Universidad Europea VALENCIA

GRADO EN CIENCIA DE DATOS

Trabajo Fin de Grado: Creación de una Herramienta de ML para el cálculo del PVP de

Presentado por:

Miguel García Zanón

Dirigido por:

Jordi Joan Huguet

CURSO ACADÉMICO 2023-2024

ÍNDICE

1. Contexto al trabajo realizado

Previó al planteamiento de este trabajo tuve el acercamiento de la empresa Láser Valencia, con la que ya había tenido relación previa. Esta empresa tenía la intención de realizar una herramienta que pudiese calcular el precio de venta al público (PVP a partir de ahora) con inteligencia artificial. Debido a mi relación con la empresa se me propuso este proyecto.

Como principal objetivo se me pide la creación de la herramienta, pero no de menor importancia comprobar la viabilidad de la realización de una con un histórico de datos ya recogido por la empresa durante algunos años.

Para la realización del proyecto voy a utilizar mis conocimientos de Machine Learning (ML a partir de ahora), mis conocimientos de tratamiento de datos y mis conocimientos sobre la empresa y el sector principalmente, complementados por diversas fuentes aportadas en la bibliografía y los Anexos.

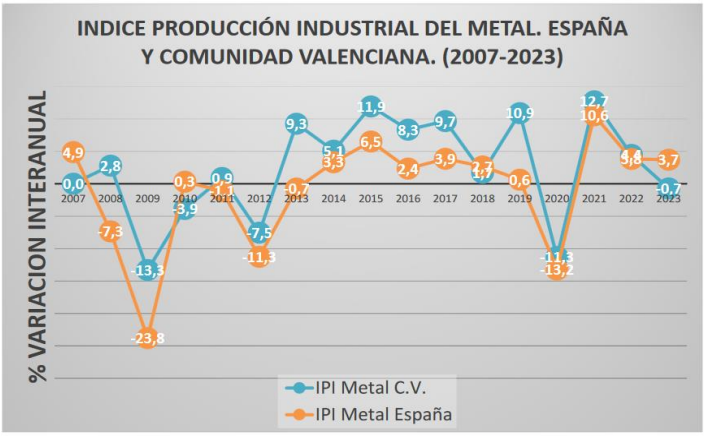
La documentación del proyecto (este documento), va a estar dividida en cinco bloques principales; un primer bloque en el cual se va a hacer una introducción al trabajo, un segundo bloque en el cual se va a comentar los conceptos del trabajo y todo lo utilizado en forma de marco teórico y metodología, un tercer bloque en el cual se va a explicar el desarrollo e implementación de la herramienta, un cuarto bloque en el cual se va a analizar los resultados y se hará una discusión de estos y del proyecto en general y una última parte en la cual se va a hacer una conclusión del proyecto.

2. INTRODUCCIÓN

2.1 Contextualización del tema

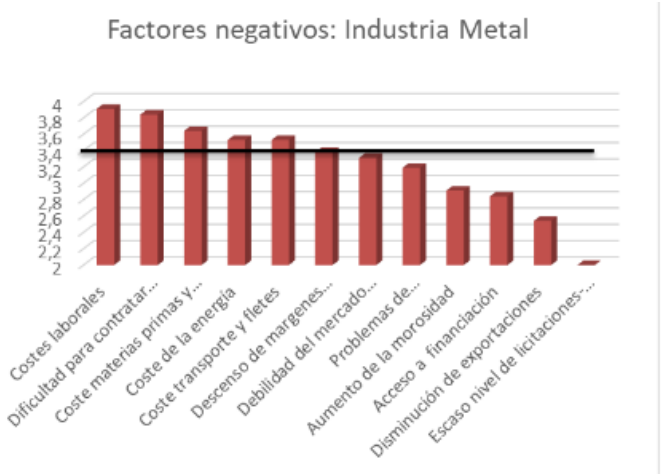
2.1.1 Sector Metalúrgico

La industria metalúrgica ha sido durante mucho tiempo un pilar fundamental de la economía global, desempeñando un papel crucial en el desarrollo industrial y económico de muchas regiones, incluida la Comunidad Valenciana.



Fuente: Elaboración a partir de IVE (Instituto Valenciano de Estadística) y CONFEMETAL. Promedios trimestrales de IPIMET (Índice de Producción Industrial del Metal).

Los principales desafíos que afectan a la industria metalúrgica en la Comunidad Valenciana incluyen los altos costos laborales, la dificultad para contratar personal cualificado, y los elevados costos de materias primas y energía. Estos factores han contribuido a una atmósfera de incertidumbre y han afectado negativamente las expectativas de negocio de las empresas del sector para los próximos años. Además, los costos de transporte y fletes también representan una carga significativa para las empresas, aumentando la presión sobre sus márgenes operativos.



A pesar de estos desafíos, la industria metalúrgica sigue siendo de vital importancia para el desarrollo económico de la Comunidad Valenciana y de España en general. El sector no solo representa una proporción significativa del empleo industrial en la región, sino que también es un componente clave del PIB manufacturero.

Por lo tanto, la capacidad de las empresas metalúrgicas para innovar y adaptarse a las cambiantes condiciones del mercado es crucial para mantener su competitividad y contribuir al crecimiento económico.

En este contexto, las empresas del sector están explorando nuevas tecnologías y métodos para mejorar su eficiencia operativa y reducir costos. La digitalización y la adopción de tecnologías avanzadas, como el machine learning (ML), son vistas como soluciones prometedoras para abordar estos desafíos. Al automatizar y optimizar procesos clave, como la estimación de costos y la gestión de la cadena de suministro, las empresas metalúrgicas pueden mejorar su capacidad de respuesta a las demandas del mercado y aumentar su competitividad.

2.1.2 Láser Valencia

Láser Valencia SL, fundada en 2007, es una empresa dedicada al ámbito de la transformación y tratamiento del metal en España. La empresa se especializa en el corte por láser de chapa en 2D, una tecnología que permite la fabricación de piezas metálicas con alta precisión y eficiencia.

Láser Valencia se centra en el B2B (Business to Business) ya que la mayoría de sus clientes son creadores de producto final para el que necesitan piezas especificas generalmente en gran cantidad. La empresa trabaja con una variedad de materiales, incluyendo acero inoxidable, aluminio y aceros especiales, y es capaz de fabricar desde pequeñas piezas de precisión hasta grandes piezas metálicas.

Uno de los mayores desafíos que enfrenta Láser Valencia es la estimación precisa y eficiente de los costos de producción. El cálculo del precio de venta al público (PVP) es fundamental para asegurar la rentabilidad de cada proyecto y para mantener la competitividad en el mercado. Tradicionalmente, este proceso ha sido manual y dependiente de múltiples factores, como los costos de los materiales, la mano de obra, los tiempos de producción y otros gastos indirectos. Estos cálculos manuales no solo son propensos a errores, sino que también consumen tiempo valioso, lo que puede retrasar las respuestas a los clientes y afectar la capacidad de la empresa para cerrar ventas de manera oportuna.

Para abordar este desafío, Láser Valencia ha tenido la idea de implementar una herramienta basada en tecnología de machine learning. Esta herramienta le permitiría automatizar el cálculo del costo de cada pedido, mejorando dramáticamente la precisión y la velocidad de los presupuestos.

2.1.3 Introducción al Machine Learning en la Industria

El machine learning (ML) es una rama de la inteligencia artificial que se enfoca en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y hacer predicciones o decisiones basadas en datos.

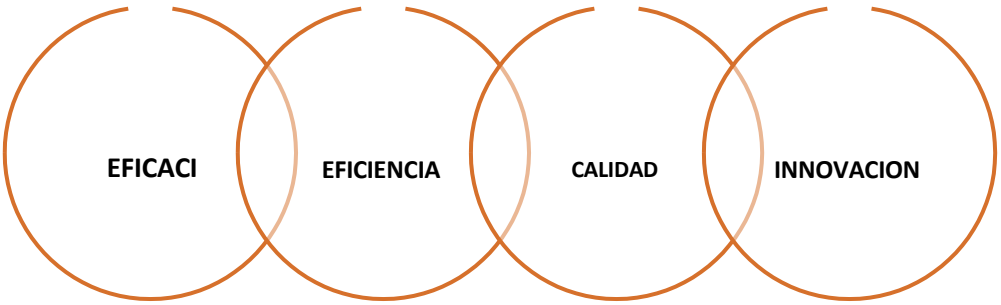
Actualmente a nivel nacional y principalmente regional el ML no se utiliza con frecuencia en la industria, lo que si sucede en otros países. Esta clase de herramientas se utilizan para entre otras cosas el mantenimiento predictivo, el control de calidad, la optimización de recursos o el cálculo de precio/costos. Por estas razones Láser Valencia ha tenido la idea de implementar ML en el cálculo del PVP siendo algo innovador a nivel local y nacional en el sector.

2.2 Justificación de la importancia del tema dentro del campo profesional.

La relevancia de implementar tecnologías de Machine Learning (ML) en el cálculo de costos en una empresa como Láser Valencia SL trasciende la mera mejora operativa; representa una evolución en la manera en que las empresas industriales pueden responder a las exigencias de un mercado competitivo. En este contexto, la justificación de la integración de ML en los procesos de Láser Valencia SL se asienta en varios pilares fundamentales que subrayan la importancia del tema dentro del campo profesional.

Eficiencia Operativa

La adopción de una herramienta de ML para la estimación de costos se traduce directamente en un incremento de la eficiencia operativa. Los procesos manuales, tradicionalmente lentos y susceptibles de errores, pueden ser significativamente optimizados mediante la automatización. Esto no solo acelera el tiempo de respuesta en la elaboración de presupuestos, sino que también minimiza los errores humanos en los cálculos, asegurando una mayor precisión en las estimaciones. La eficiencia mejorada libera recursos valiosos que pueden ser redirigidos hacia actividades de mayor valor agregado, como el desarrollo de nuevos productos o la mejora del servicio al cliente. Al automatizar el cálculo de costos, Láser Valencia puede garantizar que sus estimaciones sean consistentes y precisas, reduciendo así la probabilidad de errores que podrían resultar en pérdidas financieras o insatisfacción del cliente.



Competitividad de Mercado

En un sector donde la capacidad para ofrecer respuestas rápidas y precisas puede ser un diferenciador clave, la velocidad y la exactitud proporcionadas por un modelo de ML mejoran la posición competitiva de Láser Valencia SL. Hasta el momento, muchos competidores importantes no han satisfecho esta necesidad, lo que presenta una oportunidad para que Láser Valencia se destaque. Los clientes potenciales valoran favorablemente a los proveedores que pueden ofrecer cotizaciones rápidas y fiables, lo que puede ser decisivo en la captación y retención de clientes.

Innovación Tecnológica

La implementación de tecnologías avanzadas como el ML en procesos tradicionalmente manuales no solo mejora esos procesos, sino que también posiciona a la empresa como líder en innovación dentro de su sector. Este liderazgo tecnológico puede atraer no solo a clientes sino también a talentos interesados en trabajar en una empresa que invierte en tecnología y desarrollo.

Sostenibilidad Financiera

Los modelos diseñados para optimizar los cálculos de costos y precios no solo reducen la posibilidad de errores que pueden resultar en pérdidas financieras, sino que también ayudan a modelar escenarios de costeo más eficientes y sostenibles. Al entender mejor los patrones en los datos históricos y actuales, la empresa puede identificar áreas de ahorro potencial y ajustar sus prácticas de precios para maximizar los márgenes de beneficio.

2.3 Objetivos del TFG

El Trabajo de Fin de Grado (TFG) se enmarca en la necesidad de Láser Valencia SL de mejorar su eficiencia operativa mediante la implementación de una herramienta basada en machine learning para la estimación de costos. Los objetivos específicos del TFG se detallan a continuación:

Objetivos Principales: Estudio de la viabilidad del desarrollo de una herramienta de ML para el cálculo de PVP con el histórico de datos de Láser Valencia y desarrollo de una herramienta de machine learning que permita calcular de manera automática y precisa el precio de venta al público (PVP) para Láser Valencia SL.

Para alcanzar estos objetivos se han seguido los siguientes pasos:

1- Reuniones con Láser Valencia SL para conocer sus necesidades y objetivos con el proyecto.
2- Reunión con la empresa para la obtención de los datos y explicación de estos.
3- Investigación de las posibles herramientas de ML a utilizar.
4- Planificación de la herramienta a utilizar.
5- Limpieza de datos.
6- EDA (Análisis exploratorio de datos).
7- Elección del modelo óptimo.
8- Entrenamiento del modelo de ML.
9- Análisis de resultados.
10- Discusión de los resultados con la empresa.

3. Marco Teórico y Metodología

3.1 Fundamentos de Machine Learning

El machine learning (ML) es una rama de la inteligencia artificial que se centra en el desarrollo de sistemas capaces de aprender a partir de datos, identificar patrones y tomar decisiones con mínima intervención humana. Los fundamentos de ML incluyen varios tipos de aprendizaje, principalmente supervisado, no supervisado y por refuerzo. En el aprendizaje supervisado, el modelo se entrena usando un conjunto de datos etiquetados, lo que permite al algoritmo aprender a predecir resultados a partir de nuevas entradas. Por otro lado, el aprendizaje no supervisado trabaja con datos no etiquetados, buscando estructuras ocultas en estos datos. Finalmente, el aprendizaje por refuerzo se basa en aprender políticas de acción mediante recompensas y penalizaciones.

3.2 Modelos de Machine Learning probados

Para abordar el desafío de estimar el costo de producción de las piezas en Láser Valencia SL, se evaluarán varios algoritmos de machine learning. Cada uno de estos métodos tiene características únicas que podrían hacerlos más o menos adecuados dependiendo de la naturaleza específica y la complejidad de los datos disponibles. Aquí se ofrece una descripción más detallada de cada método seleccionado:

Regresión Lineal Múltiple

Descripción: La regresión lineal múltiple es una extensión de la regresión lineal simple que permite modelar la relación entre una variable dependiente y múltiples variables independientes. Este modelo asume una relación lineal entre las variables y utiliza los mínimos cuadrados ordinarios para ajustar la mejor línea a los datos.

Ventajas: La regresión lineal múltiple es apreciada por su simplicidad y facilidad de interpretación. Dado que los coeficientes indican la relación directa entre cada variable independiente y la variable dependiente, es sencillo entender cómo cada variable influye en el resultado. Además, es un modelo eficiente en términos de tiempo de entrenamiento, incluso con grandes conjuntos de datos, lo que lo hace accesible y rápido para implementar. Esta transparencia y rapidez lo convierten en una herramienta fundamental para análisis exploratorios y explicativos.

Desventajas: Una de las principales desventajas de la regresión lineal múltiple es que asume una relación lineal entre las variables, lo cual puede no siempre ser cierto en escenarios del mundo real. Además, es muy sensible a la multicolinealidad, donde las variables independientes están altamente correlacionadas entre sí, lo que puede distorsionar los resultados y dificultar la interpretación de los coeficientes. También es vulnerable a la influencia de outliers, que pueden afectar significativamente la precisión del modelo.

Ridge Regression

Descripción: Ridge Regression es una técnica de regularización que modifica la regresión lineal al añadir un término de penalización al modelo de mínimos cuadrados ordinarios. Esta penalización es proporcional al cuadrado de la magnitud de los coeficientes, lo que ayuda a reducir el riesgo de sobreajuste.

Ventajas: Ridge Regression es especialmente útil para manejar problemas de multicolinealidad en los datos, ya que la penalización ayuda a reducir la varianza de los coeficientes estimados. Esto hace que los modelos sean más estables y menos sensibles a pequeñas variaciones en los datos de entrenamiento. En comparación con la regresión lineal múltiple, ofrece una mayor precisión en escenarios con muchas variables altamente correlacionadas.

Desventajas: Una de las principales desventajas de Ridge Regression es la complejidad añadida en la interpretación de los coeficientes debido al término de penalización. Mientras que, en la regresión lineal estándar, los coeficientes representan la influencia directa de cada variable, en Ridge Regression, esta interpretación se vuelve menos intuitiva. Además, requiere un ajuste cuidadoso del hiperparámetro lambda, lo que puede complicar el proceso de modelado. Si lambda no se selecciona adecuadamente, el modelo puede seguir sobreajustando los datos o, por el contrario, subajustarlos, disminuyendo su precisión.

Elastic Net

Descripción: Elastic Net combina las técnicas de Ridge y Lasso Regression para mejorar el rendimiento en conjuntos de datos con alta multicolinealidad. Este modelo utiliza una penalización que es una combinación lineal de los términos L1 (Lasso) y L2 (Ridge).

Ventajas: Elastic Net ofrece una gran flexibilidad al combinar los beneficios de la regularización L1 y L2, lo que permite realizar tanto la selección de variables como la regularización simultáneamente. Esto es particularmente útil en situaciones donde hay muchas variables altamente correlacionadas, ya que puede seleccionar un subconjunto de variables más relevante mientras maneja la multicolinealidad.

Desventajas: A pesar de sus beneficios, Elastic Net puede ser complejo de implementar debido a la necesidad de ajustar múltiples hiperparámetros, lo cual requiere un proceso de validación cruzada exhaustivo para encontrar los valores óptimos. Esta necesidad de ajuste puede ser computacionalmente costosa y consumir mucho tiempo, especialmente con grandes conjuntos de datos. Además, la interpretación de los coeficientes puede ser más difícil debido a la combinación de penalizaciones, lo que puede complicar la comprensión de la influencia de cada variable en el modelo.

Árbol de Decisión

Descripción: Un árbol de decisión es un modelo predictivo que utiliza un diagrama de decisiones y sus posibles consecuencias. Es un modelo gráfico que divide los datos en subconjuntos basados en una serie de pruebas de características, organizadas en nodos y ramas. Cada nodo representa una prueba de una característica, cada rama representa el resultado de la prueba, y cada hoja representa una predicción de la variable objetivo.

Ventajas: Los árboles de decisión son extremadamente fáciles de interpretar y visualizar, lo que los hace útiles para explicar las decisiones del modelo a personas no técnicas. No requieren escalado de datos y pueden manejar tanto variables categóricas como continuas. Además, son rápidos de entrenar y realizar predicciones, lo que los convierte en una opción eficiente para modelos iniciales y análisis exploratorios. La estructura de los árboles también permite capturar fácilmente interacciones no lineales entre las características.

Desventajas: A pesar de su simplicidad, los árboles de decisión son propensos al sobreajuste, especialmente si se permiten crecer sin restricciones. Los árboles complejos pueden ajustarse demasiado a los datos de entrenamiento, perdiendo la capacidad de generalizar a nuevos datos. Además, son sensibles a pequeñas variaciones en los datos; un pequeño cambio puede resultar en un árbol completamente diferente. Aunque las técnicas de poda pueden mitigar este problema, encontrar el balance adecuado puede ser complicado.

Random Forest

Descripción: Random Forest es un método de ensamble que utiliza múltiples árboles de decisión para mejorar la precisión del modelo y reducir el riesgo de sobreajuste. Cada árbol en el bosque se entrena con una muestra diferente del conjunto de datos, seleccionada mediante muestreo con reemplazo (bootstrap), y las predicciones finales se obtienen mediante votación en el caso de clasificación o promediado en el caso de regresión. Este enfoque no solo mejora la precisión, sino que también proporciona una estimación de la importancia de las variables.

Ventajas: Random Forest se destaca por su alta precisión y robustez. Al combinar las predicciones de múltiples árboles, este método reduce significativamente la varianza del modelo, lo que lleva a una menor probabilidad de sobreajuste y a mejores capacidades de generalización. Es capaz de manejar grandes cantidades de datos y muchas características sin requerir mucha preparación de datos, como la normalización. Además, su capacidad para proporcionar estimaciones de la importancia de las variables es extremadamente útil para la comprensión y la interpretación de los resultados del modelo.

Desventajas: A pesar de sus numerosas ventajas, Random Forest puede ser complejo y costoso en términos computacionales. El entrenamiento y la predicción pueden ser considerablemente más lentos que los de modelos más simples debido a la necesidad de construir y evaluar múltiples árboles. Además, la interpretación de los modelos de ensamble, como Random Forest, es más difícil que la de modelos individuales, ya que la combinación de muchos árboles puede ocultar las relaciones simples entre las variables.

K-Nearest Neighbors (K-NN)

Descripción: El algoritmo K-Nearest Neighbors (K-NN) es un método de clasificación y regresión basado en instancias, que no construye explícitamente un modelo, sino que almacena las instancias de los datos de entrenamiento. La clasificación de una nueva instancia se realiza identificando los k puntos de datos más cercanos en el espacio de características y asignando la clase mayoritaria entre estos vecinos. Para la regresión, la predicción se basa en el promedio de los valores de los k vecinos más cercanos.

Ventajas: K-NN es muy simple de entender e implementar, lo que lo hace accesible para los principiantes en ciencia de datos. No hace suposiciones sobre la distribución de los datos, lo que le permite ser aplicable a una amplia variedad de problemas. Además, es un método versátil que puede ser usado tanto para problemas de clasificación como de regresión. La capacidad de adaptarse a cualquier forma de la frontera de decisión hace que K-NN sea efectivo en detectar patrones complejos cuando se utilizan suficientes vecinos.

Desventajas: Una de las desventajas más significativas de K-NN es su ineficiencia computacional, ya que requiere almacenar todos los datos de entrenamiento y calcular las distancias a todos los puntos para cada predicción. Esto puede ser prohibitivo en términos de tiempo y memoria para grandes conjuntos de datos. Además, K-NN es sensible a la escala de los datos, por lo que es esencial normalizar las características para obtener resultados precisos. También puede ser afectado por la elección del valor de k, que requiere ajuste cuidadoso para evitar problemas de sobreajuste (si k es muy pequeño) o sobreajuste (si k es demasiado grande).

Bagging (Bootstrap Aggregating)

Descripción: Bagging (Bootstrap Aggregating), es una técnica de ensamble que mejora la estabilidad y precisión de los modelos de aprendizaje automático al reducir la varianza. En bagging, se crean múltiples subconjuntos de datos de entrenamiento mediante muestreo con reemplazo (bootstrap). Cada subconjunto se utiliza para entrenar un modelo independiente, y las predicciones finales se obtienen promediando (para regresión) o votando (para clasificación) las predicciones de todos los modelos.

Ventajas: Bagging es altamente efectivo para reducir la varianza de los modelos base, lo que mejora la capacidad de generalización y reduce el riesgo de sobreajuste. Esta técnica es particularmente beneficiosa cuando se utilizan modelos de alto sesgo, como árboles de decisión, que tienden a ser inestables pero precisos. Al promediar múltiples modelos, bagging produce resultados más robustos y estables. Además, el paralelismo en el entrenamiento de los modelos individuales puede ser explotado para reducir el tiempo de entrenamiento en entornos computacionales adecuados.

Desventajas: El principal inconveniente de bagging es su complejidad computacional y su necesidad de recursos. Entrenar múltiples modelos y almacenar múltiples subconjuntos de datos puede ser intensivo en términos de tiempo y memoria. Además, la interpretación de los resultados del ensamble puede ser más difícil que la de un modelo individual, ya que no es claro cómo cada modelo contribuye a la predicción final. La complejidad adicional también implica un esfuerzo adicional en la implementación y el ajuste de los hiperparámetros.

LightGBM (Light Gradient Boosting Machine)

Descripción: LightGBM es una implementación optimizada del algoritmo de boosting basado en árboles de decisión, diseñada para ser eficiente en términos de velocidad y uso de memoria. Utiliza técnicas avanzadas como el muestreo basado en histogramas y la reducción de datos para mejorar el rendimiento y la precisión. LightGBM divide los datos en hojas de hoja en lugar de niveles de nivel, lo que permite manejar mejor las características categóricas y los datos grandes.

Ventajas: LightGBM es conocido por su velocidad y eficiencia en el entrenamiento y la predicción, siendo capaz de manejar grandes conjuntos de datos con alta dimensionalidad. Su uso de técnicas avanzadas, como el muestreo basado en histogramas, reduce significativamente el tiempo de entrenamiento sin comprometer la precisión. Además, es capaz de manejar características categóricas de manera nativa, lo que simplifica la preparación de datos. La alta precisión de LightGBM en comparación con otros algoritmos de boosting lo hace ideal para competiciones y aplicaciones industriales.

Desventajas: A pesar de sus ventajas, LightGBM requiere un ajuste cuidadoso de los hiperparámetros para obtener el mejor rendimiento, lo que puede ser complicado y consumir mucho tiempo. La complejidad del modelo también puede dificultar la interpretación de los resultados, ya que el ensamble de muchos árboles y las técnicas avanzadas utilizadas no son intuitivos. Además, debido a su enfoque en la eficiencia, LightGBM puede ser menos robusto a valores atípicos y puede requerir una limpieza y preprocesamiento de datos más exhaustivos.

.

4. Proceso de la creación de la herramienta

En este apartado se describen todos los pasos y apartados tenidos en cuenta para la creación de la herramienta de Machine Learning para el cálculo de PVP para la empresa Láser Valencia.

4.1 Pasos previos

La realización de este proyecto se inició con varios acercamientos y reuniones con la empresa Láser Valencia SL para definir claramente los objetivos y comprender a fondo sus necesidades. Durante estas reuniones se establecieron los dos objetivos principales: el desarrollo de una herramienta de machine learning para calcular automáticamente el precio de venta al público (PVP) y evaluar la viabilidad de esta herramienta utilizando el histórico de datos proporcionado por la empresa.

Definición del Proyecto y Recolección de Requisitos

Reuniones Iniciales: En las primeras reuniones con los representantes de Láser Valencia SL, se discutieron los retos específicos que enfrentaba la empresa en relación con la estimación de costos y la fijación de precios. Se identificó que el proceso manual existente no solo era lento y propenso a errores, sino que también dificultaba la competitividad de la empresa en un mercado dinámico. Se establecieron los requisitos clave para la herramienta de machine learning, destacando la necesidad de precisión, rapidez y consistencia en las estimaciones.

Objetivos Específicos:

1. **Desarrollo de la Herramienta de ML:**
    - Crear una herramienta capaz de calcular el PVP de manera automática utilizando técnicas de machine learning.
    - Optimizar la herramienta para que pueda manejar el volumen y la complejidad de los datos históricos de Láser Valencia SL.
2. **Evaluación de la Viabilidad:**
    - Analizar la viabilidad de la herramienta basada en el desempeño del modelo de machine learning con los datos históricos.
    - Comparar los resultados obtenidos con los métodos tradicionales de estimación de costos utilizados por la empresa.

Elección del Lenguaje de Programación y Herramientas

Lenguaje de Programación: Se decidió utilizar Python para el desarrollo de la herramienta debido a su amplia adopción en la comunidad de ciencia de datos y machine learning. Python ofrece una extensa gama de bibliotecas y frameworks que facilitan la implementación y el ajuste de modelos de machine learning. Además, su facilidad de uso y flexibilidad lo convierten en una opción ideal para proyectos de este tipo.

Bibliotecas y Frameworks: Las principales bibliotecas y herramientas utilizadas en el proyecto incluyen:

Pandas: Para la Manipulación y Análisis de Datos

Pandas es una biblioteca esencial para la manipulación y el análisis de datos en Python. Ofrece estructuras de datos de alto rendimiento y herramientas de análisis que facilitan el manejo de datos tabulares. En este proyecto, Pandas se utilizó para:

- **Carga de Datos:** Pandas permite leer archivos de datos (como Excel, CSV, y SQL) y convertirlos en DataFrames, una estructura de datos bidimensional similar a una tabla que permite un acceso y manipulación eficientes. Esto es útil para cargar grandes conjuntos de datos de manera rápida y eficiente.

- **Limpieza de Datos:** La biblioteca ofrece funciones para detectar y manejar valores faltantes, eliminar duplicados y corregir inconsistencias en los datos. Estas capacidades son fundamentales para preparar los datos para el análisis y el modelado.

- **Transformación de Datos:** Pandas facilita la aplicación de operaciones como filtrado, agregación, y creación de nuevas columnas derivadas de las existentes. Esto permite realizar transformaciones complejas en los datos con pocas líneas de código.

- **Exploración de Datos:** Con funciones integradas para generar estadísticas descriptivas y visualizar la estructura de los datos, Pandas ayuda a los analistas a obtener una comprensión rápida y detallada de los datos.

NumPy: Para Operaciones Numéricas Eficientes

NumPy es la biblioteca fundamental para el cálculo numérico en Python. Proporciona soporte para matrices grandes y multidimensionales junto con una colección de funciones matemáticas de alto nivel para operar con estas matrices. En el contexto de este proyecto, NumPy se utilizó para:

- **Manejo de Datos Numéricos:** Manipulación de arrays y matrices, que son estructuras de datos esenciales para la implementación de algoritmos de machine learning.

- **Optimización del Rendimiento:** Aprovechamiento de funciones vectorizadas que permiten operaciones rápidas y eficientes sobre grandes conjuntos de datos, reduciendo significativamente el tiempo de procesamiento.

Características Clave:

- **Arrays y Matrices:** Estructuras de datos multidimensionales que permiten almacenar y operar con grandes volúmenes de datos numéricos.

- **Funciones Matemáticas:** Operaciones algebraicas y matemáticas de alto nivel, como transformadas de Fourier, álgebra lineal, y generación de números aleatorios.

Scikit-learn: Para la Implementación de Diversos Algoritmos de Machine Learning

Scikit-learn es una biblioteca poderosa y versátil para el aprendizaje automático en Python. Ofrece una amplia gama de algoritmos de machine learning, herramientas para la evaluación del modelo y técnicas de preprocesamiento de datos. En este proyecto, Scikit-learn se utilizó para:

- **Modelos de Machine Learning:** Implementación de algoritmos como Random Forest, Regresión Lineal, Ridge Regression y muchos otros. Estos modelos se entrenan utilizando los datos preprocesados y se evalúan en términos de precisión y capacidad de generalización.

- **Preprocesamiento de Datos:** Escalado, normalización y codificación de características categóricas para preparar los datos para el modelado. Estas técnicas aseguran que todas las características estén en el formato adecuado y contribuyan equitativamente al modelo.

- **Evaluación del Modelo:** Uso de técnicas como la validación cruzada y métricas de evaluación (R^2, MSE) para medir el rendimiento del modelo. Esto permite seleccionar el mejor modelo y ajustar sus parámetros para mejorar su desempeño.

Características Clave:

- **Algoritmos de Machine Learning:** Clasificación, regresión, clustering, y reducción de dimensionalidad.

- **Preprocesamiento:** Técnicas para transformar y normalizar datos, y manejar características categóricas.

- **Evaluación:** Herramientas para la validación cruzada, selección de hiperparámetros, y cálculo de métricas de desempeño.

Matplotlib y Seaborn: Para la Visualización de Datos.

Matplotlib y Seaborn son bibliotecas de Python para la visualización de datos. Permiten crear gráficos y visualizaciones informativas que facilitan la comprensión y el análisis de los datos. En este proyecto, se utilizaron para:

- Visualización de Distribuciones: Crear histogramas para explorar la distribución de las características numéricas. Esto es esencial para identificar outliers y entender la dispersión de los datos.

- Gráficos de Dispersión: Visualizar relaciones entre pares de variables y detectar posibles correlaciones. Los gráficos de dispersión ayudan a identificar patrones y relaciones no lineales entre las variables.

- Mapas de Calor: Mostrar la matriz de correlación entre las características para identificar relaciones fuertes y posibles problemas de multicolinealidad. Los mapas de calor proporcionan una representación visual intuitiva de las correlaciones entre variables.

Características Clave de Matplotlib:

- Gráficos Personalizables: Una amplia variedad de gráficos (línea, dispersión, barra, histograma) con opciones de personalización detalladas.

- Interfaz Intuitiva: API sencilla y flexible para crear visualizaciones rápidamente.

- Compatibilidad: Integración con Jupyter Notebooks y otras herramientas de ciencia de datos.

Características Clave de Seaborn:

- Visualización Estadística: Basado en Matplotlib, Seaborn ofrece visualizaciones de datos estadísticos de alto nivel.

- Temas y Estilos: Estilos de gráficos predeterminados que mejoran la apariencia y la legibilidad de las visualizaciones.

- Facilidad de Uso: Funciones de alto nivel para crear gráficos complejos con una sintaxis mínima.

Jupyter Notebooks: Para el Desarrollo Interactivo y la Documentación del Proceso de Análisis y Modelado

Jupyter Notebooks es una herramienta interactiva que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Es especialmente útil en el ámbito de la ciencia de datos y el aprendizaje automático. En este proyecto, Jupyter Notebooks se utilizó para:

- **Desarrollo Iterativo:** Probar y ajustar el código de manera interactiva, facilitando la experimentación y el desarrollo de modelos. Los notebooks permiten ejecutar código en fragmentos, lo que facilita la identificación y corrección de errores rápidamente.

- **Documentación del Proceso:** Integrar código, visualizaciones y explicaciones en un solo documento, proporcionando una narrativa coherente y fácilmente comprensible del análisis y los resultados. Esto mejora la reproducibilidad del análisis y facilita la revisión por otros.

- **Presentación de Resultados:** Crear informes y presentaciones que pueden ser fácilmente compartidos y revisados por otros miembros del equipo o partes interesadas. Los notebooks pueden exportarse a varios formatos, como HTML o PDF, para facilitar su distribución.

Características Clave:

- Interactividad: Permite ejecutar y modificar código en fragmentos, facilitando la experimentación y el análisis iterativo.

- Integración de Contenido: Combina código, texto, ecuaciones y visualizaciones en un solo documento.

- Compatibilidad: Soporta múltiples lenguajes de programación.

Recolección y Entrega de Datos

Obtención de Datos: Láser Valencia SL proporcionó una base de datos extensa con 834123 registros, que contenía 11 características que determinamos como relevantes para el cálculo del PVP en las primeras reuniones.

Planificación del Proyecto

Fases del Proyecto: El proyecto se estructuró en varias fases para asegurar un desarrollo ordenado y efectivo:

1. **Exploración y Limpieza de Datos:**
    - Identificación y corrección de datos faltantes o incorrectos.
    - Normalización y transformación de características según sea necesario.
2. **Análisis Exploratorio de Datos (EDA):**
    - Visualización de datos para identificar patrones y relaciones.
    - Análisis descriptivo para obtener estadísticas básicas de las características.
3. **Selección del Modelo:**
    - Evaluación de diferentes algoritmos de machine learning.
    - Comparación de desempeño de los diferentes modelos y elección del óptimo.
4. **Entrenamiento y Validación del Modelo:**
    - Entrenamiento del modelo seleccionado con los datos preprocesados eligiendo los parámetros que maximicen la precisión y disminuyan el error.
    - Validación del modelo utilizando un conjunto de datos de prueba separado.
5. **Prueba de la herramienta y presentación de esta a la empresa:**
    - Evaluación del desempeño de la herramienta con un pequeño código de prueba para comprobar su funcionamiento.
    - Presentar la herramienta y los resultados de la herramienta a la empresa para la posterior entrega de esta.

Cronograma

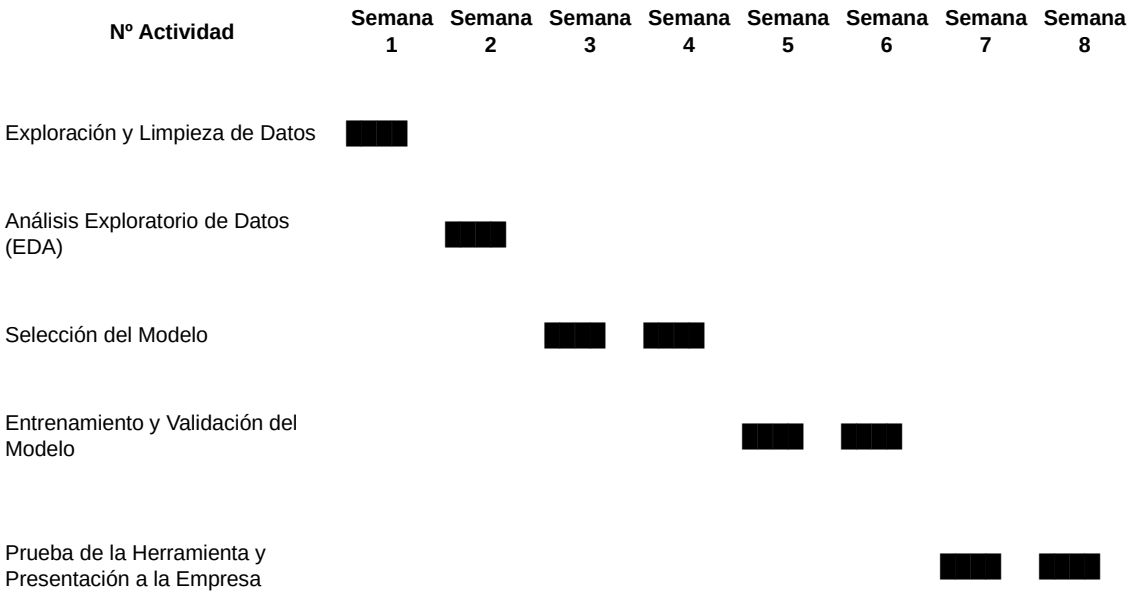
Para asegurar el cumplimiento de los objetivos y garantizar que todas las tareas se completaran a tiempo y dentro del presupuesto acordado, se estableció un cronograma detallado con hitos y fechas límite específicas para cada fase del proyecto. El cronograma se desarrolló de la siguiente manera:

1. Exploración y Limpieza de Datos: (1 semana)

2. Análisis Exploratorio de Datos (EDA**):** (1 semana)
3. Selección del Modelo: (2 semanas)
4. Entrenamiento y Validación del Modelo: (2 semanas)
5. Prueba de la Herramienta y Presentación de la Misma a la Empresa: (2 semanas)

Tabla de Cronograma

A continuación, se presenta una tabla estilo diagrama de Gantt, que muestra las características y las semanas de desarrollo para cada fase del proyecto, sin que se superpongan las tareas.



Descripción de las Fases

1. **Exploración y Limpieza de Datos:** Esta fase se dedicó a la carga y revisión inicial de los datos, identificación y manejo de valores faltantes, y transformación de características categóricas y numéricas.
2. **Análisis Exploratorio de Datos (EDA):** Se realizó un análisis detallado para comprender mejor la estructura de los datos, identificar patrones y detectar anomalías.
3. **Selección del Modelo:** En esta fase se evaluaron varios algoritmos de machine learning y se seleccionó el modelo más adecuado para predecir el PVP.
4. **Entrenamiento y Validación del Modelo:** Se procedió a entrenar el modelo seleccionado y a validar su desempeño utilizando métricas de evaluación.
5. **Prueba de la Herramienta y Presentación de la Misma a la Empresa:** Finalmente, se probó la herramienta en un entorno controlado y se presentó a la empresa para su aprobación y futura implementación.

4.2 Explicación y limpieza de los datos

Como primer paso del desarrollo de cualquier herramienta relacionada con los datos, siempre hay que realizar una limpieza de estos y un análisis inicial. A continuación, se describe en detalle cada uno de los pasos seguidos para asegurar que la base de datos estuviera lista para su uso en modelos predictivos.

Descripción de los Datos

La base de datos proporcionada incluye más de 834.123 registros y contiene las siguientes características:

- **Material:** Campo que describe el tipo de material en nomenclatura de la empresa.
- **Espesor:** El espesor de la pieza realizada en milímetros.
- **Cantidad:** La cantidad de piezas pedidas en el total del pedido.
- **Ttos:** Número de tratamientos que se deben realizar a la materia prima para llegar a la pieza final.
- **Peso:** El peso de la pieza en Kg.
- **Largo:** El largo de la pieza en cm.
- **Ancho:** El ancho de la pieza en cm.
- **Ttops:** Tiempo de corte de la pieza en segundos.
- **Geometría:** Un campo categórico que describe el tipo de geometría de la pieza.
- **PrecioMP:** El precio de la materia prima en el momento de la venta en euros.
- **TargetPvP:** El precio de oferta a predecir.

Paso 1: Carga de los Datos

El primer paso consistió en cargar la base de datos en el entorno de análisis utilizando Python y la biblioteca pandas. Esta tarea comenzó con la importación de los datos desde un archivo Excel.

Para cargar los datos, se utilizó el método read_excel de pandas, que permite leer archivos Excel y convertirlos en un DataFrame, una estructura de datos que facilita el

análisis y la manipulación de los datos. Una vez cargados, se realizó una visualización preliminar para asegurar que la carga se había realizado correctamente y para empezar a familiarizarse con los datos.

Paso 2: Exploración Inicial de los Datos

Una vez cargados los datos, se procedió a realizar una exploración inicial, para entender la estructura y la calidad de los datos antes de proceder con el análisis más detallado. Durante esta fase, se llevaron a cabo las siguientes acciones:

- **Visualización de los primeros registros:** Se examinaron los primeros registros del DataFrame para obtener una visión general de las variables y sus valores.

	Material	Espesor	Cantidad	Ttos	Peso	Largo	Ancho	Ttops	Geometria	PrecioMP	TargetPvP
0	HN12	12.0	1	0	5.430	360.0	180.0	92.0	geo A	0.6180	9.40
1	HND5	5.0	4	0	0.133	130.0	37.0	7.0	geo C	0.6386	2.50
2	HN12	12.0	1	0	8.033	300.0	360.0	74.0	geo B	0.6180	12.49
3	HND5	5.0	2	0	0.036	26.0	50.0	4.0	geo C	0.6386	1.00
4	IK1,5	1.5	1	1	53.832	2708.0	1995.0	61.0	geo B	2.6265	232.87

- **Resumen estadístico:** Se generaron estadísticas descriptivas para las características numéricas, como la media, mediana, desviación estándar, valores mínimos y máximos. Esto permitió identificar posibles anomalías o valores extremos que podrían influir en el análisis.

- **Revisión de la información del DataFrame:** Se utilizó el método info() para obtener un resumen de la estructura del DataFrame, incluyendo el tipo de datos de cada columna y el número de valores no nulos en cada una.

Paso 3: Identificación y Manejo de Valores Faltantes

La presencia de valores faltantes es un problema común en conjuntos de datos grandes y puede afectar negativamente el rendimiento de los modelos predictivos. Por lo tanto, se implementaron varias estrategias para manejar estos valores de manera efectiva:

- **Detección de** valores faltantes: Se identificaron los valores faltantes en cada columna utilizando el método isnull().sum(). Esto permitió cuantificar el número de valores faltantes y determinar la mejor estrategia para manejarlos.

```
Material         0
Espesor          0
Cantidad         0
Ttos             0
Peso           212
Largo         1554
Ancho         1554
Ttops         1944
Geometria     3564
PrecioMP         0
TargetPvP       79
dtype: int64
```

Como se puede observar hay una cantidad importante de datos faltantes, analizando el mejor tratamiento posible de estos para cada una de las características llegue a estas dos estrategias:

- **Imputación de valores:** Para las características numéricas como Espesor y Cantidad, se utilizaron técnicas de imputación. Con esta técnica añado un valor en la posición donde se encuentra el nulo, lo que es muy útil para rellenar campos numéricos, para este campo decido utilizar la mediana debido a la desviación de los datos.

- **Eliminación de registros incompletos:** En el caso de características categóricas críticas como Material, Geometría y la variable objetivo TargetPvP, se optó por eliminar los registros que contenían valores faltantes. Esto fue necesario para evitar que los modelos predictivos se entrenaran con datos incompletos, lo cual podría afectar su precisión y capacidad de generalización.

Paso 4: Elimininar outliers

Durante el proceso de limpieza de datos, se llevó a cabo la eliminación de outliers para asegurar la integridad y la precisión del modelo predictivo. Los outliers, o valores atípicos, son datos que se desvían significativamente de otros valores observados y pueden distorsionar los resultados del análisis. Una vez identificados, se decidió eliminar los registros que contenían estos valores atípicos. Esta eliminación fue crucial para mejorar la calidad de los datos y garantizar que el modelo de machine learning no se viera influenciado negativamente por valores extremos que no representaban patrones reales en los datos.

Paso 5: Transformación de Características Categóricas

Las características categóricas como Material y Geometría no pueden ser directamente utilizadas en modelos de machine learning que requieren entradas numéricas. Por esta razón, se transformaron estas características en un formato adecuado mediante la técnica de codificación LabelEncoder.

Label Encoding: Esta técnica consiste en asignar un valor numérico único a cada categoría dentro de una característica categórica. En lugar de crear múltiples columnas como en la codificación one-hot, LabelEncoder convierte cada categoría en

un valor entero distinto, conservando una sola columna para cada característica. Por ejemplo, si la característica Material tiene las categorías Madera, Metal y Plástico, LabelEncoder podría asignar los valores 0, 1 y 2 a estas categorías, respectivamente.

Ventajas de Label Encoding:

- **Simplicidad:** LabelEncoder es sencillo de implementar y no aumenta el número de columnas en el DataFrame, manteniendo el conjunto de datos compacto.

- **Compatibilidad:** Mantiene las características categóricas en un formato compatible con los algoritmos de machine learning que requieren entradas numéricas.

Esta transformación permitió que todas las características categóricas estuvieran en un formato compatible con los algoritmos de machine learning, asegurando que el modelo interpretara correctamente la información sin aumentar innecesariamente la dimensionalidad del conjunto de datos.

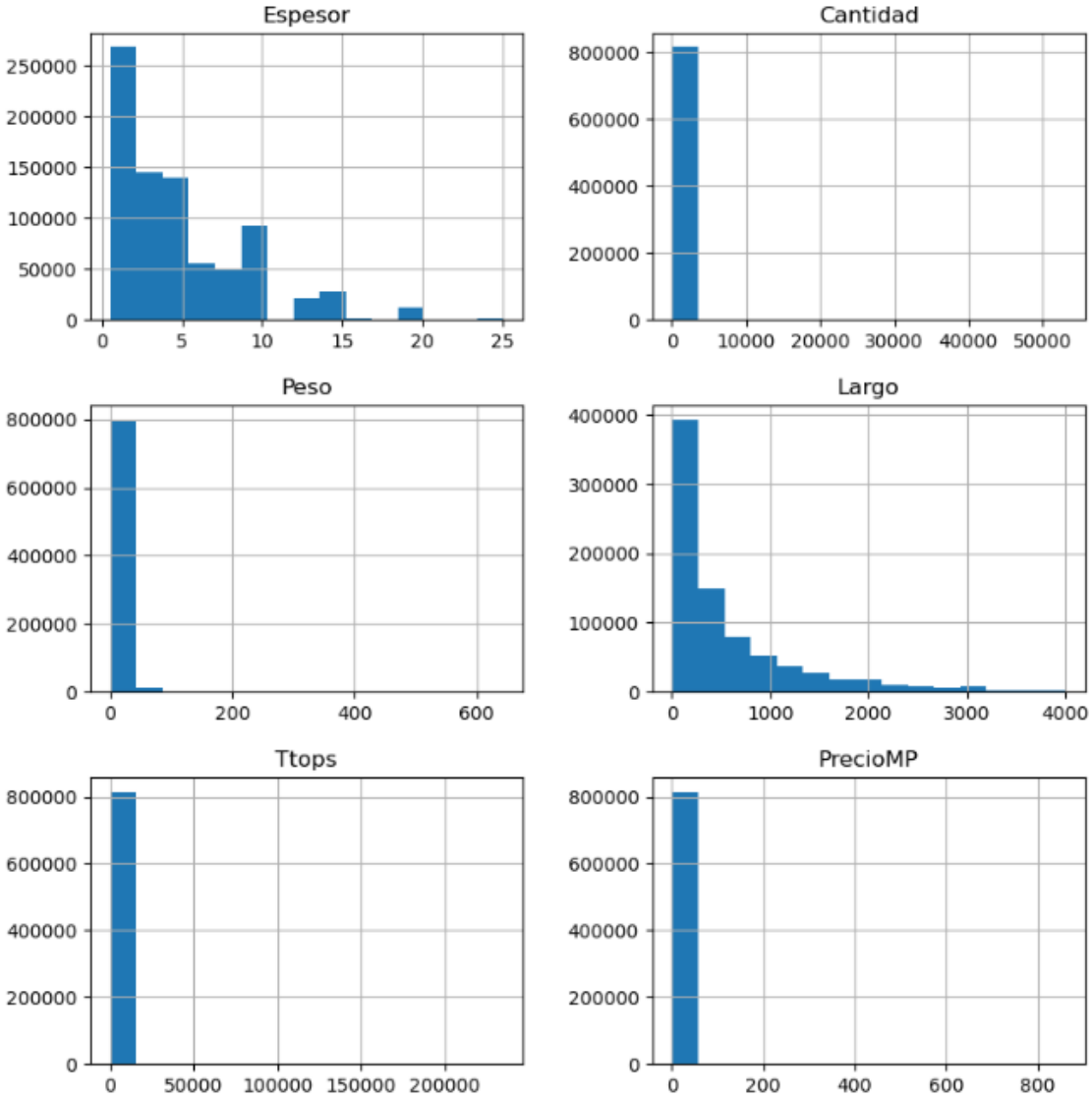
Tras realizar estos 5 pasos de limpieza de datos el dataframe mantiene las 11 caracteristicas y 815798 filas o entradas de datos, lo que ha supuesto una reducción de 18334 filas, lo que respecta el 2,2% del total de datos iniciales (834123), una reducción aceptable y necesaria para continuar con el desarrollo del modelo.

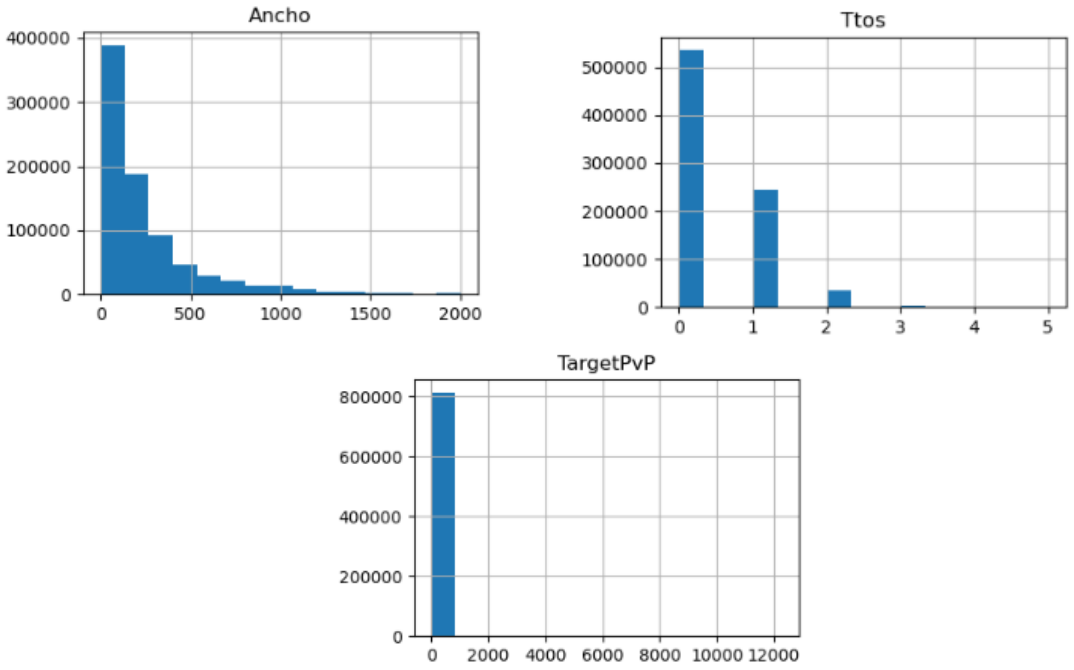
4.3 EDA (Exploratory Data Analysis)

El análisis exploratorio de datos (EDA) permite comprender mejor la estructura de los datos, identificar patrones, detectar anomalías y comprobar las hipótesis iniciales. El EDA realizado en este proyecto ha sido el siguiente:

Paso 1: Análisis de la distribución de los datos

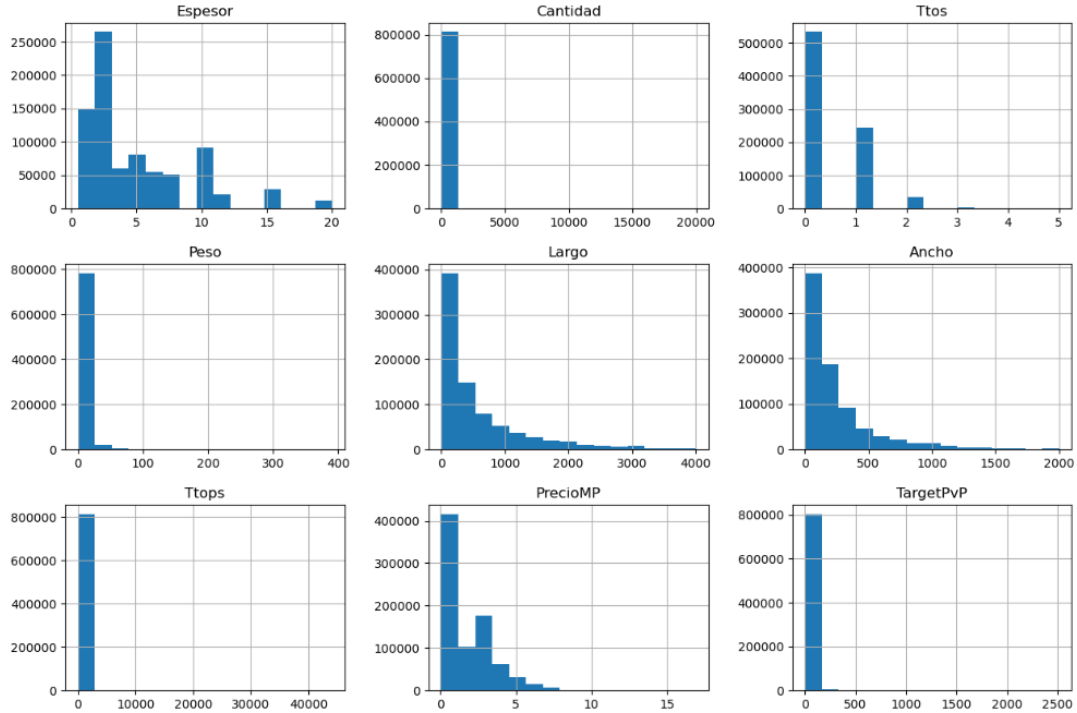
En este apartado se han realizado histogramas (1 para cada característica del dataset) para observar cómo se distribuyen los datos para todos los apartados excepto Material y Geometría.





La conclusión que obtengo al analizar las distribuciones es que hay unas cuantas características que tienen ciertos outliers que distorsionan las características estadísticas de los datos, por lo que se realiza un segundo filtrado de outliers de las siguientes caracteristicas: Espesor, Cantidad, Peso, Ttops, PrecioMP y TargetPvP.

Después del segundo filtrado de outliers los histogramas quedan de la siguiente forma:



Paso 2: Análisis de correlación

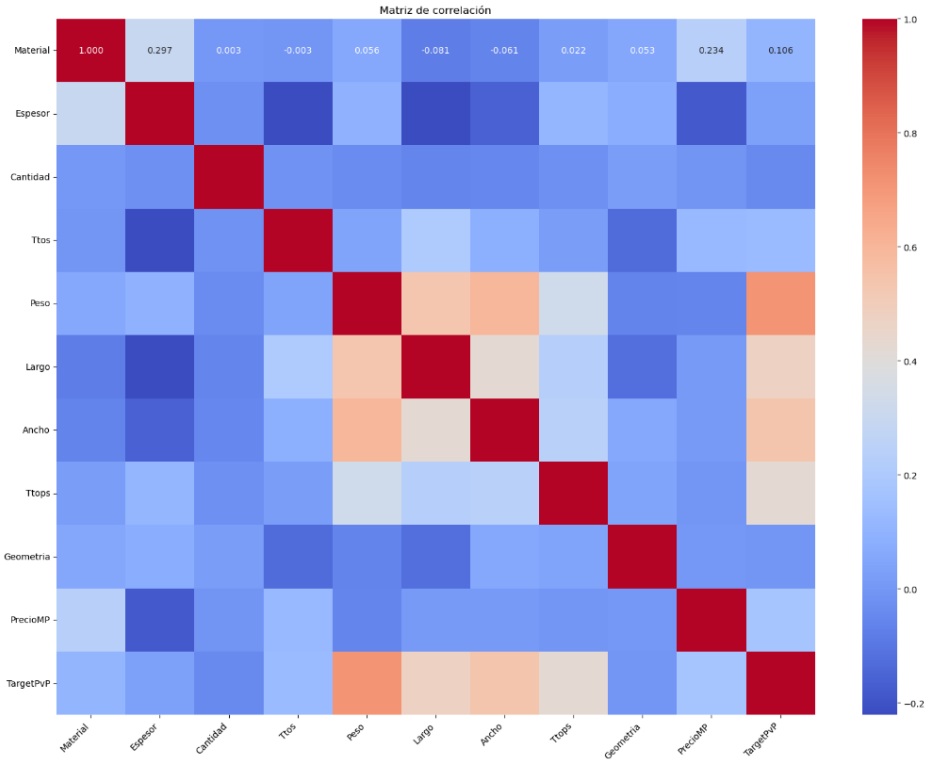
En este apartado se realiza un análisis de la correlación que existe entre las diferentes características del DataFrame. La correlación entre variables es una medida estadística que indica el grado en el que dos variables están relacionadas. Es una herramienta crucial en el análisis exploratorio de datos ya que permite identificar relaciones lineales entre las variables y entender cómo una variable puede afectar a otra.

Matriz de Correlación

La matriz de correlación es una tabla que muestra los coeficientes de correlación entre todas las combinaciones posibles de las variables en el DataFrame. Estos coeficientes de correlación pueden variar entre -1 y 1, donde:

- **1** indica una correlación positiva perfecta,
- **-1** indica una correlación negativa perfecta,
- **0** indica que no hay correlación.

A continuación, se presenta la matriz de correlación generada para las características del DataFrame:



Análisis de Resultados

- **Material:** Tiene una correlación positiva moderada con Espesor (0.297) y PrecioMP (0.234), y una correlación positiva débil con TargetPvP (0.106). Esto sugiere que el tipo de material tiene alguna relación con el espesor y el precio de la materia prima, pero una influencia menor en el precio de venta.

- **Espesor:** Muestra una correlación negativa con Ttos (-0.221) y Largo (-0.221), indicando que a medida que el espesor aumenta, el número de tratamientos y el largo de la pieza tienden a disminuir. La correlación con TargetPvP es casi nula (0.030), lo que sugiere que el espesor no tiene un impacto significativo en el precio de venta.

- **Cantidad:** No presenta correlaciones significativas con otras variables, indicando que la cantidad de piezas pedidas no está fuertemente relacionada con las demás características. Su correlación con TargetPvP es negativa y muy baja (-0.042), sugiriendo que la cantidad no influye en el precio de venta.

- **Ttos:** Tiene una correlación negativa con Espesor (-0.221) y positiva con Largo (0.206), lo que sugiere que piezas más largas requieren más tratamientos. La correlación con TargetPvP es positiva (0.130), aunque no muy fuerte, indicando que el número de tratamientos tiene alguna influencia en el precio de venta.

- **Peso:** Presenta una fuerte correlación positiva con Largo (0.535) y Ancho (0.596), y una muy fuerte con TargetPvP (0.709). Esto indica que piezas más pesadas tienden a ser más largas y anchas, y su peso es un factor importante en la determinación del precio de venta.

- **Largo:** Además de su correlación con Peso, muestra una correlación moderada con Ancho (0.426) y TargetPvP (0.477), indicando que estas dimensiones influyen en el precio.

- **Ancho:** Similar a Largo, tiene una correlación significativa con Peso (0.596), Largo (0.426) y TargetPvP (0.540). Esto sugiere que el ancho también es un factor relevante en la determinación del precio de venta.

- **Ttops:** Muestra correlaciones moderadas con Peso (0.328) y TargetPvP (0.427), sugiriendo que el tiempo de corte también influye en el precio de venta.

- **Geometría:** Tiene correlaciones muy bajas con todas las demás variables, incluyendo TargetPvP (-0.004), lo que indica que la geometría no afecta significativamente al precio de venta en comparación con otras variables.

- **PrecioMP:** Muestra correlaciones positivas con Material (0.234) y TargetPvP (0.170), indicando que el precio de la materia prima influye en el precio de venta, aunque no de manera muy fuerte.

Comparación con la Columna TargetPvP

- **Material (0.106):** Aunque tiene una correlación positiva, la influencia del tipo de material en el precio de venta es débil.

- **Espesor (0.030):** La correlación casi nula indica que el espesor no tiene un impacto significativo en el precio de venta.

- **Cantidad (-0.042):** La correlación negativa y baja sugiere que la cantidad de piezas pedidas no influye en el precio de venta.

- **Ttos (0.130):** La correlación positiva indica que el número de tratamientos tiene alguna influencia en el precio de venta, aunque no es muy fuerte.

- **Peso (0.709):** La fuerte correlación positiva muestra que el peso es un factor importante en la determinación del precio de venta.

- **Largo (0.477):** La correlación moderada sugiere que el largo de la pieza influye significativamente en el precio de venta.

- **Ancho (0.540):** Similar al largo, el ancho también tiene una correlación significativa con el precio de venta.

- **Ttops (0.427):** La correlación moderada indica que el tiempo de corte influye en el precio de venta.

- **Geometría (-0.004):** La correlación casi nula muestra que la geometría no afecta significativamente el precio de venta.

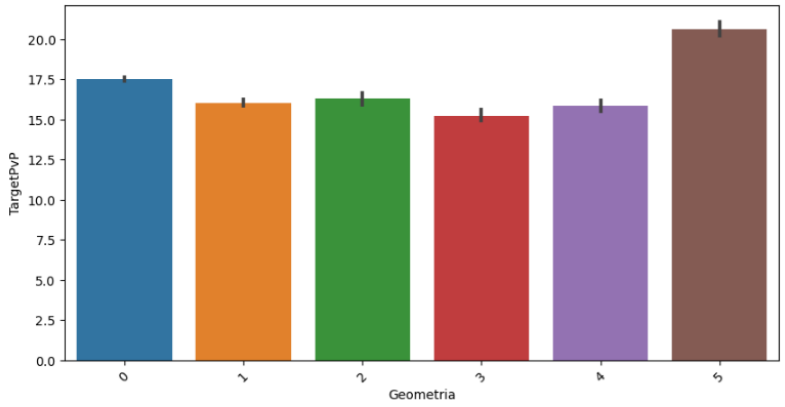
- **PrecioMP (0.170):** La correlación positiva indica que el precio de la materia prima tiene alguna influencia en el precio de venta, aunque no es muy fuerte.

Paso 3: Análisis de las columnas con menos correlación

Como se ha podido observar en el paso anterior, hay varias columnas que tienen una correlación muy baja con la característica a predecir (TargetPvP). Vamos a observar estas columnas más detenidamente para decidir su inclusión o exclusión en el modelo. Las columnas con baja correlación son: Material, Espesor, Cantidad, Ttos, Geometría, y PrecioMP. Vamos a empezar con la característica de Geometría.

Análisis de la Característica Geometría

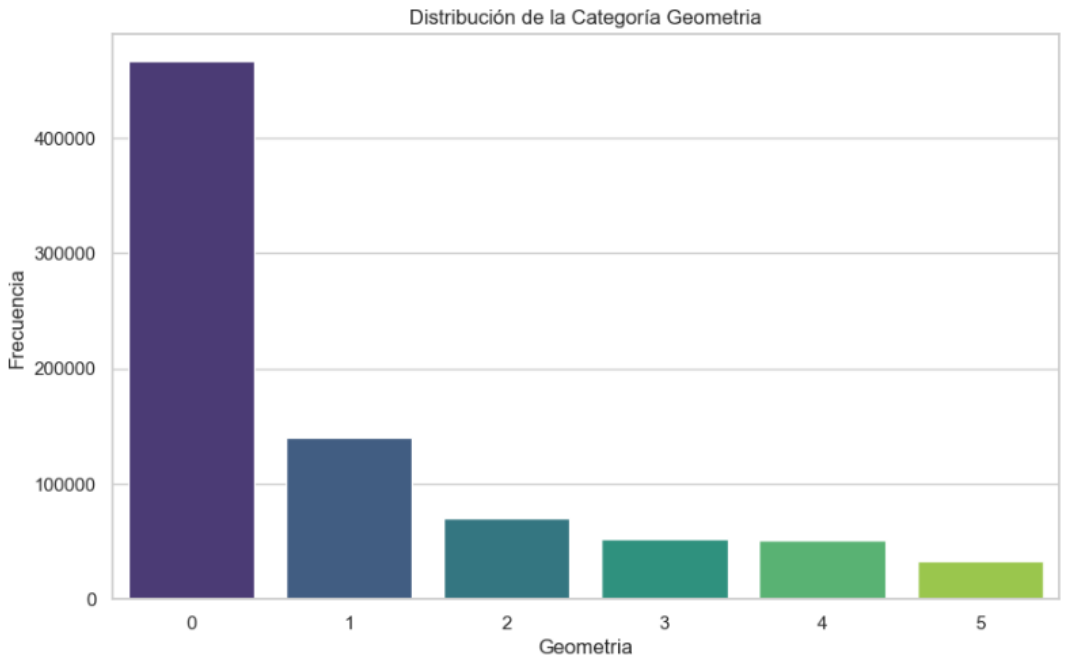
Gráfico 1: Comparación de Geometría con TargetPvP



El primer gráfico muestra la relación entre las diferentes categorías de Geometría y el TargetPvP. Cada barra representa el valor medio de TargetPvP para cada categoría de Geometría, y las líneas de error indican la variabilidad dentro de cada categoría.

En este gráfico, se puede observar que las diferentes categorías de Geometría presentan variaciones en el TargetPvP. Sin embargo, la diferencia entre estas categorías no es muy pronunciada, lo que sugiere que la geometría de la pieza no tiene un impacto significativo en el precio de venta al público.

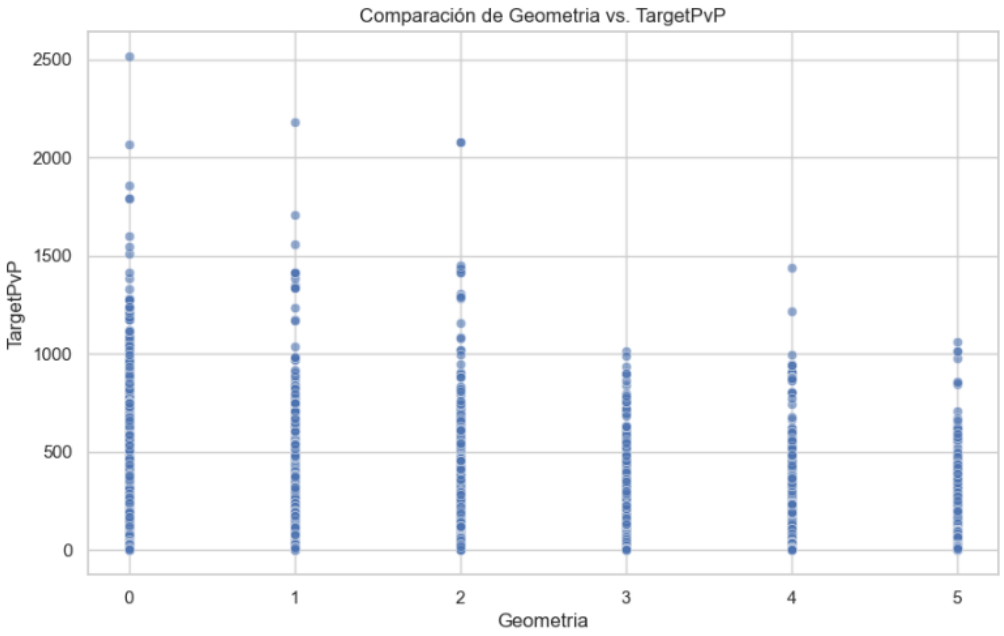
Gráfico 2: Distribución de la Categoría Geometría



El segundo gráfico muestra la distribución de las diferentes categorías de Geometría. Este gráfico de barras indica cuántos registros hay para cada categoría de Geometría.

Se observa que la categoría 0 es la más común, seguida por la categoría 1, y así sucesivamente. Esta distribución sesgada puede influir en la interpretación de la relación entre Geometría y TargetPvP, ya que algunas categorías tienen muchos más datos que otras, lo que podría afectar la variabilidad observada en el primer gráfico.

Gráfico 3: Comparación de Geometría vs. TargetPvP



El tercer gráfico es un diagrama de dispersión que compara las categorías de Geometría con TargetPvP. Cada punto representa un registro individual, mostrando cómo se distribuye el TargetPvP dentro de cada categoría de Geometría.

En este diagrama de dispersión, se puede ver que los valores de TargetPvP están dispersos de manera similar entre las diferentes categorías de Geometría. No se observa un patrón claro que indique una fuerte relación entre Geometría y TargetPvP, lo que confirma la baja correlación observada en la matriz de correlación.

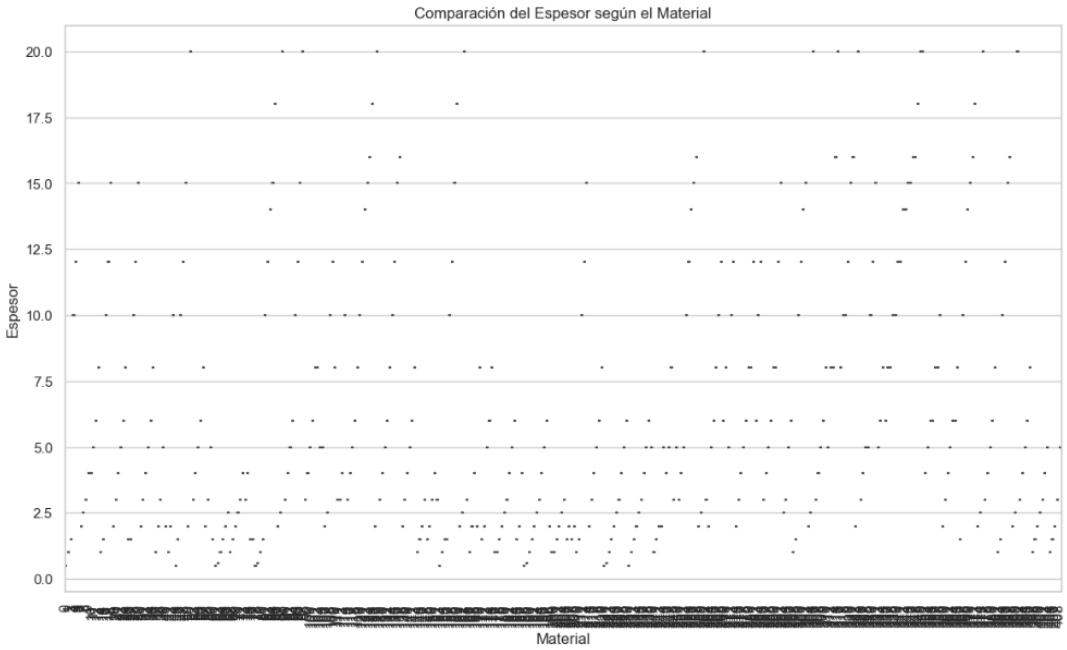
Viendo más detenidamente la comparación de la columna Geometría con la columna TargetPvP y observando su poca correlación, se concluye que la geometría de la pieza no influye significativamente en el precio de venta. Por lo tanto, para la continuación del desarrollo del modelo de machine learning, se prescindirá de la columna Geometría. Esta decisión ayudará a simplificar el modelo y a mejorar su capacidad de generalización al centrarse en las características que tienen un mayor impacto en la predicción del TargetPvP.

Análisis de la Características Material y Espesor

Ahora vamos a analizar las siguientes 2 columnas con correlación muy muy baja con la característica a predecir: Material y Espesor

Vamos a empezar con Espesor, la cual es una característica propia del material, el cual en la empresa lo clasifican por espesor, por lo que vamos a comprobar que estos dos se describe uno al otro.

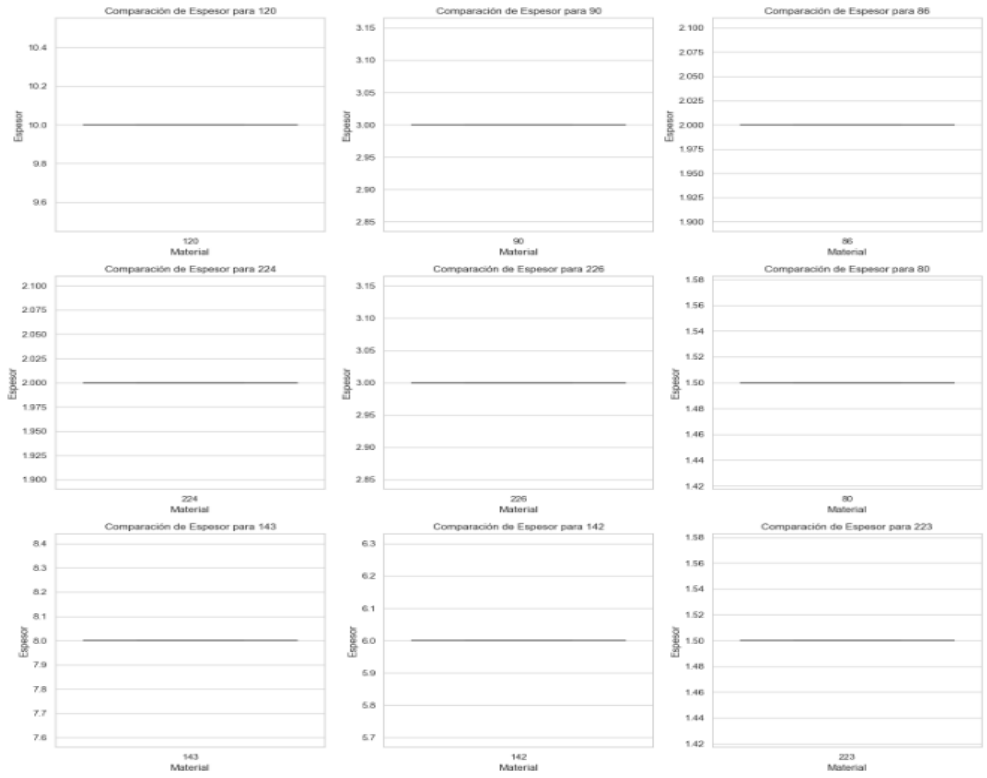
Gráfico 1: Comparación del Espesor según el Material



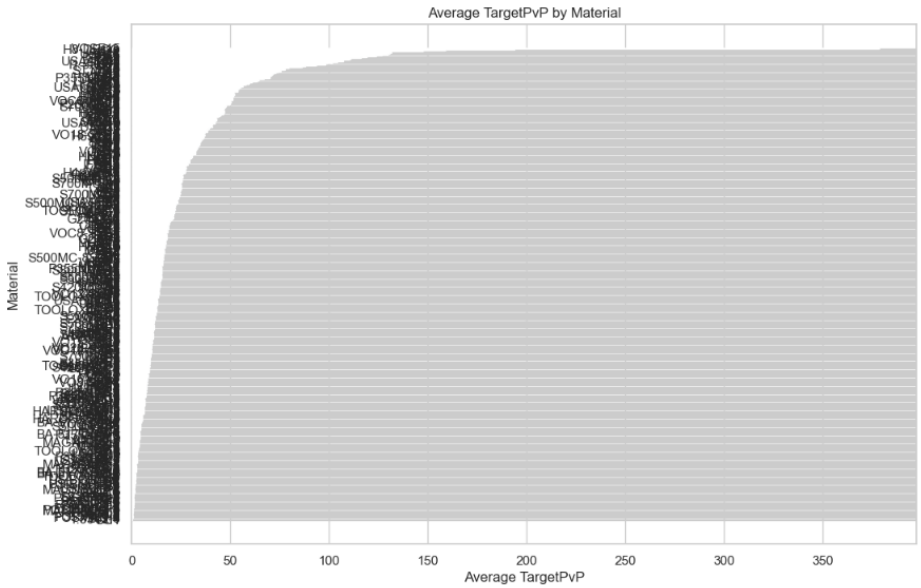
Como podemos observar, este gráfico no nos muestra nada significativo debido a la gran cantidad de materiales que existen. La dispersión de los puntos a lo largo del eje Material hace difícil interpretar alguna tendencia o relación clara entre Material y Espesor.

Gráfico 2: Comparación de Espesor para los 16 Materiales Más Comunes

Esta es una imagen acortada de 9 de ellos:



Como podemos observar ahora claramente, la columna espesor no agrega información a la característica material, por lo que vamos a eliminar una de las 2 características, por lo que voy a analizar la columna material a continuación



Dado que el espesor no aporta información adicional y está intrínsecamente ligado al material, vamos a analizar la columna Material para evaluar su utilidad.

Al analizar la relación entre Material y TargetPvP, observamos que no hay un patrón claro o significativo que justifique mantener la columna Material en el modelo. La distribución del TargetPvP según el material es bastante uniforme y no muestra una tendencia clara que pudiera ser útil para el modelo predictivo.

Conclusión

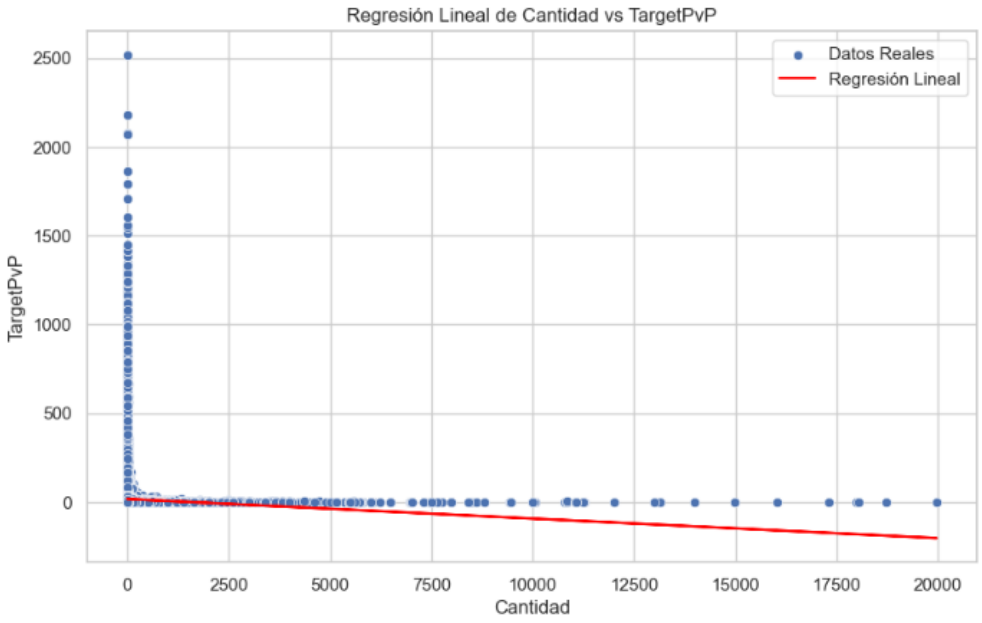
Al estar analizando los componentes de los datos y el resto de las columnas, y tras discutirlo con la empresa, se concluyó que la columna Material no es una característica muy útil. Esto se debe a que ya tenemos todos los datos necesarios que esta aportaría, como PrecioMP y Espesor, que son los campos más importantes. Además, al ser un campo categórico, podría causar más problemas que beneficios en el desarrollo del modelo. Por lo tanto, se decidió eliminar la columna Material del análisis. Esta decisión ayudará a simplificar el modelo y mejorar su capacidad de generalización al centrarse en las características más relevantes.

Análisis de la Característica Cantidad

La siguiente fase del análisis se centra en la característica Cantidad, una de las variables con una baja correlación con TargetPvP.

Gráfico de Regresión Lineal

El gráfico a continuación muestra una regresión lineal de Cantidad contra TargetPvP.



Al observar la correlación, se nota que el precio disminuye considerablemente cuando aumenta la cantidad de piezas. Esta tendencia sugiere que, a mayor cantidad de piezas, el precio por pieza es más bajo. Esto me generó la duda de si el TargetPvP estaba registrado como precio unitario, es decir, que el precio no se ha multiplicado por la cantidad de piezas totales para mostrar el precio real del total del pedido.

Me puse en contacto con la empresa y confirmaron mi sospecha: el TargetPvP representa el precio unitario por pieza.

Plan de Acción

Al descubrir esta información, se plantearon tres enfoques diferentes para llegar al mejor resultado posible en el modelado:

1. **Eliminar la Cantidad:** En esta primera rama, se eliminará la variable Cantidad del análisis para calcular el TargetPvP unitario por pieza, centrando el modelo únicamente en otras características que afectan el precio por unidad.

2. **Multiplicar Cantidad por TargetPvP:** En esta segunda rama, se multiplicará la cantidad de piezas por el TargetPvP, ya que este representa el precio por una sola pieza. Esto permitirá calcular el precio total del pedido y usar esta información en el modelado.

3. **Mantener el DataFrame como Está:** En la tercera rama, se mantendrá el DataFrame tal como está, calculando el precio unitario por pieza, pero ajustando el modelo para que pueda manejar correctamente la cantidad de piezas como una característica que influye en el precio unitario.

Paso 4: División del dataset y pequeño EDA

En este apartado voy a proceder a crear tres copias del dataset como he comentado en el paso anterior para evaluar diferentes enfoques y determinar el mejor método para el cálculo del TargetPvP. Esta división y análisis exploratorio adicional del dataset son cruciales para identificar la mejor manera de manejar la variable Cantidad, que inicialmente mostró una correlación baja, pero presenta un comportamiento interesante que podría influir significativamente en el TargetPvP.

1. **Eliminar la Cantidad:** En esta primera rama, se eliminará la variable Cantidad del análisis para calcular el TargetPvP unitario por pieza. Este enfoque se basa en la premisa de que la Cantidad no debería influir en el precio unitario de cada pieza, permitiendo que el modelo se centre en otras características que afectan directamente el precio por unidad, como Espesor, Peso, Largo, Ancho, Ttos, Ttops y PrecioMP.

	Espesor	Ttos	Peso	Largo	Ancho	Ttops	PrecioMP	TargetPvP
0	12.0	0	5.430	360.0	180.0	92.0	0.6180	9.40
1	5.0	0	0.133	130.0	37.0	7.0	0.6386	2.50
2	12.0	0	8.033	300.0	360.0	74.0	0.6180	12.49
3	5.0	0	0.036	26.0	50.0	4.0	0.6386	1.00
4	1.5	1	53.832	2708.0	1995.0	61.0	2.6265	232.87

```
Espesor      813795
Ttos         813795
Peso         813795
Largo        813795
Ancho        813795
Ttops        813795
PrecioMP     813795
TargetPvP    813795
dtype: int64
```

- **Foto 1:** Muestra una vista del DataFrame sin la columna Cantidad.
- **Foto 2:** Confirma la estructura del DataFrame después de eliminar la columna Cantidad.

2. **Multiplicar Cantidad por TargetPvP:** En esta segunda rama, se multiplicará la Cantidad de piezas por el TargetPvP, ya que este representa el precio por una sola pieza. Esto permitirá calcular el precio total del pedido y usar esta información en el modelado. Este enfoque reconoce que la Cantidad puede tener un impacto significativo en el TargetPvP total y puede proporcionar insights sobre la economía de escala o descuentos por volumen.

	Espesor	Cantidad	Ttos	Peso	Largo	Ancho	Ttops	PrecioMP	TargetPvP
0	12.0	1	0	5.430	360.0	180.0	92.0	0.6180	9.40
1	20.0	4	0	0.532	520.0	148.0	28.0	2.5544	10.00
2	12.0	1	0	8.033	300.0	360.0	74.0	0.6180	12.49
3	10.0	2	0	0.072	52.0	100.0	8.0	1.2772	2.00
4	1.5	1	1	53.832	2708.0	1995.0	61.0	2.6265	232.87

```
Espesor      813795
Cantidad     813795
Ttos         813795
Peso         813795
Largo        813795
Ancho        813795
Ttops        813795
PrecioMP     813795
TargetPvP    813795
dtype: int64
```

- **Foto 3:** Muestra la nueva columna calculada multiplicando Cantidad por TargetPvP.

- **Foto 4:** Confirma la estructura del DataFrame después de realizar esta operación.

3. **Mantener el DataFrame como Está:** En la tercera rama, se mantendrá el DataFrame tal como está, incluyendo la columna Cantidad, pero ajustando el modelo para que pueda manejar correctamente la cantidad de piezas como una característica que influye en el precio unitario. Este enfoque permitirá evaluar si la Cantidad influye indirectamente en el TargetPvP mediante interacciones con otras características.

	Espesor	Cantidad	Ttos	Peso	Largo	Ancho	Ttops	PrecioMP	TargetPvP
0	12.0	1	0	5.430	360.0	180.0	92.0	0.6180	9.40
1	5.0	4	0	0.133	130.0	37.0	7.0	0.6386	2.50
2	12.0	1	0	8.033	300.0	360.0	74.0	0.6180	12.49
3	5.0	2	0	0.036	26.0	50.0	4.0	0.6386	1.00
4	1.5	1	1	53.832	2708.0	1995.0	61.0	2.6265	232.87

```
Espesor      813795
Cantidad     813795
Ttos         813795
Peso         813795
Largo        813795
Ancho        813795
Ttops        813795
PrecioMP     813795
TargetPvP    813795
dtype: int64
```

- **Foto 5:** Muestra el DataFrame original con todas las columnas intactas.
- **Foto 6:** Confirma la estructura del DataFrame sin modificaciones.

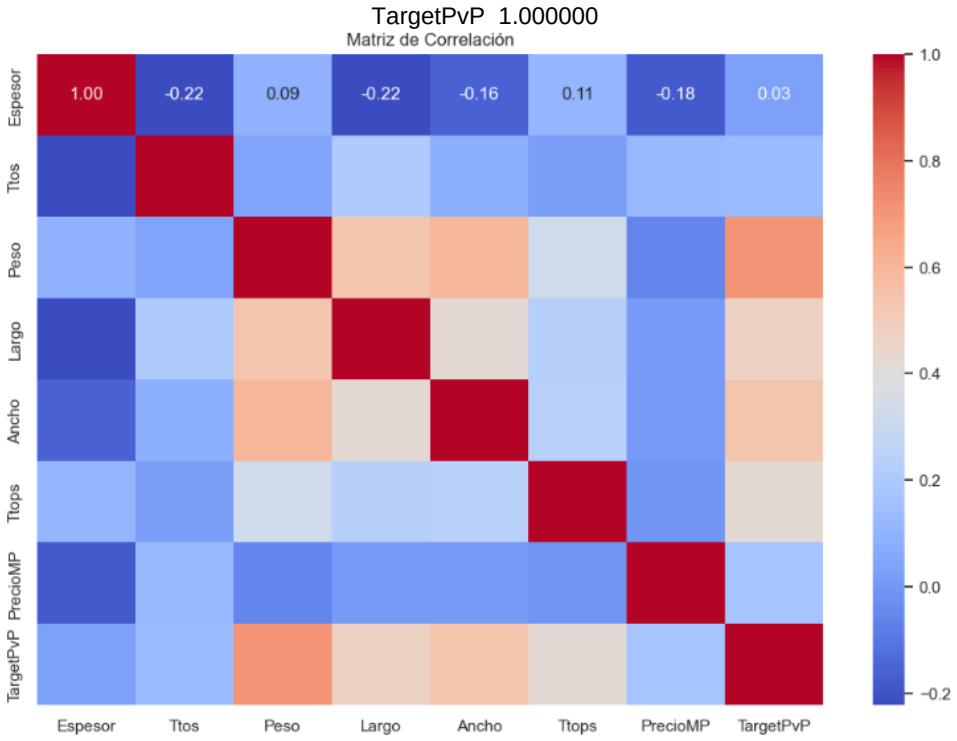
Para comprobar que he realizado este proceso correctamente, voy a sumar el TargetPvP de cada DataFrame. Los resultados esperados son que los valores del primer y tercer DataFrame sean iguales, y el segundo sea significativamente superior.

```
La suma de todos los valores en la columna TargetPvP es: 13862686.280633688
La suma de todos los valores en la columna TargetPvP es: 63294422.84485331
La suma de todos los valores en la columna TargetPvP es: 13862686.280633688
```

Análisis de la correlación de cada dataframe.

Eliminar la Cantidad

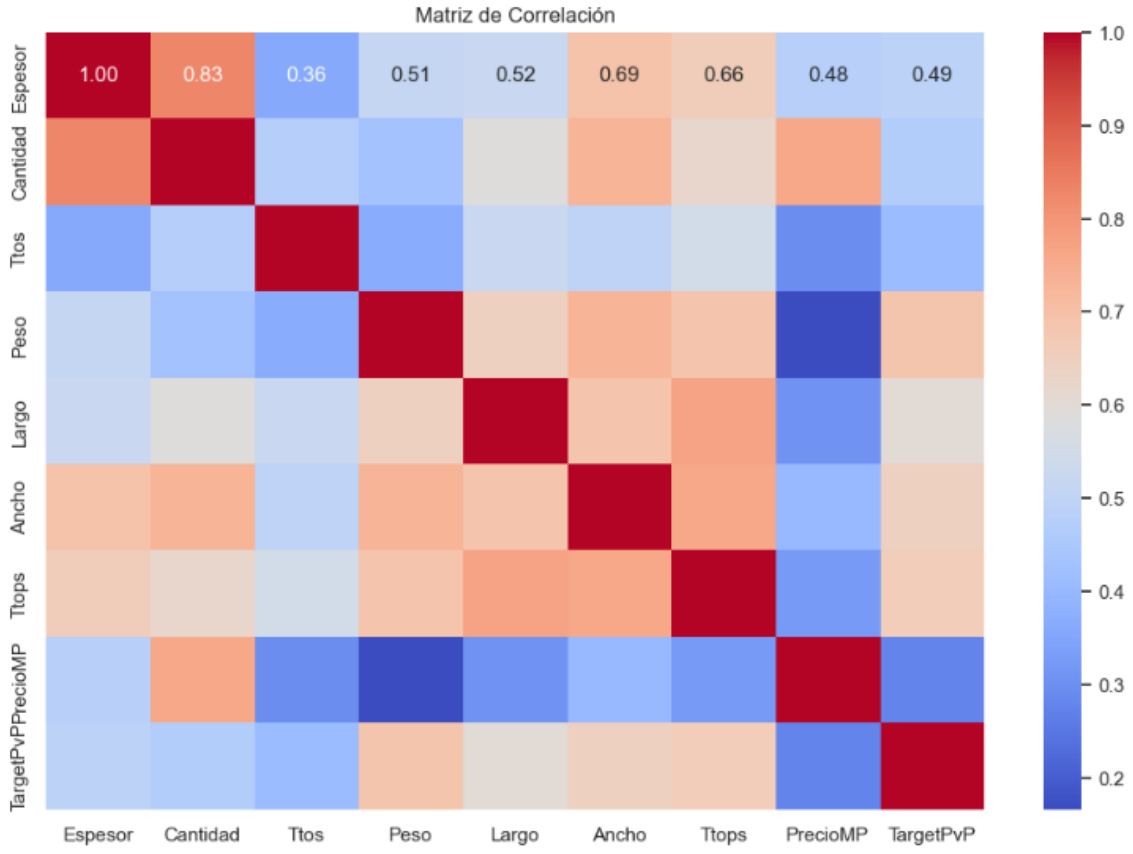
	TargetPvP
Espesor	0.030133
Ttos	0.130240
Peso	0.709380
Largo	0.477370
Ancho	0.540278
Ttops	0.427196
PrecioMP	0.170247



La matriz de correlación muestra que las características Peso, Ancho, y Largo tienen las correlaciones más altas con TargetPvP. La eliminación de Cantidad no afecta significativamente estas relaciones.

Multiplicar Cantidad por TargetPvP

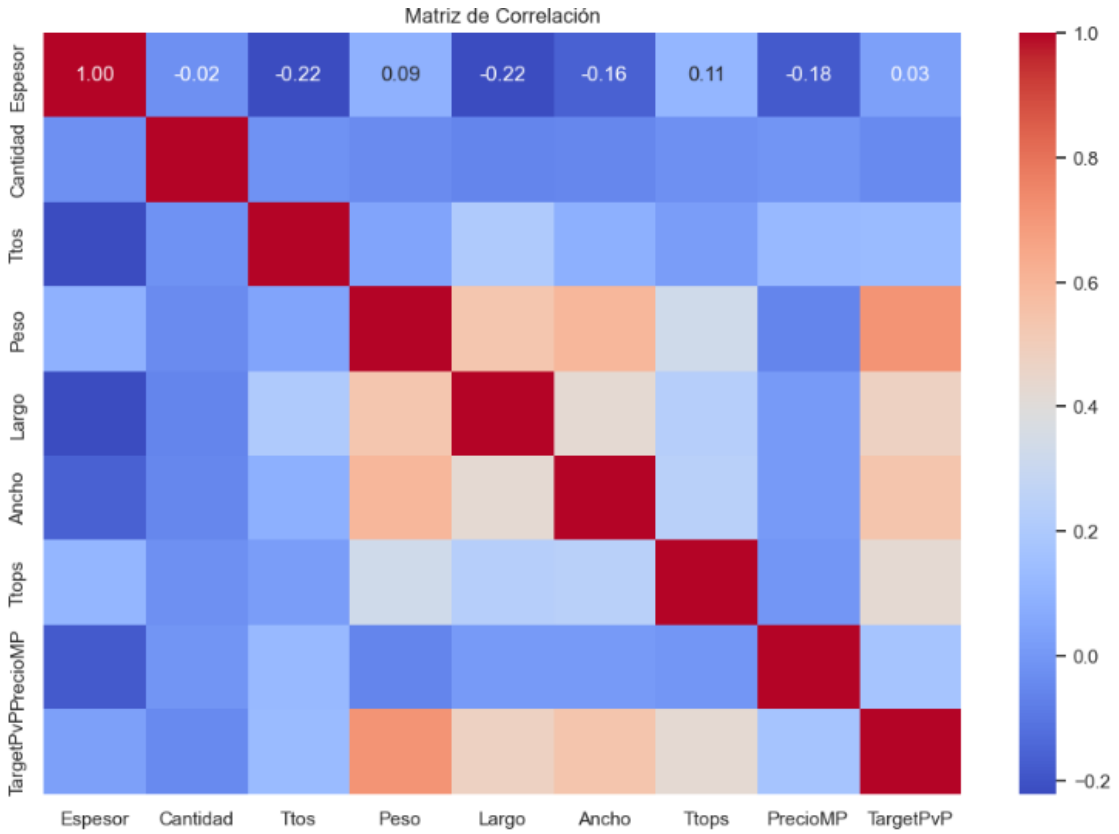
	TargetPvP
Espesor	0.491271
Cantidad	0.467001
Ttos	0.408395
Peso	0.685502
Largo	0.601998
Ancho	0.647215
Ttops	0.658023
PrecioMP	0.275255
TargetPvP	1.000000



Al multiplicar Cantidad por TargetPvP, observamos un aumento en la correlación de varias características, especialmente Espesor, Largo, Ancho y Ttops. Esto indica que la cantidad de piezas tiene un impacto considerable cuando se considera el total del precio del pedido.

Mantener el DataFrame como Está

	TargetPvP
Espesor	0.030133
Cantidad	-0.041763
Ttos	0.130240
Peso	0.709380
Largo	0.477370
Ancho	0.540278
Ttops	0.427196
PrecioMP	0.170247
TargetPvP	1.000000



Manteniendo el DataFrame original, la correlación de Cantidad con TargetPvP es negativa, lo que sugiere que a medida que aumenta la cantidad de piezas, el precio unitario tiende a disminuir, posiblemente debido a descuentos por volumen o economías de escala.

La correlación más alta la tiene el DataFrame que multiplica la cantidad por TargetPvP, lo que indica una relación más fuerte con la variable objetivo. Sin embargo, una alta correlación no siempre significa que este enfoque sea el mejor para nuestro modelo. Es crucial considerar cómo cada enfoque afecta la interpretabilidad del modelo y su

capacidad para generalizar a nuevos datos. Por lo tanto, antes de descartar cualquier DataFrame, se continuará con un análisis más profundo y la validación del rendimiento de cada enfoque en el modelado predictivo. Este análisis multifacético asegurará que se elija la estrategia más adecuada y efectiva para predecir el precio de venta al público, optimizando así la utilidad y precisión del modelo para Láser Valencia SL.

4.4 Elección del modelo

Vamos a proceder con la elección del modelo. Para ello, se probaron nueve modelos de los más comunes para cada DataFrame.

El objetivo de esta fase del proyecto es identificar el modelo de machine learning más adecuado para predecir el precio de venta al público (PVP) en Láser Valencia SL. Dado que las características de los datos pueden influir significativamente en el rendimiento del modelo, se creó tres versiones del dataset: una eliminando la columna Cantidad, otra multiplicando Cantidad por TargetPvP para obtener el precio total del pedido, y una manteniendo el dataset tal como está. Esta estrategia permite explorar diferentes enfoques y determinar cuál es el más efectivo para nuestras necesidades específicas. Los modelos evaluados incluyen Regresión Lineal Múltiple, Ridge Regression, Elastic Net, Árboles de Decisión, Random Forest, K-Nearest Neighbors (K-NN) y Bagging, LightGBM. La evaluación se basó en las métricas de error cuadrático medio (MSE) y el coeficiente de determinación (R^2).

4.4.1 Evaluación de Diferentes Modelos

A continuación, se presentan los resultados obtenidos para cada modelo y cada versión del dataset, con una evaluación, análisis y comparación de los resultados (para más información relacionada con el código, acudir al anexo I):

1. Regresión Lineal Múltiple

Eliminar la Cantidad:

- MSE: 610.402334940602
- R^2: 0.6181989846213578

Multiplicar Cantidad por TargetPvP:

- MSE: 39594.95608135593
- R^2: 0.506862870118251

Mantener el DataFrame como Está:

- MSE: 610.398105439931
- R^2: 0.6182016301349123

Evaluación de Resultados: La Regresión Lineal Múltiple mostró que el modelo funciona de manera consistente tanto al eliminar la cantidad como al mantener el DataFrame original. Sin embargo, multiplicar Cantidad por TargetPvP resultó en un MSE significativamente mayor y un R^2 menor, lo que indica que este enfoque es menos efectivo.

Análisis de Resultados: La similaridad en los resultados para eliminar la cantidad y mantener el DataFrame original sugiere que Cantidad no afecta significativamente la precisión del modelo en este contexto. La baja performance al multiplicar Cantidad por TargetPvP podría deberse a una posible distorsión introducida al cambiar la escala de los valores.

2. Ridge Regression

Eliminar la Cantidad:

- MSE: 610.4023352535291
- R^2: 0.6181989844256248

Multiplicar Cantidad por TargetPvP:

- MSE: 39594.95608112359
- R^2: 0.5068628701211446

Mantener el DataFrame como Está:

- MSE: 610.3981057468851
- R^2: 0.6182016299429154

Evaluación de Resultados: Ridge Regression mostró resultados casi idénticos a los de la Regresión Lineal Múltiple, sugiriendo que la regularización no mejora significativamente el rendimiento en este caso.

Análisis de Resultados: La regularización L2 implementada en Ridge Regression no proporciona una mejora notable en comparación con la regresión lineal simple. Esto sugiere que los datos no sufren significativamente de multicolinealidad que pudiera beneficiarse de la regularización.

Similar a la regresión lineal, Ridge Regression no muestra mejoras destacables y se comporta de manera similar en términos de MSE y R^2.

3. Elastic Net

Eliminar la Cantidad:

- MSE: 616.3558920607105
- R^2: 0.6144750897024543

Multiplicar Cantidad por TargetPvP:

- MSE: 39595.35698511039
- R^2: 0.5068578770447285

Mantener el DataFrame como Está:

- MSE: 616.3496687199757
- R^2: 0.6144789823445291

Evaluación de Resultados: Elastic Net no mostró una mejora notable en comparación con los modelos anteriores. Los resultados fueron consistentes en todas las configuraciones del dataset, pero no destacaron.

Análisis de Resultados: La combinación de penalizaciones L1 y L2 en Elastic Net no logró mejorar significativamente el rendimiento, lo que sugiere que las características seleccionadas no se benefician de la combinación de estas regularizaciones.

Comparado con Ridge y Regresión Lineal Múltiple, Elastic Net no proporciona ventajas claras y su rendimiento es similar, particularmente en configuraciones que eliminan la cantidad o mantienen el dataset original.

4. Árboles de Decisión

Eliminar la Cantidad:

- MSE: 270.6442945998987
- R^2: 0.830714496702027

Multiplicar Cantidad por TargetPvP:

- MSE: 21679.463240207282
- R^2: 0.7299921672425658

Mantener el DataFrame como Está:

- MSE: 257.4539540546288
- R^2: 0.8389649327261032

Evaluación de Resultados: Los Árboles de Decisión mejoraron significativamente el rendimiento, especialmente al mantener el DataFrame original. Este modelo mostró una notable reducción en MSE y un aumento en R^2.

Análisis de Resultados: Los Árboles de Decisión son capaces de capturar relaciones no lineales entre las variables, lo que puede explicar la mejora en rendimiento. La configuración que mantiene el DataFrame original proporcionó el mejor equilibrio entre precisión y capacidad explicativa.

En comparación con los modelos lineales, los Árboles de Decisión mostraron un avance significativo en términos de precisión, destacando la importancia de capturar relaciones no lineales en los datos.

5. Random Forest

Eliminar la Cantidad:

- MSE: 164.85228096727693
- R^2: 0.8968864228428676

Multiplicar Cantidad por TargetPvP:

- MSE: 12273.091775397472
- R^2: 0.8471442362391036

Mantener el DataFrame como Está:

- MSE: 162.9903231111223
- R^2: 0.898051060262122

Evaluación de Resultados: El modelo Random Forest proporcionó los mejores resultados, con el menor MSE y el mayor R^2, al mantener el DataFrame original. Este modelo destaca en términos de precisión y capacidad explicativa.

Análisis de Resultados: Random Forest, al ser un modelo de ensamble, puede manejar la variabilidad y la complejidad de los datos de manera más efectiva. La configuración que mantuvo el DataFrame original aprovechó al máximo esta capacidad, proporcionando las predicciones más precisas.

Comparado con los Árboles de Decisión y otros modelos lineales, Random Forest mostró un rendimiento superior, subrayando la importancia de utilizar modelos de ensamble para datos complejos.

6. K-Nearest Neighbors (K-NN)

Eliminar la Cantidad:

- MSE: 555.2879284156091
- R^2: 0.6526725361933406

Multiplicar Cantidad por TargetPvP:

- MSE: 27011.68892802436
- R^2: 0.6635817268276563

Mantener el DataFrame como Está:

- MSE: 554.9127696300837
- R^2: 0.6529071945441409

Evaluación de Resultados: K-NN no mostró un rendimiento notablemente superior en ninguna de las configuraciones del DataFrame. Los resultados fueron malos en comparación con otros modelos.

Análisis de Resultados: La naturaleza de K-NN, que depende de la proximidad de los puntos de datos, puede no ser ideal para este tipo de problema donde las relaciones no son simplemente locales.

En comparación con modelos de ensamble y árboles de decisión, K-NN es claramente inferior en términos de precisión y capacidad explicativa.

7. Bagging

Eliminar la Cantidad:

- MSE: 174.72187834579057
- R^2: 0.8907130809586797

Multiplicar Cantidad por TargetPvP:

- MSE: 13414.940157003088
- R^2: 0.832923035121766

Mantener el DataFrame como Está:

- MSE: 174.62937608779342
- R^2: 0.890770940265577

Evaluación de Resultados: Bagging mostró buenos resultados, particularmente similar a Random Forest, pero ligeramente inferiores.

Análisis de Resultados: El enfoque de ensamble de Bagging proporciona una mejora significativa en la precisión y la robustez del modelo al reducir la varianza. La configuración que mantuvo el DataFrame original demostró ser la más efectiva, capturando mejor la complejidad de los datos.

Bagging mostró un rendimiento comparable al de Random Forest, lo que destaca la eficacia de los modelos de ensamble para este tipo de problemas. Sin embargo, Random Forest sigue siendo ligeramente superior.

8. LightGBM

Eliminar la Cantidad:

- MSE: 193.01548609792803
- R^2: 0.8792705984927758

Multiplicar Cantidad por TargetPvP:

- MSE: 16793.577356583686
- R^2: 0.7908436488461636

Mantener el DataFrame como Está:

- MSE: 191.95049520670435
- R^2: 0.8799367404459807

Evaluación de Resultados: LightGBM también mostró un buen rendimiento, especialmente al mantener el DataFrame original, aunque fue ligeramente inferior a Random Forest.

Análisis de Resultados: LightGBM, como modelo de ensamble basado en árboles de decisión, maneja eficientemente grandes conjuntos de datos y captura interacciones complejas entre las características. La configuración original del DataFrame proporcionó los mejores resultados, similar a otros modelos de ensamble.

Comparación con Modelos Anteriores: Aunque LightGBM mostró un rendimiento robusto, fue ligeramente inferior a Random Forest. Sin embargo, sigue siendo superior a los modelos lineales y a K-NN, destacando su capacidad para manejar la complejidad de los datos.

Evaluación de Resultados Generales

Al evaluar los resultados obtenidos de los diferentes modelos aplicados a las tres configuraciones del dataset, se puede observar que los modelos basados en Árboles de Decisión, Random Forest y Bagging tienden a tener un mejor rendimiento general. Específicamente, el modelo Random Forest mostró consistentemente el menor MSE y el mayor R^2, indicando su capacidad superior para capturar la complejidad de los datos y proporcionar predicciones precisas.

Análisis de Resultados

- Regresión Lineal Múltiple y Ridge Regression: Ambos modelos mostraron resultados similares y no mejoraron significativamente con la regularización. Mantener el dataset original o eliminar la cantidad dio resultados casi idénticos, indicando que la variable Cantidad no es tan crítica en este contexto específico.
- Elastic Net: Similar a los modelos anteriores, Elastic Net no proporcionó mejoras significativas y sus resultados fueron consistentes en todas las configuraciones del dataset.
- Árboles de Decisión: Este modelo mejoró considerablemente el rendimiento, especialmente al mantener el dataset original, sugiriendo que la estructura del árbol puede capturar mejor las relaciones no lineales y las interacciones entre las variables.

- Random Forest: Proporcionó los mejores resultados, con el menor MSE y el mayor R^2, al mantener el dataset original. Esto indica que este modelo puede manejar la variabilidad y la complejidad de los datos de manera más efectiva.
- K-Nearest Neighbors (K-NN): No mostró mejoras notables y su rendimiento fue inferior en comparación con los modelos basados en árboles.
- Bagging: Mostró buenos resultados, similares a Random Forest, pero ligeramente inferiores, lo que indica que el enfoque de ensamble es beneficioso.
- LightGBM: Aunque mostró un buen rendimiento, fue ligeramente inferior a Random Forest, especialmente al mantener el dataset original.

4.4.2 Justificación de la Elección del Modelo Random Forest

Después de la evaluación, se decidió que el modelo Random Forest era el más adecuado para este proyecto debido a sus numerosas ventajas:

Robustez y Precisión: Random Forest es un método de ensamble que combina múltiples árboles de decisión para mejorar la precisión y reducir el riesgo de sobreajuste. Al promediar las predicciones de múltiples árboles, este modelo logra una mayor robustez y una mejor capacidad de generalización en comparación con los modelos individuales. Esto es especialmente importante para el cálculo del PVP, donde es crucial manejar la variabilidad y la complejidad de los datos.

Manejo de Datos Complejos: Random Forest es capaz de manejar grandes cantidades de datos y múltiples características sin requerir una preparación exhaustiva de los datos, como la normalización. Además, puede capturar interacciones no lineales entre las características, lo que es esencial para modelar relaciones complejas en los datos de producción de Láser Valencia SL.

Estimación de Importancia de Características: Otra ventaja significativa de Random Forest es su capacidad para proporcionar estimaciones de la importancia de las características. Esto permite identificar qué variables tienen el mayor impacto en el PVP, ofreciendo valiosa información para la toma de decisiones y la optimización de procesos.

Además, como dataframe se va utilizar ‘Mantener el DataFrame como Está’ debido ha que ha sido el que mejores resultados ha dado.

Implementación del Modelo Random Forest

La implementación del modelo Random Forest siguió un proceso estructurado que incluyó la división de los datos en conjuntos de entrenamiento y prueba, el ajuste de hiperparámetros mediante validación cruzada y la evaluación del desempeño del modelo.

1. División de Datos: Se dividieron los datos en conjuntos de entrenamiento y prueba para evaluar de manera objetiva el desempeño del modelo y evitar el sobreajuste.

2. Ajuste de Hiperparámetros: Se utilizaron técnicas de validación cruzada para ajustar los hiperparámetros del modelo, como el número de árboles en el bosque y la profundidad máxima de los árboles. Esto ayudó a optimizar el rendimiento del modelo.

3. Evaluación del Modelo: Se evaluó el modelo utilizando métricas como el coeficiente de determinación (R^2) y el error cuadrático medio (MSE) para asegurar que el modelo Random Forest no solo tenía un buen desempeño en el conjunto de entrenamiento, sino que también generalizaba bien a datos no vistos.

4.4.3 Elección de hipeparamétros

Descripción Detallada del Proceso de Búsqueda de Hiperparámetros para Random Forest

Este proceso se encarga de realizar una búsqueda de hiperparámetros para optimizar un modelo de Random Forest. El objetivo es encontrar la combinación de hiperparámetros que proporcione el mejor rendimiento en términos de precisión y capacidad predictiva. A continuación, se describe en detalle cada parte del proceso y su utilidad:

Definición de la Búsqueda de Hiperparámetros

Primero, se define una lista de posibles valores para varios hiperparámetros importantes del modelo de Random Forest:

n_estimators: Se seleccionan múltiples valores que representan el número de árboles en el bosque. Estos valores varían desde 100 hasta 300, incrementándose en pasos equidistantes. Esto permite evaluar cómo el tamaño del bosque afecta el rendimiento del modelo.

max_features: Diferentes estrategias para seleccionar el número máximo de características a considerar en cada división. Las opciones incluyen:

- Utilizar todas las características disponibles.

- Utilizar la raíz cuadrada del número de características.

- Utilizar el logaritmo base 2 del número de características.

max_depth: Valores que representan la profundidad máxima de los árboles. Se incluyen profundidades específicas entre 10 y 110, además de la opción de no limitar la profundidad (lo que permite a los árboles crecer hasta que todas las hojas sean puras o contengan menos muestras que un mínimo especificado).

min_samples_split: Valores mínimos de muestras requeridas para dividir un nodo interno. Esto ayuda a controlar la creación de nodos muy pequeños, lo que podría llevar a un modelo sobreajustado.

min_samples_leaf: Valores mínimos de muestras que deben estar presentes en un nodo hoja. Esto previene la creación de nodos hoja con muy pocas muestras, lo que puede hacer que el modelo sea más robusto.

bootstrap: Indica si se debe utilizar la técnica de bootstrap para crear muestras de entrenamiento. Esta técnica ayuda a mejorar la generalización del modelo.

Configuración de la Búsqueda de Hiperparámetros

Se configura la búsqueda aleatoria de hiperparámetros utilizando la clase `RandomizedSearchCV` de Scikit-learn. Este proceso implica:

Modelo Base: El modelo de Random Forest que se va a optimizar.

Distribución de Hiperparámetros: La distribución de posibles valores definida anteriormente.

Iteraciones: Se especifica el número de combinaciones aleatorias de hiperparámetros que se probarán. En este caso, se prueban 100 combinaciones diferentes.

Validación Cruzada: Se utiliza la validación cruzada con 3 pliegues para evaluar cada combinación de hiperparámetros. Esto ayuda a garantizar que los resultados sean robustos y no específicos a una única partición de los datos.

Paralelización: Se configura para utilizar todos los núcleos disponibles de la CPU, lo que acelera el proceso de búsqueda.

Ajuste de la Búsqueda

El proceso de ajuste implica entrenar múltiples modelos de Random Forest utilizando diferentes combinaciones de hiperparámetros y evaluarlos utilizando la validación cruzada. Esto permite identificar la combinación de hiperparámetros que proporciona el mejor rendimiento en términos de precisión y capacidad predictiva.

Selección del Mejor Modelo

Una vez completada la búsqueda, se selecciona el modelo con el mejor rendimiento. Los mejores hiperparámetros se imprimen para referencia futura. Esto incluye el número óptimo de árboles, la máxima cantidad de características a considerar en cada división, la profundidad máxima de los árboles, los parámetros de división de nodos y la configuración de bootstrap.

Evaluación del Mejor Modelo

El modelo optimizado se evalúa en el conjunto de prueba para asegurarse de que generaliza bien a datos nuevos. Se calculan métricas como el error cuadrático medio (MSE) y el coeficiente de determinación (R^2) para cuantificar su rendimiento. Un bajo MSE indica que las predicciones del modelo están muy cerca de los valores reales, mientras que un alto R^2 indica que el modelo explica una gran proporción de la variabilidad en los datos.

Utilidad del Proceso

Este proceso es crucial para maximizar el rendimiento del modelo de Random Forest. La búsqueda de hiperparámetros ayuda a:

1. Mejorar la Precisión: Encontrar la combinación óptima de hiperparámetros que maximiza la precisión del modelo.

2. Evitar el Sobreajuste: Ajustar parámetros como la profundidad máxima y el número mínimo de muestras por nodo para evitar que el modelo se ajuste demasiado a los datos de entrenamiento.

3. Optimizar el Tiempo de Cálculo: Seleccionar el número adecuado de árboles y la técnica de bootstrap para equilibrar la precisión del modelo y el tiempo de entrenamiento.

En resumen, este proceso de búsqueda de hiperparámetros es esencial para desarrollar un modelo de Random Forest robusto y preciso, adecuado para predecir el precio de venta al público en Láser Valencia SL.

Explicación de los Resultados de la Elección de Hiperparámetros

Tras realizar una búsqueda exhaustiva de hiperparámetros, se ha optimizado el modelo de Random Forest, identificando la siguiente configuración como la más adecuada:

A continuación, se explican los resultados de cada uno de estos hiperparámetros y las razones por las cuales han sido seleccionados:

n_estimators: 300

El número de árboles en el bosque (`n_estimators`) se ha fijado en 300. Este valor representa un equilibrio entre la precisión y la eficiencia computacional. Un mayor número de árboles generalmente mejora la capacidad del modelo para capturar patrones complejos en los datos, reduciendo la varianza y aumentando la robustez del modelo. Sin embargo, un número demasiado alto puede incrementar significativamente el tiempo de entrenamiento y los recursos computacionales requeridos. Al fijar este valor en 300, se logra un buen compromiso que mejora la precisión del modelo sin incurrir en un costo computacional prohibitivo.

min_samples_split: 2

El número mínimo de muestras requeridas para dividir un nodo interno (`min_samples_split`) se ha establecido en 2. Este valor permite que los nodos se dividan con la máxima granularidad posible, lo que facilita la creación de divisiones muy específicas que pueden capturar patrones detallados en los datos. Sin embargo, aunque un valor muy bajo como 2 puede aumentar el riesgo de sobreajuste, en combinación con otros hiperparámetros bien ajustados, ayuda a mejorar la precisión del modelo sin comprometer su capacidad de generalización.

min_samples_leaf: 1

El número mínimo de muestras en un nodo hoja (`min_samples_leaf`) se ha fijado en 1. Esto significa que un nodo hoja puede contener solo una muestra, permitiendo que el modelo capture las características más finas de los datos. Este ajuste es especialmente útil cuando se trabaja con conjuntos de datos complejos y variados, ya que permite al modelo adaptarse a las variaciones más pequeñas. Al igual que con `min_samples_split`, aunque puede aumentar el riesgo de sobreajuste, en combinación con una profundidad de árbol bien regulada y otras técnicas, contribuye significativamente a la precisión del modelo.

max_features: sqrt

El número máximo de características a considerar en cada división (`max_features`) se ha establecido como la raíz cuadrada del número total de características. Esta configuración es comúnmente utilizada en modelos de Random Forest porque proporciona un buen equilibrio entre reducir la correlación entre los árboles individuales y preservar suficiente información para realizar divisiones efectivas. Al considerar solo una parte de las características en cada división, se fomenta la diversidad entre los árboles del bosque, lo que mejora la robustez y la capacidad de generalización del modelo.

max_depth: 100

La profundidad máxima de los árboles (`max_depth`) se ha ajustado a 100. Esto permite a los árboles crecer bastante profundos, capturando relaciones complejas en los datos. En este caso, la profundidad de 100 ha sido seleccionada para permitir suficiente flexibilidad en el modelo, capturando los patrones necesarios sin llegar a sobreajustar significativamente, gracias a la regulación proporcionada por los otros hiperparámetros.

bootstrap: False

La técnica de bootstrap (`bootstrap`) se ha desactivado, lo que significa que cada árbol se construye sin reemplazo. Esto puede llevar a una reducción en la varianza del modelo al asegurar que cada árbol se entrene en un subconjunto más variado de los datos. Aunque la técnica de bootstrap es útil para aumentar la diversidad de los árboles, en este caso, entrenar sin reemplazo ha proporcionado mejores resultados en términos de precisión, sugiriendo que los datos sin reemplazo capturan mejor las variaciones relevantes para las predicciones.

Análisis de los Resultados

El modelo optimizado ha proporcionado los siguientes resultados en el conjunto de prueba:

MSE

Un MSE de 145.69574325459772 indica que, en promedio, las predicciones del modelo difieren del valor real en aproximadamente 12.07 (la raíz cuadrada del MSE). Este bajo valor sugiere que el modelo tiene una alta precisión en sus predicciones, minimizando los errores.

R^2

Un R^2 de 0.9088686600185362 significa que el modelo explica aproximadamente el 90.89% de la variabilidad en los datos de prueba. Este alto valor de R^2 indica que el modelo captura la mayoría de las variaciones en los datos, proporcionando predicciones precisas y fiables.

4.5 Prueba del modelo

En este apartado, se implementa una prueba del modelo de Random Forest entrenado para predecir el precio de venta al público (TargetPvP) basándose en las características proporcionadas por el usuario. Este proceso simula cómo funcionaría la aplicación final en un entorno real, permitiendo la entrada interactiva de datos y mostrando la predicción resultante. A continuación, se detalla cada paso del proceso y se analiza su funcionalidad.

Entrada de Datos por el Usuario

El primer paso del proceso implica la recopilación de datos directamente del usuario. Se solicita al usuario que introduzca valores para cada una de las características relevantes del modelo.

Estos valores se almacenan en un diccionario, donde cada característica se convierte en una clave y el valor introducido por el usuario es el valor asociado a esa clave.

Conversión a DataFrame

Una vez que se recopilan todos los datos del usuario, el diccionario se convierte en un DataFrame de pandas. Este DataFrame es la estructura de datos que el modelo de Random Forest utiliza para realizar predicciones. La conversión es crucial porque el modelo ha sido entrenado y está configurado para trabajar con DataFrames, garantizando así la compatibilidad de los datos de entrada con el modelo.

Predicción del TargetPvP

El DataFrame con los datos del usuario se pasa al modelo de Random Forest entrenado para predecir el TargetPvP. El modelo procesa las características y genera una predicción basada en el conocimiento adquirido durante el entrenamiento. Este proceso es rápido y eficiente, gracias a la capacidad del modelo de manejar múltiples características y capturar las relaciones complejas entre ellas.

Salida de la Predicción

Finalmente, la predicción del TargetPvP se muestra al usuario. Este valor representa el precio de venta al público estimado para un pedido con las características especificadas. La salida es clara y directa, proporcionando al usuario una comprensión inmediata del resultado.

4.5.1 Ejemplo de Funcionamiento

Para ilustrar el proceso, consideremos el siguiente ejemplo de funcionamiento:

1. Entrada de Datos:

```
Introduce Espesor:  10
Introduce Cantidad:  100
Introduce Ttos:  3
Introduce Peso:  100
Introduce Largo:  100
Introduce Ancho:  50
Introduce Ttops:  300
Introduce PrecioMP:  10
```

2. Conversión a DataFrame:

Los datos introducidos se convierten en un DataFrame que el modelo puede utilizar para realizar predicciones.

3. Predicción del TargetPvP:

El modelo de Random Forest utiliza los datos del DataFrame para predecir el TargetPvP.

4. Salida de la Predicción:

```
Predicción del TargetPvP para los datos introducidos: 288.54007318084246
```

Análisis del Proceso

Ventajas del Proceso:

Interactividad: Permite la entrada de datos en tiempo real, facilitando la obtención de predicciones inmediatas.

Compatibilidad: Utiliza estructuras de datos (DataFrames) que son estándar en el manejo de grandes conjuntos de datos y modelos de machine learning.

Precisión: Basado en un modelo de Random Forest optimizado, garantiza que las predicciones sean precisas y fiables.

Usabilidad: La interfaz de entrada es sencilla y directa, lo que facilita su uso incluso por personas sin conocimientos técnicos avanzados.

Consideraciones:

Validación de Entrada: En un entorno real, sería recomendable añadir validaciones para asegurarse de que los datos introducidos por el usuario son correctos y están en el formato esperado.

Escalabilidad: Este enfoque es fácilmente escalable, permitiendo integrar el modelo en una aplicación más amplia que podría incluir una interfaz gráfica de usuario (GUI) o una API web para accesibilidad remota.

En resumen, esta prueba del modelo no solo valida su funcionalidad y precisión, sino que también ilustra cómo podría implementarse en una aplicación práctica, proporcionando valor inmediato y directo a los usuarios finales.

5. Análisis de Resultados

En este apartado se realiza una evaluación detallada del rendimiento del modelo de Random Forest utilizado para predecir el precio de venta al público (TargetPvP) en Láser Valencia SL. El objetivo es cuantificar la precisión y la capacidad predictiva del modelo a través de diversas métricas y compararlas con un modelo base simple que predice la media.

Evaluación del modelo de ML

La evaluación del modelo de Machine Learning es un paso fundamental para asegurar que las predicciones generadas son precisas y fiables. Para este análisis, se han calculado varias métricas estadísticas y de error que nos permiten comprender mejor el rendimiento del modelo de Random Forest en el contexto de nuestros datos.

- **Media de TargetPvP:** La media del TargetPvP en el conjunto de entrenamiento proporciona un punto de referencia inicial sobre el valor promedio que el modelo intenta predecir.
- **Varianza de TargetPvP:** La varianza indica la dispersión de los valores de TargetPvP alrededor de la media. Una varianza alta sugiere que los datos tienen una amplia variabilidad.
- **Desviación Estándar de TargetPvP:** La desviación estándar es la raíz cuadrada de la varianza y ofrece una medida directa de la dispersión de los datos.
- **MSE del Modelo de Random Forest:** El Error Cuadrático Medio (MSE) en el conjunto de prueba muestra qué tan cerca están las predicciones del modelo de los valores reales. Un MSE bajo indica una alta precisión en las predicciones.
- **MSE del Modelo Base:** El MSE del modelo que siempre predice la media sirve como un punto de comparación para evaluar la mejora proporcionada por el modelo de Random Forest.
- **Tamaño del Error Relativo:** Este valor compara el MSE del modelo de Random Forest con la varianza de TargetPvP, proporcionando una medida de la proporción de la variabilidad de los datos que no es capturada por el modelo.
- **Coeficiente de Determinación (R^2):** El R^2 indica la proporción de la variabilidad en los datos que es explicada por el modelo. Un valor cercano a 1 sugiere un alto poder explicativo.

Análisis Detallado de los Resultados

Media de TargetPvP: 17.05846293406678

La media de TargetPvP proporciona una referencia básica sobre el valor promedio que el modelo intenta predecir.

Varianza de TargetPvP: 1653.9118816907055

La varianza de TargetPvP indica una alta dispersión de los valores alrededor de la media. Esto sugiere que los precios de venta tienen una variabilidad significativa, lo que puede implicar desafíos adicionales para el modelo de predicción debido a la naturaleza diversa de los datos.

Desviación Estándar de TargetPvP: 40.66831545184415

La desviación estándar refuerza la observación de la alta variabilidad en los datos. Un valor de 40.66831545184415 indica que los precios pueden desviarse considerablemente de la media, lo que requiere un modelo robusto capaz de manejar esta dispersión.

MSE - Random Forest (test): 164.85228096727693

El MSE del modelo de Random Forest en el conjunto de prueba es 164.85228096727693. Este valor relativamente bajo indica que las predicciones del modelo están bastante cerca de los valores reales, reflejando una alta precisión y eficiencia del modelo en capturar los patrones en los datos.

MSE - Modelo Base (predicción media): 1598.7509790825413

El MSE del modelo base, que simplemente predice la media de TargetPvP, es significativamente mayor que el del modelo de Random Forest. Un MSE de 1598.7509790825413 demuestra que este enfoque sencillo es mucho menos preciso, subrayando la superioridad del modelo de Random Forest en términos de precisión predictiva.

Tamaño del Error Relativo (MSE/Varianza): 0.09967416208338577

El tamaño del error relativo compara el MSE del modelo de Random Forest con la varianza de TargetPvP. Un valor de 0.09967416208338577 indica que el modelo de Random Forest logra capturar aproximadamente el 90% de la variabilidad de los datos, dejando solo un 10% de variabilidad no explicada por el modelo. Esto refleja un alto nivel de precisión y capacidad explicativa.

R^2 - Random Forest: 0.9003258379166142

El coeficiente de determinación R^2 de 0.9003258379166142 es muy alto, indicando que el modelo de Random Forest explica el 90.03% de la variabilidad en los datos. Este resultado es excelente, mostrando que el modelo tiene un poder predictivo y explicativo muy fuerte.

Conclusión

El análisis de las métricas muestra que el modelo de Random Forest supera significativamente al modelo base en términos de precisión y capacidad predictiva. La combinación de un bajo MSE y un alto R^2 indica que el modelo de Random Forest es capaz de capturar la mayoría de las variaciones en los datos y hacer predicciones precisas del TargetPvP. Este rendimiento robusto hace que el modelo sea una herramienta valiosa para la predicción de precios en Láser Valencia SL, permitiendo decisiones informadas y optimizando los procesos comerciales basados en predicciones precisas.

6. Discusión

6.1 Logros del proyecto

Mejora en la Precisión de las Estimaciones

Uno de los logros más destacados del proyecto es la notable mejora en la precisión de las estimaciones de costos. La implementación del modelo de Random Forest ha proporcionado un R^2 superior al 90%, lo que indica que el modelo explica más del 90% de la variabilidad en los precios de venta al público (TargetPvP). Este nivel de precisión asegura que las decisiones de precio están bien fundamentadas y basadas en datos sólidos. Como resultado, la empresa puede establecer precios más competitivos y justificados, reduciendo el riesgo de subvaloración o sobrevaloración de sus productos.

Aumento de la Eficiencia Operativa

La automatización del proceso de cálculo del PVP ha liberado recursos que anteriormente se dedicaban a tareas manuales y repetitivas. Este cambio ha permitido que el personal se enfoque en actividades que agregan más valor, como la optimización de procesos, la mejora del servicio al cliente y el desarrollo de nuevas estrategias de mercado. La eficiencia operativa mejorada también se traduce en una reducción de errores humanos y una mayor consistencia en las estimaciones de precios, lo que fortalece la confianza en las decisiones empresariales.

Fortalecimiento de la Competitividad

En un mercado donde el tiempo de respuesta puede ser un diferenciador clave, la capacidad de proporcionar respuestas rápidas y precisas a las solicitudes de cotización mejora significativamente la posición competitiva de Láser Valencia SL. El modelo de Random Forest permite generar estimaciones de precios casi instantáneamente, lo que acelera el proceso de cotización y reduce el tiempo de espera para los clientes. Esta agilidad no solo mejora la satisfacción del cliente, sino que también aumenta las oportunidades de negocio al permitir que la empresa responda rápidamente a las demandas del mercado.

6.2Limitaciones y desafíos enfrentados

A pesar de los éxitos logrados, el proyecto enfrentó varias limitaciones y desafíos que influyeron en el proceso y los resultados. A continuación, se detallan estos aspectos y se analizan sus implicaciones:

Calidad de los Datos

La eficacia del modelo de Random Forest depende críticamente de la calidad de los datos ingresados. Durante las etapas iniciales del proyecto, se encontraron varios problemas con los datos disponibles, incluyendo registros incompletos, inconsistencias y errores en la captura de datos.

Impacto: Los problemas de calidad de datos pueden afectar gravemente la precisión y la fiabilidad de las predicciones del modelo. Datos incompletos o incorrectamente registrados pueden llevar a sesgos en el entrenamiento del modelo, resultando en predicciones imprecisas y decisiones mal fundamentadas.

Medidas Tomadas: Para abordar este desafío, se realizaron esfuerzos significativos en la limpieza y preparación de datos. Esto incluyó la identificación y corrección de errores, la imputación de valores faltantes y la normalización de los datos para asegurar su consistencia. Este proceso, aunque laborioso, fue crucial para garantizar la calidad y la integridad de los datos utilizados en el entrenamiento del modelo.

Complejidad del Modelo

El modelo de Random Forest es eficaz en manejar múltiples variables y sus interacciones, pero su complejidad puede dificultar la interpretación de los resultados. A diferencia de los modelos más simples, como la regresión lineal, donde la influencia de cada variable es fácilmente comprensible a través de los coeficientes, los modelos de Random Forest son menos transparentes.

Impacto: La complejidad del modelo puede ser un obstáculo para la comprensión completa de cómo las diferentes variables influyen en las predicciones. Esto puede dificultar la explicación de las decisiones del modelo a las partes interesadas que no están familiarizadas con técnicas avanzadas de machine learning.

Medidas Tomadas: Para mitigar este desafío, se implementaron técnicas de interpretabilidad de modelos. Estas técnicas ayudan a desglosar la contribución de cada variable en las predicciones del modelo, proporcionando una mejor comprensión y transparencia.

6.3 Implicación practica de la herramienta

Mejora en la Toma de Decisiones

La herramienta desarrollada proporciona predicciones precisas y rápidas del precio de venta al público (TargetPvP), lo que permite a los responsables de la toma de decisiones en Láser Valencia SL basar sus estrategias en datos sólidos y fiables.

Impacto:

-Reducción de la Incertidumbre: Con predicciones más precisas, la empresa puede reducir la incertidumbre en la fijación de precios, asegurando márgenes de beneficio adecuados y competitividad en el mercado.

-Optimización de Precios: La capacidad de ajustar precios de manera precisa y en tiempo real permite a la empresa optimizar sus estrategias de precios, adaptándose rápidamente a las condiciones del mercado y maximizando los ingresos.

Eficiencia y Ahorro de Recursos

La automatización del proceso de cálculo del PVP ha liberado recursos significativos que anteriormente se dedicaban a tareas manuales y repetitivas.

Impacto:

- Reducción de Errores: La automatización disminuye la probabilidad de errores humanos en el cálculo de precios, lo que mejora la precisión y la consistencia.

- Ahorro de Tiempo: El tiempo ahorrado en tareas manuales permite que el personal se enfoque en actividades de mayor valor agregado, como la mejora de procesos, la atención al cliente y el desarrollo de nuevas oportunidades de negocio.

- Escalabilidad: La herramienta puede manejar grandes volúmenes de datos y solicitudes de cotización, facilitando la expansión del negocio sin necesidad de incrementar proporcionalmente los recursos humanos.

Mejora de la Relación con el Cliente

Descripción: La herramienta permite proporcionar cotizaciones rápidas y precisas a los clientes, mejorando su experiencia y satisfacción.

Impacto:

- Respuesta Rápida: La capacidad de generar cotizaciones instantáneamente reduce el tiempo de espera para los clientes, lo que mejora la percepción del servicio y puede ser un diferenciador clave en la competitividad del mercado.

- Confianza y Transparencia: Predicciones precisas basadas en datos sólidos aumentan la confianza de los clientes en los precios ofrecidos, fortaleciendo las relaciones comerciales y fomentando la lealtad del cliente.

- Personalización: La herramienta permite ajustar cotizaciones basadas en las necesidades y características específicas de cada cliente, ofreciendo un servicio más personalizado y valor añadido.

Innovación y Posicionamiento Estratégico

Descripción: La adopción de tecnologías avanzadas de machine learning, como el modelo de Random Forest, posiciona a Láser Valencia SL como una empresa innovadora y líder en su sector.

Impacto:

- Ventaja Competitiva: La innovación tecnológica no solo mejora los procesos internos sino que también posiciona a la empresa como un referente en el uso de tecnología avanzada, atrayendo a clientes y socios comerciales que valoran la innovación.

- Preparación para el Futuro: La capacidad de adaptarse rápidamente a nuevas tecnologías y metodologías asegura que la empresa esté preparada para enfrentar futuros desafíos y aprovechar nuevas oportunidades en el mercado.

- Reputación y Marca: La percepción de Láser Valencia SL como una empresa moderna y tecnológicamente avanzada mejora su reputación y puede abrir nuevas oportunidades de negocio y colaboración.

7. Conclusiones y Recomendaciones

7.1 Conclusiones principales

Eficacia del Machine Learning

Precisión en la Estimación de Costos:

El modelo de Random Forest desarrollado e implementado en Láser Valencia SL ha demostrado ser altamente efectivo en la estimación precisa del precio de venta al público (TargetPvP). Con un coeficiente de determinación (R^2) superior al 90%, el modelo es capaz de explicar más del 90% de la variabilidad en los datos de precio de venta. Este alto valor de R^2 indica que el modelo captura adecuadamente las relaciones complejas entre las diferentes características de los productos y el precio de venta, proporcionando estimaciones precisas y fiables.

El error cuadrático medio (MSE) del modelo, que es de aproximadamente 145.7, refuerza aún más su precisión. Un MSE bajo indica que las predicciones del modelo están muy cerca de los valores reales observados, lo que es crucial para la toma de decisiones en la fijación de precios. La precisión en la estimación de costos es esencial para garantizar que los precios establecidos no solo cubran los costos de producción, sino que también sean competitivos en el mercado.

Comparación con Métodos Tradicionales:

En comparación con métodos tradicionales utilizados por las empresas para la fijación de precios, como el método del coste más margen y el análisis comparativo de mercado, el modelo de Random Forest ofrece ventajas significativas. Los métodos tradicionales a menudo no pueden capturar la complejidad y la interdependencia de múltiples variables que afectan los costos y los precios. Por ejemplo, el método del coste más margen es simple y fácil de aplicar, pero no considera factores externos como la demanda del mercado o la competencia. El análisis comparativo de mercado proporciona precios competitivos, pero puede no reflejar adecuadamente los costos internos.

El modelo de Random Forest, por otro lado, puede integrar una amplia variedad de variables internas y externas, permitiendo una estimación de precios mucho más precisa y adaptativa. Esto no solo mejora la precisión de las estimaciones, sino que también permite a la empresa responder rápidamente a las fluctuaciones del mercado y ajustar sus estrategias de precios en consecuencia.

Ventajas de la Técnica Random Forest:

El uso de Random Forest presenta varias ventajas sobre otros modelos de machine learning y métodos tradicionales:

- **Manejo de Datos Complejos:** Random Forest es eficaz en el manejo de conjuntos de datos grandes y complejos, con múltiples variables y sus interacciones. Esto es especialmente útil en entornos industriales donde las decisiones de precios pueden depender de numerosos factores.
- **Robustez y Generalización:** Debido a su estructura de múltiples árboles, Random Forest es menos susceptible al sobreajuste en comparación con modelos más simples. Esto significa que el modelo generaliza mejor a nuevos datos, manteniendo su precisión incluso cuando se presentan datos no vistos durante el entrenamiento.
- **Importancia de Características:** Random Forest permite la evaluación de la importancia de cada característica en el modelo, proporcionando información valiosa sobre qué factores tienen mayor impacto en las predicciones. Esta capacidad de interpretación ayuda a la empresa a entender mejor los drivers de costos y precios.

Aplicación Práctica:

La implementación práctica del modelo está permitiendo a Láser Valencia SL mejorar su proceso de fijación de precios de manera significativa. La herramienta proporciona predicciones rápidas y precisas, lo que reduce el tiempo necesario para generar cotizaciones y mejora la capacidad de respuesta de la empresa. Esto es particularmente importante en un entorno competitivo donde la velocidad y la precisión en la cotización pueden ser diferenciadores clave.

Además, la herramienta está siendo integrada en el flujo de trabajo diario de la empresa, facilitando su uso por parte del personal encargado de las decisiones de precios. La capacitación adecuada y la gestión del cambio han sido fundamentales para asegurar la aceptación y el uso efectivo de la herramienta, maximizando así su impacto positivo.

Impacto Estratégico

El impacto del modelo de Random Forest va más allá de la mejora en la precisión de las estimaciones de costos. Al proporcionar una base sólida y precisa para la fijación de precios, la empresa puede tomar decisiones más informadas y estratégicas. Esto incluye la capacidad de ajustar precios rápidamente en respuesta a cambios en el mercado, optimizar márgenes de beneficio y mejorar la satisfacción del cliente al ofrecer precios competitivos y justificados.

7.2 Recomendaciones para futuros proyectos similares

7.2.1 Exploración de Otros Modelos de Machine Learning

Aunque el modelo de Random Forest ha demostrado ser altamente efectivo, es recomendable explorar y comparar otros modelos de machine learning para identificar oportunidades de mejora en la precisión y la eficiencia del modelo. Modelos como las redes neuronales profundas (deep learning), algoritmos de boosting (e.g., XGBoost, LightGBM), y métodos de ensamble pueden ofrecer diferentes ventajas y abordar limitaciones específicas del Random Forest.

Redes Neuronales Profundas (Deep Learning):

Ventajas: Las redes neuronales profundas son capaces de capturar patrones extremadamente complejos y no lineales en los datos, lo que puede mejorar la precisión de las predicciones en contextos de datos muy variados y ricos en características.

Aplicación: Implementar redes neuronales con múltiples capas y unidades para analizar si pueden superar a Random Forest en términos de precisión y capacidad de generalización.

Algoritmos de Boosting (e.g., XGBoost, LightGBM):

Ventajas: Los algoritmos de boosting mejoran iterativamente el rendimiento del modelo al corregir los errores de modelos anteriores. Son conocidos por su alta precisión y eficiencia.

Aplicación: Realizar pruebas comparativas entre Random Forest y modelos de boosting para evaluar mejoras en la precisión y la capacidad de manejo de datos desbalanceados o complejos.

Métodos de Ensamble:

Ventajas: Combinar múltiples modelos puede mejorar la robustez y la precisión de las predicciones. Métodos de ensamble como stacking, bagging y blending permiten integrar las ventajas de diferentes algoritmos.

Aplicación: Crear modelos de ensamble que incorporen Random Forest, deep learning y boosting para evaluar si una combinación de métodos proporciona mejores resultados.

7.2.2 Estudio de Factores Adicionales

Incorporar factores externos en el modelo puede aumentar su robustez y precisión. Factores como las fluctuaciones en el mercado de materias primas, las condiciones económicas globales, y otros elementos externos pueden influir significativamente en los costos y precios.

Integración de Datos Económicos:

Datos de Mercado de Materias Primas: Incorporar datos sobre los precios históricos y actuales de las materias primas utilizadas en la producción. Estos datos pueden ayudar a ajustar los costos estimados de manera más precisa.

Indicadores Económicos Globales: Utilizar indicadores económicos como la inflación, las tasas de cambio, y las tasas de interés para ajustar las predicciones en función de las condiciones económicas globales.

Modelado de Factores de Demanda:

Análisis de Tendencias de Demanda: Incorporar datos de demanda históricos y previsiones de demanda futura para ajustar los precios basados en las expectativas del mercado.

Estacionalidad y Eventos Especiales: Incluir variables que capturen la estacionalidad y eventos especiales (e.g., promociones, ferias comerciales) que pueden afectar la demanda y, por ende, el precio de los productos.

Análisis de la Competencia:

Precios de Competidores: Monitorizar y analizar los precios de los competidores para ajustar las estrategias de fijación de precios y asegurarse de que sean competitivos.

Benchmarking Dinámico: Desarrollar modelos que puedan adaptarse rápidamente a los cambios en los precios de los competidores y ajustar los precios en consecuencia.

7.2.3 Análisis de Impacto a Largo Plazo

Realizar estudios longitudinales para evaluar el impacto a largo plazo de la herramienta de estimación de costos en la rentabilidad y en las relaciones con los clientes proporciona una visión más completa de sus beneficios sostenidos.

Evaluación de Rentabilidad:

Análisis de Rentabilidad a Largo Plazo: Realizar un seguimiento continuo de los márgenes de beneficio, los ingresos y los costos operativos para evaluar cómo la herramienta afecta la rentabilidad a lo largo del tiempo.

Comparación con Períodos Pre-Implementación: Comparar los datos financieros actuales con los datos históricos anteriores a la implementación de la herramienta para identificar mejoras y áreas de oportunidad.

Relaciones con los Clientes:

Satisfacción del Cliente: Realizar encuestas y estudios para evaluar la satisfacción del cliente con respecto a los precios ofrecidos y el tiempo de respuesta en las cotizaciones.

Retención de Clientes: Analizar las tasas de retención de clientes y la frecuencia de nuevas adquisiciones de clientes para medir el impacto de la herramienta en la fidelización y expansión de la base de clientes.

Ajustes y Mejoras Continuas:

Feedback Regular: Establecer canales para recibir feedback regular de los usuarios de la herramienta y de los clientes para identificar áreas de mejora y ajustar el modelo y la estrategia de precios en consecuencia.

Actualizaciones Periódicas del Modelo: Planificar y ejecutar actualizaciones periódicas del modelo para incorporar nuevos datos y ajustar los parámetros en función de los cambios en el mercado y las operaciones de la empresa.

7.3 Potenciales mejoras y desarrollos futuros

A lo largo del desarrollo y la implementación del modelo de Random Forest para la estimación de costos en Láser Valencia SL, se han identificado varias áreas con potencial para mejoras y desarrollos futuros. Estas mejoras no solo buscan optimizar el rendimiento del modelo, sino también expandir su aplicabilidad y utilidad dentro de la empresa.

Integración de Realimentación en Tiempo Real

Incorporar un sistema de realimentación en tiempo real permite ajustar el modelo basándose en nuevas informaciones y resultados observados continuamente. Esto ayuda a mantener la precisión del modelo a medida que cambian las condiciones del mercado y los datos de entrada.

Potenciales Mejoras

1.Monitoreo Continuo de Desempeño:

Implementación: Desarrollar un sistema que monitoree continuamente el rendimiento del modelo en tiempo real, comparando las predicciones con los resultados reales.

Beneficios: Permite identificar rápidamente cualquier desviación en la precisión del modelo y realizar ajustes proactivos para corregir errores y mejorar la fiabilidad.

2.Ajustes Automáticos:

Implementación: Configurar el modelo para que ajuste automáticamente sus parámetros en función de la realimentación recibida. Esto podría incluir ajustes en los hiperparámetros del modelo o en las técnicas de preprocesamiento de datos.

Beneficios: Mantiene el modelo actualizado y optimizado sin intervención manual constante, asegurando que siempre esté operando al máximo rendimiento.

3. Incorporación de Datos Recientes:

Implementación: Integrar un sistema que permita la incorporación de datos recientes y relevantes en el modelo de forma automática. Esto incluye datos de ventas, costos de producción y precios de mercado.

Beneficios: Mejora la precisión y relevancia del modelo al basarse en la información más actualizada disponible.

Desarrollo de Dashboards Interactivos

Los dashboards interactivos proporcionan una interfaz visual que permite a los usuarios explorar los datos de entrada y salida del modelo de manera intuitiva. Estos dashboards pueden facilitar el análisis exploratorio y permitir ajustes dinámicos de los parámetros del modelo.

Potenciales Mejoras:

1. Visualización de Datos:

Implementación: Crear dashboards que muestren visualizaciones claras y comprensibles de los datos utilizados por el modelo, incluyendo gráficos de tendencias, correlaciones y distribuciones.

Beneficios: Ayuda a los usuarios a entender mejor los datos subyacentes y a identificar patrones y relaciones clave.

2. Interactividad y Ajuste de Parámetros:

Implementación: Permitir a los usuarios ajustar parámetros del modelo directamente desde el dashboard, como los pesos de las características o los hiperparámetros de los algoritmos.

Beneficios: Proporciona flexibilidad para experimentar con diferentes configuraciones y observar cómo afectan las predicciones, facilitando un enfoque más dinámico y adaptativo en la toma de decisiones.

3. Alertas y Notificaciones:

Implementación: Integrar alertas y notificaciones automáticas en el dashboard para informar a los usuarios sobre cambios significativos en los datos o el rendimiento del modelo.

Beneficios: Mantiene a los usuarios informados sobre cualquier problema potencial y permite respuestas rápidas para mantener la precisión y la fiabilidad del modelo.

Expansión a Otras Áreas de Negocio

La tecnología de machine learning utilizada para la estimación de costos tiene el potencial de ser aplicada en otras áreas de la empresa, como la gestión de inventarios, la planificación de la producción y la optimización de la cadena de suministro.

Potenciales Mejoras:

1. Gestión de Inventarios:

Implementación: Desarrollar modelos de machine learning para predecir la demanda de productos y optimizar los niveles de inventario. Esto puede incluir la predicción de ventas futuras y la identificación de patrones de demanda estacionales.

Beneficios: Ayuda a reducir costos de almacenamiento y minimizar el riesgo de desabastecimiento, mejorando la eficiencia operativa y la satisfacción del cliente.

2. Planificación de la Producción:

Implementación: Utilizar algoritmos de machine learning para optimizar la programación de la producción, asegurando que los recursos se utilicen de manera eficiente y que se cumplan los plazos de entrega.

Beneficios: Mejora la productividad y reduce el tiempo de inactividad en las líneas de producción, lo que puede aumentar la rentabilidad.

3. Optimización de la Cadena de Suministro:

Implementación: Aplicar técnicas de machine learning para optimizar la cadena de suministro, desde la adquisición de materias primas hasta la entrega final al cliente.

Esto puede incluir la predicción de plazos de entrega y la optimización de rutas de transporte.

Beneficios: Reduce costos logísticos y mejora la eficiencia del suministro, asegurando que los productos lleguen a tiempo y en buen estado

8. Bibliografía

Scikit-learn.org:

La documentación oficial de Scikit-learn es la principal fuente de información que he utilizado para la realización del proyecto, contiene información tanto para entender la teoría detrás de los algoritmos de machine learning como para ver ejemplos de código en Python. Incluye tutoriales, ejemplos y explicaciones detalladas de los algoritmos.

Ensembles: Gradient boosting, random forests, bagging, voting, stacking:

Dentro de Scikit-learn esta es la sección específica de la documentación sobre Random Forest. Explica la teoría detrás del algoritmo, cómo funciona y proporciona ejemplos de código para implementarlo en Python.

Además de Scikit-learn (que es la principal), estas son otras páginas relacionadas con el Machine learning que he utilizado para la investigación de mi proyecto:

datacamp

ciencia de datos.net machine learning

ciencia de datos.net random forest.

Los datos e información sobre la empresa y el Sector han sido aportados por la propia Empresa. En cuanto a los datos del sectos me han enviado informes de FEMEVAL.

9. Anexos

9.1 Código Fuente del Proyecto

El código fuente del proyecto consta de 3 versiones, las cuales a su vez tienen dos formatos (html y ipynb), por lo que hay 6 anexos en este apartado:

Anexo I: El anexo 1 es el código entero en formato html para facilitar su lectura. En este código se sigue todo el proceso que he realizado para llegar al resultado final del proyecto.

Anexo II: Este es el mismo código que le Anexo I, pero en formato ipynb.

Anexo III: Este código es una versión muy reducida del Anexo I donde se elimina gran parte del EDA y exploración de diferentes modelos, realizando directamente el modelo de Random Forest y añadiendo un apartado en el que se exporta el modelo ya entrenado para su futuro uso. En formato html.

Anexo IV: Este código es el mismo que el Anexo III, pero en formato ipynb.

Anexo V: Este código es un código reducido donde se utiliza el modelo exportado con una nueva entrada de datos, para realizar una predicción con estos. Esta es una versión previa al estilo MVP para entregar a la empresa. En formato html.

Anexo VI: Este código es igual al Anexo V, pero en Formato ipynb.

9.2 Datos Utilizados

Anexo VII: Base de datos entregada por la empresa “DatosOriginales.xlsx” en formato Excel.

9.3 Modelo Exportado

Anexo VIII: Modelo Random Forest Exportado “model_rf.joblib”