1. Introducción

En el contexto actual, la **predicción de ingresos** se ha convertido en una herramienta clave para la toma de decisiones estratégicas en entornos empresariales. Anticipar la evolución de las ventas permite planificar recursos, ajustar costes operativos y evaluar distintos escenarios futuros con mayor fiabilidad.

Este proyecto tiene como objetivo el **desarrollo de un sistema de predicción de ingresos mensuales** basado en datos históricos de ventas, incorporando técnicas de análisis exploratorio de datos, *feature engineering* y modelos de *machine learning*. La solución desarrollada no se limita únicamente al entrenamiento de un modelo, sino que abarca todo el ciclo de vida del dato: desde la ingestión y limpieza de información hasta la generación de predicciones accesibles a través de una interfaz web.

Los datos utilizados corresponden a registros de ventas a nivel diario, segmentados por tienda y canal de venta. A partir de esta información, se construye una serie temporal mensual que permite capturar tendencias, estacionalidades y patrones recurrentes del negocio.

Antes de abordar la fase de modelado, se realiza un **Análisis Exploratorio de Datos (EDA)** con el fin de comprender el comportamiento de las variables, evaluar la calidad de los datos y orientar las decisiones técnicas posteriores. Este análisis inicial resulta fundamental para justificar tanto la selección de variables como el enfoque de modelado adoptado.

El resultado final del proyecto es una aplicación web que permite:

- Subir nuevos datos de ventas.
- Reentrenar el modelo de forma controlada.
- Generar predicciones futuras por tienda, canal o de forma agregada.
- Consultar métricas de evaluación del modelo.

De este modo, el proyecto presenta una solución completa y extensible para la predicción de ingresos, combinando análisis de datos, modelado predictivo y despliegue práctico.

2. Análisis Exploratorio de Datos (EDA)

Como primer paso del proyecto se realizó un **Análisis Exploratorio de Datos (EDA)** con el objetivo de comprender la estructura de los datos disponibles, evaluar su calidad y detectar patrones relevantes antes de abordar cualquier fase de modelado.

Este análisis inicial permitió validar supuestos, identificar posibles problemas en los datos y orientar las decisiones posteriores de preprocesado y selección de variables.

El EDA se dividió en dos bloques principales, en función de la naturaleza de la información analizada:

- **EDA de Ventas**
- **EDA de Payroll**

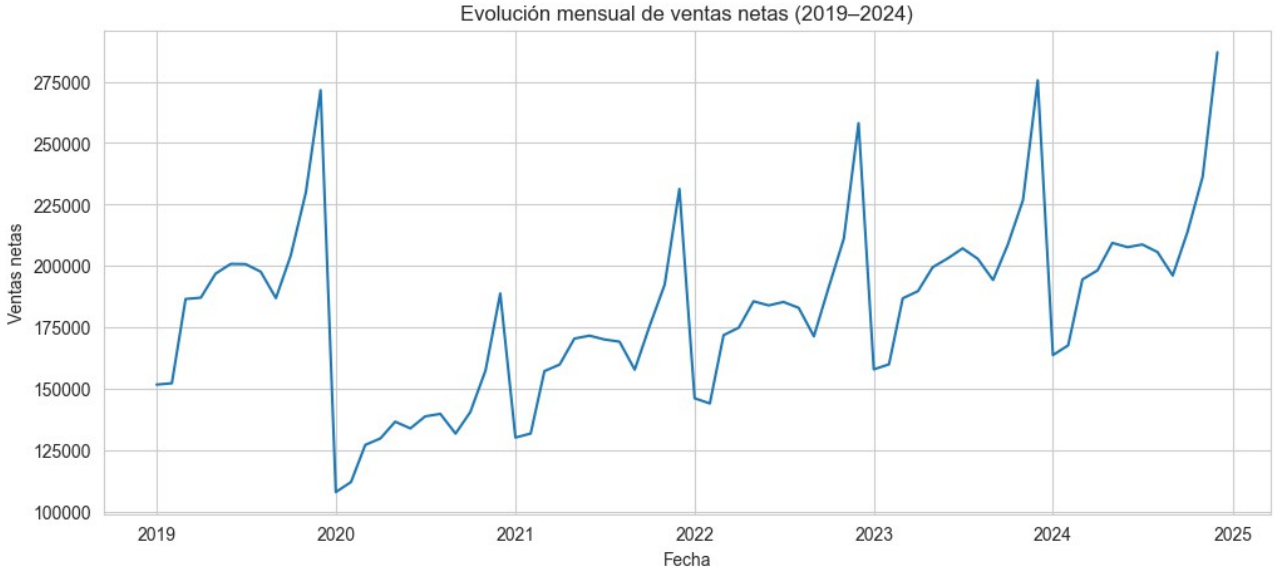
2.1 EDA de Ventas

El EDA de ventas se centró en analizar el comportamiento de los ingresos y transacciones a lo largo

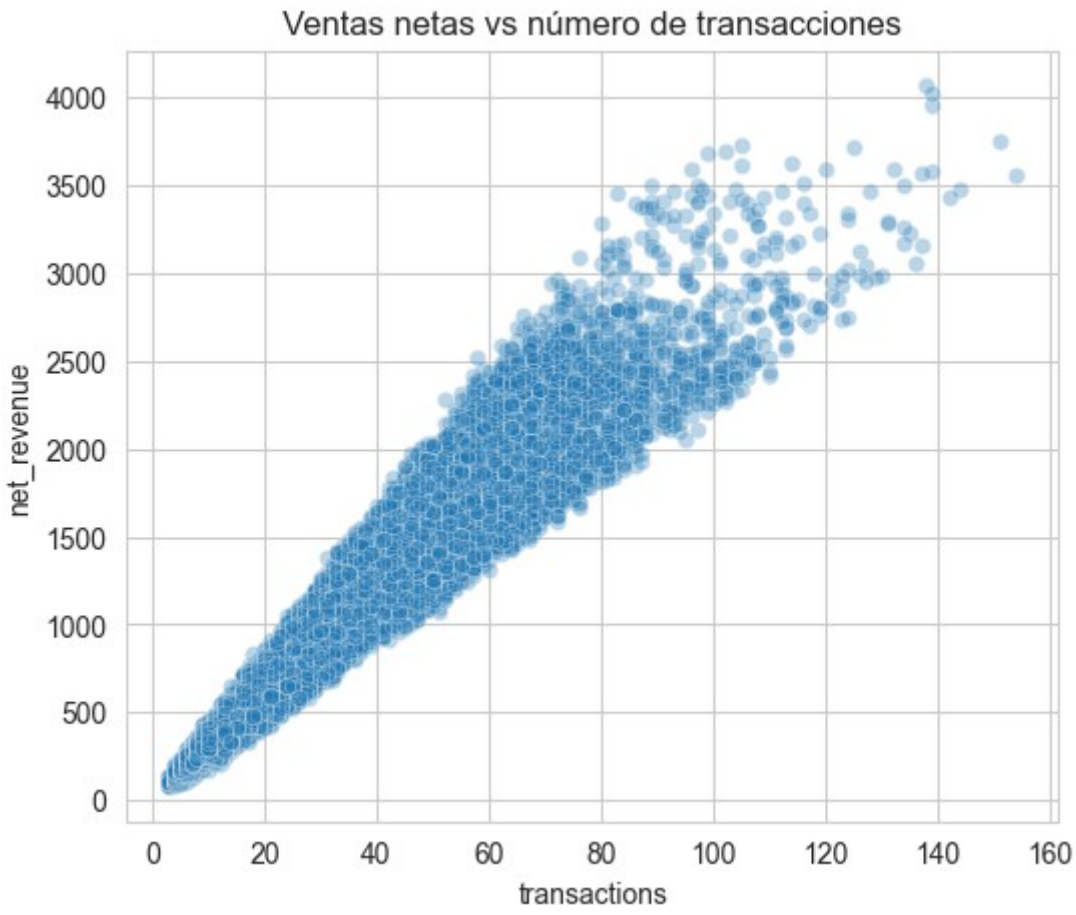
del tiempo, así como su distribución por tiendas y canales de venta.

Objetivos principales

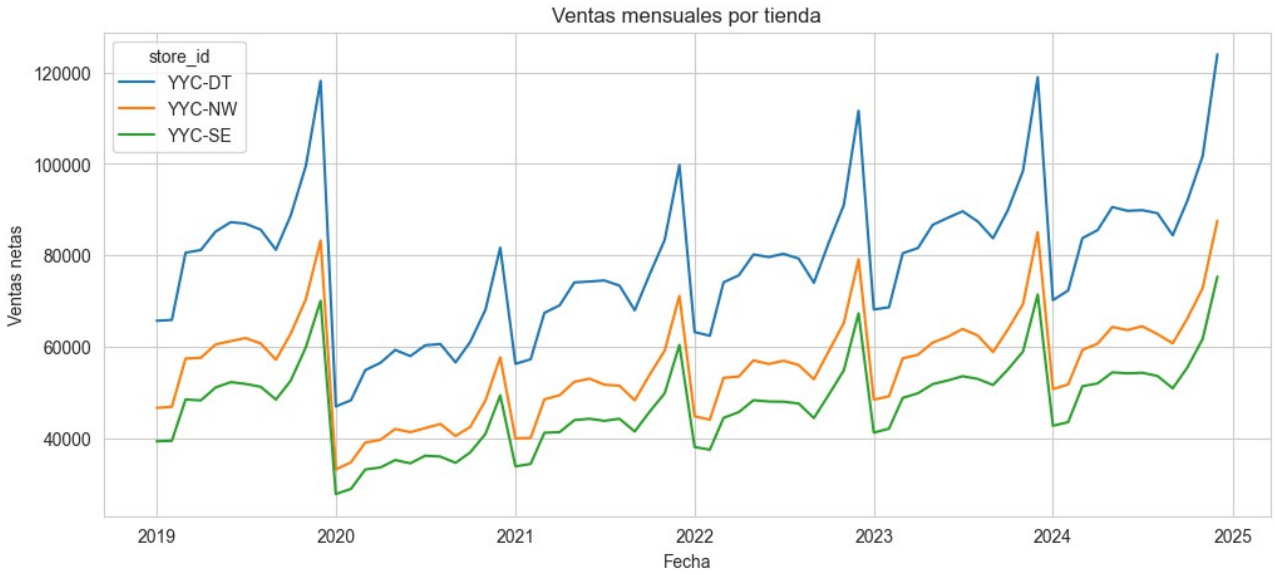
- Comprender la evolución temporal de las ventas.
- Detectar estacionalidades, tendencias y posibles anomalías.



- Analizar la relación entre ingresos y número de transacciones.



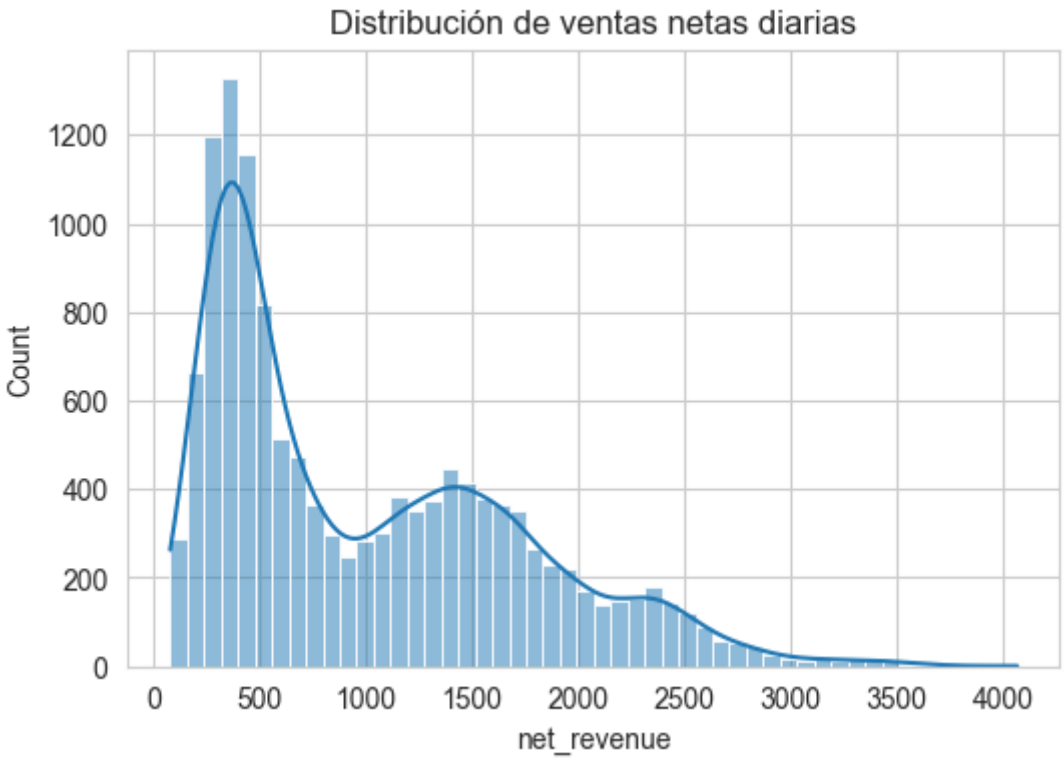
- Evaluar diferencias entre tiendas y canales.



Análisis realizados

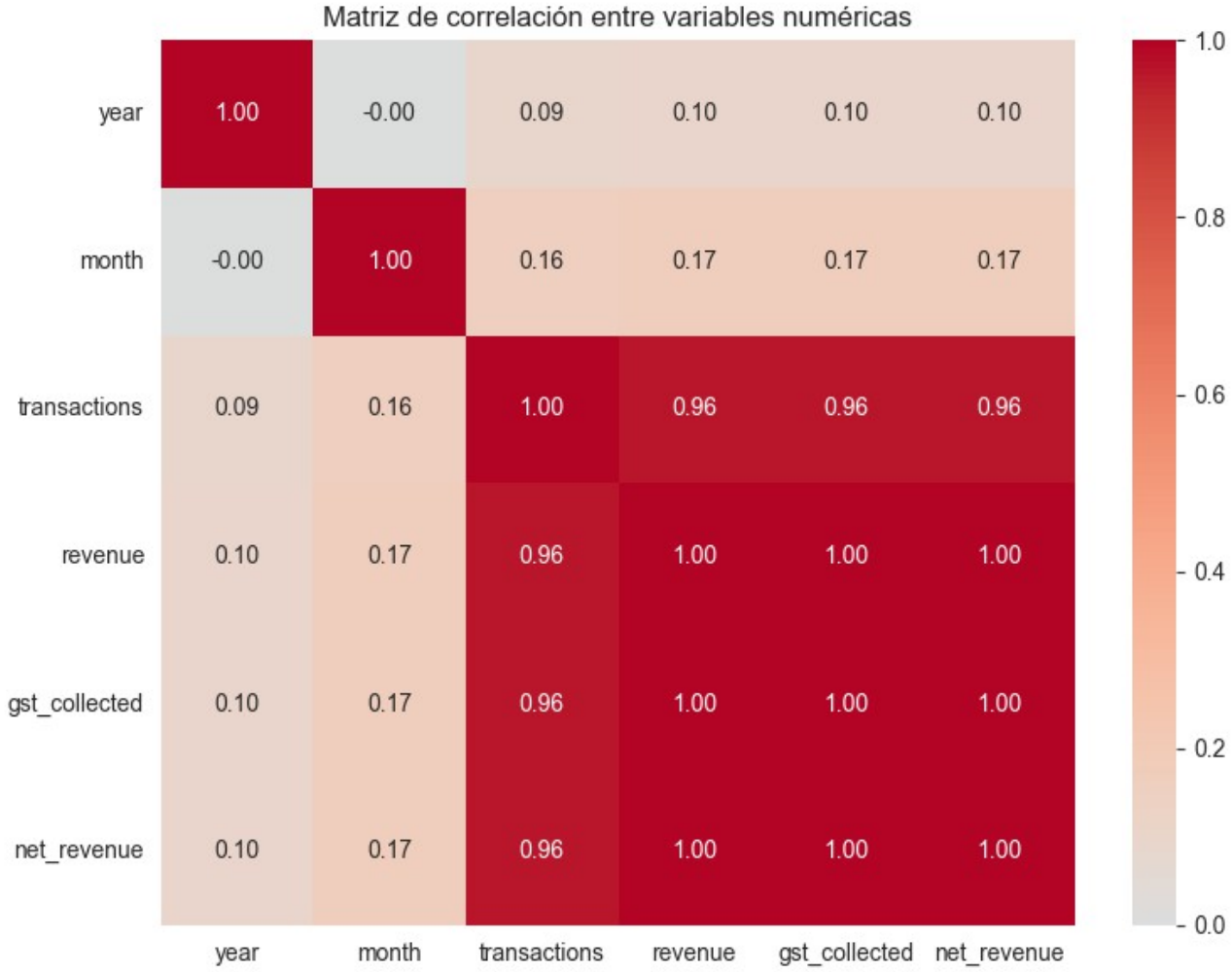
Durante este EDA se llevaron a cabo, entre otros, los siguientes análisis:

- Revisión de la estructura del dataset (tipos de datos, valores nulos, rangos).
- Análisis temporal de las ventas agregadas.
- Distribución de ingresos (`net_revenue`) y transacciones.



- Comparación entre canales de venta y tiendas.

- Análisis de correlaciones entre variables clave (ingresos, transacciones, revenue bruto).



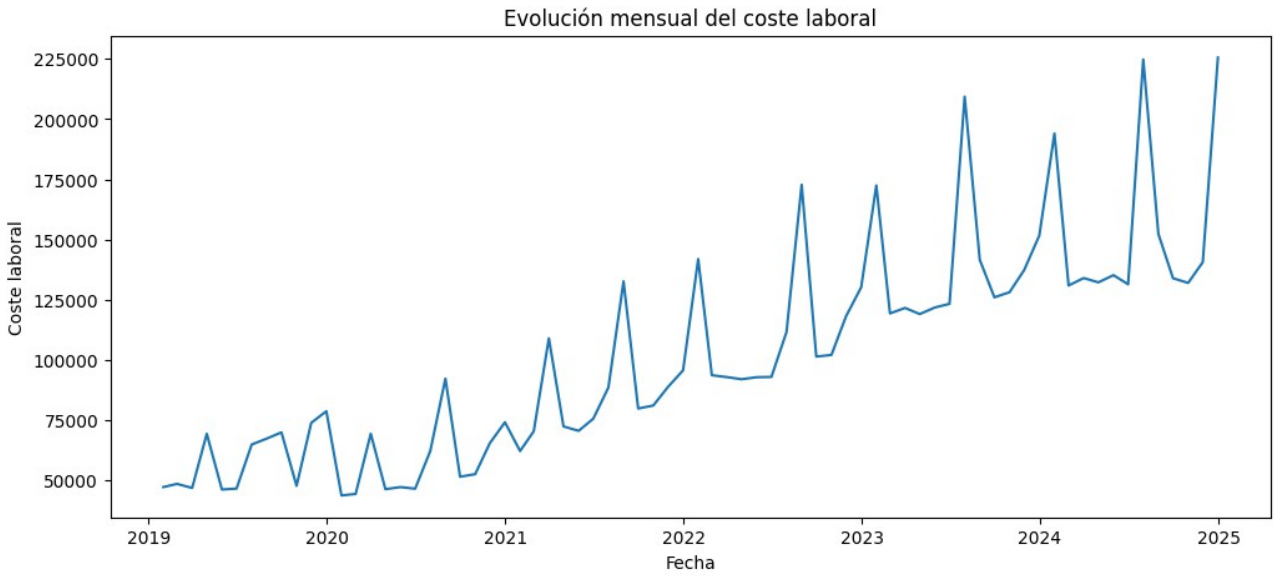
Estos análisis permitieron identificar una **alta correlación entre ingresos y número de transacciones**, así como patrones de estacionalidad claros en determinados periodos del año, especialmente asociados a campañas comerciales y periodos festivos.

2.2 EDA de Payroll

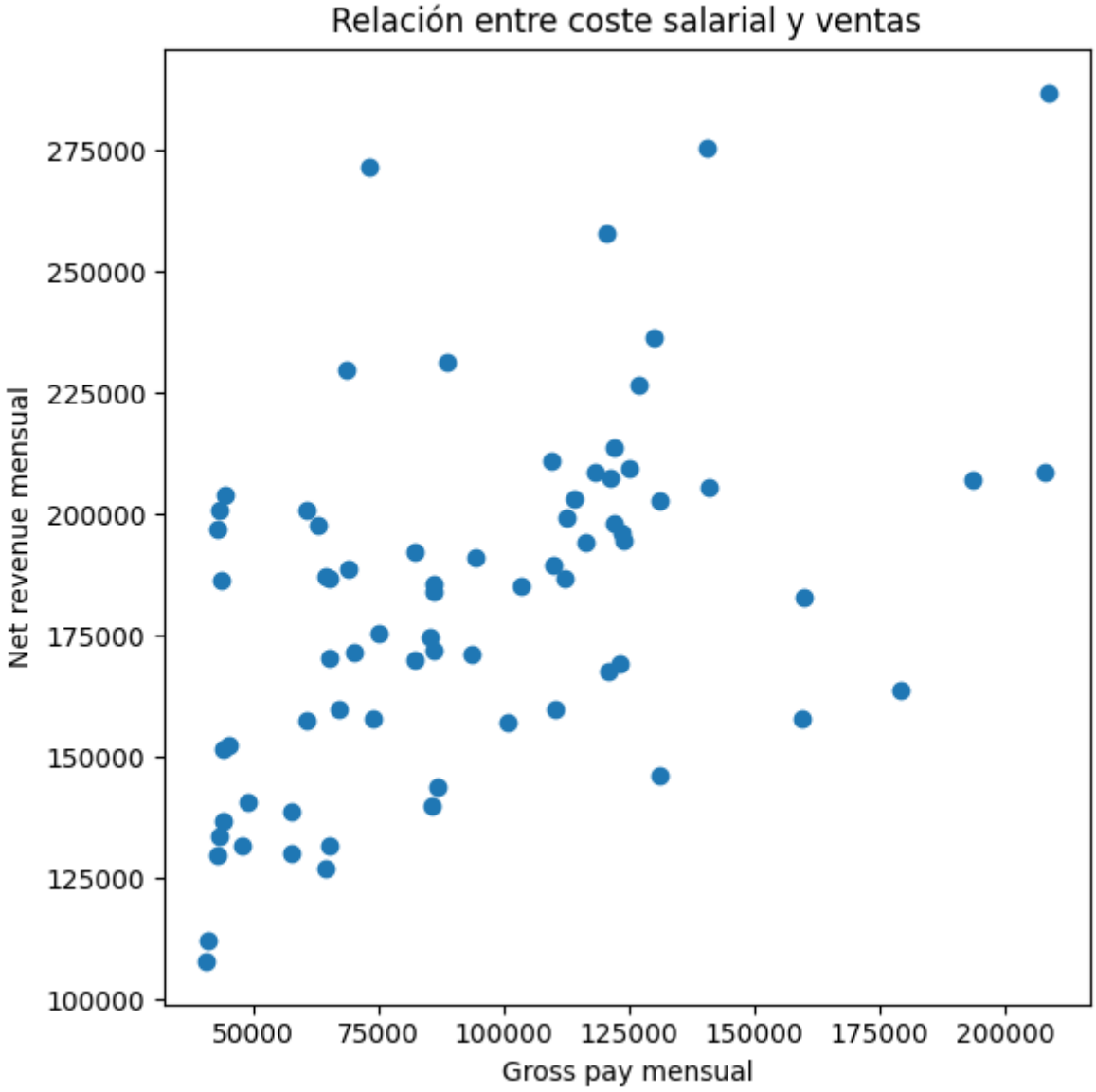
El EDA de payroll se realizó con el objetivo de analizar la evolución de los costes de personal y su posible relación con el desempeño económico del negocio.

Objetivos principales

- Analizar la evolución temporal de los costes salariales.
- Detectar cambios estructurales o picos anómalos.



- Evaluar la relación entre payroll y métricas de ventas.

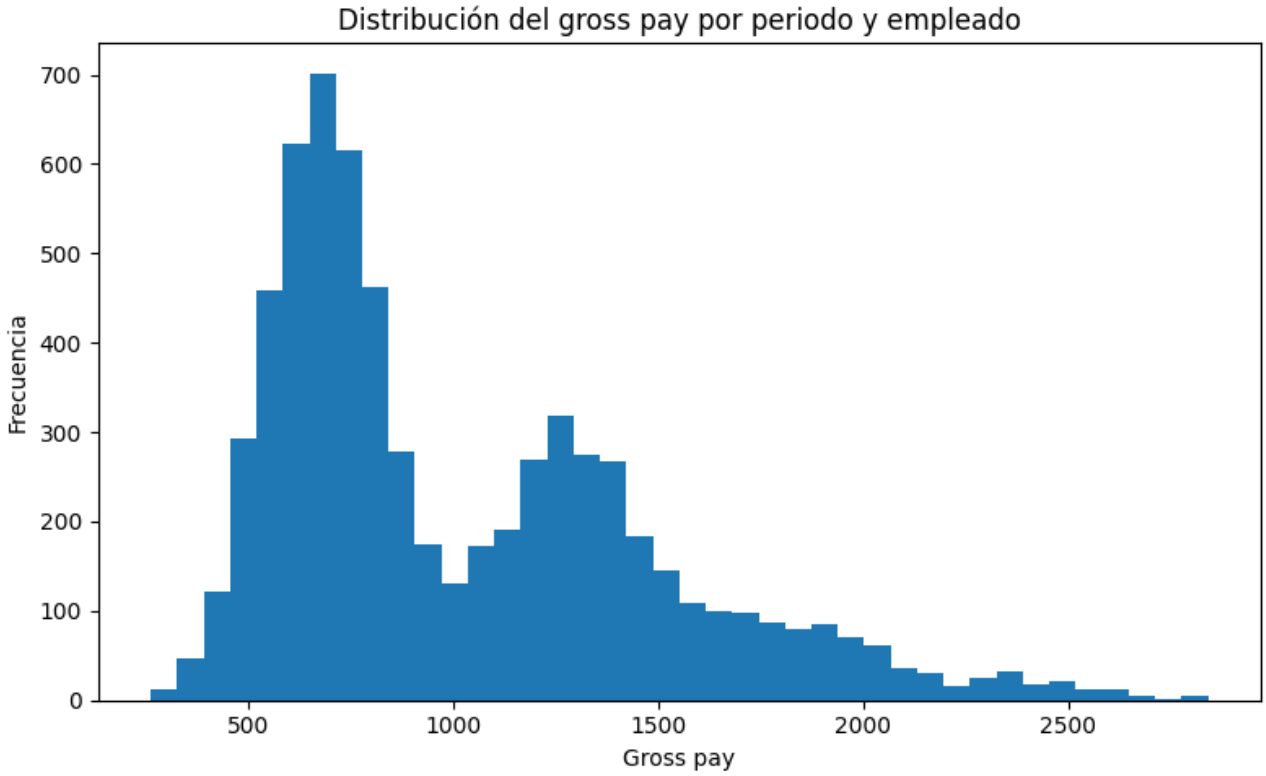


- Determinar la utilidad del payroll como variable explicativa.

Análisis realizados

Entre los análisis más relevantes destacan:

- Evolución temporal del payroll total.
- Distribución de costes salariales.



- Comparación entre periodos.
- Análisis de correlación entre payroll y métricas de ventas agregadas.

A partir de este EDA se observó que, si bien el payroll presenta una evolución relativamente estable, **su capacidad explicativa directa sobre las ventas es limitada** en comparación con otras variables, lo que influyó en su uso posterior dentro del proyecto.

2.3 Conclusiones del análisis exploratorio

Los EDAs realizados permitieron extraer varias conclusiones clave:

- Los datos presentan una estructura coherente y consistente para su uso en modelos de predicción.
- Existen patrones temporales claros que justifican un enfoque de series temporales.
- La variable `net_revenue` se identifica como la métrica principal a predecir.
- Algunas variables, aunque relevantes desde el punto de vista del negocio, no aportan una mejora significativa al modelo predictivo y se consideran secundarias.

Este análisis exploratorio sirvió como base para las decisiones de **preprocesado, feature engineering y modelado** desarrolladas en las siguientes fases del proyecto.

3. Modelos de predicción evaluados

3.1 Modelo SARIMA como baseline de series temporales

Como primera aproximación al problema de predicción de ingresos, se implementó un modelo clásico de **series temporales SARIMA** con el objetivo de establecer un *baseline* inicial y evaluar la capacidad de este tipo de modelos para capturar la dinámica temporal de las ventas.

Preparación de la serie temporal

A partir del dataset de ventas, se seleccionó la variable `net_revenue` y se realizó una agregación temporal. Inicialmente se exploraron distintas granularidades (semanal y mensual), observándose que la serie mensual presentaba un comportamiento más estable y menos ruidoso, lo que la hacía más adecuada para este tipo de modelado.

Además, se restringió el histórico al periodo posterior a enero de 2020. Esta decisión permitió eliminar un cambio estructural previo que afectaba a la estabilidad de la serie, mejorando significativamente los resultados de los tests de estacionariedad.

La serie final se construyó mediante una agregación mensual de las ventas totales:

- Frecuencia: mensual
- Variable objetivo: `net_revenue`
- Periodo: enero de 2020 – diciembre de 2023

Análisis de estacionariedad

Se evaluó la estacionariedad de la serie mediante el test de Dickey-Fuller aumentado (ADF). Los resultados mostraron que, tras filtrar el histórico a partir de 2020, la serie podía considerarse estacionaria en nivel, con un p-valor muy inferior al umbral habitual de significancia.

No obstante, al tratarse de una serie mensual, se observó la presencia de estacionalidad anual, lo que justificó el uso de un modelo SARIMA en lugar de un ARIMA simple.

Adicionalmente, se aplicó una transformación logarítmica sobre la serie con el objetivo de estabilizar la varianza y suavizar picos, lo que facilitó el ajuste del modelo y mejoró el comportamiento de los residuos.

Selección del modelo

Para la selección de los parámetros del modelo se utilizó el procedimiento automático `auto_arima`, restringiendo el espacio de búsqueda a configuraciones razonables y fijando explícitamente los órdenes de diferenciación en base al análisis previo:

- Diferenciación no estacional: `d = 0`
- Diferenciación estacional: `D = 1`
- Periodicidad estacional: `m = 12`

El modelo seleccionado por `auto_arima`, en base al criterio AIC, fue:

SARIMA(1,0,0)(0,1,0)[12] con intercept

Este modelo fue posteriormente reproducido utilizando la implementación `SARIMAX` de *statsmodels*, permitiendo un mayor control sobre el ajuste y el análisis de los resultados.

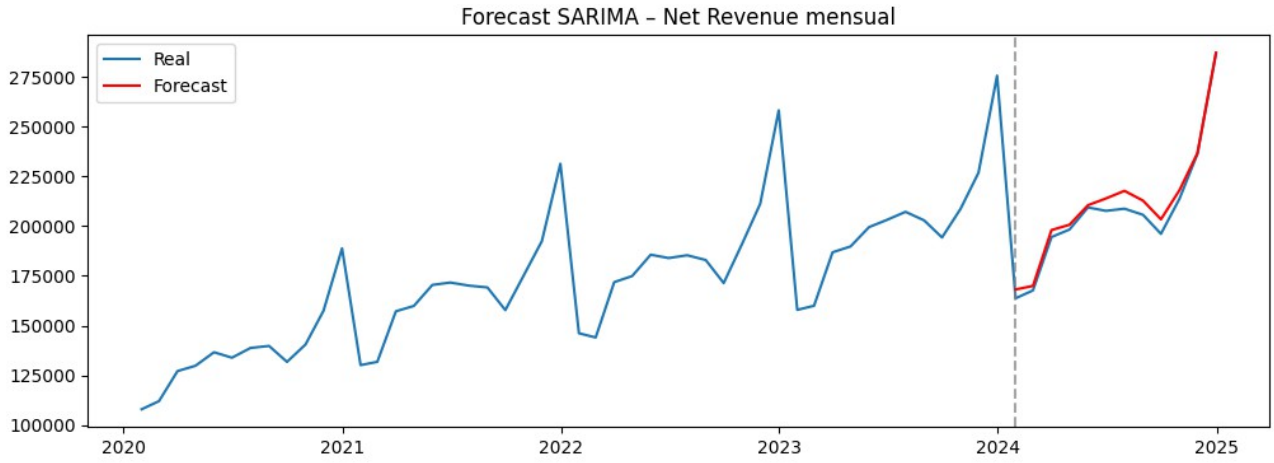
Entrenamiento y validación

El conjunto de datos se dividió de forma temporal, reservando los últimos 12 meses como conjunto de test. El modelo se entrenó sobre la serie transformada logarítmicamente y las predicciones se revirtieron posteriormente a la escala original mediante la función exponencial.

Las métricas obtenidas sobre el conjunto de test fueron:

- **RMSE** ≈ 4.879
- **MAE** ≈ 4.002

El análisis visual de las predicciones mostró que el modelo era capaz de capturar correctamente la tendencia general y la estacionalidad, aunque presentaba limitaciones para adaptarse a cambios más rápidos o a patrones específicos de negocio.



Conclusiones del enfoque SARIMA

El modelo SARIMA permitió validar la presencia de estacionalidad anual y sirvió como un baseline sólido para la predicción agregada de ingresos. Sin embargo, este enfoque presenta varias limitaciones relevantes para el contexto del proyecto:

- No escala de forma natural a múltiples series simultáneas (tienda y canal).
- Dificulta la incorporación de variables adicionales.
- Requiere mantener modelos independientes para cada segmentación.

Estas limitaciones motivaron la exploración de enfoques alternativos basados en *machine learning* supervisado, descritos en las siguientes secciones.

3.2 Enfoque supervisado (Machine Learning) para predicción mensual multiserie

Tras la evaluación inicial de un enfoque clásico de series temporales (SARIMA), se exploró un enfoque alternativo basado en **aprendizaje supervisado**, con el objetivo de construir un modelo escalable a múltiples series simultáneas (combinaciones de **tienda** y **canal**) y con capacidad de incorporar variables explicativas adicionales mediante *feature engineering*.

Preparación del dataset y limpieza

Se cargó el dataset de ventas y se realizaron las siguientes transformaciones:

- Conversión de la columna `date` a formato datetime.
- Ordenación temporal del dataset.
- Eliminación de la columna `dataset` por no aportar información útil al modelado.
- Se realizaron pruebas con y sin datos anteriores a 2020. La inclusión del periodo pre-2020 altera de forma notable el comportamiento estadístico de la serie (cambio estructural), por lo que en la versión final del pipeline se priorizó el periodo posterior a 2020 para mantener consistencia temporal.

Exploración de granularidades y enfoque final

Aunque se realizaron pruebas a nivel diario (incluyendo lags 1, 7 y 30 y medias móviles 7/30), el enfoque final se centró en la **agregación mensual**, ya que:

- reduce ruido y volatilidad del dato diario,
- captura mejor patrones estacionales,
- y es computacionalmente más eficiente para entrenamiento y despliegue.

La agregación mensual se realizó a nivel de **(mes, tienda, canal)**:

- `net_revenue`: suma mensual
- `transactions`: suma mensual

Se añadió además la variable derivada:

- **`avg_ticket`** = `net_revenue / transactions` (controlando divisiones por cero)

Feature engineering (variables explicativas)

El modelo supervisado se construyó a partir de variables que capturan inercia temporal y estacionalidad:

Variables temporales y categóricas

- `store_id` (categórica)
- `channel` (categórica)
- `month` (categórica)

Variables numéricas

- Lags: `lag_1`, `lag_3`, `lag_6`, `lag_12`

- Rolling means (con shift para evitar leakage):
    - `roll_3`, `roll_6`, `roll_12`
- `avg_ticket`

Todas las features se calcularon **por grupo (store_id, channel)** para evitar contaminación entre series.

Pipeline de preprocesado

Se implementó un pipeline robusto usando `ColumnTransformer`:

- **OneHotEncoder** para variables categóricas (`store_id`, `channel`, `month`)
- **StandardScaler** para variables numéricas

Esto permite entrenar modelos diferentes sobre el mismo esquema de datos sin modificar el preprocesado.

Estrategia de validación (split temporal)

Se utilizó una validación temporal reservando los **últimos 12 meses** como conjunto de test:

- `H = 12`
- `train`: meses anteriores al cutoff
- `test`: últimos 12 meses

Este enfoque evita el error común de mezclar periodos pasados y futuros en el split y simula un escenario real de forecasting.

Modelos evaluados

Se entrenaron y compararon distintos modelos dentro del mismo pipeline:

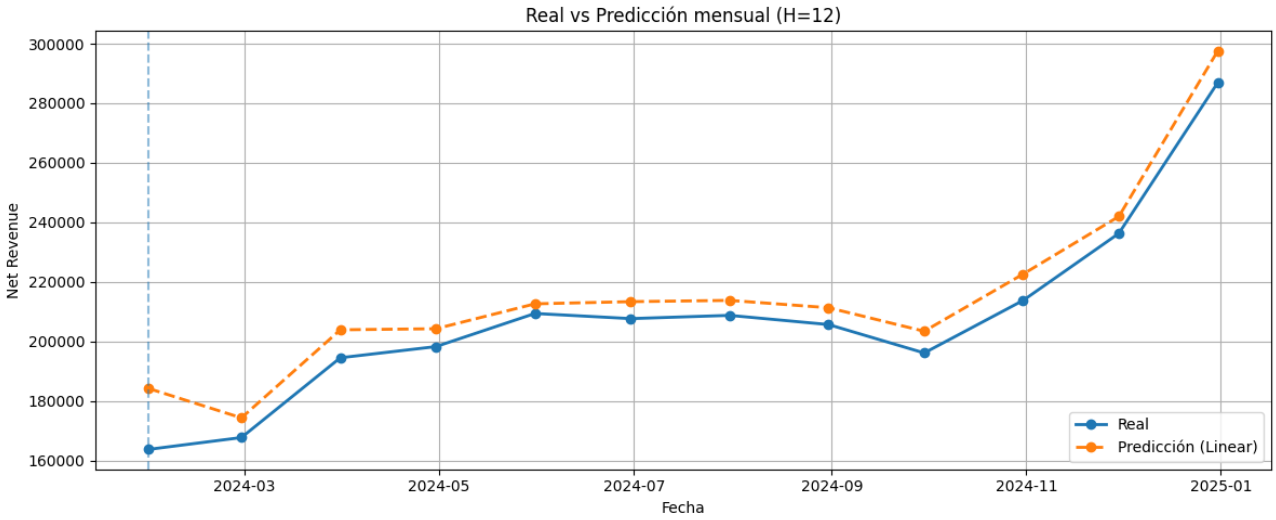
1. **Linear Regression** (baseline supervisado)
2. **Random Forest Regressor**
3. **XGBoost Regressor** (con y sin optimización de hiperparámetros)

Las métricas utilizadas fueron:

- **RMSE**
- **MAE**

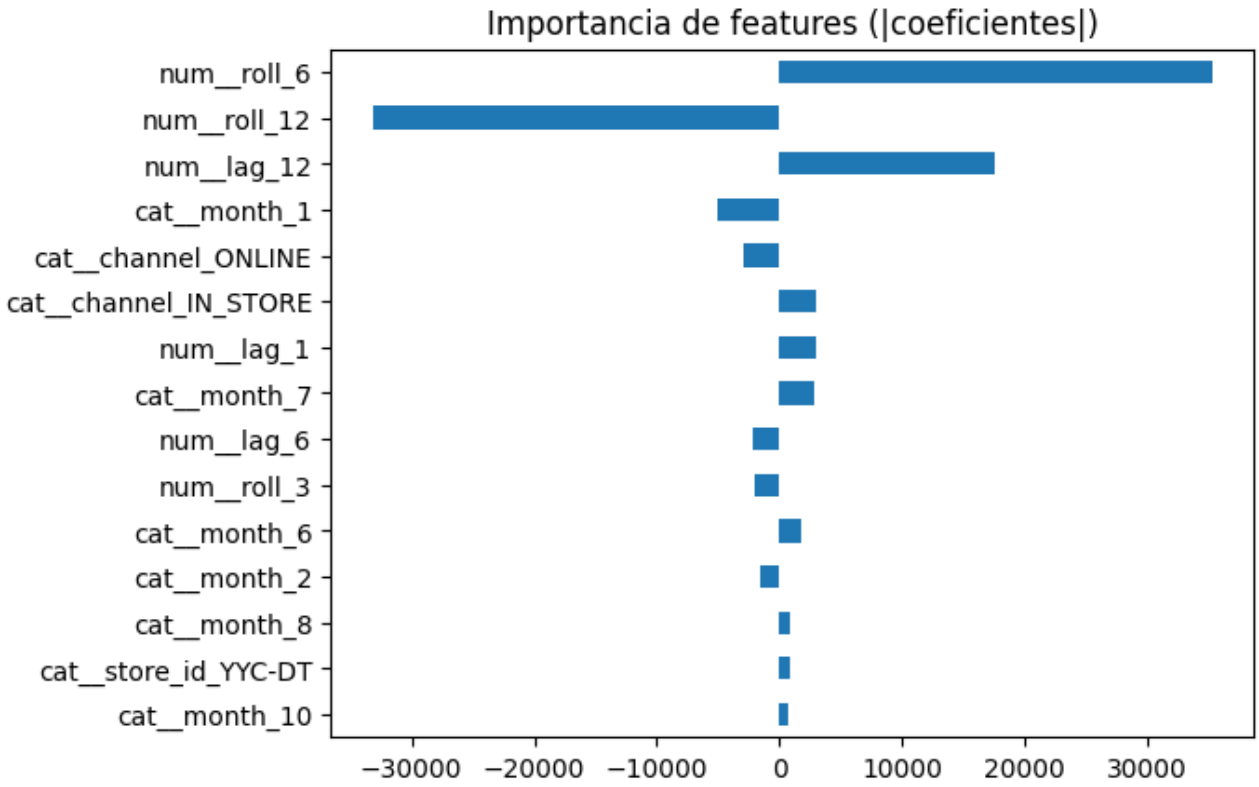
Los resultados mostraron que los modelos más complejos tendían a **sobreajustar** (error muy bajo en entrenamiento, pero degradación clara en test), especialmente en RandomForest y XGBoost. Por el contrario, **Linear Regression** ofreció un equilibrio más estable entre rendimiento y generalización en los periodos más recientes.

En consecuencia, se seleccionó **Linear Regression** como modelo final del pipeline, priorizando robustez y consistencia en predicción futura.



Interpretabilidad del modelo (coeficientes)

Una ventaja adicional del modelo lineal es su interpretabilidad. Se extrajeron los coeficientes tras el OneHotEncoding y se ordenaron por valor absoluto para identificar las variables con mayor impacto, destacando especialmente el peso de las ventanas móviles (`roll_6`, `roll_12`) y los lags de largo plazo (`lag_12`), lo que refuerza la idea de que la serie presenta una componente estacional y memoria temporal significativa.



Entrenamiento final y persistencia

Tras seleccionar el modelo, se reentrenó con todos los datos disponibles (post-2020) y se guardó mediante `joblib`:

- `monthly_linear_model.joblib`

Esto facilita su uso posterior en una aplicación web y su reentrenamiento controlado.

Forecasting a 12 meses (enfoque recursivo)

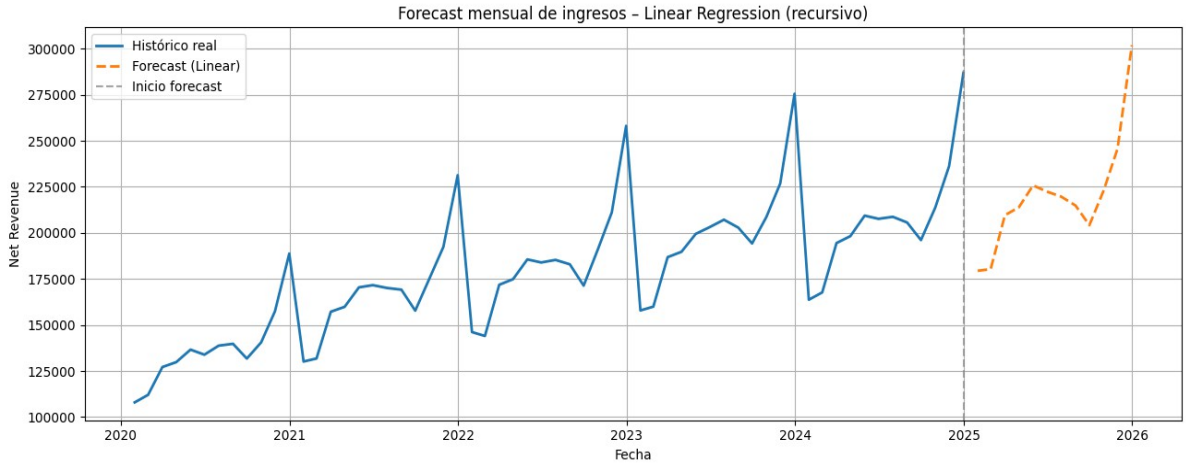
Para generar predicciones futuras se implementó un procedimiento de forecasting recursivo:

- se generan filas futuras por combinación (store_id, channel),
- se calculan lags y rollings usando histórico + predicciones anteriores,
- se predice el siguiente mes,
- y se alimenta el histórico con el valor predicho.

Finalmente se obtienen dos salidas:

- predicción futura por serie (tienda/canal),
- predicción agregada total por mes.

Este enfoque permite construir un forecast operativo a 12 meses sin depender de variables futuras externas.



3.3 Análisis avanzado de Payroll: detección de anomalías y segmentación de empleados

Además del modelado predictivo de ventas, se desarrolló un bloque específico de análisis sobre el dataset de **payroll** con dos objetivos complementarios:

1. **Control de calidad y detección de registros anómalos**, susceptibles de corresponder a errores de carga, pagos extraordinarios o situaciones fuera de lo normal.
2. **Segmentación de empleados por perfil salarial**, con el fin de caracterizar diferentes grupos laborales y facilitar análisis posteriores de costes.

Este apartado se aborda como un análisis independiente del forecasting de ventas, dado que el payroll mensual futuro no es una variable disponible de forma directa para predicción (lo que limita su uso como feature en producción). Aun así, resulta altamente relevante desde la perspectiva de negocio y trazabilidad del dato.

Dataset y variables utilizadas

El dataset de payroll cuenta con 6.640 registros a nivel de periodo quincenal y empleado, incluyendo variables salariales y de retenciones. Para los análisis posteriores se seleccionaron variables numéricas representativas:

- `hourly_rate`
- `hours_biweekly`
- `gross_pay`
- `net_pay`
- `employee_benefits`
- `income_tax_withheld`

Antes de modelar, se convirtió `pay_period_start` a formato datetime y se estandarizaron las variables numéricas mediante `StandardScaler` para evitar que escalas diferentes afectasen a los algoritmos.

3.3.1 Detección global de anomalías (Isolation Forest)

Como primera aproximación, se aplicó **Isolation Forest** sobre el conjunto completo de registros, comparando a todos los empleados entre sí. Se configuró el porcentaje esperado de anomalías en torno al 2% (`contamination=0.02`), una cifra razonable para un primer análisis exploratorio.

El modelo identificó aproximadamente **132 registros anómalos**. El análisis posterior mostró que dichas anomalías estaban fuertemente concentradas en determinados empleados y roles, destacando especialmente el rol de **Store Manager**, con valores de `gross_pay` y `hours_biweekly` claramente superiores a la media del dataset. Este resultado es coherente con diferencias salariales estructurales y evidencia una limitación del enfoque global: los perfiles salariales de roles senior pueden ser catalogados como "anómalos" simplemente por pertenecer a un grupo salarial distinto.

Para visualizar estos patrones, se representó la relación entre `hours_biweekly` y `gross_pay`, destacando los puntos clasificados como anomalía.

3.3.2 Detección de anomalías por rol (Isolation Forest condicionado)

Para evitar falsos positivos derivados de comparar roles salariales incomparables, se implementó una segunda estrategia: entrenar un **Isolation Forest por rol**, de forma que cada empleado se compara únicamente con perfiles equivalentes.

Se entrenó un modelo por cada rol con suficiente volumen de datos (se excluyeron roles con menos de 50 registros) y se almacenaron los modelos y escaladores por rol en un fichero persistente:

- `payroll_iso_by_role.pkl`

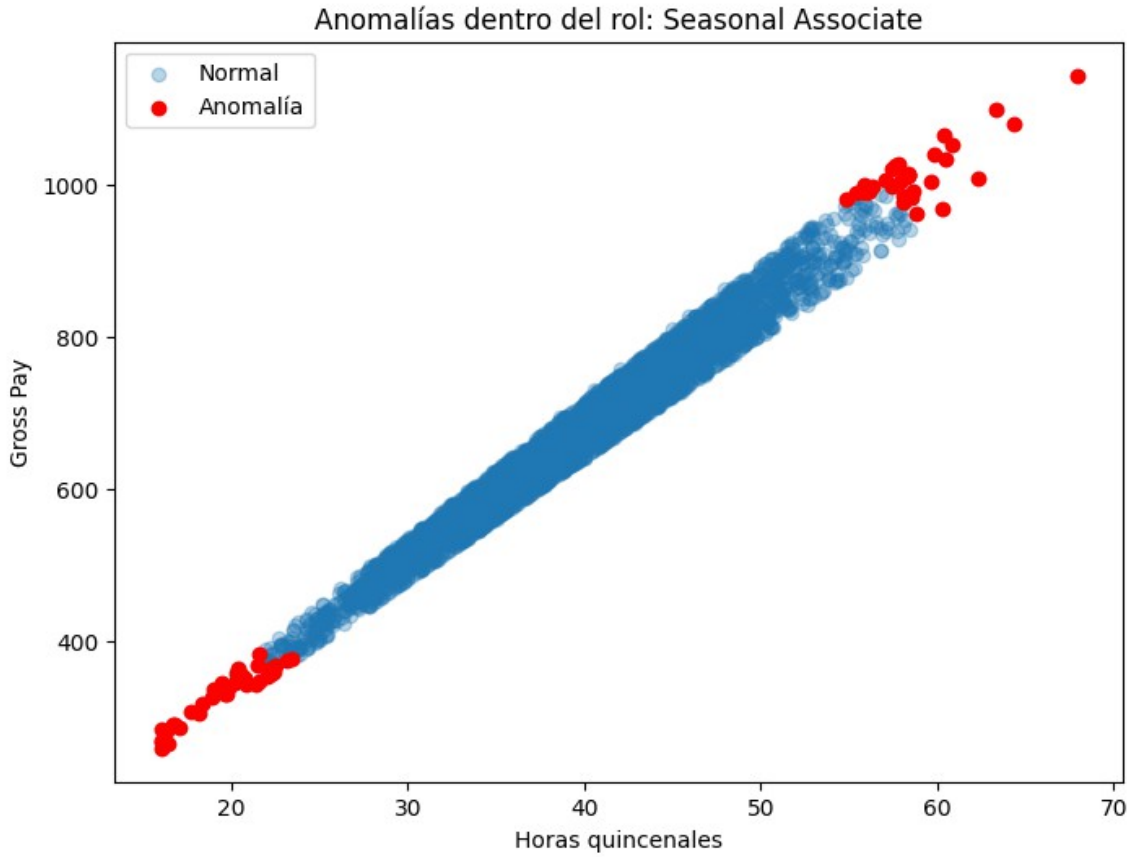
Este enfoque identificó aproximadamente **135 registros anómalos**, distribuidos de forma más razonable entre roles operativos, con especial concentración en:

- Seasonal Associate
- Bookseller

- Warehouse Clerk

Además, se identificaron empleados concretos con recurrencia de anomalías dentro del mismo rol (casos repetidos), lo que permite focalizar auditorías o revisiones específicas.

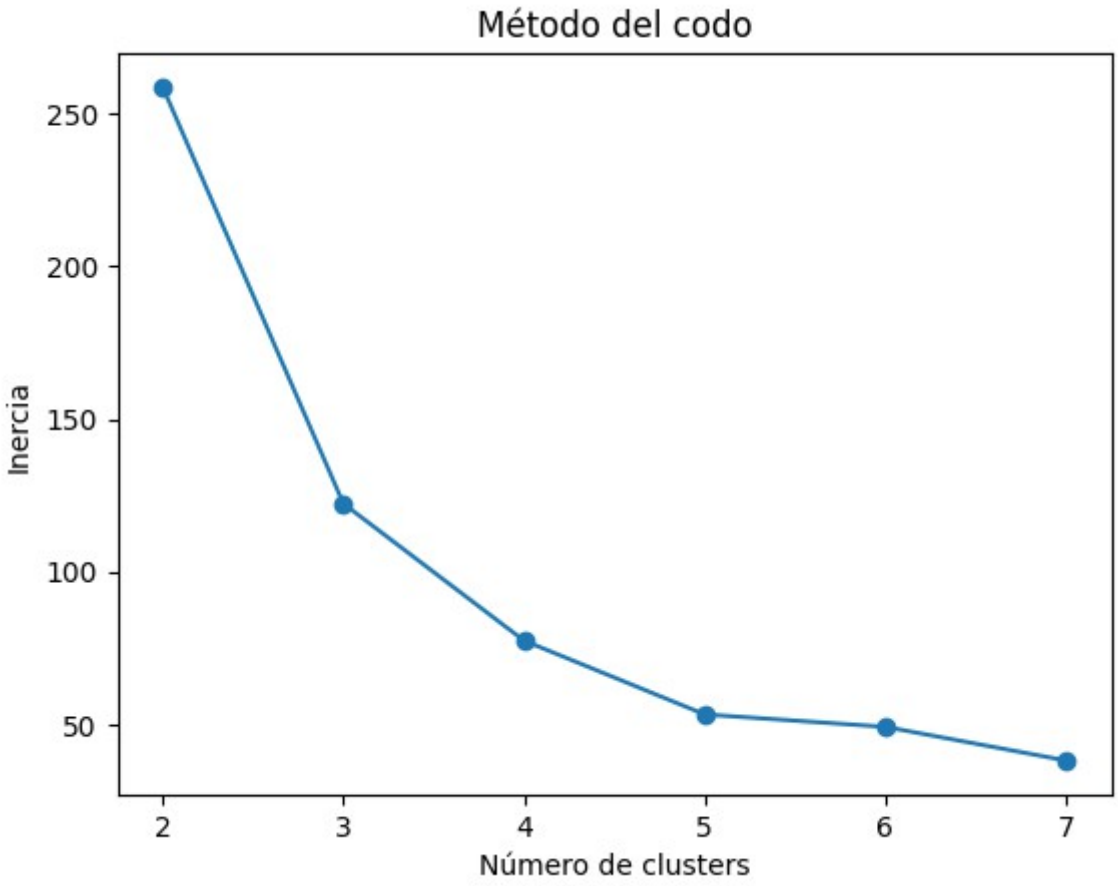
Se generaron visualizaciones por rol (ejemplo: Seasonal Associate) para interpretar mejor los valores extremos dentro de un grupo homogéneo.



3.3.3 Segmentación de empleados (Clustering con K-Means)

Como análisis complementario, se realizó una **segmentación de empleados** mediante clustering, agregando previamente la información a nivel de empleado (media de las variables seleccionadas por `employee_id` y `role`). Esta agregación permite obtener un "perfil salarial medio" por empleado y reducir el ruido de periodos individuales.

El número de clusters se evaluó con el **método del codo** (inercia), seleccionando finalmente **k=3** como compromiso razonable entre simplicidad e interpretabilidad. El modelo se entrenó sobre variables estandarizadas y se guardaron los artefactos:

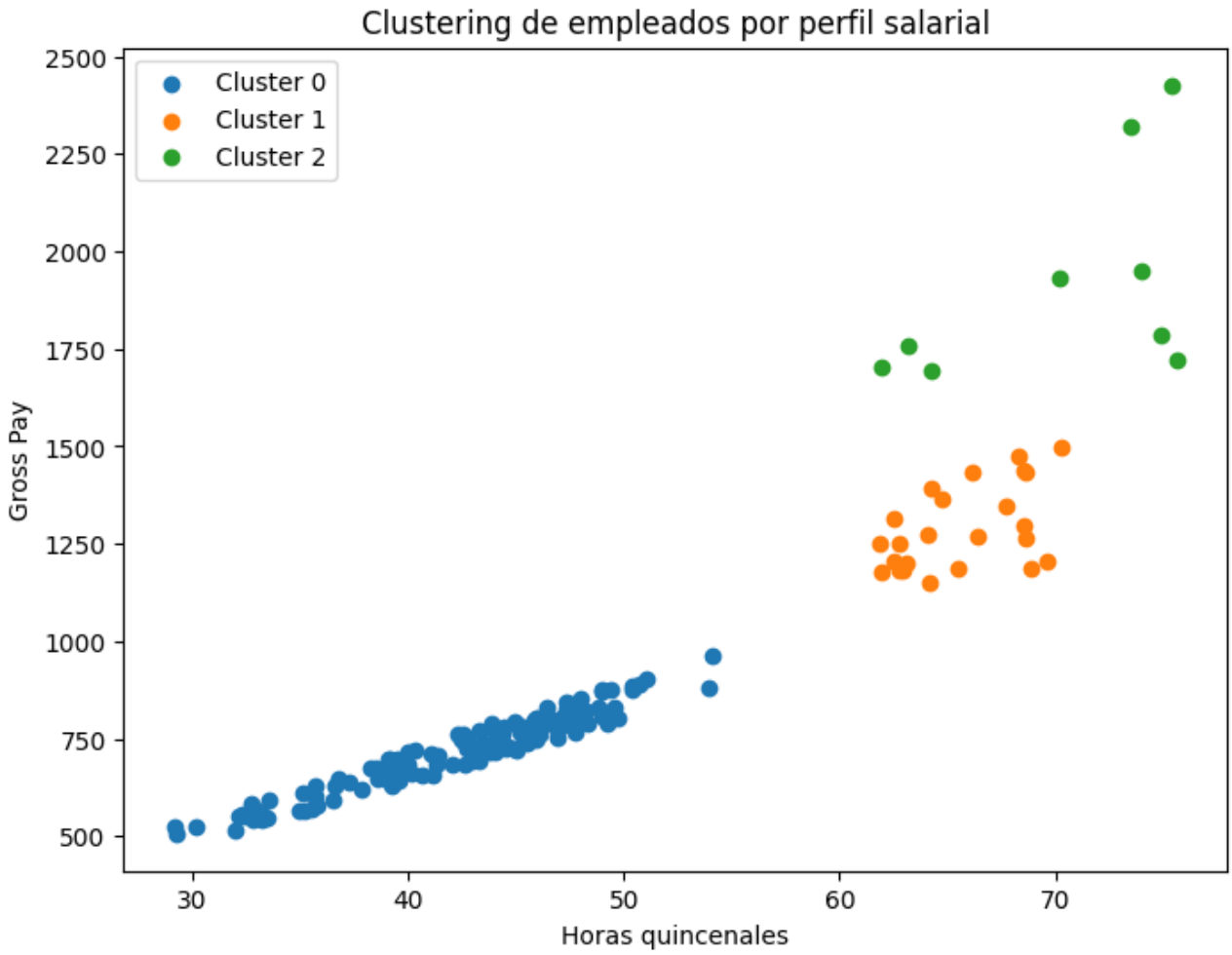


- `payroll_scaler.pkl`
- `payroll_kmeans.pkl`

Los clusters obtenidos reflejan tres perfiles salariales bien diferenciados:

- **Cluster 0**: perfiles de bajo coste (menor `hourly_rate`, menos horas y sin beneficios; predominan perfiles temporales)
- **Cluster 1**: perfiles intermedios (más horas, mayor coste y presencia de beneficios)
- **Cluster 2**: perfiles de alto coste (salario/hora más alto, mayor `gross_pay` y retenciones más elevadas)

La tabla cruzada `cluster` vs `role` confirmó que los clusters capturan estructura real del negocio (por ejemplo, roles temporales concentrados en el cluster de menor coste, y roles de responsabilidad en clusters superiores). Finalmente, se representaron visualmente los clusters usando `hours_biweekly` vs `gross_pay`.



Conclusiones del análisis de payroll

El análisis de payroll aportó valor en dos dimensiones:

1. **Calidad y control del dato:** identificación de registros potencialmente anómalos, primero a nivel global y después con un enfoque más robusto condicionado por rol.
2. **Comprensión del coste laboral:** segmentación interpretable de empleados por perfil salarial, alineada con roles y estructura del negocio.

Aunque este bloque no se integró directamente en el modelo final de forecasting (por limitaciones de disponibilidad futura del payroll), constituye un componente relevante del proyecto desde el punto de vista analítico y de negocio, y deja preparada una base sólida para futuras extensiones (por ejemplo, análisis de eficiencia coste/ventas o planificación de recursos).

3.4 Comparativa de enfoques y decisión final

Una vez evaluados los distintos enfoques de modelado y análisis, se realizó una comparación global atendiendo a tres criterios principales: **capacidad predictiva**, **escalabilidad** y **viabilidad en un entorno real de producción**.

Comparativa de resultados

Modelo SARIMA (series temporales clásicas)

- Enfoque: predicción agregada de la serie total de ingresos.
- Métricas aproximadas en test:
    - RMSE ≈ 4.879
    - MAE ≈ 4.002
- Fortalezas:
    - Captura correctamente tendencia y estacionalidad anual.
    - Modelo interpretable y adecuado como baseline.
- Limitaciones:
    - No escala de forma natural a múltiples series (tienda/canal).
    - Difícil incorporación de variables adicionales.
    - Requiere mantener modelos separados para cada segmentación.

Modelo supervisado (Machine Learning – Linear Regression)

- Enfoque: predicción mensual multiserie (tienda + canal) mediante feature engineering.
- Métricas aproximadas en test:
    - RMSE ≈ 2.663
    - MAE ≈ 1.803
- Fortalezas:
    - Mejor rendimiento predictivo en el horizonte de test.
    - Escalable a múltiples combinaciones de tienda y canal en un único modelo.
    - Fácil incorporación de nuevas variables explicativas.
    - Integración directa en un pipeline reproducible y desplegable.
- Limitaciones:
    - Requiere un diseño cuidadoso del forecasting recursivo.
    - Dependencia de la calidad de las features temporales.

Análisis de Payroll (no integrado directamente en el forecast)

- Enfoque: detección de anomalías y segmentación salarial.
- Aportación principal:
    - Control de calidad del dato.
    - Comprensión de la estructura de costes laborales.
- Decisión:
    - No se integra como feature principal en el modelo final debido a la falta de disponibilidad futura de payroll real, evitando así *data leakage* y supuestos poco realistas.

Decisión final

A la vista de los resultados, se seleccionó el **modelo supervisado basado en Linear Regression** como solución final del proyecto. Esta decisión se fundamenta en:

- Un **mejor rendimiento predictivo** en el conjunto de test.
- Su **robustez y estabilidad** frente a modelos más complejos que mostraron sobreajuste.
- Su **capacidad de generalización** a múltiples series en un único pipeline.
- La **facilidad de despliegue** y mantenimiento en un entorno real.

El modelo SARIMA se mantiene como un **baseline de referencia**, útil para validar patrones globales de la serie, mientras que el análisis de payroll se considera un **bloque analítico complementario**, orientado a control y comprensión del negocio más que a predicción directa.

4. Implementación de una aplicación web para explotación del modelo

4.1 Motivación y objetivo

Para completar el proyecto y acercarlo a un escenario de uso real, se desarrolló una **aplicación web ligera** que permite interactuar con los modelos entrenados sin necesidad de ejecutar notebooks manualmente.
El objetivo de esta capa no es introducir nuevas técnicas de modelado, sino **operacionalizar** el trabajo realizado previamente y facilitar:

- la carga de nuevos datos,
- el reentrenamiento controlado del modelo,
- la consulta de métricas de evaluación,
- y la visualización de resultados tanto de ventas como de payroll.

De este modo, el proyecto cubre no solo la fase analítica, sino también un primer nivel de **despliegue y explotación del modelo**.

4.2 Arquitectura general de la solución

La aplicación se ha diseñado siguiendo una arquitectura sencilla y modular:

- **Backend**: aplicación Python que carga los modelos persistidos (`joblib`), ejecuta el preprocesado y realiza predicciones o reentrenamientos bajo demanda.
- **Frontend**: interfaz HTML minimalista, orientada a claridad y funcionalidad, que permite interactuar con el sistema sin conocimientos técnicos.
- **Persistencia**:
    - modelos entrenados (`monthly_linear_model.joblib`, modelos de payroll),
    - métricas de validación asociadas a cada entrenamiento.

Este diseño permite reproducir los experimentos del proyecto de forma controlada y coherente con

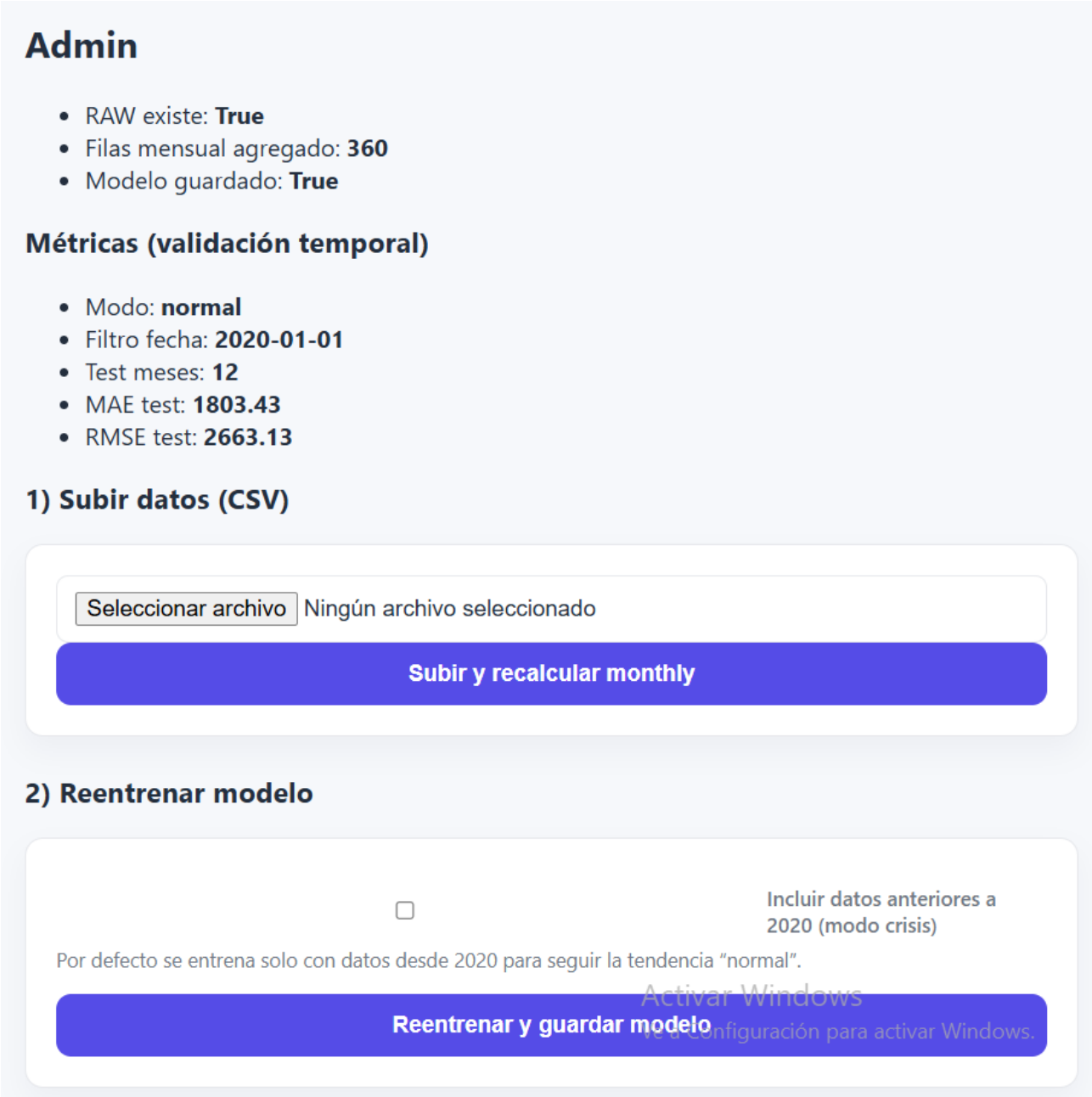
el pipeline descrito en secciones anteriores.

4.3 Panel de administración (Admin)

Se implementó un panel de administración que centraliza las operaciones principales del sistema. Sus funcionalidades incluyen:

- **Subida de nuevos datos de ventas (CSV)**
  Al subir un nuevo fichero, el sistema recalcula automáticamente la agregación mensual y deja los datos listos para reentrenar.
- **Reentrenamiento del modelo de ventas**
  El usuario puede decidir si:
    - entrenar solo con datos posteriores a 2020 (escenario “normal”), o
    - incluir datos anteriores a 2020 para simular escenarios de crisis o cambios estructurales.
- **Visualización de métricas de validación temporal**
  Tras cada entrenamiento se muestran métricas clave:
    - MAE
    - RMSE
    - número de meses usados como test

Estas métricas permiten evaluar rápidamente la estabilidad del modelo sin acceder al entorno de desarrollo.



4.4 Visualización de resultados de Payroll

Como complemento al análisis desarrollado en el apartado 3.3, la aplicación incluye una vista específica para **explorar los resultados de payroll**:

- Filtros por:
    - rol,
    - localización,
    - empleado,
    - cluster,
    - y anomalías.

- Visualizaciones dinámicas:
    - anomalías por rol,
    - anomalías por periodo,
    - dispersión entre horas trabajadas y salario bruto.
- Tablas resumen:
    - registros anómalos detectados,
    - asignación de clusters salariales por empleado.

Esta vista permite **interpretar visualmente** los resultados de los modelos de Isolation Forest y K-Means entrenados previamente, reforzando el carácter analítico del bloque de payroll.

4.5 Alcance y limitaciones

Es importante destacar que esta aplicación web:

- **no pretende ser un producto final**,
- ni sustituye a una arquitectura completa de producción,

sino que actúa como una **prueba de concepto funcional**, orientada a demostrar:

- la reproducibilidad del pipeline,
- la integración de modelos de machine learning en un flujo usable,
- y la viabilidad de extender el sistema con nuevos datos y modelos.

Este enfoque refuerza el carácter práctico del proyecto y conecta el análisis realizado con un contexto real de uso empresarial.

4.6 Conclusión del apartado

La inclusión de una interfaz web permite cerrar el proyecto de forma coherente, mostrando no solo la capacidad de construir modelos predictivos, sino también de **integrarlos en un sistema accesible y reutilizable**.
De este modo, el proyecto abarca todo el ciclo: análisis, modelado, evaluación y explotación práctica de los resultados.

5. Conclusiones y líneas futuras

5.1 Conclusiones

En este proyecto se ha abordado un problema realista de análisis y predicción de ventas a partir de datos históricos, combinando enfoques clásicos de series temporales, modelos de aprendizaje supervisado y análisis avanzado de datos auxiliares (payroll).

El trabajo comenzó con una exploración exhaustiva de los datos mediante EDA, lo que permitió comprender la estructura del negocio, identificar patrones temporales relevantes y detectar cambios estructurales en el histórico. Este análisis inicial fue clave para tomar decisiones metodológicas

posteriores, como la restricción del periodo de entrenamiento a partir de 2020 o la elección de la granularidad mensual.

Como primer baseline, se implementó un modelo SARIMA sobre la serie agregada de ingresos. Este enfoque permitió validar la presencia de estacionalidad anual y sirvió como punto de referencia, aunque mostró limitaciones claras en términos de escalabilidad y capacidad predictiva en comparación con enfoques más flexibles.

Posteriormente, se desarrolló un modelo supervisado de forecasting basado en *feature engineering* temporal (lags, medias móviles y variables estacionales), capaz de predecir de forma simultánea múltiples series (tienda y canal). Este enfoque obtuvo mejores resultados en las métricas de validación temporal y demostró una mayor robustez y adaptabilidad, motivo por el cual fue seleccionado como modelo final del proyecto.

De forma complementaria, se realizó un análisis avanzado del dataset de payroll, aplicando técnicas de detección de anomalías y clustering para mejorar la comprensión del coste laboral y la calidad del dato. Aunque este bloque no se integró directamente en el modelo de predicción de ventas, aportó valor analítico y mostró la capacidad de extender el proyecto a otras áreas del negocio.

Finalmente, se implementó una aplicación web sencilla que permite cargar datos, reentrenar modelos y consultar resultados, cerrando el ciclo completo desde el análisis hasta la explotación práctica del modelo. Este componente refuerza el carácter aplicado del proyecto y demuestra la viabilidad de integrar modelos de machine learning en un flujo operativo.

5.2 Líneas futuras de trabajo

A partir de la base desarrollada, existen múltiples líneas de mejora y extensión del proyecto:

- Incorporación de **variables exógenas** adicionales (campañas, festivos, promociones, indicadores macroeconómicos).
- Evaluación de modelos más avanzados para series temporales multivariantes, como **XGBoost con variables futuras estimadas** o **modelos híbridos** (SARIMA + ML).
- Análisis conjunto de **ventas y payroll** para estudiar relaciones de eficiencia (coste laboral vs ingresos).
- Mejora del sistema de forecasting recursivo incorporando intervalos de confianza.
- Evolución de la aplicación web hacia un sistema de despliegue más completo (API, control de versiones de modelos, monitorización).

5.3 Cierre final

En conjunto, el proyecto demuestra una aproximación completa y progresiva al análisis de datos y al modelado predictivo, abarcando desde la exploración inicial hasta la integración en un entorno funcional.
La solución final prioriza realismo, interpretabilidad y aplicabilidad práctica, alineándose con escenarios habituales en proyectos de ciencia de datos en entorno empresarial.