola clase. Por el contrario, valores más elevados reflejan una mayor mezcla de clases y, por tanto, un mayor nivel de incertidumbre. Durante el proceso de construcción del árbol, se seleccionan las divisiones que minimizan el Índice de Gini en los nodos hijos, con el objetivo de lograr particiones cada vez más homogéneas.

Otra medida de impureza ampliamente utilizada en los árboles de decisión es la entropía. La entropía cuantifica el grado de desorden o incertidumbre presente en un nodo, interpretándose como la cantidad promedio de información necesaria para identificar correctamente la clase de una observación.

La entropía de un nodo t se define matemáticamente como:

$$H(t) = -\sum_{k=1}^{K} p_k \log_2(p_k)$$

donde:

- $H(t)$: entropía del nodo t.
- K: número de clases.
- p_k: probabilidad de pertenecer a la clase $k$dentro del nodo.
- $\log_2(p_k)$: logaritmo en base 2, que expresa la cantidad de información asociada a la ocurrencia de la clase k.

La entropía toma el valor cero cuando el nodo es completamente puro, es decir, cuando todas las observaciones pertenecen a una sola clase. En cambio, alcanza su valor máximo cuando las clases se encuentran distribuidas de manera uniforme, lo que implica el máximo nivel de incertidumbre.

En los árboles de decisión, la entropía se utiliza junto con el concepto de ganancia de información, la cual mide la reducción de incertidumbre lograda al realizar una partición. El algoritmo selecciona la división que maximiza dicha ganancia, favoreciendo estructuras de árbol que separen de forma más efectiva las clases.

9.20 Criterio de división

Una vez definida la impureza de un nodo, se evalúa todas las posibles divisiones candidatas $(X_j \, , s)$, donde:

- X_j es una variable explicativa
- s es un valor de corte.

La división genera dos nodos hijos: t_{izq} y t_{der}.

La impureza total después de la división se calcula como una combinación ponderada:

$$Gini_{post} = \frac{n_{izq}}{n_t} Gini(t_{izq}) + \frac{n_{der}}{n_t} Gini(t_{der})$$

La mejora obtenida por la división se define como la **reducción de impureza**:

$$\Delta Gini = Gini(t) - Gini_{post}$$

El Árbol de Decisión selecciona la variable y el punto de corte que **maximizan** $\Delta Gini$, ya que esta división produce el mayor aumento en la homogeneidad de las clases.

Todo este proceso se convierte en la construcción del árbol en un proceso el cual se repite cada nodo.

9.21 Regla de división

El método de división es la forma en la cual un árbol de decisión decide de qué forma dividir las observaciones que hay en un nodo en los nodos hijos de aquel. La razón principal para hacer esto es reducir la impureza del nodo padre, y conseguir que los nodos hijos sean más homogéneos con respecto a la variable objetivo.

En cada nodo, un conjunto de posibles cortes es evaluado a partir de las variables explicativas que hay. Para cada una de las variables se considerarán diferentes puntos de corte, criterios de separación, en función de si la variable independiente es cuantitativa o cualitativa. Para cada uno de los cortes posibles se derivan un conjunto de nodos hijos y la calidad de ellos se mide a base de una medida de impureza como el Índice de Gini o la entropía antes mencionados.

Si tenemos un nodo t que contiene N_t observaciones. Una división $s$sobre dicho nodo lo separa en J nodos hijos $t_1, t_2, \dots, t_J$. La calidad de esta división se evalúa comparando la impureza del nodo padre con la impureza ponderada de los nodos hijos.

Se expresa como:

$$\Delta I(s,t) = I(t) - \sum_{j=1}^{J} \frac{N_{t_j}}{N_t} I(t_j)$$

donde:

- $\Delta I(s,t)$: reducción de impureza producida por la división $s$en el nodo t.
- $I(t)$: impureza del nodo padre (Gini o entropía).

- $I(t_j)$: impureza del nodo hijo t_j.
- N_t: número total de observaciones en el nodo padre.
- N_{t_j}: número de observaciones en el nodo hijo t_j.
- $\frac{N_{t_j}}{N_t}$: peso relativo del nodo hijo en función de su tamaño.

Se escoge como lo más optimo a la división que maximiza la reducción de impureza o que logra la mayor ganancia informativa.

9.22 Ganancia de información

Es uno de los criterios que se utilizan en los árboles de decisión cuando se mide la impureza de un nodo mediante las medidas de impureza. Esta consiste en cuantificar cuánto se reduce la incertidumbre sobre la variable objetivo gracias a una regla de división determinada. Por lo que podemos decir que, es un criterio que nos permite seleccionar la división que mejor separa las observaciones en función de sus clases.

La ganancia de información asociada a una división s en el nodo t se define como:

$$IG(s,t) = H(t) - \sum_{j=1}^{J} \frac{N_{t_j}}{N_t} H(t_j)$$

$IG(s,t)$: Ganancia de información generada por la división s en el nodo t.

$H(t)$: Entropía del nodo padre antes de realizar la división.

$\sum_{j=1}^{J}$: Suma de las entropías de todos los nodos hijos generados por la división.

$\frac{N_{t_j}}{N_t}$: Este es el peso del nodo hijo t_j, que indica el porcentaje de observaciones.

$H(t_j)$: Entropía del nodo hijo t_j, calculada respecto a la nueva distribución de clases después de la división.

Este proceso se repite en los nodos hijos hasta que se cumpla cualquiera de los criterios de parada.

9.23 Criterios de parada

Una vez seleccionada la mejor regla de división en cada nodo mediante la ganancia de información, el algoritmo continúa dividiendo los nodos de manera recursiva. Sin embargo, este proceso no puede extenderse indefinidamente, ya que un árbol excesivamente profundo puede ajustarse en exceso a los datos de entrenamiento y causar sobreajuste.

Los **criterios de parada** establecen las condiciones bajo las cuales el algoritmo deja de realizar nuevas particiones y declara un nodo como nodo terminal u hoja.

9.24 Principales criterios de parada

1. **Pureza del nodo**

   El proceso de división se detiene cuando todas las observaciones del nodo pertenecen a una misma clase. Lo que indica que no existe incertidumbre en la clasificación.

2. **Número mínimo de observaciones por nodo**

   Se implementa un umbral mínimo de observaciones para permitir una nueva partición. Si un nodo tiene menos observaciones que el umbral, entonces no se parte.

3. **Profundidad máxima del árbol**

   Se define una profundidad máxima del árbol. Esto sirve para limitar la que tan complejo puede ser el modelo y evitar que el árbol tenga una mucha longitud.

4. **Ganancia de información mínima**

Si la mejor división de un nodo produce una ganancia de información inferior a un umbral mínimo el cual ya se ha establecido anteriormente, entonces no se considera como información útil, y el nodo se convierte en una hoja.

9.25 Metodología aplicativa para la realización del Random Forest

El Random Forest sirve como un método con ensambles, ya que combina varios árboles de decisión con el objetivo de aumentar el poder de predicción del modelo y de disminuir el sobreajuste. De esta forma, a diferencia de un solo árbol de decisión, que puede ser sensible a cambios en los datos, el Random Forest construye una colección de árboles independientes, que combina el resultado a través de un mecanismo de votación.

Random Forest está fundamentado por dos ideas básicas:

1. **Muestra Bootstrap**: esto genera varias muestras aleatorias que vienen directo del dataset original.
2. **Aleatoriedad en la selección de variables**: no se evalúan todas las variables, sino solo un subconjunto.

Al combinar estos dos puntos, nos ayuda a reducir la correlación entre los árboles y mejora la generalización del modelo.

Si tenemos $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ el conjunto de datos original con N observaciones.

Una muestra bootstrap D_b se define como:

$$D_b = \{(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \dots, (x_{i_N}, y_{i_N})\}$$

donde cada índice i_j es seleccionado de forma aleatoria e independiente del conjunto $\{1,2,\dots,N\}$.

9.26 Propiedades

1. **Repetición de observaciones**

   Esto se refiere a que una sola observación puede aparecer varias veces dentro de una misma muestra bootstrap.

2. **Observaciones no seleccionadas (Out-of-Bag)**

   La probabilidad de que una observación no sea seleccionada en una muestra bootstrap es:

$$\left(1-\frac{1}{N}\right)^N \approx e^{-1} \approx 0.368$$

Esto implica que aproximadamente el **36.8%** de las observaciones quedan fuera de cada muestra y pueden utilizarse para validación.

Mientras la muestra sea más grande, se acercará cada vez más al valor de 0.368

3. **Variabilidad entre muestras**

   Cada muestra bootstrap es diferente, lo que introduce diversidad entre los modelos entrenados sobre ellas.

9.27 Bagging

Es una técnica de combinación de modelos que utiliza el ajuste de varios modelos generados a partir de diferentes muestras bootstrap con el objetivo de mejorar el rendimiento general de un modelo, es decir, la idea fundamental es que, al hacer el promedio o la suma de varios modelos

inestables, el estimador que se obtiene es más robusto y menos sensible a las variaciones de los datos.

Este algoritmo sigue estos pasos:

1. Generar B muestras bootstrap independientes derivados del conjunto de datos original.
2. Entrenar un modelo base (en este caso, arboles de decisiones) sobre cada muestra Bootstrap hecha.
3. Combinar las predicciones de todos los modelos mediante un mecanismo de agregación y decidiendo mediante una votación mayoritaria.

Regla de votación mayoritaria

$$\hat{y} = \arg\max_{k} \sum_{b=1}^{B} \mathbb{I}(\, h_b(x) = k)$$

esto significa lo siguiente:

- $\hat{y}$: clase predicha final.
- B: número total de árboles del bosque.
- $h_b(x)$: clase predicha por el árbol b.
- $\mathbb{I}(\cdot)$: función indicadora:

vale 1 si la condición se cumple,

vale 0 si no se cumple.

- $\arg\max$: selecciona la clase con mayor número de votos.

10 Resultados

La implementación del modelo de clasificación para la asignación del **sector destino en importaciones del mercado ecuatoriano** se desarrolló utilizando el lenguaje de programación Python, en el entorno **Google Colab**, lo que permitió ejecutar el análisis de datos y los algoritmos de Machine Learning de forma eficiente y reproducible. La metodología aplicada sigue una secuencia estructurada que abarca desde la carga de datos hasta la evaluación de los modelos de clasificación.

10.1 Carga de datos

Para el desarrollo del modelo de clasificación del sector destino, la base de datos fue cargada desde **Google Drive**, utilizando el entorno de **Google Colab**, lo que permitió acceder de forma directa al archivo que contiene los registros históricos de importaciones.

En primera instancia, se realizó el montaje de Google Drive con el fin de habilitar el acceso a los archivos almacenados:

```
from google.colab import drive
drive.mount('/content/drive')
```

Una vez establecida la conexión, se procedió a la carga del archivo de datos, el cual contiene 304 observaciones correspondientes a importaciones del mercado ecuatoriano, obtenidas desde la plataforma COBUS y datos internos de la empresa:

```
df = pd.read_excel("/content/drive/MyDrive/BI A-
2025/asignacion del sector destino en importaciones del
Ecuador.xlsx")
```

Este procedimiento permitió importar la información en un **DataFrame**, donde cada fila representa una operación de importación y cada columna corresponde a una variable relevante para la clasificación del sector destino.

Posteriormente, se verificó la correcta carga de los datos mediante la visualización directa del conjunto de datos utilizando el comando `df`. Este paso permitió comprobar que la información fue importada correctamente desde el archivo original y que los registros se encontraban disponibles para su análisis.

10.2 Preparación y limpieza de los datos (Data Clean & Data Preparation)

La preparación y limpieza de los datos constituye una etapa fundamental en el desarrollo del modelo de clasificación, ya que permite garantizar la calidad, coherencia y representatividad del conjunto de datos utilizado para el entrenamiento de los algoritmos, considerando que los datos reales suelen presentar inconsistencias, ruido y valores incompletos (Han, Kamber & Pei, 2012). En esta investigación, el proceso de limpieza se realizó de acuerdo con las características específicas de la base de datos y las necesidades del modelo.

10.3 Balanceo de la variable objetivo

Inicialmente, se identificó un **desbalance en la variable objetivo "SECTOR DESTINO"**, particularmente en la categoría correspondiente al sector químico. Con el objetivo de mejorar el equilibrio entre clases y evitar sesgos en el entrenamiento del modelo, se procedió a eliminar de forma aleatoria el **50% de los registros pertenecientes a dicha categoría**.

```
quimico_indices = df[df['SECTOR DESTINO'] == 3].index
    num_to_drop = int(len(quimico_indices) * 0.5)
```

```
indices_to_drop = np.random.choice(quimico_indices,
            num_to_drop, replace=False)
df = df.drop(indices_to_drop)
```

Este procedimiento permitió obtener una distribución más equilibrada de las clases, contribuyendo a mejorar el desempeño y la estabilidad de los modelos de clasificación.

10.4 Eliminación de variables no relevantes

Posteriormente, se procedió a la eliminación de variables que no aportaban información relevante para la clasificación del sector destino o que podían generar ruido en el modelo.

La selección adecuada de variables contribuye a mejorar el desempeño y la precisión del modelo de clasificación, ya que disminuye la complejidad del espacio de búsqueda y favorece una mejor capacidad de generalización sobre datos no vistos. Esto permite obtener resultados más estables y confiables durante el proceso de entrenamiento y validación del modelo (Hastie, Tibshirani & Friedman, 2009).

En este caso, se eliminó la variable **"ESPECIE OBJETIVO PRINCIPAL"**, al no ser considerada determinante para el objetivo del estudio.

```
df = df.drop(columns=['ESPECIE OBJETIVO PRINCIPAL'])
```

Una vez realizada la depuración inicial de la base de datos mediante la eliminación de variables no relevantes, se utilizó el comando `df.info()` con el fin de obtener un resumen estructural del conjunto de datos resultante. Este procedimiento permitió verificar el número total de registros y columnas, así como identificar el tipo de dato asociado a cada variable y la presencia de valores nulos.

Además, se aplicó el comando `df = df.round(2)` con el objetivo de redondear a dos decimales las variables de tipo numérico presentes en el conjunto de datos. Esto permitió estandarizar la precisión de los valores, facilitando su lectura, interpretación y presentación, sin alterar de manera significativa la información contenida en las variables.

10.4.1 Definición de variables predictoras y variable objetivo

Una vez depurada la base de datos, se realizó la separación de las variables independientes (predictores) y la variable dependiente. La variable **"SECTOR DESTINO"** fue definida como la variable objetivo del modelo, mientras que el resto de las variables conformaron el conjunto de predictores.

```
X = df.drop(columns=['SECTOR DESTINO'])
y = df['SECTOR DESTINO']
```

Este proceso permitió estructurar correctamente los datos para su posterior uso en el entrenamiento y evaluación de los algoritmos de clasificación.

10.5 Mapeo de la variable objetivo (Sector Destino)

El mapeo de la variable objetivo se utilizó con el propósito de **transformar los valores numéricos de la variable "sector destino" en etiquetas categóricas legibles**, facilitando la interpretación de los resultados obtenidos durante el análisis exploratorio y la visualización de datos.

Código:

```
sector_map = {1: 'ACUICOLA', 2: 'PECUARIO', 3: 'QUIMICO', 4:
                        'VETERINARIO'}
y_labels = y.map(sector_map)
```

En el conjunto de datos, los sectores destino se encuentran codificados numéricamente. Si bien esta codificación es adecuada para el entrenamiento de los modelos de Machine Learning, no resulta intuitiva para el análisis descriptivo ni para la presentación gráfica de los resultados. Por esta razón, se definió un diccionario de correspondencia (`sector_map`) que asocia cada código numérico con el nombre del sector económico correspondiente.

La aplicación de la función:

```
y_labels = y.map(sector_map)
```

permite generar una nueva variable con etiquetas textuales, la cual se emplea exclusivamente para visualización, análisis descriptivo y comunicación de resultados, sin alterar la variable original utilizada por los algoritmos de clasificación.

10.6 División del conjunto de datos

La división del conjunto de datos se realizó con el objetivo de separar la información en datos de entrenamiento y datos de prueba, permitiendo evaluar de forma objetiva el desempeño de los modelos de clasificación.

Código:

```
X_train, X_test, y_train, y_test = train_test_split(
  X, y, test_size=0.25, random_state=42, stratify=y
)
```

El parámetro:

```
test_size=0.25
```

indica que el 25 % de las observaciones se destina al conjunto de prueba, mientras que el 75 % restante se utiliza para el entrenamiento del modelo.

Esta proporción es ampliamente utilizada en estudios de aprendizaje automático, ya que permite un balance adecuado entre el aprendizaje del algoritmo y la evaluación objetiva de su desempeño sobre datos no vistos (James, G., Witten, D., Hastie, T., & Tibshirani, R.2013).

Esta proporción permite que el modelo disponga de una cantidad suficiente de datos para aprender los patrones subyacentes, al mismo tiempo que se reserva un subconjunto representativo para evaluar su capacidad de generalización.

El uso del parámetro:

```
stratify=y
```

permite que la distribución de la variable objetivo (sector destino) se mantenga proporcional tanto en el conjunto de entrenamiento como en el de prueba.

Kuhn y Johnson (2013) señalan que la estratificación en la partición de los datos permite reducir el sesgo de muestreo y obtener métricas de desempeño más confiables, especialmente cuando se trabaja con variables objetivo-categóricas.

Según Hastie, Tibshirani y Friedman (2009), en los problemas de clasificación es esencial que la proporción de cada clase se conserve al momento de dividir el conjunto de datos, ya que una distribución desbalanceada entre los subconjuntos de entrenamiento y prueba puede distorsionar la evaluación del desempeño del modelo y generar resultados poco representativos de su capacidad predictiva real.

El parámetro:

```
random_state=42
```

se emplea para garantizar la **reproducibilidad de los resultados**, asegurando que la partición de los datos sea la misma cada vez que se ejecute el código, lo cual es fundamental en investigaciones académicas.

James et al. (2013) explican que fijar un valor para la semilla aleatoria durante la partición de los datos asegura que la división entre los conjuntos de entrenamiento y prueba sea consistente en cada ejecución del experimento, permitiendo comparar resultados de manera objetiva y confiable a lo largo del proceso de validación del modelo.

La variable `X` corresponde al conjunto de **variables predictoras**, mientras que `Y` representa la **variable objetivo (sector destino)**. Esta separación es un paso crítico previo a la implementación del Árbol de Decisión y Random Forest, ya que permite medir métricas como exactitud, matriz de confusión y curva ROC sobre datos no utilizados durante el entrenamiento.

10.7 Distribución de clases

El análisis de la **distribución de la variable objetivo (sector destino)** permitió identificar cómo se encuentran representadas las distintas clases dentro del conjunto de datos, aspecto fundamental previo a la aplicación de modelos de clasificación.

Código:

```
plt.figure(figsize=(8, 5))
unique, counts = np.unique(y_labels, return_counts=True)
```

```
plt.bar(unique, counts,
color=['#a6cee3','#1f78b4','#b2df8a','#33a02c'])
plt.title('Distribución de la variable objetivo: SECTOR
DESTINO')
for i, v in enumerate(counts):
plt
plt.text(i,
v + 1,
str(round(v / len(y_labels) * 100,
2)) + '%',
ha='center',
fontsize=10)
plt.xlabel('Sector Destino')
plt.ylabel('Frecuencia')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

Para saber la frecuencia de cada clase dentro de la variable destinada, se usa este código:

```
unique, counts = np.unique(y_labels, return_counts=True)
```

Esto es importante, ya que con esto se puede obtener la cantidad de observaciones que se asocian a cada sector de destino, permitiendo la detección de posibles desbalances de clases, que afectan el desempeño del algoritmo.

La representación gráfica de la distribución se obtiene a través del gráfico de barras:

```
plt.bar(unique, counts,
color=['#a6cee3','#1f78b4','#b2df8a','#33a02c'])
```

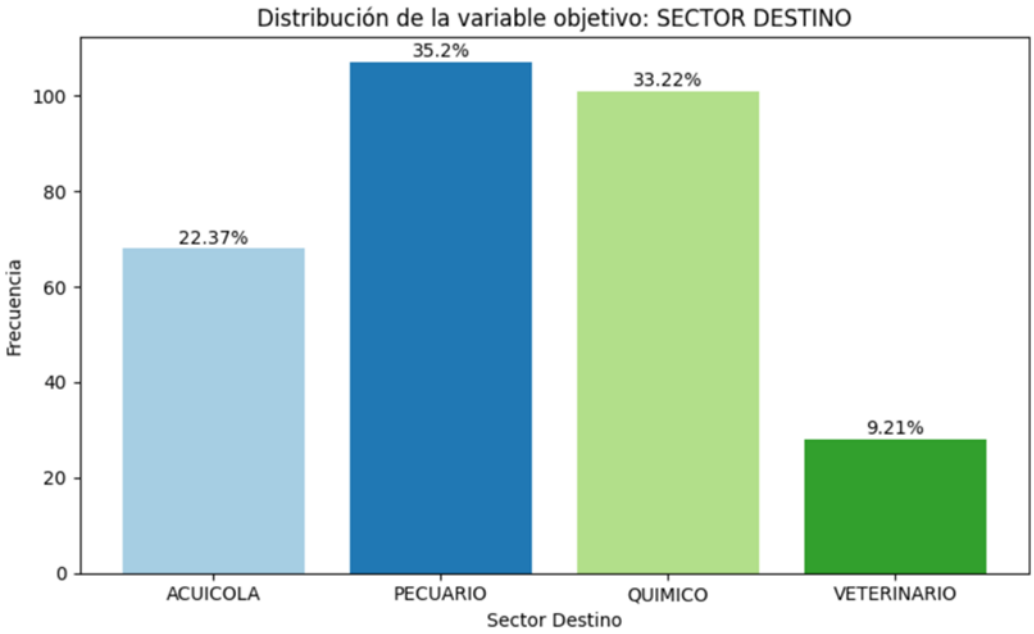
Este código presenta el porcentaje observado por cada sector, ayudándonos a comparar las clases e interpretar el peso que existe en las clases del conjunto de datos.

Finalmente, la inclusión de los porcentajes sobre cada barra, calculados mediante:

```
v / len(y_labels) * 100
```

proporciona una visión más precisa de la **participación porcentual de cada sector destino**, lo que refuerza el análisis descriptivo de la variable objetivo.

Figura 19
Distribución de la variable objetivo.



10.8 Implementación del modelo Árbol de Decisión

Una vez hecha la división del conjunto de datos en entrenamiento y testeo, se procedió a realizar el modelo de árbol de decisión, el cual permitió construir reglas de clasificación de la combinación de las variables para comprender los criterios que derivan hasta llegar a la asignación del sector destino.

10.9 Construcción y entrenamiento del modelo

Para la implementación del Árbol de Decisión se utilizó el clasificador `DecisionTreeClassifier`, configurando parámetros específicos con el fin de controlar la complejidad del modelo y mejorar su capacidad de generalización. En particular, se definió una profundidad máxima del árbol, un número mínimo de observaciones por hoja y un balanceo de clases automático.

Código:

```
dt = DecisionTreeClassifier(
        max_depth=4,
    min_samples_leaf=15,
      random_state=42,
   class_weight='balanced'
              )
  dt.fit(X_train, y_train)
```

El uso de estos parámetros permite evitar el sobreajuste del modelo, garantizando que las reglas de clasificación obtenidas sean representativas del comportamiento general de los datos.

10.10 Visualización del Árbol de Decisión

Posteriormente, se realizó la visualización del Árbol de Decisión entrenado, con el propósito de analizar gráficamente las reglas generadas y los perfiles que conducen a cada sector destino.

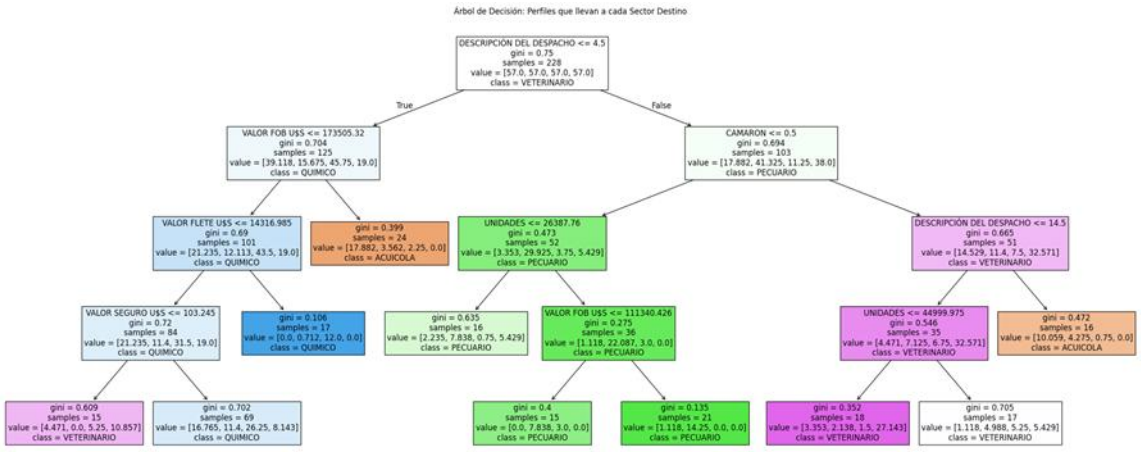
Código:

```
plt.figure(figsize=(25, 10))
        plot_tree(
            dt,
```

```
                feature_names=X.columns,
        class_names=['ACUICOLA', 'PECUARIO', 'QUIMICO',
                     'VETERINARIO'],
                filled=True,
                fontsize=12
                )
plt.title("Árbol de Decisión: Perfiles que llevan a cada
                     Sector Destino")
            plt.tight_layout()
            plt.show()
```

Esta representación gráfica permite identificar las variables más relevantes en cada nivel del árbol y comprender de manera intuitiva el proceso de decisión del modelo.

Figura 20
Árbol de decisión resultante.



Análisis:

Todo inicia con la descripción del despacho, que funciona como la primera gran decisión del árbol. Dependiendo de si esta condición se cumple o no, el modelo avanza hacia la izquierda o hacia la derecha, separando desde el inicio dos grupos con características distintas. Esto muestra que el árbol primero intenta entender qué tipo de producto se está comercializando, marcando la dirección inicial del análisis. En pocas palabras, antes de

considerar cantidades, el modelo establece una clasificación preliminar basada en la naturaleza del despacho, reflejando una lógica muy cercana a la forma en que operan los sectores en la práctica.

Una vez que el árbol se dirige hacia la izquierda, comienza un proceso donde intervienen variables económicas como el valor FOB, el valor del flete y el seguro. En cada nodo se realiza una nueva pregunta, y dependiendo de la respuesta avanza hacia la izquierda o gira hacia la derecha, reduciendo poco a poco la mezcla entre sectores. Este recorrido muestra cómo el árbol va afinando la clasificación paso a paso, agrupando operaciones que tienen costos similares. A medida que se desciende, los nodos finales presentan menor impureza, indicando que el modelo logra identificar perfiles más iguales, principalmente asociados al sector químico, evidenciando que este sector mantiene patrones comerciales más definidos dentro del conjunto de datos.

Por el contrario, cuando desde el nodo inicial el árbol avanza hacia la derecha, la lógica de decisión cambia y el peso recae principalmente en la variable unidades. En esta parte del árbol, cada nodo utiliza el volumen comercializado como criterio para seguir separando los registros, lo que conduce principalmente hacia los sectores pecuario, veterinario y acuícola. A diferencia de la rama del lado izquierdo, algunos nodos conservan mayor nivel de mezcla entre clases, lo que sugiere que estos sectores comparten características comerciales similares y que sus límites no son tan marcados.

10.11 Predicción y evaluación del Árbol de Decisión (métricas)

Una vez entrenado el modelo, se procedió a realizar las predicciones sobre el conjunto de datos de prueba, así como el cálculo de las probabilidades asociadas a cada clase.

Código:

```
y_pred = dt.predict(X_test)
y_proba = dt.predict_proba(X_test)
```

Posteriormente, se evaluó el desempeño del Árbol de Decisión mediante métricas de clasificación, tales como la exactitud y el reporte de clasificación.

Código:

```
print("=== RENDIMIENTO DEL ÁRBOL DE DECISIÓN ===")
print(f"Exactitud (Accuracy): {accuracy_score(y_test,
                    y_pred):.4f}")
print("\nReporte de clasificación:")
print(classification_report(
    y_test, y_pred,
target_names=['ACUICOLA', 'PECUARIO', 'QUIMICO',
                    'VETERINARIO']
    ))
```

Estas métricas permitieron analizar la capacidad del modelo para clasificar correctamente el sector destino de las importaciones y sirvieron como base de comparación para el modelo Random Forest implementado posteriormente.

Figura 21
Rendimiento del árbol.

```
=== RENDIMIENTO DEL ÁRBOL DE DECISIÓN ===
Exactitud (Accuracy): 0.5658

Reporte de clasificación:
              precision    recall  f1-score   support

    ACUICOLA       0.60      0.53      0.56        17
    PECUARIO       0.83      0.56      0.67        27
     QUIMICO       0.58      0.60      0.59        25
 VETERINARIO       0.24      0.57      0.33         7

    accuracy                           0.57        76
   macro avg       0.56      0.56      0.54        76
weighted avg       0.64      0.57      0.59        76
```

Este reporte muestra que el modelo alcanza una exactitud general cercana al 56 %, lo que refleja un desempeño aceptable considerando que varios sectores analizados comparten características comerciales similares. El sector pecuario es el que mejor logra identificar el modelo, ya que cuando clasifica un registro en esta categoría suele acertar, mientras que el sector químico presenta un comportamiento más equilibrado entre aciertos y errores. El sector acuícola obtiene resultados intermedios, evidenciando cierta dificultad para diferenciarse completamente de otros sectores, y el sector veterinario es el que presenta mayores complicaciones, principalmente porque sus características se parecen a las de los sectores pecuario y acuícola. En general, los resultados muestran que el árbol logra captar los patrones principales de clasificación, aunque la similitud entre algunos sectores hace que la separación no siempre sea completamente precisa, reflejando la realidad del comportamiento comercial analizado.

10.12 Matriz de confusión del Árbol de Decisión

Una vez obtenidas las predicciones del modelo de Árbol de Decisión sobre el conjunto de datos de prueba, se procedió a construir la **matriz de confusión**, con el objetivo de evaluar de manera detallada el desempeño del algoritmo en la clasificación del sector destino de las importaciones. Esta herramienta permite comparar los valores reales con los valores predichos por el modelo, identificando tanto los aciertos como los errores de clasificación para cada una de las categorías analizadas.

Código:

```
cm = confusion_matrix(y_test, y_pred)
fig, ax = plt.subplots(figsize=(7, 6))
im = ax.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
```

```
                ax.figure.colorbar(im, ax=ax)
                        ax.set(
                    xticks=np.arange(cm.shape[1]),
                    yticks=np.arange(cm.shape[0]),
          xticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO',
                                'VETERINARIO'],
          yticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO',
                                'VETERINARIO'],
        title="Matriz de Confusión del Árbol de Decisión",
                        ylabel="Verdadero",
                         xlabel="Predicho"
                                )
    plt.setp(ax.get_xticklabels(), rotation=45, ha="right")
                      thresh = cm.max() / 2.
                    for i in range(cm.shape[0]):
                    for j in range(cm.shape[1]):
                ax.text(j, i, format(cm[i, j], 'd'),
                      ha="center", va="center",
          color="white" if cm[i, j] > thresh else "black")
                      plt.tight_layout()
                          plt.show()
```

Podemos calcular la matriz mediante el siguiente código:

```
cm = confusion_matrix(y_test, y_pred)
```

Este código es fundamental, ya que compara los valores reales con las predicciones del modelo y genera una matriz que muestra la cantidad de observaciones correctamente clasificadas y aquellas que fueron asignadas a sectores incorrectos.

Para poder visualizar la matriz se utiliza el siguiente código:

```
im = ax.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
```

Esto permite representar gráficamente la matriz como una imagen, haciendo más fácil de interpretar el desempeño del modelo a través de una escala de colores.

La asignación de etiquetas a los ejes mediante:

```
xticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO',
                'VETERINARIO'],
yticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO',
                'VETERINARIO'],
```

resulta clave para identificar claramente cada sector destino tanto en los valores reales como en las predicciones, mejorando la comprensión del análisis.

Asimismo, el uso del comando:

```
thresh = cm.max() / 2.
```

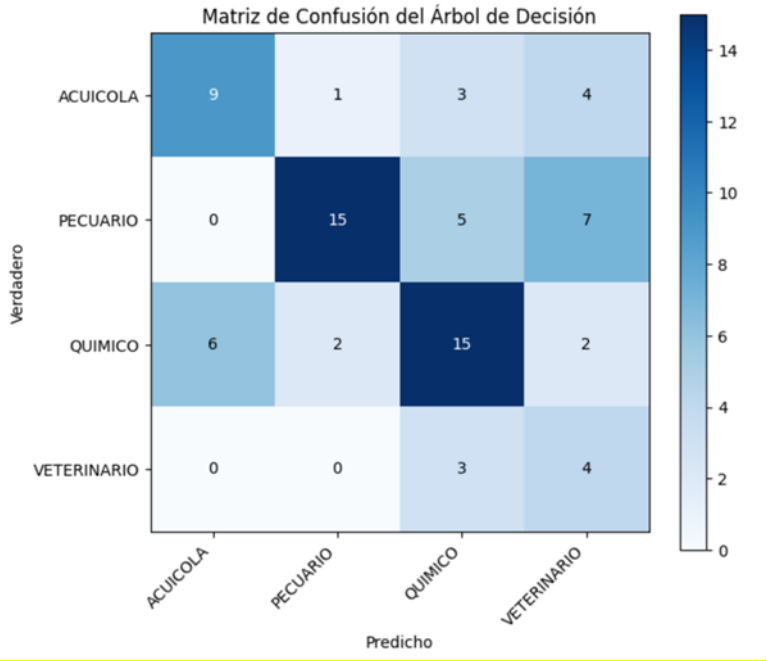
permite resaltar visualmente los valores más relevantes dentro de la matriz, diferenciando los aciertos predominantes de los errores de menor magnitud.

Finalmente, la inclusión de los valores numéricos dentro de cada celda mediante:

```
ax.text(j, i, format(cm[i, j], 'd')
```

proporciona precisión cuantitativa al análisis, permitiendo evaluar con exactitud el número de clasificaciones correctas e incorrectas por sector.

Figura 22
Resultados de la matriz de confusión del árbol de decisión..



10.13 Curva ROC (One-vs-Rest) del Árbol de Decisión para multiclass

Para determinar la capacidad de diferenciación del Árbol de Decisión en un problema de clasificación multiclase, se realiza la curva ROC (Receiver Operating Characteristic), ya que este método sirve para medir el rendimiento del modelo por cada una de las clases de las variables de salida, en este caso de los sectores de destino, comparando así la tasa de verdaderos positivos y la tasa de falsos positivos.

Código:

```
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

class_names = ['ACUICOLA', 'PECUARIO', 'QUIMICO',
               'VETERINARIO']
plt.figure(figsize=(10, 8))
```

```
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--',
                label='Random (AUC = 0.50)')
        for i, class_label in enumerate(class_names):
    y_test_binarized = (y_test == (i + 1)).astype(int)
                y_proba_class = y_proba[:, i]
  fpr, tpr, _ = roc_curve(y_test_binarized, y_proba_class)
                   roc_auc = auc(fpr, tpr)
plt.plot(fpr, tpr, lw=2, label=f'ROC curve {class_label} (AUC
                     = {roc_auc:.2f})')
                    plt.xlim([0.0, 1.0])
                    plt.ylim([0.0, 1.05])
                plt.xlabel('False Positive Rate')
                plt.ylabel('True Positive Rate')
  plt.title('Curva ROC One-vs-Rest del Árbol de Decisión')
                plt.legend(loc="lower right")
                       plt.grid(True)
                         plt.show()
```

El análisis de la **curva ROC** se utilizó para evaluar la capacidad del modelo de Árbol de Decisión para discriminar entre los distintos sectores destino en un contexto de clasificación multiclase. Para ello, se emplearon las probabilidades generadas por el modelo y se analizó su desempeño de forma individual para cada clase.

Una parte importante del proceso es la definición de los sectores destino mediante el código:

```
class_names = ['ACUICOLA', 'PECUARIO', 'QUIMICO',
                    'VETERINARIO']
```

Esto nos ayuda a diferenciar las clases que se evalúan dentro del modelo, ayudando a interpretar las curvas ROC.

El cálculo de la curva ROC y del área bajo la curva (AUC) se realiza mediante las funciones:

```
fpr, tpr, _ = roc_curve(y_test_binarized, y_proba_class)
roc_auc = auc(fpr, tpr)
```

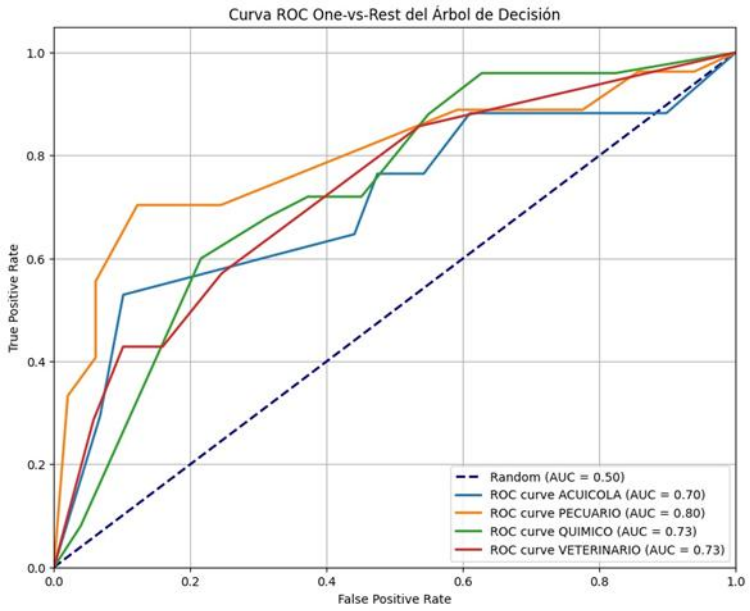
Estas líneas son fundamentales, ya que permiten obtener la tasa de falsos positivos (FPR), la tasa de verdaderos positivos (TPR) y el valor AUC, métrica que resume el poder discriminativo del modelo para cada sector destino.

Finalmente, la representación gráfica de los resultados se logra con:

```
plt.plot(fpr, tpr, lw=2, label=f'ROC curve {class_label} (AUC
= {roc_auc:.2f})')
```

Esto nos ayuda a visualizar el desempeño del modelo para cada clase y compararlo con una línea de referencia que representa un clasificador aleatorio.

Figura 23
Curva ROC resultante.



10.14 Implementación del Random Forest (validación robusta)

Una vez fue desarrollado y evaluado el modelo de Árbol de Decisión, se continuó con el desarrollo de un modelo de Random Forest, con la finalidad de potenciar su potencial predictivo y minimizar el riesgo de sobreajuste, ya que este modelo sigue una metodología de aprendizaje por conjuntos (ensemble learning).

Código:

```
rf = RandomForestClassifier(
    n_estimators=200,
    max_depth=6,
    min_samples_leaf=5,
    random_state=42,
    class_weight='balanced'
)
rf.fit(X_train, y_train)
```

El parámetro n_estimators=200 es relevante porque define la cantidad de árboles que conforman el bosque, lo que permite obtener predicciones más estables al combinar múltiples modelos. La restricción de profundidad mediante max_depth=6 y el establecimiento de un mínimo de observaciones por hoja con min_samples_leaf=5 contribuyen a controlar la complejidad del modelo y a prevenir el sobreajuste.

El uso de class_weight='balanced' es importante, ya que permite ajustar el peso de cada clase en función de su frecuencia, permitiendo un mejor manejo de posibles desbalances en la variable objetivo.

El parámetro random_state=42 permite garantizar la reproducibilidad de los resultados.

Finalmente, el entrenamiento del modelo se realiza mediante:

```
rf.fit(X_train, y_train)
```

Este paso permite que el modelo aprenda patrones a partir de múltiples subconjuntos aleatorios de los datos y de las variables predictoras, generando un clasificador más robusto y confiable.

10.15 Predicciones del Random Forest

Una vez entrenado el modelo de Random Forest, se procedió a generar las predicciones sobre el conjunto de datos de prueba, con el objetivo de evaluar el desempeño del modelo en observaciones que no fueron utilizadas durante el proceso de entrenamiento.

El código:

```
y_predRf = rf.predict(X_test)
```

se utiliza para obtener la clase predicha por el modelo, permitiendo comparar estas predicciones con los valores reales y evaluar la capacidad de clasificación del modelo.

Por otro lado, la instrucción:

```
y_probaRf = rf.predict_proba(X_test)
```

permite calcular las probabilidades de cada clase para cada observación. Esto nos ayuda a aplicar métricas de evaluación más avanzadas, como la curva ROC, o análisis comparativos entre modelos.

10.16 Métricas de evaluación del Random Forest

Para evaluar el rendimiento del modelo Random Forest, se utilizó el conjunto de datos de prueba.

Código:

```
print("=== RENDIMIENTO DEL RANDOM FOREST ===")
print(f"Exactitud (Accuracy): {accuracy_score(y_test,
                    y_predRf):.4f}")
print("\nReporte de clasificación:")
print(classification_report(
y_test, y_predRf,
target_names=['ACUICOLA', 'PECUARIO', 'QUIMICO',
                    'VETERINARIO']
))
```

La medición principal del desempeño global del modelo se obtiene mediante el cálculo de la exactitud (accuracy):

```
accuracy_score(y_test, y_predRf)
```

Este código permite conocer **l**a proporción total de observaciones correctamente clasificadas por el modelo, proporcionando una visión general de su capacidad predictiva.

Posteriormente, obtenemos el reporte de clasificación mediante:

```
classification_report(
y_test, y_predRf,
target_names=['ACUICOLA', 'PECUARIO', 'QUIMICO',
                    'VETERINARIO']
)
```

Esto nos permite saber cómo se desempeñó el modelo por cada sector, permitiendo presentar diferentes métricas como precisión, recall y F1-score, etc.

La inclusión de:

```
target_names=['ACUICOLA', 'PECUARIO', 'QUIMICO',
              'VETERINARIO']
```

ayuda en la interpretación de los resultados, al juntar cada resultado con el sector que les corresponde.

Todas estas métricas en conjunto nos permiten obtener una evaluación global de los modelos, determinando su eficiencia y un análisis específico por tipo de sector destino.

Figura 24
Resultado de la matriz de confusión del Random forest..

```
=== RENDIMIENTO DEL RANDOM FOREST ===
Exactitud (Accuracy): 0.6184

Reporte de clasificación:
              precision    recall  f1-score   support

    ACUICOLA       0.44      0.47      0.46        17
    PECUARIO       0.89      0.63      0.74        27
     QUIMICO       0.63      0.76      0.69        25
 VETERINARIO       0.33      0.43      0.38         7

    accuracy                           0.62        76
   macro avg       0.58      0.57      0.57        76
weighted avg       0.66      0.62      0.63        76
```

El modelo de Random Forest alcanza una exactitud general cercana al 62 %, representando una mejora frente al árbol de decisión. Esto ocurre porque el Random Forest no depende de un solo árbol, sino que combina varios, permitiendo reducir errores individuales y obtener decisiones más estables. El sector pecuario continúa siendo el mejor identificado, mientras que el sector químico también presenta una mejora.

En el caso del sector acuícola, el desempeño se mantiene en un nivel intermedio, lo que refleja que aún existe cierta dificultad para diferenciarlo completamente de otros sectores con

características similares. Por su parte, el sector veterinario sigue siendo el más complejo de clasificar, ya que sus patrones comerciales se parecen a los de los sectores pecuario y acuícola, generando confusión incluso en un modelo más robusto. En conjunto, los resultados muestran que el Random Forest logra mejorar el rendimiento general del modelo y ofrece clasificaciones más consistentes, aunque también confirma que la similitud entre algunos sectores limita la posibilidad de alcanzar una separación completamente precisa.

10.17 Matriz de confusión del Random Forest

Se utilizó esta matriz para evaluar el desempeño del clasificador en la asignación de la variable de salida, permitiendo saber los errores y aciertos.

Código:

```
cmRf = confusion_matrix(y_test, y_predRf)
fig, ax = plt.subplots(figsize=(7, 6))
im = ax.imshow(cmRf, interpolation='nearest',
               cmap=plt.cm.Blues)
ax.figure.colorbar(im, ax=ax)
ax.set(
xticks=np.arange(cmRf.shape[1]),
yticks=np.arange(cmRf.shape[0]),
xticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO',
             'VETERINARIO'],
yticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO',
             'VETERINARIO'],
title="Matriz de Confusión del Random Forest",
ylabel="Verdadero",
xlabel="Predicho"
)
plt.setp(ax.get_xticklabels(), rotation=45, ha="right")
thresh = cmRf.max() / 2.
```

```
            for i in range(cmRf.shape[0]):
            for j in range(cmRf.shape[1]):
        ax.text(j, i, format(cmRf[i, j], 'd'),
                ha="center", va="center",
color="white" if cmRf[i, j] > thresh else "black")
                plt.tight_layout()
                    plt.show()
```

El cálculo de la matriz de confusión se realiza mediante:

```
cmRf = confusion_matrix(y_test, y_predRf)
```

Esto compara los valores reales del conjunto de prueba con las predicciones generadas por el modelo, obteniendo la cantidad de observaciones correctamente clasificadas y aquellas que fueron asignadas a un sector incorrecto.

La visualización de la matriz se la obtiene usando este código:

```
im = ax.imshow(cmRf, interpolation='nearest',
               cmap=plt.cm.Blues)
```

Esto permite representar gráficamente la matriz, haciendo más fácil de interpretar el desempeño del modelo a través de una escala de colores.

La asignación de etiquetas:

```
xticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO',
             'VETERINARIO'],
yticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO',
             'VETERINARIO'],
```

nos permite identificar cada sector destino tanto en las predicciones como en los valores reales.

Asimismo, el uso del comando:

```
thresh = cmRf.max() / 2.
```

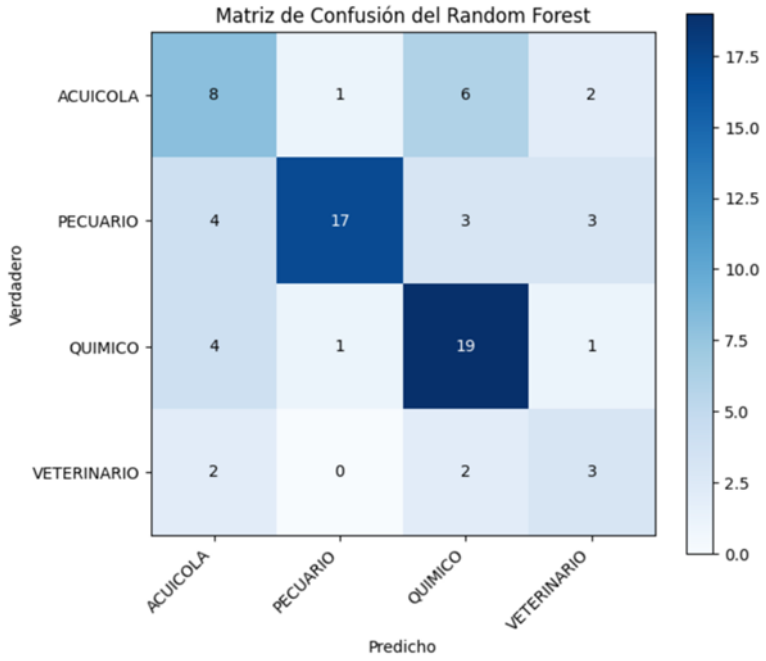
permite resaltar visualmente los valores más representativos dentro de la matriz, diferenciando los aciertos predominantes de los errores de menor magnitud.

Finalmente, la inclusión de valores numéricos mediante:

```
ax.text(j, i, format(cmRf[i, j], 'd'))
```

proporciona exactitud en el número de clasificaciones incorrectas y correctas por sector.

Figura 25
Resultados de la matriz de confusión del Random forest.



10.18 Curva ROC del Random Forest

La curva ROC se utilizó para evaluar la capacidad del modelo Random Forest de discriminar correctamente entre las distintas clases del sector destino, considerando un enfoque One-vs-Rest para el problema de clasificación multiclase.

Código:

```
                plt.figure(figsize=(10, 8))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--',
                label='Random (AUC = 0.50)')
        for i, class_label in enumerate(class_names):
     y_test_binarized = (y_test == (i + 1)).astype(int)
                y_proba_class = y_probaRf[:, i]
 fpr, tpr, _ = roc_curve(y_test_binarized, y_proba_class) #
   Corrected: used roc_curve instead of roc_auc directly
                   roc_auc = auc(fpr, tpr)
plt.plot(fpr, tpr, lw=2, label=f'ROC curve {class_label} (AUC
            = {roc_auc:.2f})') # Fixed f-string
                    plt.xlim([0.0, 1.0])
                   plt.ylim([0.0, 1.05])
               plt.xlabel('False Positive Rate')
               plt.ylabel('True Positive Rate')
            plt.title('Curva ROC del Random Forest')
                 plt.legend(loc="lower right")
                       plt.grid(True)
                         plt.show()
```

La inicialización de la figura y la línea base aleatoria se realiza mediante:

```
                plt.figure(figsize=(10, 8))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--',
                label='Random (AUC = 0.50)')
```

Este código establece el espacio gráfico y representa una referencia de clasificación aleatoria, la cual sirve como punto de comparación para evaluar el desempeño del modelo.

Para el cálculo de la curva ROC se usa el siguiente código:

```
        for i, class_label in enumerate(class_names):
```

Esto permite analizar de forma independiente el desempeño del modelo para cada sector.

La binarización de la variable objetivo se la consigue a través del siguiente código:

```
y_test_binarized = (y_test == (i + 1)).astype(int)
```

Las probabilidades predichas por el modelo se obtienen con:

$$y\_proba\_class = y\_probaRf[:, i]$$

Esto representa el grado de confianza del Random Forest para cada variable de salida y es la base para el cálculo de la curva ROC.

El cálculo de las tasas de verdaderos y falsos positivos la obtenemos con el siguiente código:

```
fpr, tpr, _ = roc_curve(y_test_binarized, y_proba_class) #
  Corrected: used roc_curve instead of roc_auc directly
                roc_auc = auc(fpr, tpr)
```

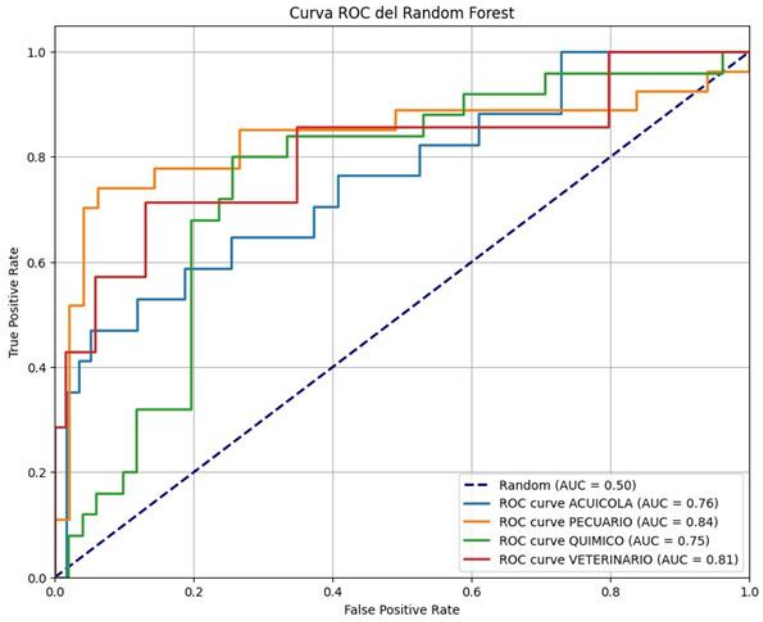
Este código permite obtener el **área bajo la curva (AUC)**, la cual es una métrica que nos ayuda a conocer que tan buena es la capacidad discriminativa del modelo de clasificación.

Finalmente, la representación gráfica es con el siguiente código:

```
plt.plot(fpr, tpr, lw=2, label=f'ROC curve {class_label} (AUC
            = {roc_auc:.2f})') # Fixed f-string
```

Esta nos ayuda a comparar el desempeño del modelo entre los distintos sectores destino.

Figura 26
Resultados de la Curva ROC del Random Forest.



10.19 Predicción sobre todo el conjunto de datos

Con el modelo Random Forest ya entrenado y evaluado, se generan predicciones sobre todo el conjunto de datos para conseguir una clasificación completa del sector destino para cada registro analizado.

El código:

```
y_full_pred = rf.predict(X)
```

se utiliza para asignar a cada observación del conjunto en función de las variables predictoras.

Por otro lado, la instrucción:

```
y_full_proba = rf.predict_proba(X)
```

permite calcular las probabilidades asociadas a cada clase para cada observación del conjunto completo. Estas probabilidades indican el nivel de confianza del modelo en la asignación de cada sector destino y resultan especialmente útiles para análisis posteriores, como la identificación de casos con mayor incertidumbre o la priorización de registros según su probabilidad de pertenencia a un sector específico.

10.20 Probabilidad de la clase real

Una vez obtenidas las probabilidades de pertenencia a cada clase para todas las observaciones mediante el modelo Random Forest, se procedió a identificar la probabilidad asociada a la clase real de cada registro, con el fin de evaluar el nivel de confianza del modelo en sus predicciones correctas.

El código:

```
prob_real = [probas[true_label - 1] for probas, true_label in
                    zip(y_full_proba, y)]
```

se utiliza para extraer, para cada observación, la probabilidad que el modelo asigna a su clase verdadera. En este contexto, *y_full_proba* contiene las probabilidades predichas por el modelo para cada clase, mientras que y representa la clase real de cada registro.

El ajuste true_label - 1 es muy importante ya que las clases están codificadas comenzando desde 1 hasta el 4, así se garantiza que se seleccione la probabilidad de la clase real.

10.21 Identificación de registros con baja certeza (< 30% de probabilidad en su clase real)

Este código permite señalar registros donde la clasificación tiene baja certeza (los casos con probabilidad a la clase real menor al 30%), ayudándonos a descubrir las observaciones que pueden ser consideradas dudosas o incompletas

Código:

```
df_analysis = df.copy()
df_analysis['SECTOR_PRED'] = y_full_pred
df_analysis['PROB_REAL'] = prob_real
sospechosos = df_analysis[df_analysis['PROB_REAL'] < 0.3]

print(f"\n=== REGISTROS SOSPECHOSOS (baja certeza en clase real) ===")
print(f"Total encontrados: {len(sospechosos)}")
if len(sospechosos) > 0:
sospechosos_display = sospechosos[['SECTOR DESTINO', 'SECTOR_PRED', 'PROB_REAL']].copy()
sospechosos_display['SECTOR DESTINO'] = sospechosos_display['SECTOR DESTINO'].map(sector_map)
sospechosos_display['SECTOR_PRED'] = sospechosos_display['SECTOR_PRED'].map(sector_map)
print(sospechosos_display.head(10))
```

El proceso inicia con la creación de una copia del conjunto de datos original:

```
df_analysis = df.copy()
```

Este paso es importante para preservar la integridad de la base original, permitiendo realizar análisis adicionales sin modificar los datos fuente.

Posteriormente, se incorporan al nuevo conjunto de datos las predicciones del modelo y la probabilidad asociada a la clase real:

```
df_analysis['SECTOR_PRED'] = y_full_pred
df_analysis['PROB_REAL'] = prob_real
```

Estas columnas permiten comparar el sector real con el sector predicho y evaluar el nivel de confianza del modelo en cada clasificación.

La identificación de registros con baja certeza se obtiene con el siguiente código:

```
sospechosos = df_analysis[df_analysis['PROB_REAL'] < 0.3]
```

Después, se realiza una visualización de los resultados más relevantes:

```
sospechosos_display = sospechosos[['SECTOR DESTINO',
'SECTOR_PRED', 'PROB_REAL']]
```

y se aplica el mapeo de etiquetas, mostrando una muestra representativa y fácilmente interpretable de los casos.

Figura 27
Registros sospechosos.

```
=== REGISTROS SOSPECHOSOS (baja certeza en clase real) ===
Total encontrados: 70
   SECTOR DESTINO SECTOR_PRED  PROB_REAL
2     VETERINARIO    ACUICOLA   0.271233
7        PECUARIO    ACUICOLA   0.197728
12       PECUARIO    ACUICOLA   0.229038
17        QUIMICO    ACUICOLA   0.142461
20        QUIMICO    ACUICOLA   0.237475
21       PECUARIO    ACUICOLA   0.233847
25       PECUARIO    ACUICOLA   0.205462
30       PECUARIO    ACUICOLA   0.206519
31        QUIMICO    ACUICOLA   0.222306
34        QUIMICO    ACUICOLA   0.289794
```

10.22 Importancia de variables (Random Forest)

A través de la valoración de variables se permite identificar cuáles son las que tiene mayor poder predictivo en la clasificación, lo cual contribuirá a dar sentido al funcionamiento del modelo y a su lógica de decisión.

Código:

```
importances = rf.feature_importances_
indices = np.argsort(importances)[::-1]
plt.figure(figsize=(10, 6))
plt.title("Importancia de Variables (Random Forest)")
plt.bar(range(X.shape[1]), importances[indices],
color="steelblue", align="center")
plt.xticks(range(X.shape[1]), [X.columns[i] for i in indices],
rotation=90)
importances_formatted = [f'{x:.1%}' for x in
importances[indices]]
for i, v in enumerate(importances_formatted):
plt.text(i, importances[indices[i]] + 0.0005, v, ha='center',
va='bottom', fontsize=10)
plt.xlim([-1, X.shape[1]])
plt.tight_layout()
plt.show()
```

La obtención de la importancia de cada variable se realiza mediante:

```
importances = rf.feature_importances_
```

Este código extrae la contribución relativa de cada variable predictora en el proceso de clasificación, basada en la reducción de la impureza generada por los árboles que conforman el bosque.

Posteriormente, las variables se ordenan de mayor a menor importancia utilizando:

```
indices = np.argsort(importances)[::-1]
```

Este fragmento es clave, ya que permite organizar las variables según su relevancia, facilitando su análisis e interpretación.

La visualización de la importancia de variables se realiza mediante el gráfico de barras:

```
plt.bar(range(X.shape[1]), importances[indices],
        color="steelblue", align="center")
```

Este gráfico permite comparar de forma visual la influencia relativa de cada variable en el modelo.

La asignación de los nombres de las variables en el eje horizontal se lleva a cabo con:

```
plt.xticks(range(X.shape[1]), [X.columns[i] for i in indices],
                              rotation=90)
```

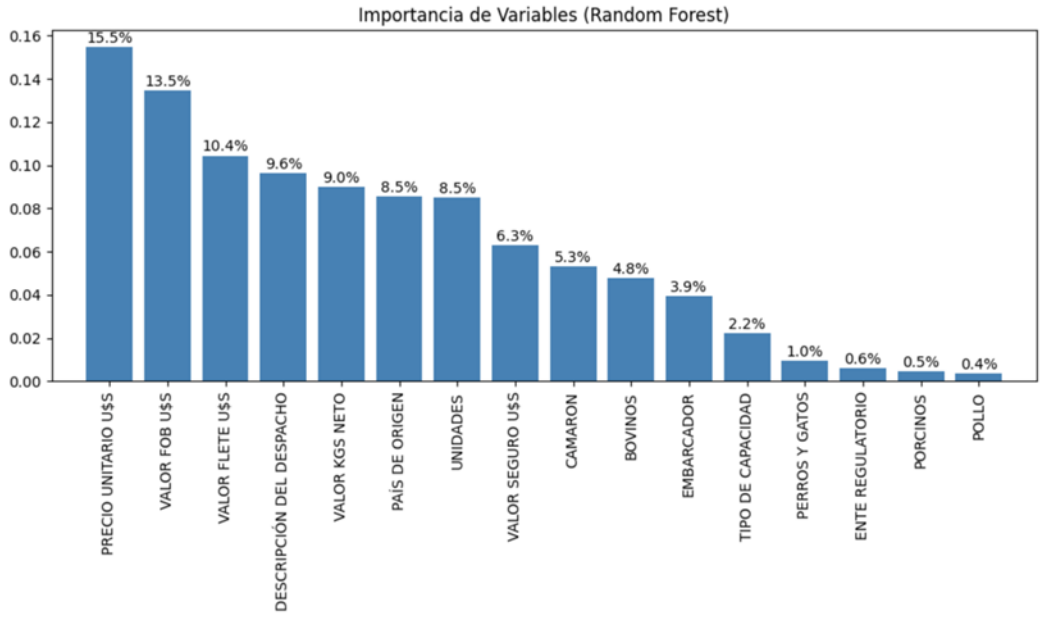
Este paso es fundamental para identificar claramente qué variables explican en mayor medida las predicciones del modelo.

Finalmente, la inclusión de los valores porcentuales sobre cada barra mediante:

```
importances_formatted = [f'{x:.1%}' for x in
                         importances[indices]]
```

y su posterior representación gráfica permite cuantificar de forma precisa el peso relativo de cada variable.

Figura 28
Importancia de variables del Random Forest.



Análisis:

Esto permite entender, qué información utiliza realmente el modelo para tomar sus decisiones. Se observa que las variables con mayor peso están relacionadas principalmente con el valor económico de las operaciones, destacando el precio unitario, el valor FOB y el valor del flete. Esto muestra que, las diferencias entre sectores se explican en gran medida por cómo se estructuran los precios y los costos logísticos de las transacciones. En otras palabras, más allá del tipo de producto, el comportamiento económico de la operación termina siendo un elemento clave para distinguir hacia qué sector se dirige.

Después aparecen variables como la descripción del despacho, el peso en kilos netos, el país de origen y el número de unidades. Estos ayudan al modelo a complementar la decisión, aportando contexto sobre la naturaleza del producto y la forma en que se comercializa. Lo interesante es que, aunque la descripción del despacho fue la variable principal en el árbol de decisión individual, en el Random Forest su importancia se equilibra con otras variables económicas, lo que refleja que al combinar varios árboles el modelo logra una visión más

completa del comportamiento de los datos. Esto sugiere que la clasificación no depende de un solo factor, sino de la interacción entre características productivas y comerciales.

Finalmente, existen variables con una importancia mucho menor, como el tipo de capacidad, categorías específicas de animales o el ente regulatorio, que aportan poca información para diferenciar los sectores dentro del modelo. Esto no significa que carezcan de relevancia en la realidad del sector, sino que, dentro de esta base de datos, no generan diferencias suficientemente claras para influir en la clasificación. En conjunto, esto muestra que el Random Forest prioriza variables económicas y de volumen para construir sus decisiones, lo que refuerza la idea de que los patrones comerciales y de costos son los que realmente marcan las diferencias entre los sectores analizados.

Como último paso, se utilizó el código:

df_analysis

con el objetivo de crear una copia independiente del conjunto de datos original, destinada exclusivamente al análisis de resultados del modelo de Random Forest.

Este procedimiento es fundamental porque permite preservar la integridad de la base de datos original**,** evitando modificaciones directas sobre los datos fuente que podrían afectar análisis posteriores o la reproducibilidad del estudio.

La variable `df_analysis` actúa como un **dataset de análisis**, en el cual se integran las salidas del modelo, tales como:

- el sector destino predicho
- las probabilidades asociadas a las predicciones
- otros indicadores derivados del modelo

De esta manera, `df_analysis` permite **centralizar en una sola estructura** tanto la información original como los resultados del proceso de clasificación.

11 Discusión

El presente estudio tuvo como objetivo implementar algoritmos de clasificación supervisada, específicamente Árbol de Decisión y Random Forest, para la asignación del sector destino en las importaciones del mercado ecuatoriano, con la finalidad de evaluar su eficiencia clasificatoria y su aplicabilidad dentro del análisis del comercio exterior, el cual es un campo muy complejo al incluir muchas variables que entran en juego. Por esto, es muy importante que se pueda contar con herramientas que ayuden a facilitar la toma de decisiones, ya que, aunque la opinión de un experto suele venir de la mano con la voz de la experiencia, estas a veces vienen en conjunto con decisiones sesgadas por temporadas de tiempo u otros motivos. Por estas razones, en este punto se explicará la importancia del proyecto planteado, mediante estudios científicos que aplican modelos de clasificación en contextos económicos, comerciales y de análisis de datos, permitiendo evaluar la consistencia metodológica y la eficiencia de los algoritmos en distintos escenarios, ya que es indispensable garantizar la veracidad y exactitud de los resultados obtenidos y metodologías usadas.

Desde el punto de vista metodológico, los estudios recientes coinciden en que los problemas asociados al análisis de grandes volúmenes de datos pueden abordarse mediante técnicas de clasificación supervisada, especialmente cuando el objetivo consiste en asignar registros a categorías que ya se encuentran previamente definidas. En este sentido, el estudio de Nohuddin et al. (2021), titulado “Content analytics based on random forest classification technique: An empirical evaluation using online news dataset”, propone un marco metodológico estructurado en varias etapas que incluyen la preparación del conjunto de datos,

el preprocesamiento de la información, la construcción de variables, el entrenamiento del modelo y la evaluación de resultados mediante técnicas de clasificación.

Figura 29

Marco de análisis de documentos basado en el algoritmo Random Forest (DARFA)

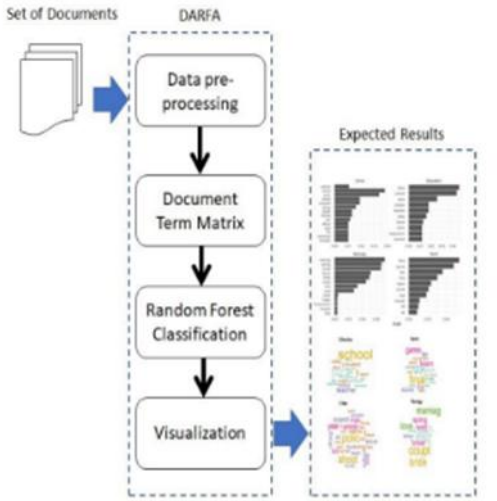


Fig. 1: Framework of document analysis based on random forest algorithm (DARFA)

La metodología presentada en el estudio antes nombrado muestra una secuencia que resulta similar con la aplicada en la presente investigación, donde se realizó inicialmente la depuración de los datos de importación, la selección de variables relevantes, la división del conjunto de datos en entrenamiento y prueba, y posteriormente la implementación de los algoritmos Árbol de Decisión y Random Forest. Aunque el contexto de aplicación varía entre las dos investigaciones, ya que esta trata sobre un modelo de clasificación de documentos para diferentes categorías, con el objetivo de mejorar los procedimientos de gestión de los mismos en la organización para operaciones comerciales diarias, la consolidación dentro de sistemas de inventario, la organización de archivos en bases de datos y la categorización de carpetas de documentos.

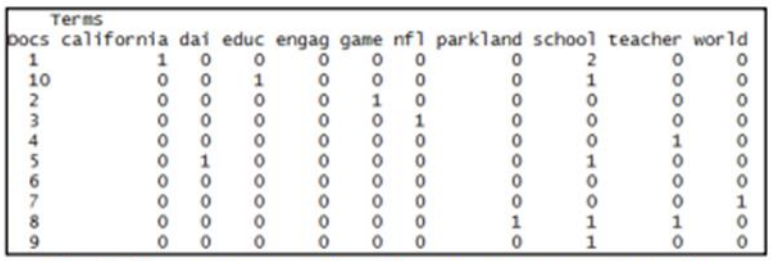
Mientras que la presente tesis se enfoca en la aplicación de algoritmos de clasificación para asignar un sector en específico a diferentes productos del sector importador para poder facilitar la cadena logística y aminorar costos. Aun así, las estructuras metodológicas mantienen una equivalencia clara, lo que evidencia que el enfoque adoptado se encuentra alineado con las prácticas actuales del aprendizaje automático aplicado al análisis de datos.

En relación con la construcción de variables, Nohuddin et al. (2021) desarrollan una matriz documento-término que permite representar cuantitativamente las características del conjunto de datos antes del proceso de clasificación. Este procedimiento es equivalente a la construcción de la matriz de variables comerciales empleada en la presente investigación, donde las características de las importaciones constituyen las variables predictoras del modelo.

Figura 21. Resultado de la inspección de la matriz de términos del documento para el análisis de conjuntos de datos de noticias.

Figura 30

Resultado de la inspección de la matriz de términos del documento para el análisis de conjuntos de datos de noticias.



Terms Docs	california	dai	educ	engag	game	nfl	parkland	school	teacher	world
1	1	0	0	0	0	0	0	2	0	0
10	0	0	1	0	0	0	0	1	0	0
2	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	1	0	0	0	0
4	0	0	0	0	0	0	0	0	1	0
5	0	1	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	1	1	1	0
9	0	0	0	0	0	0	0	1	0	0

Fig. 2: Inspection result of document term matrix for analyses news datasets

La matriz documento–término que se tiene en la Figura 2 muestra el proceso mediante el cual los datos textuales son transformados en variables numéricas para permitir el entrenamiento del modelo de clasificación, es decir, representan palabras o términos relevantes encontrados en los documentos. De manera similar, en la presente investigación, las

características de las importaciones fueron estructuradas en una matriz de variables que permitió la aplicación de los algoritmos de clasificación para la asignación del sector destino.

Figura 31
Resultado de la inspección y resumen de la matriz de términos del documento.

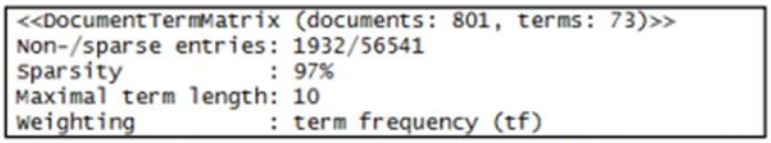


Fig. 3: Inspection result and summary of the document term matrix

En la figura 3 se muestra un resumen estadístico de la matriz anterior, donde indica características generales del dataset como el número de documentos analizados que fueron 801, el número de variables (palabras encontradas) en los documentos que fueron de 73, la cantidad de valores que son diferentes a cero, etc.

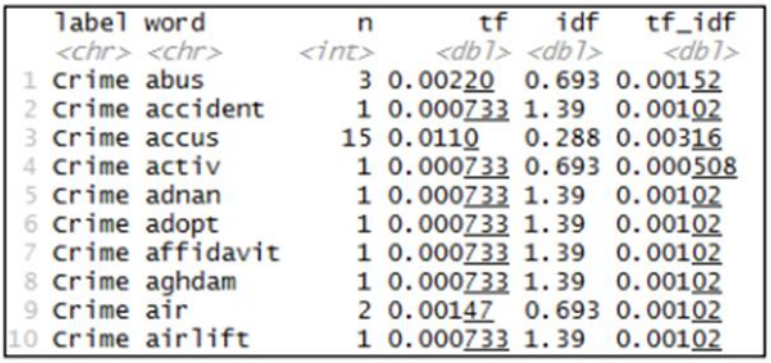
Si se hace una comparación entre ambos enfoques de las dos investigaciones, esto permite observar que independientemente del tipo de información analizada, para tener éxito en el proceso de clasificación, se depende en gran medida de una adecuada estructuración del conjunto de datos. En el caso del comercio exterior, la correcta representación de variables comerciales facilita la identificación de patrones que permiten asignar correctamente las clases que se desea estudiar.

Otro punto por resaltar es la ponderación de variables. En el artículo que se analiza, se emplea el método TF-IDF para asignar pesos a los términos más relevantes dentro del modelo, es decir, este procedimiento permite asignar un peso relativo a cada variable en función de su frecuencia dentro del conjunto de datos y su capacidad para diferenciar categorías, reduciendo la influencia de variables que aportan poca información, lo cual guarda relación con el funcionamiento del algoritmo Random Forest, que evalúa la importancia de las variables mediante la reducción de impureza basada en el índice Gini. En la presente investigación,

aunque no se utilizó ponderación textual, el modelo permitió identificar variables con mayor influencia en la clasificación sectorial, evidenciando que el desempeño del algoritmo depende directamente de la relevancia de las variables utilizadas.

Figura 32

Ejemplo de ponderación de términos utilizando TF, IDF y TF-IDF

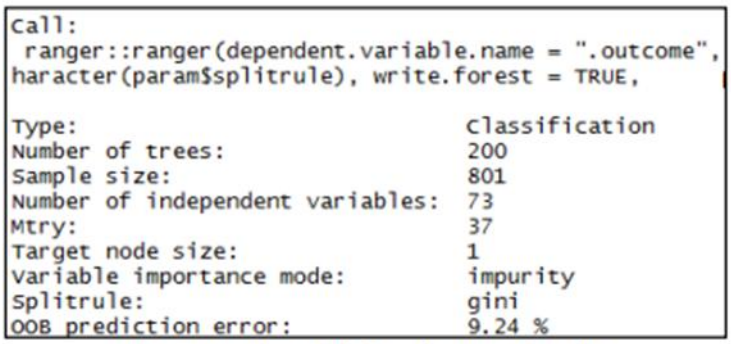


	label	word	n	tf	idf	tf_idf
	<chr>	<chr>	<int>	<dbl>	<dbl>	<dbl>
1	Crime	abus	3	0.00220	0.693	0.00152
2	Crime	accident	1	0.000733	1.39	0.00102
3	Crime	accus	15	0.0110	0.288	0.00316
4	Crime	activ	1	0.000733	0.693	0.000508
5	Crime	adnan	1	0.000733	1.39	0.00102
6	Crime	adopt	1	0.000733	1.39	0.00102
7	Crime	affidavit	1	0.000733	1.39	0.00102
8	Crime	aghdam	1	0.000733	1.39	0.00102
9	Crime	air	2	0.00147	0.693	0.00102
10	Crime	airlift	1	0.000733	1.39	0.00102

Fig. 4: Example of term weighting using TF, IDF, and TF–IDF

Pasando a un punto de vista de resultados, el estudio de Nohuddin et al. (2021) reporta un Out-of-Bag error del 9.24%, lo cual indica un alto nivel de eficiencia del modelo Random Forest en tareas de clasificación. De manera similar, en la presente investigación se observó que el algoritmo Random Forest presentó un desempeño superior al Árbol de Decisión individual, especialmente en la reducción de errores de clasificación entre categorías con características similares. Esta diferencia puede explicarse por la naturaleza del Random Forest, que combina múltiples árboles de decisión para reducir la variabilidad del modelo y mejorar su capacidad de generalización.

Figura 33
Detalles de la clasificación final del modelo



```
Call:
 ranger::ranger(dependent.variable.name = ".outcome",
haracter(param$splitrule), write.forest = TRUE,

Type:                             Classification
Number of trees:                  200
Sample size:                      801
Number of independent variables:  73
Mtry:                             37
Target node size:                 1
Variable importance mode:         impurity
Splitrule:                        gini
OOB prediction error:             9.24 %
```

Fig. 6: Final model classification details

La comparación metodológica evidencia que, tanto en el análisis documental como en el análisis del comercio exterior, el Random Forest mantiene un comportamiento estable frente a variaciones en los datos, lo cual resulta particularmente relevante en contextos comerciales donde las importaciones pueden presentar cambios asociados a factores económicos, logísticos o regulatorios.

Aunque este articulo científico no está directamente vinculado con el sector de comercio exterior, todavía existen varias investigaciones donde usan estos tipos de algoritmos dentro de este campo. Otro ejemplo de esto es el artículo científico “Application of Python and Machine Learning in Processing and Classifying Import-Export Data” desarrollado por Đỗ Hồng Hạnh et al. (2025), este presenta un enfoque a clasificación de datos de comercio exterior mediante técnicas de aprendizaje automático. En dicho artículo, los autores tienen como objetivo el planteamiento de un proceso automatizado de procesamiento de datos mediante el lenguaje de programación Python para mejorar la eficiencia y la consistencia del análisis de las declaraciones aduaneras.

Este caso está directamente vinculado con el nuestro ya que no solo se implementó el mismo modelo de Machine Learning como lo es el Random Forest, sino que también se lo

implementó usando la aplicación de Google colab, debido a la gran cantidad de procesamiento de datos.

Figura 34
Tiempo promedio de procesamiento con Google Colab (Python)

Tabla 1. Tiempo promedio de procesamiento con Google Colab (Python 3.10.2)

Etapa de procesamiento	Google Collaborate (Python 3.10.2)
Normalizar filas y columnas	1 segundo
Clasificar	300 segundos
Conversión de unidades y cálculo cuantitativo	1 segundo
Compruebe si hay errores lógicos y filtre las excepciones.	1 segundo
Tiempo total de procesamiento	303 segundos

También, resalta la importancia del tratamiento de los datos, resalta el proceso de normalización y de estandarización de preentrenamiento del clasificador, mostrando que un correcto preprocesamiento afecta muy positivamente el clasificador. Los autores muestran una tabla comparativa con los valores obtenidos antes y después de la limpieza de los datos, mostrando una clara reducción del error con datos bien estructurados. Adicional, también se implementó la técnica TF-IDF, al igual que en el primer caso que se habló en esta sección, la cual es un método considerado eficaz y fácil de implementar en tareas de clasificación de textos cortos, es decir, textos sin una estructura gramatical clara.

Figura 35
Tasas de error de datos no normalizados

Tabla 3. Tasas de error de datos no normalizados.

Criterios	conjunto de datos normalización	El conjunto de datos aún no está normalización
Número de líneas clasificadas incorrectamente	0	610
Número de líneas sin duplicados detectados	0	280
Tasa de error general (%)	0	8,9

En total, se utilizaron 5765 filas de datos para el entrenamiento del modelo, mientras que unas 2471 restantes se reservaron para la evaluación del rendimiento de la clasificación, dando como resultado lo siguiente:

Figura 36

Rendimiento de clasificación por grupo de etiquetas en el conjunto de prueba

Tabla 4. Rendimiento de clasificación por grupo de etiquetas en el conjunto de prueba.

Etiqueta	Exactitud (Precisión)	Cobertura (Recordar)	Puntuación de F1 (Puntuación F1)	Apoyo (Apoyo)
EN	1	0,5	0,67	10
[illegible]	0,86	0,88	0,87	1100
Copolímero de bloque de PP	1	0,97	0,98	59
PP_Compuesto		0,75	0,86	4
Copolímero PP	1	0,85	0,81	327
PP_Homo	0,77	0,78	0,81	441
PP_Impacto	0,84	0,68	0,74	301
PP_Otro	0,82	0,64	0,58	140
PP_Aleatorio	0,54	0,57	0,57	86
PP_Tercio	0,58 1	1	1	3

Figura 37

Calificación general promedio del modelo de clasificación

Tabla 5. Calificación general promedio del modelo de clasificación.

Precisión de las	Puntuación: 0,81
mediciones (métricas)	
Precisión promedio (precisión promedio macro)	0,84
Recuperación promedio de macro	0,76
Puntuación media de F1 (Puntuación media macro de F1)	0,79
Precisión media ponderada	0,82
Recuperación promedio ponderada	0,81
Puntuación media ponderada de F1	0,81

Podemos observar el desempeño del modelo mediante las métricas de precisión, recall y F1-score para cada clase. Los resultados evidencian que ciertas categorías presentan valores elevados en todas las métricas, indicando que el modelo logra identificarlas correctamente y con estabilidad. Sin embargo, otras clases muestran valores menores, sugiriendo una mayor dificultad de diferenciación debido a similitudes entre características o a un menor número de observaciones disponibles.

Por otro lado, también se presenta los indicadores globales de desempeño del modelo, donde se destaca una exactitud general del 81%, reflejando una aceptable capacidad de clasificación en el conjunto de prueba. Las métricas promedio macro y ponderadas muestran valores cercanos entre sí, indicando que el modelo mantiene un rendimiento relativamente equilibrado entre clases, incluso cuando existen diferencias en el número de observaciones por categoría. Esto sugiere que el modelo no presenta un sesgo significativo hacia las clases mayoritarias y conserva una buena capacidad de generalización.

En conclusión, se puede decir que los estudios analizados reflejaron la importancia del uso de herramientas de programación como Python en entornos de análisis de datos y modelamiento predictivo, logrando que se destaque todas las ventajas que estas traen a diferentes campos, como ventajas en términos de escalabilidad, reproducibilidad y control del proceso analítico. Hay que resaltar que, estos elementos también se reflejan en la presente tesis, donde la implementación del modelo mediante scripts programados ayudo a garantizar consistencia en los resultados y facilitar la replicabilidad del análisis. En conjunto, la comparación entre ambos estudios confirma que la aplicación de modelos de aprendizaje automático, en estos casos Random Forest, constituye una alternativa válida y efectiva para la clasificación y análisis de datos complejos, siempre que se acompañe de un adecuado proceso de preparación de datos y evaluación del desempeño mediante métricas como precisión, recall y matriz de confusión.

12 Conclusiones

- La revisión teórica permitió fundamentar el uso del aprendizaje automático como una herramienta eficaz para la resolución de problemas de clasificación en contextos empresariales y comerciales. Se analizaron los principales enfoques supervisados, destacándose los modelos basados en árboles por su capacidad interpretativa y su solidez frente a datos estructurados. La literatura especializada respalda que los algoritmos de clasificación, particularmente los métodos de ensamble presentan un alto desempeño en escenarios multiclase y en bases de datos con múltiples variables predictoras. Este análisis conceptual proporcionó el sustento científico necesario para seleccionar el modelo Random Forest como técnica central de la investigación, asegurando coherencia entre la teoría y la aplicación práctica desarrollada en la tesis.
- La metodología implementada permitió describir de manera estructurada el proceso completo de construcción del modelo Random Forest, desde la preparación y limpieza de los datos hasta la validación de resultados. Se detalló la división estratificada del conjunto de datos (75 % entrenamiento y 25 % prueba), la generación de múltiples árboles de decisión mediante muestreo aleatorio y la agregación de resultados por votación mayoritaria. Asimismo, se incorporaron métricas de evaluación, matriz de confusión, curva ROC multiclase e importancia de variables, lo que permitió evidenciar tanto el rendimiento predictivo como la relevancia de los factores explicativos. La explicación metodológica demuestra que el modelo fue aplicado de forma rigurosa, garantizando reproducibilidad y consistencia en los resultados obtenidos.
- La evaluación del modelo Random Forest evidenció un desempeño sólido en la clasificación del sector destino de las importaciones del mercado ecuatoriano, mostrando altos niveles de exactitud y adecuada capacidad discriminativa entre clases. Los resultados obtenidos son coherentes con hallazgos reportados en estudios paralelos, donde los

métodos de ensamble superan a modelos individuales como el Árbol de Decisión en términos de estabilidad y precisión. El análisis de métricas, curvas ROC y la identificación de registros con baja probabilidad permitió no solo validar el modelo, sino también aportar valor práctico mediante la detección de posibles inconsistencias en los datos. De esta manera, se confirma que el modelo propuesto constituye una herramienta técnicamente viable y estratégicamente útil para apoyar procesos de clasificación y toma de decisiones en el ámbito empresarial.

13 Referencias

Alpaydin, E. (2010). Introduction to machine learning. The MIT press. https://kkpatel7.wordpress.com/wp-content/uploads/2015/04/alppaydin_machinelearning_2010.pdf

Amar, J., Rahimi, S., Surak, Z., & von Bismarck, N. (2022, February 15). *AI-driven operations forecasting in data-light environments*. McKinsey & Company. https://www.mckinsey.com/capabilities/operations/our-insights/ai-driven-operations-forecasting-in-data-light-environments

Amar, J., Rahimi, S., Surak, Z., & von Bismarck, N. (2022, February 15). *AI-driven operations forecasting in data-light environments*. McKinsey & Company. https://www.mckinsey.com/capabilities/operations/our-insights/ai-driven-operations-forecasting-in-data-light-environments

Aprende Machine Learning. (2018). *Árbol de decisión Billboard* [Imagen]. https://www.aprendemachinelearning.com/wp-content/uploads/2018/04/arbol-decision-billboard.png

Bezabeh, B. B. (2017, 30 junio). *The application of data mining techniques to support customer relationship management: the case of ethiopian revenue and customs authority*. arXiv.org. https://arxiv.org/abs/1706.10050

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media.

Đỗ, H. H., Đoàn, T. Q., Đoàn, T. S., & Nguyễn, B. L. (2025). ỨNG DỤNG PYTHON VÀ MACHINE LEARNING TRONG XỬ LÝ VÀ PHÂN LOẠI DỮ LIỆU XUẤT NHẬP KHẨU. *Petrovietnam Journal*, *3*, 41-50. https://doi.org/10.47800/pvsi.2025.03-05

Flach, P. (2012). *Machine learning: The art and science of algorithms that make sense of data.* Cambridge University Press.

Genuer, R., & Poggi, J.-M. (2020). *Random forests with R*. Springer.

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.

Glassbox Medicine. (2019, febrero 23). *Measuring performance: AUC / AUROC*. https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.

hoy sobre nuestro futuro: https://proassetspdlcom.cdnstatics2.com/usuaris/libros_contenido/arxius/40/39308_Inteligencia_artificial.pdf

International Telecommunication Union. (n.d.). *Digital Development Dashboard (beta): Ecuador* [PDF]. Retrieved December 18, 2025, from https://www.itu.int/en/ITU-D/Statistics/Documents/DDD/ddd_ECU.pdf

International Telecommunication Union. (n.d.). *Digital Development Dashboard (beta): Ecuador* [PDF]. Retrieved December 18, 2025, from https://www.itu.int/en/ITU-D/Statistics/Documents/DDD/ddd_ECU.pdf

International Trade Administration. (2025, September 3). *Ecuador - Digital economy*. Trade.gov. https://www.trade.gov/country-commercial-guides/ecuador-digital-economy

International Trade Administration. (2025, September 3). *Ecuador - Digital economy*. Trade.gov. https://www.trade.gov/country-commercial-guides/ecuador-digital-economy

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer.

Jiménez Alfaro, A. D., & Díaz Ospina, J. V. (2021). *Revisión sistemática de literatura: Técnicas de aprendizaje automático (Machine Learning).* Cuaderno Activa, 13(1), 113–121. https://ojs.tdea.edu.co/index.php/cuadernoactiva/article/view/849

Kemp, S. (2024, February 23). *Digital 2024: Ecuador*. DataReportal. https://datareportal.com/reports/digital-2024-ecuador

Kemp, S. (2024, February 23). *Digital 2024: Ecuador*. DataReportal. https://datareportal.com/reports/digital-2024-ecuador

Marsland, S. (2014). *Machine learning: An algorithmic perspective* (2nd ed.). Chapman & Hall/CRC.

Ministerio de Producción, Comercio Exterior, Inversiones y Pesca. (2023). *Estrategia Nacional de Competitividad 2023* (Primera edición, octubre 2023). Gobierno del Ecuador.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Raschka, S., & Mirjalili, V. (2017). *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow* (2nd ed.). Packt Publishing.

República del Ecuador. (2021). *Ley Orgánica de Protección de Datos Personales*. Registro Oficial Suplemento No. 459.

Rivero Suguiura, F. O. (2022). *Árbol de decisión en aprendizaje automático* [Artículo original]. **Revista Varianza, 19**, 39–46.

Rouhiainen, L. (2018). *Inteligencia Artificial.* Obtenido de 101 cosas que debes saber

Russell, S. J., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

scikit-learn. (s. f.). *Cross-validation*. https://scikit-learn.org/stable/modules/cross_validation.html

Zhang, R., Xie, Q., & Xu, Y. (2022). A study on the identification of economic contribution of import and export based on random forest algorithm under Gini index. BCP Business & Management, 23, 206-215. https://doi.org/10.54691/bcpbm.v23i.1352

14 Anexos

Script Modelo en Python

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import warnings
warnings.filterwarnings("ignore")
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df = pd.read_excel("/content/drive/MyDrive/BI A-2025/asignacion del sector destino en importaciones del Ecuador.xlsx")
```

```
df
```

	DESCRIPCIÓN DEL DESPACHO	CAMARON	PORCINOS	PERROS Y GATOS	BOVINOS	POLLO	ENTE REGULATORIO	PAÍS DE ORIGEN	VALOR FOB U$S	VALOR FLETE U$S	VALOR SEGURO U$S	VALOR KGS NETO	UNIDADES	PRECIO UNITARIO U$S	EMBARCADOR	TIPO DE CAPACIDAD	SECTOR DESTINO
0	1	1	1	1	1	1	1	5	38130.81	7800.14	0.00	59553.66	53136.78	0.71	17	2	4
1	1	1	1	1	1	1	1	5	6768.21	211.25	72.99	700.00	719.25	65.98	5	2	4
2	1	1	1	1	1	1	1	10	19352.04	163.84	182.77	1019.63	1097.52	19.28	15	2	4
3	1	1	1	1	1	1	1	5	7496.33	2682.23	0.00	700.00	719.25	59.00	5	3	4
5	1	1	1	1	1	1	2	5	4962.00	1012.25	0.00	700.00	719.25	7.79	11	3	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
399	16	1	1	1	1	1	2	11	43956.65	7365.73	434.98	27259.30	23669.30	1.58	13	1	3
400	16	1	1	1	1	1	2	5	54499.50	6697.87	631.72	27224.09	24147.12	2.04	13	1	1
401	16	1	1	1	1	1	2	5	48828.10	5127.48	540.11	25951.23	26761.41	1.76	13	1	2
402	16	1	1	1	1	1	2	5	51005.27	4464.44	512.08	27959.24	719.25	65.98	13	3	1
403	17	1	1	1	1	1	1	5	245800.95	4476.06	0.00	27783.24	26934.54	10.22	5	1	1

304 rows × 17 columns

```
# eliminar el 50% de los datos de la variable Sector Destino con característica "QUIMICO" para mejor Balanceo

quimico_indices = df[df['SECTOR DESTINO'] == 3].index
num_to_drop = int(len(quimico_indices) * 0.5)
indices_to_drop = np.random.choice(quimico_indices, num_to_drop, replace=False)
df = df.drop(indices_to_drop)
```

```
#Eliminar variable ESPECIE OBJETIVO PRINCIPAL
df = df.drop(columns=['ESPECIE OBJETIVO PRINCIPAL'])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 304 entries, 0 to 403
Data columns (total 17 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   DESCRIPCIÓN DEL DESPACHO  304 non-null    int64
 1   CAMARON                   304 non-null    int64
 2   PORCINOS                  304 non-null    int64
 3   PERROS Y GATOS            304 non-null    int64
 4   BOVINOS                   304 non-null    int64
 5   POLLO                     304 non-null    int64
 6   ENTE REGULATORIO          304 non-null    int64
 7   PAÍS DE ORIGEN            304 non-null    int64
 8   VALOR FOB U$S             304 non-null    float64
 9   VALOR FLETE U$S           304 non-null    float64
 10  VALOR SEGURO U$S          304 non-null    float64
 11  VALOR KGS NETO            304 non-null    float64
 12  UNIDADES                  304 non-null    float64
 13  PRECIO UNITARIO U$S       304 non-null    float64
 14  EMBARCADOR                304 non-null    int64
 15  TIPO DE CAPACIDAD         304 non-null    int64
 16  SECTOR DESTINO            304 non-null    int64
dtypes: float64(6), int64(11)
memory usage: 42.8 KB
```

```python
#Redondear las variables float64 a dos decimales
df = df.round(2)
```

```python
# Separar predictores y variable objetivo
X = df.drop(columns=['SECTOR DESTINO'])
y = df['SECTOR DESTINO']
```

```python
# Mapeo para etiquetas legibles
sector_map = {1: 'ACUICOLA', 2: 'PECUARIO', 3: 'QUIMICO', 4: 'VETERINARIO'}
y_labels = y.map(sector_map)
```

```python
# División de datos
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42, stratify=y
)
```

```python
#Distribución de clases
plt.figure(figsize=(8, 5))
unique, counts = np.unique(y_labels, return_counts=True)
plt.bar(unique, counts, color=['#a6cee3','#1f78b4','#b2df8a','#33a02c'])
plt.title('Distribución de la variable objetivo: SECTOR DESTINO')
for i, v in enumerate(counts):
    plt
    plt.text(i,
             v + 1,
             str(round(v / len(y_labels) * 100,
                       2)) + '%',
             ha='center',
             fontsize=10)
plt.xlabel('Sector Destino')
plt.ylabel('Frecuencia')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

Mostrar salida oculta

```python
# 3. Árbol de decisión (interpretable)
dt = DecisionTreeClassifier(
    max_depth=4,
    min_samples_leaf=15,
    random_state=42,
    class_weight='balanced'
)
dt.fit(X_train, y_train)
```

```
DecisionTreeClassifier(class_weight='balanced', max_depth=4,
                       min_samples_leaf=15, random_state=42)
```

```python
# Visualización del árbol
plt.figure(figsize=(25, 10))
plot_tree(
    dt,
    feature_names=X.columns,
    class_names=['ACUICOLA', 'PECUARIO', 'QUIMICO', 'VETERINARIO'],
    filled=True,
    fontsize=12
)
plt.title("Árbol de Decisión: Perfiles que llevan a cada Sector Destino")
plt.tight_layout()
plt.show()
```

Mostrar salida oculta

```python
# Métricas del árbol
y_pred = dt.predict(X_test)
y_proba = dt.predict_proba(X_test)
print("=== RENDIMIENTO DEL ÁRBOL DE DECISIÓN ===")
print(f"Exactitud (Accuracy): {accuracy_score(y_test, y_pred):.4f}")
print("\nReporte de clasificación:")
print(classification_report(
    y_test, y_pred,
    target_names=['ACUICOLA', 'PECUARIO', 'QUIMICO', 'VETERINARIO']
))
```

```
=== RENDIMIENTO DEL ÁRBOL DE DECISIÓN ===
Exactitud (Accuracy): 0.5658

Reporte de clasificación:
              precision    recall  f1-score   support

    ACUICOLA       0.60      0.53      0.56        17
    PECUARIO       0.83      0.56      0.67        27
     QUIMICO       0.58      0.60      0.59        25
 VETERINARIO       0.24      0.57      0.33         7

    accuracy                           0.57        76
   macro avg       0.56      0.56      0.54        76
weighted avg       0.64      0.57      0.59        76
```

```python
#Predicciones del árbol de decisión
y_pred = dt.predict(X_test)
y_proba = dt.predict_proba(X_test)
```

```python
#Matriz de confusión del árbol
cm = confusion_matrix(y_test, y_pred)
fig, ax = plt.subplots(figsize=(7, 6))
im = ax.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
ax.figure.colorbar(im, ax=ax)
ax.set(
    xticks=np.arange(cm.shape[1]),
    yticks=np.arange(cm.shape[0]),
    xticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO', 'VETERINARIO'],
    yticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO', 'VETERINARIO'],
    title="Matriz de Confusión del Árbol de Decisión",
    ylabel="Verdadero",
    xlabel="Predicho"
)
plt.setp(ax.get_xticklabels(), rotation=45, ha="right")
thresh = cm.max() / 2.
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        ax.text(j, i, format(cm[i, j], 'd'),
                ha="center", va="center",
                color="white" if cm[i, j] > thresh else "black")
plt.tight_layout()
plt.show()
```

```python
# Curva ROC (One-vs-Rest) del Árbol de Decisión para multiclass
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

class_names = ['ACUICOLA', 'PECUARIO', 'QUIMICO', 'VETERINARIO']
plt.figure(figsize=(10, 8))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--', label='Random (AUC = 0.50)')
for i, class_label in enumerate(class_names):
    y_test_binarized = (y_test == (i + 1)).astype(int)
    y_proba_class = y_proba[:, i]
    fpr, tpr, _ = roc_curve(y_test_binarized, y_proba_class)
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, lw=2, label=f'ROC curve {class_label} (AUC = {roc_auc:.2f})')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Curva ROC One-vs-Rest del Árbol de Decisión')
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```

Mostrar salida oculta

```python
#Random Forest (validación robusta)
rf = RandomForestClassifier(
    n_estimators=200,
    max_depth=6,
    min_samples_leaf=5,
    random_state=42,
    class_weight='balanced'
)
rf.fit(X_train, y_train)
```

```
RandomForestClassifier(class_weight='balanced', max_depth=6, min_samples_leaf=5,
                       n_estimators=200, random_state=42)
```

```python
# Predicciones del random forest
y_predRf = rf.predict(X_test)
y_probaRf = rf.predict_proba(X_test)
```

```python
# Métricas del random forest
print("=== RENDIMIENTO DEL RANDOM FOREST ===")
print(f"Exactitud (Accuracy): {accuracy_score(y_test, y_predRf):.4f}")
print("\nReporte de clasificación:")
print(classification_report(
    y_test, y_predRf,
    target_names=['ACUICOLA', 'PECUARIO', 'QUIMICO', 'VETERINARIO']
))
```

```
=== RENDIMIENTO DEL RANDOM FOREST ===
Exactitud (Accuracy): 0.6184

Reporte de clasificación:
              precision    recall  f1-score   support

    ACUICOLA       0.44      0.47      0.46        17
    PECUARIO       0.89      0.63      0.74        27
     QUIMICO       0.63      0.76      0.69        25
 VETERINARIO       0.33      0.43      0.38         7

    accuracy                           0.62        76
   macro avg       0.58      0.57      0.57        76
weighted avg       0.66      0.62      0.63        76
```

```python
# Matriz de confusión del random forest
cmRf = confusion_matrix(y_test, y_predRf)
fig, ax = plt.subplots(figsize=(7, 6))
im = ax.imshow(cmRf, interpolation='nearest', cmap=plt.cm.Blues)
ax.figure.colorbar(im, ax=ax)
ax.set(
    xticks=np.arange(cmRf.shape[1]),
    yticks=np.arange(cmRf.shape[0]),
    xticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO', 'VETERINARIO'],
    yticklabels=['ACUICOLA', 'PECUARIO', 'QUIMICO', 'VETERINARIO'],
    title="Matriz de Confusión del Random Forest",
    ylabel="Verdadero",
    xlabel="Predicho"
)
plt.setp(ax.get_xticklabels(), rotation=45, ha="right")
thresh = cmRf.max() / 2.
for i in range(cmRf.shape[0]):
    for j in range(cmRf.shape[1]):
        ax.text(j, i, format(cmRf[i, j], 'd'),
                ha="center", va="center",
                color="white" if cmRf[i, j] > thresh else "black")
plt.tight_layout()
plt.show()
```

Mostrar salida oculta

```python
#Curva ROC del Random Forest
plt.figure(figsize=(10, 8))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--', label='Random (AUC = 0.50)')
for i, class_label in enumerate(class_names):
    y_test_binarized = (y_test == (i + 1)).astype(int)
    y_proba_class = y_probaRf[:, i]
    fpr, tpr, _ = roc_curve(y_test_binarized, y_proba_class) # Corrected: used roc_curve instead of roc_auc directly
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr, tpr, lw=2, label=f'ROC curve {class_label} (AUC = {roc_auc:.2f})') # Fixed f-string
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Curva ROC del Random Forest')
    plt.legend(loc="lower right")
    plt.grid(True)
plt.show()
```

Mostrar salida oculta

```python
# Predecir sobre todo el conjunto con el Random forest
y_full_pred = rf.predict(X)
y_full_proba = rf.predict_proba(X)
```

```python
# Agregar probabilidad de la clase real
prob_real = [probas[true_label - 1] for probas, true_label in zip(y_full_proba, y)]
```

+ Código + Texto

```python
# Identificar registros con baja certeza (< 30% de probabilidad en su clase real)
df_analysis = df.copy()
df_analysis['SECTOR_PRED'] = y_full_pred
df_analysis['PROB_REAL'] = prob_real
sospechosos = df_analysis[df_analysis['PROB_REAL'] < 0.3]

print(f"\n=== REGISTROS SOSPECHOSOS (baja certeza en clase real) ===")
print(f"Total encontrados: {len(sospechosos)}")
if len(sospechosos) > 0:
    sospechosos_display = sospechosos[['SECTOR DESTINO', 'SECTOR_PRED', 'PROB_REAL']].copy()
    sospechosos_display['SECTOR DESTINO'] = sospechosos_display['SECTOR DESTINO'].map(sector_map)
    sospechosos_display['SECTOR_PRED'] = sospechosos_display['SECTOR_PRED'].map(sector_map)
    print(sospechosos_display.head(10))
```

Mostrar salida oculta

```python
#Importancia de variables (Random Forest)

importances = rf.feature_importances_
indices = np.argsort(importances)[::-1]
plt.figure(figsize=(10, 6))
plt.title("Importancia de Variables (Random Forest)")
plt.bar(range(X.shape[1]), importances[indices], color="steelblue", align="center")
plt.xticks(range(X.shape[1]), [X.columns[i] for i in indices], rotation=90)
importances_formatted = [f'{x:.1%}' for x in importances[indices]]
for i, v in enumerate(importances_formatted):
    plt.text(i, importances[indices[i]] + 0.0005, v, ha='center', va='bottom', fontsize=10)
plt.xlim([-1, X.shape[1]])
plt.tight_layout()
plt.show()
```

Mostrar salida oculta

```python
df_analysis
```

Mostrar salida oculta

DECLARACIÓN Y AUTORIZACIÓN

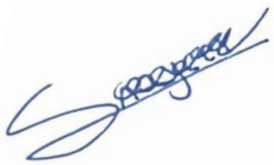
Nosotros, **Campoverde Cedeño, Snaycer Rodrigo**, con C.C: # **0706189420** y **Miranda Yunda, Alisson Melina** C.C: # **0932341345, autor**/a del trabajo de titulación: **Implementación de un algoritmo de clasificación para la asignación del sector destino en importaciones del mercado ecuatoriano,** previo a la obtención del título de **Licenciado en Negocios Internacionales** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, **12 del mes de febrero del año 2026**

AUTORES:



Campoverde Cedeño, Snaycer Rodrigo
C.C: # 0706189420

Miranda Yunda, Alisson Melina
C.C: # 0932341345

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TEMA Y SUBTEMA:	**Implementación de un algoritmo de clasificación para la asignación del sector destino en importaciones del mercado ecuatoriano.**		
AUTOR(ES)	**Campoverde Cedeño, Snaycer Rodrigo** **Miranda Yunda, Alisson Melina**		
REVISOR(ES)/TUTOR(ES)	**Carrera Buri, Félix Miguel**		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	**Economía y Empresa**		
CARRERA:	**Negocios Internacionales**		
TITULO OBTENIDO:	**Licenciados en Negocios Internacionales**		
FECHA DE PUBLICACIÓN:	**12** de **febrero** de **2026**	**No. DE PÁGINAS:**	132 p.
ÁREAS TEMÁTICAS:	**Análisis de datos, Importaciones, Mercados, Producción acuícola, Producción pecuaria.**		
PALABRAS CLAVES/ KEYWORDS:	**Aprendizaje automático, Clasificación, Random Forest, Importaciones, Sector destino, Comercio exterior, Datos históricos, Árbol de Decisión, Evaluación, Predicción.**		

RESUMEN/ABSTRACT (150-250 palabras):

La presente investigación tuvo como objetivo desarrollar y evaluar un modelo de clasificación basado en técnicas de aprendizaje automático para la asignación del sector destino de las importaciones del mercado ecuatoriano. El estudio se enmarca dentro de un enfoque cuantitativo de tipo categórico, al centrarse en la clasificación de registros en categorías específicas (Acuícola, Pecuario, Químico y Veterinario) a partir del análisis de datos históricos provenientes de la plataforma COBUS y de información interna empresarial. Metodológicamente, se realizó una revisión de literatura sobre los principales enfoques de machine learning aplicables a problemas de clasificación, destacando los métodos basados en árboles. Posteriormente, se llevó a cabo la preparación y depuración de la base de datos, garantizando su calidad, coherencia y consistencia mediante procesos de limpieza, eliminación de variables irrelevantes y estandarización de formatos. El modelo principal implementado fue Random Forest, complementado con un modelo de Árbol de Decisión para fines comparativos. Se dividió el conjunto de datos en 75 % para entrenamiento y 25 % para prueba, utilizando estratificación para mantener la proporción de clases. La evaluación del desempeño se realizó mediante métricas de exactitud, reporte de clasificación, matriz de confusión y curva ROC multiclase, lo que permitió medir la capacidad discriminativa del modelo. Los resultados evidenciaron que el modelo Random Forest presentó mayor estabilidad y precisión en la clasificación del sector destino en comparación con el Árbol de Decisión individual. Asimismo, el análisis de importancia de variables permitió identificar los factores más influyentes en el proceso de clasificación, mientras que la detección de registros con baja probabilidad aportó información relevante para el control de calidad de los datos.

ADJUNTO PDF:	☒ SI	☐ NO
CONTACTO CON AUTOR/ES:	**Teléfono:** +593-4	E-mail: **snaycer.campoverde@cu.ucsg.edu.ec** Alisson.yunda@cu.ucsg.edu.ec
CONTACTO CON LA INSTITUCIÓN (C00RDINADOR DEL PROCESO UTE)::	**Nombre: Freire Quintero Cesar Enrique**	
	Teléfono: +593 990090702	
	E-mail: cesar.freire@cu.ucsg.edu.ec	

SECCIÓN PARA USO DE BIBLIOTECA	
Nº. DE REGISTRO (en base a datos):	
Nº. DE CLASIFICACIÓN:	
DIRECCIÓN URL (tesis en la web):	