INSTITUTO SUPERIOR POLITÉCNICO DE CÓRDOBA

TECNICATURA SUPERIOR EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

Módulo de Práctica Profesionalizante

Entrega Final

04/11/2024

Docente:

- Charletti, Carlos

Integrantes:

- López, Erick
- Nüesch, Christian
- Zurita Rojo, Débora

ÍNDICE

Enlace al documento en Google Drive: Entrega Final - PPI

INTRODUCCIÓN AL ESPACIO CURRICULAR

Desarrollo de la Idea de Negocio

DataVista Analytics es una empresa especializada en análisis de datos, y ha identificado varias oportunidades de negocio basadas en el análisis del dataset df_autos. Este dataset, recopilado a principios de 2023, incluye información detallada sobre ventas de automóviles, como marca, modelo, tipo de chasis, precio de venta y la moneda en la que se cotiza (pesos o dólares).

La riqueza y variedad de estos datos proporcionan una base sólida para la generación y desarrollo de nuevas ideas de negocio. Al analizar este dataset, podemos obtener insights valiosos y detectar oportunidades en el mercado automotriz, facilitando la identificación de tendencias y áreas de oportunidad que pueden ser explotadas para el crecimiento y éxito de nuestra empresa.

A continuación, se presentan cuatro ideas principales de negocio que podrían desarrollarse utilizando este dataset:

- **Informes de mercado personalizados:**
  Idea: Ofrecer informes a medida para concesionarios, fabricantes o empresas de seguros, basados en el análisis del dataset **df_autos**.
  Ejemplo: Un informe podría analizar las preferencias de los clientes en cuanto a marcas, modelos o características de seguridad, ayudando a un concesionario a optimizar su inventario.

- **Predicción de precios:**
  Idea: Desarrollar un modelo de Machine Learning que prediga el precio de un automóvil usado basándose en las características del dataset.

Ejemplo: Ofrecer este modelo como un servicio a sitios web de compraventa de automóviles o a empresas de tasación.

- **Detección de anomalías:**
  Idea: Analizar el dataset para identificar posibles fraudes o inconsistencias en los datos de los vehículos.
  Ejemplo: Detectar anuncios de autos con kilometraje sospechosamente bajo o precios inusualmente altos.

- **Segmentación de clientes:**
  Idea: Agrupar a los clientes potenciales en función de sus preferencias automotrices utilizando técnicas de clustering.
  Ejemplo: Dirigir campañas de marketing personalizadas a cada segmento, ofreciendo vehículos que se ajusten a sus necesidades.

Idea principal

En base a los datos disponibles, la opción con mayores posibilidades de implementación es la de “Informes de mercado personalizados”.

Problema que se aborda: En el competitivo mercado automotriz, concesionarios, fabricantes y empresas de seguros enfrentan el desafío de tomar decisiones informadas en un entorno dinámico y en constante cambio. La falta de acceso a datos precisos y análisis detallados puede llevar a decisiones subóptimas, afectando la rentabilidad y el crecimiento.

Solución propuesta: DataVista Analytics propone la creación de informes de mercado personalizados, basados en el análisis exhaustivo del dataset df_autos. Estos informes proporcionarán insights valiosos sobre tendencias del mercado, preferencias de los clientes, análisis de la competencia y oportunidades de mercado. La solución incluye:

- Identificación de Necesidades de los Clientes Potenciales:
  - Determinar qué tipo de información requieren los concesionarios, fabricantes y empresas de seguros.
  - Identificar las preguntas clave que pueden responderse utilizando los datos del dataset.

- Definición de los Tipos de Informes a Ofrecer:
    - Tendencias del mercado (marcas, modelos, precios).
    - Preferencias de los clientes (características, tipo de vehículo).
    - Análisis de la competencia.
    - Oportunidades de mercado.

- Desarrollo de Herramientas de Análisis:
    - Escribir funciones en Python para procesar y analizar los datos del dataset.
    - Utilizar librerías como Pandas y Matplotlib para generar visualizaciones en forma de gráficos y tablas.

- Creación de Plantillas de Informes:
    - Diseñar informes con una estructura clara y visualmente atractiva.
    - Incluir gráficos, tablas y resúmenes de los hallazgos principales.

- Automatización de la Generación de Informes:
    - Escribir un script en Python que genere informes a partir de los datos del dataset y las plantillas diseñadas.
    - Permitir la personalización de los informes según las necesidades específicas del cliente.

Rol de la ciencia de datos: La ciencia de datos jugará un papel crucial en la implementación de esta idea, permitiendo el procesamiento y análisis eficiente de grandes volúmenes de datos. Utilizando técnicas avanzadas de análisis y visualización de datos, DataVista Analytics podrá transformar datos brutos en insights accionables, facilitando la toma de decisiones informadas.

Propuesta de valor

Los beneficios esperados al implementar los informes de mercado personalizados incluyen:

- Aumento de ingresos: Ofrecer un nuevo servicio genera una nueva fuente de ingresos para la empresa.
- Posicionamiento como experto: **DataVista Analytics** se consolidaría como un referente en análisis de datos para el sector automotriz.
- Mayor visibilidad: Atraer nuevos clientes y consolidar la relación con los existentes.

- Ventaja competitiva: Ofrecer un servicio innovador y de alto valor agregado.
- Expansión a nuevos mercados: Los informes pueden adaptarse a diferentes necesidades y mercados.
- Mejora de la toma de decisiones: Tanto para DataVista como para sus clientes.
- Fidelización de clientes: al ofrecer un servicio personalizado, DataVista podría fidelizar clientes y generar un flujo de ingresos constante.

La implementación de informes de mercado personalizados no solo resolverá problemas críticos para los clientes, sino que también ofrecerá múltiples beneficios estratégicos para **DataVista Analytics**, posicionándola como líder en el análisis de datos en el mercado automotriz.

Expectativas y Objetivos del Proyecto

Objetivos a corto plazo

Semana 1: Preparación y análisis inicial

- Revisión del dataset:
  - Revisar y limpiar el dataset df_autos para asegurar la calidad de los datos.
  - Identificar las variables clave y posibles inconsistencias.
  - Entregable: Informe de calidad de datos y plan de limpieza.
- Definición de requisitos:
  - Reunirse con los stakeholders (concesionarios, fabricantes, empresas de seguros) para entender sus necesidades y expectativas.
  - Definir las preguntas clave que los informes deben responder.
  - Entregable: Documento de requisitos y expectativas del cliente.

Semana 2: Desarrollo de herramientas de análisis

- Desarrollo de funciones en python:
  - Escribir funciones en Python para procesar y analizar los datos del dataset.
  - Utilizar librerías como Pandas para manipulación de datos.

- Entregable: Código funcional para el procesamiento de datos.

- Generación de visualizaciones:
  - Crear visualizaciones iniciales utilizando Matplotlib y Seaborn.
  - Generar gráficos y tablas que representen las tendencias y patrones identificados.
  - Entregable: Conjunto de visualizaciones preliminares.

Semana 3: Diseño de plantillas de informes

- Diseño de plantillas:
  - Diseñar plantillas de informes con una estructura clara y visualmente atractiva.
  - Incluir secciones para gráficos, tablas y resúmenes de hallazgos.
  - Entregable: Plantillas de informes en formato editable.

- Revisión y feedback:
  - Presentar las plantillas a los stakeholders para obtener feedback.
  - Ajustar el diseño según las sugerencias recibidas.
  - Entregable: Plantillas revisadas y aprobadas.

Semana 4: Desarrollo de prototipos de informes

- Creación de prototipos:
  - Generar prototipos de informes utilizando los datos del dataset y las plantillas diseñadas.
  - Incluir ejemplos de gráficos y análisis detallados.
  - Entregable: Prototipos de informes listos para revisión.

- Validación de prototipos:
  - Validar los prototipos con los stakeholders para asegurar que cumplen con los requisitos.
  - Realizar ajustes necesarios basados en el feedback.
  - Entregable: Prototipos validados y ajustados.

Semana 5: Automatización de la generación de informes

- Desarrollo del script de automatización:
    - Escribir un script en Python que automatice la generación de informes a partir de los datos del dataset y las plantillas diseñadas.
    - Incluir opciones de personalización según las necesidades específicas del cliente.
    - Entregable: Script de automatización funcional.

- Pruebas de automatización:
    - Realizar pruebas exhaustivas del script para asegurar su correcto funcionamiento.
    - Ajustar y optimizar el script según los resultados de las pruebas.
    - Entregable: Script probado y optimizado.

Semana 6-7: Implementación y pruebas piloto

- Implementación inicial:
    - Implementar el sistema de generación de informes en un entorno de prueba.
    - Realizar pruebas piloto con clientes seleccionados.
    - Entregable: Sistema implementado y pruebas piloto realizadas.

- Recopilación de feedback:
    - Recopilar feedback de los clientes piloto sobre la utilidad y precisión de los informes.
    - Realizar ajustes y mejoras basadas en el feedback recibido.
    - Entregable: Informe de feedback y plan de mejoras.

Semana 8-11: Lanzamiento y comercialización

- Lanzamiento del servicio:
    - Lanzar el servicio de informes de mercado personalizados al mercado.
    - Establecer relaciones con concesionarios, fabricantes y empresas de seguros para ofrecerles los informes.
    - Entregable: Servicio lanzado y disponible para clientes.

- Monitoreo y optimización:
    - Monitorear el uso y la efectividad de los informes.

- Continuar optimizando el servicio basado en el feedback continuo de los clientes.
- Entregable: Informes de monitoreo y optimización continua.

Objetivos a largo plazo

- Impacto en la empresa:
  - Posicionar a DataVista Analytics como líder en el análisis de datos dentro del sector automotriz.
  - Diversificar las fuentes de ingresos mediante la venta de informes personalizados.
  - Aumentar la fidelización y satisfacción de los clientes actuales.
- Impacto en el mercado:
  - Proveer a los actores del mercado automotriz con herramientas de análisis avanzadas para tomar decisiones informadas.
  - Facilitar la identificación de tendencias y oportunidades de mercado, mejorando la competitividad del sector.

Objetivos del Equipo

Fase 1: Análisis y Desarrollo (Semanas 1-3)

- Hito 1: Completar el análisis inicial del dataset df_autos.
- Hito 2: Desarrollar las funciones de análisis en Python.
- Entregable: Código funcional para el procesamiento y análisis de datos.

Fase 2: Diseño y Prototipado (Semanas 4-5)

- Hito 3: Diseñar plantillas de informes con gráficos y tablas.
- Hito 4: Crear prototipos de informes personalizados.
- Entregable: Plantillas de informes y prototipos listos para revisión.

Fase 3: Automatización y Personalización (Semanas 6-7)

- Hito 5: Desarrollar el script de automatización para la generación de informes.
- Hito 6: Implementar opciones de personalización en los informes.
- Entregable: Script de automatización y opciones de personalización funcionales.

Fase 4: Implementación y Lanzamiento (Semanas 8-11)

- Hito 7: Realizar pruebas piloto con clientes seleccionados.
- Hito 8: Ajustar y optimizar los informes basados en feedback.
- Hito 9: Lanzar el servicio de informes de mercado personalizados.
- Entregable: Servicio de informes lanzado y disponible para clientes.

Metas Individuales

Roles y Responsabilidades

Chief Data Officer (CDO):

- Descripción del rol: La Coordinadora de la Tecnicatura Superior en Ciencia de Datos e Inteligencia Artificial será responsable de la visión integral del proyecto. Su función principal será asegurar que la solución esté alineada con los objetivos educativos y tecnológicos del equipo. Además, supervisará el progreso general, proporcionando orientación estratégica para garantizar el éxito del proyecto.
- Tareas y responsabilidades:
    - Monitorear y supervisar el progreso del equipo.
    - Asegurar que la solución propuesta esté alineada con los objetivos educativos y tecnológicos.
    - Proveer retroalimentación estratégica y garantizar la calidad del trabajo final.

Supervisor:

- Descripción del rol: El profesor del curso servirá como supervisor, guiando al equipo en la correcta implementación del proyecto. Será responsable de asegurar que los métodos utilizados sean académicamente válidos y que se mantenga la calidad en el desarrollo del modelo predictivo y la documentación.
- Tareas y responsabilidades:
    - Proveer retroalimentación en cada fase del proyecto.
    - Monitorear y asegurar la calidad técnica y documental de los entregables.

Responsable de Equipo de Trabajo:

- Descripción del rol: Este rol será rotativo entre los miembros del equipo y garantizará que todas las tareas se completen a tiempo y que la comunicación entre los roles del proyecto fluya correctamente.
- Tareas y responsabilidades:
    - Coordinar las actividades diarias del equipo.
    - Asegurar que los hitos del proyecto se cumplan según el cronograma establecido.
    - Gestionar la comunicación entre el equipo y los supervisores.
- Metas individuales:
    - Mantener un flujo de trabajo eficiente y organizado.
    - Asegurar la entrega de los hitos del proyecto en el tiempo estimado.
- Plan de desarrollo personal:
    - Desarrollo de liderazgo: Mejorar las habilidades de liderazgo en la gestión de equipos multifuncionales, aprendiendo a delegar tareas de manera efectiva y asegurar una comunicación clara y precisa.
    - Gestión de proyectos: Aumentar la capacidad de seguimiento y control de múltiples fases del proyecto, dominando herramientas de gestión de proyectos y optimizando la coordinación del equipo para asegurar entregables de alta calidad.

Primer Reemplazo:

- Descripción del rol: Asumirá las responsabilidades del Responsable de Equipo en su ausencia, garantizando la continuidad del proyecto.
- Tareas y responsabilidades:
    - Tomar decisiones operativas en ausencia del Responsable de Equipo.
    - Mantener al equipo informado sobre el progreso del proyecto.
- Metas individuales:
    - Asegurar la continuidad en la gestión y el progreso del proyecto.
    - Mantener una vista general clara de todas las tareas en curso.
- Plan de desarrollo personal:

- Adaptabilidad y toma de decisiones: Fortalecer las habilidades de toma de decisiones rápidas en situaciones imprevistas, mejorando la capacidad de adaptación frente a cambios en el entorno de trabajo.
- Liderazgo de emergencia: Desarrollar habilidades de liderazgo cuando se enfrenta a situaciones de presión, siendo capaz de asumir roles de responsabilidad cuando sea necesario, y optimizar la gestión del tiempo para cumplir con los plazos.

Segundo Reemplazo:

- Descripción del rol: Actuará cuando tanto el Responsable de Equipo como el Primer Reemplazo estén ausentes, asegurando que el proyecto continúe sin interrupciones.
- Tareas y responsabilidades:
  - Supervisar el progreso del equipo en caso de emergencia.
  - Mantener la coordinación entre los miembros del equipo para garantizar que se cumplan los plazos del proyecto.
- Metas individuales:
  - Estar siempre preparado para asumir responsabilidades en cualquier momento.
  - Mantener una supervisión continua del progreso del proyecto.
- Plan de desarrollo personal:
  - Preparación y proactividad: Desarrollar una mentalidad proactiva, anticipando posibles problemas y manteniéndose siempre informado sobre el progreso del proyecto.
  - Gestión del estrés y resolución de problemas: Mejorar la capacidad para gestionar el estrés en situaciones críticas, fortaleciendo la resolución de problemas y la capacidad para asumir decisiones en momentos de alta presión.

Plan de Acción

Fases del Proyecto

Fase 1: Análisis Inicial y Preparación (Semanas 1-2)

- Actividades:
  - Revisión del dataset: Identificar posibles inconsistencias, valores atípicos y datos faltantes en el dataset df_autos.
  - Limpieza de datos: Implementar técnicas de limpieza como imputación de datos faltantes, eliminación de duplicados y estandarización de variables.
  - Identificación de variables clave: Determinar las variables que serán esenciales para el análisis y la creación de informes personalizados.
- Recursos necesarios: Dataset df_autos, librerías de Python (Pandas, NumPy), equipo de analistas de datos.
- Plazos: 2 semanas.

Fase 2: Desarrollo de Herramientas de Análisis (Semanas 3-4)

- Actividades:
  - Desarrollo de funciones en Python: Creación de funciones específicas para el análisis exploratorio de datos, incluyendo estadísticas descriptivas y visualizaciones.
  - Generación de visualizaciones avanzadas: Uso de Matplotlib y Seaborn para crear gráficos que muestren tendencias, distribuciones y correlaciones clave.
  - Definición de métricas de éxito: Establecer métricas para evaluar la efectividad de los análisis y la calidad de los informes generados.
- Recursos necesarios: Python, Matplotlib, Seaborn, Jupyter Notebooks, equipo técnico.
- Plazos: 2 semanas.

Fase 3: Diseño y Prototipado de Informes (Semanas 5-7)

- Actividades:
  - Diseño de plantillas de informes: Crear plantillas estandarizadas que incluyan

gráficos, tablas y resúmenes, adaptadas a diferentes tipos de clientes.
    - Prototipado de informes: Generación de prototipos de informes utilizando datos reales del dataset para validar el diseño y el contenido.
    - Revisión con stakeholders: Presentación de prototipos a los clientes potenciales para recibir feedback y realizar ajustes necesarios.

- Recursos necesarios: Herramientas de diseño gráfico, software de edición de documentos (LaTeX, MS Word), equipo de diseño y marketing.
- Plazos: 3 semanas.

Fase 4: Automatización y Desarrollo de Personalización (Semanas 8-9)

- Actividades:
    - Desarrollo de scripts de automatización: Programación de scripts en Python para automatizar la generación de informes a partir del dataset y las plantillas diseñadas.
    - Personalización de informes: Implementación de opciones de personalización para que los informes se adapten a las necesidades específicas de cada cliente.
    - Pruebas de funcionalidad: Realización de pruebas exhaustivas para asegurar que el proceso automatizado funcione sin errores y cumpla con los requisitos.
- Recursos necesarios: Python, librerías de automatización (Jinja2, Pandas), servidores de prueba, equipo técnico.
- Plazos: 2 semanas.

Fase 5: Implementación, Pruebas Piloto y Lanzamiento (Semanas 10-12)

- Actividades:
    - Implementación en entorno de producción: Despliegue del sistema de generación de informes en un entorno real.
    - Pruebas piloto: Colaboración con un grupo selecto de clientes para realizar pruebas piloto, recopilando datos de rendimiento y feedback.
    - Lanzamiento comercial: Realización del lanzamiento oficial del servicio,

incluyendo la creación de campañas de marketing y estrategias de venta.

- Recursos necesarios: Infraestructura de servidores, herramientas de monitorización (Prometheus, Grafana), equipo de ventas y marketing.
- Plazos: 3 semanas.

Hitos y Entregables

- Hito 1: Finalización del análisis inicial y limpieza de datos (Semana 2)
    - Entregable: Informe de calidad de datos y plan de limpieza detallado.
    - Fechas de revisión: 29/08.
- Hito 2: Desarrollo y validación de funciones de análisis (Semana 4)
    - Entregable: Código Python para análisis y visualizaciones, conjunto de gráficos y tablas preliminares.
    - Fechas de revisión: 10/09.
- Hito 3: Diseño y validación de plantillas de informes (Semana 7)
    - Entregable: Plantillas de informes aprobadas por stakeholders y prototipos de informes.
    - Fechas de revisión: 28/09.
- Hito 4: Desarrollo del script de automatización y personalización (Semana 9)
    - Entregable: Script funcional de automatización con opciones de personalización.
    - Fechas de revisión: 10/10.
- Hito 5: Lanzamiento del servicio de informes personalizados (Semana 12)
    - Entregable: Servicio operativo, informes generados y entregados a clientes.
    - Fechas de revisión: 04/11.

Metodología de Trabajo

Frecuencia de Reuniones:

- Reuniones iniciales: Reuniones diarias durante las primeras dos semanas para asegurar una comprensión clara del proyecto y alineación de objetivos.

- Reuniones semanales: A partir de la tercera semana, reuniones semanales para revisar el progreso, resolver problemas y ajustar el plan según sea necesario.
- Reuniones de revisión: Reuniones al final de cada fase para evaluar los entregables y preparar la siguiente etapa.

Herramientas de Comunicación y Colaboración:

- Google Drive: Para la gestión de documentos compartidos y colaboración en tiempo real.
- Trello: Para el seguimiento de tareas y gestión de proyectos, incluyendo la asignación de tareas y la monitorización del progreso.
- GitHub: Para el control de versiones del código y la colaboración en el desarrollo de scripts.

Proceso de Seguimiento y Revisión:

- Monitoreo del progreso: Uso de herramientas como Trello para seguir el avance de cada tarea, asignando responsables y fechas límite.
- Revisión continua: Evaluaciones periódicas de los entregables por parte del equipo y los stakeholders para asegurar que se cumplen los requisitos y se mantienen los estándares de calidad.
- Gestión de riesgos: Identificación temprana de posibles obstáculos o problemas, con planes de contingencia preparados para asegurar la continuidad del proyecto.

ANÁLISIS EXPLORATORIO DE DATOS DE FLOTA DE AUTOS

Resumen Ejecutivo

En esta fase del proyecto, se llevó a cabo un **Análisis Exploratorio de Datos (EDA)** sobre un dataset de una flota de automóviles con el objetivo de entender la estructura y características de los datos, identificar patrones iniciales y detectar problemas como valores atípicos o datos faltantes. Los análisis proporcionaron insights preliminares que serán útiles para la toma de decisiones informadas en futuras etapas del proyecto.

Introducción

El objetivo del EDA es explorar los datos, generar visualizaciones descriptivas y descubrir patrones que permitan mejorar la comprensión de las variables. Esta fase es esencial para el análisis posterior, ya que establece las bases para modelado predictivo o recomendaciones estratégicas para la flota automotriz.

El dataset incluye información detallada sobre automóviles, como el precio, marca, modelo, año de fabricación, tipo de combustible, kilometraje acumulado, y otras características relevantes.

Metodología

Proceso de Análisis

El análisis se llevó a cabo utilizando librerías de Python como **Pandas** y **Seaborn** para la manipulación y visualización de datos. Los pasos realizados fueron:

1. **Carga y limpieza del dataset:** Identificación y tratamiento de valores faltantes o inconsistentes.

2. **Análisis univariado:** Exploración de las variables individuales mediante estadísticas descriptivas.
3. **Visualización:** Histogramas, boxplots y gráficos de barras para representar distribuciones y relaciones.
4. **Análisis bivariado:** Exploración de relaciones entre pares de variables, utilizando gráficos de dispersión y mapas de calor de correlaciones.

Datos Utilizados

El dataset con el que se trabajó en este análisis contiene múltiples características relacionadas con automóviles de una flota. A continuación, se describen las columnas del dataset:

- **Marca:** marca del auto. **Tipo de dato:** Cadena de caracteres.
- **Modelo:** modelo específico del auto. **Tipo de dato:** Cadena de caracteres.
- **Año:** año de fabricación del auto. **Tipo de dato:** Entero.
- **Color:** color del exterior del auto. **Tipo de dato:** Cadena de caracteres.
- **Combustible:** tipo de combustible que utiliza el vehículo**. Tipo de dato:** Cadena de caracteres.
- **Puertas:** cantidad de puertas que tiene el vehículo. **Tipo de dato:** Entero.
- **Caja:** tipo de caja de cambios del vehículo. **Tipo de dato:** Cadena de caracteres.
- **Motor:** tamaño del motor del auto,expresado en litros. **Tipo de dato:** Flotante con un decimal.
- **Carrocería:** tipo de carrocería del vehículo**. Tipo de dato:** Cadena de caracteres.
- **Kilómetros:** kilometraje acumulado por el auto.**Tipo de dato:** Entero.
- **Moneda:** moneda en la que se cotiza el precio del auto. **Tipo de dato:** Cadena de caracteres.
- **Precio:** precio del auto, expresado en la moneda especificada en la columna "Moneda". **Tipo de dato:** Entero.

Hallazgos Clave

Año

- La mayoría de los autos tienen menos de 10 años. El promedio de fabricación es 2016, lo que sugiere que **el parque automotor es relativamente moderno**.

Kilometraje

- El kilometraje promedio es de 74.732 km. La mayoría de **los autos han recorrido distancias moderadas**, con un 75% por debajo de 99.100 km, indicando que en general los vehículos están en buen estado.

Puertas

- La mayoría de **los autos tiene entre 4 y 5 puertas**, con un promedio de 4.47 puertas. Esto sugiere que el parque está compuesto principalmente por **vehículos familiares o utilitarios**, que suelen tener más puertas para comodidad de los pasajeros o para uso comercial.

Motor

- El **motor promedio es de 1.88 litros**, lo que indica que el parque está compuesto principalmente por **autos de cilindrada moderada**, comúnmente usados para tareas estándar.

- Hay autos con motores pequeños (1.0 litros), y el valor máximo de 6.4 litros probablemente corresponda a un auto de alta gama o de características muy específicas.

Carrocería y Kilometraje:

- Las **minivans presentan los kilometrajes más altos**, lo cual es coherente con su uso en trayectos largos. Los Coupés, por otro lado, tienen el kilometraje más bajo, posiblemente debido a su naturaleza recreativa.

Tipo de Combustible:

- Los vehículos con **nafta y diésel son los más comunes**, mientras que los híbridos son modelos más recientes en el mercado, con menores rangos de antigüedad.

Precio

- De momento tiene poco sentido analizar esta variable, porque hay autos nominados en pesos, y autos nominados en dólares.

Visualizaciones y Tendencias

Se utilizaron gráficos de barras para identificar tendencias en la **distribución por marca** y **año de fabricación**, mostrando que la mayor parte del parque se comercializó entre 2012 y 2020.

Un **gráfico de caja (boxplot)** reveló la gran variabilidad en kilometraje por marcas como BMW, Renault y Volkswagen, mientras que marcas como DS y Suzuki mostraron kilometrajes más uniformes y bajos.

En la **distribución de cajas** (manual vs automática), el 58.8% de los autos en el mercado tienen caja manual, lo que sugiere una preferencia por vehículos más económicos en cuanto a mantenimiento.

Conclusión del Análisis Exploratorio

El análisis exploratorio de datos permitió identificar patrones relevantes en el parque automotor, destacando que los vehículos con mayor kilometraje, como las minivans, requieren una **atención especial en mantenimiento preventivo**. Esto es fundamental para reducir costos operativos a largo plazo y mejorar la eficiencia del uso de estos autos.

Además, se evidenció que ciertos tipos de vehículos, **según su año** de fabricación **y tipo de uso**, presentan una mayor durabilidad, lo que puede ayudar a los clientes a optimizar sus decisiones de compra y mantenimiento. **La segmentación** por tipo de combustible y caja de cambios también proporciona una herramienta valiosa para elegir autos más eficientes y adecuados a las necesidades específicas de los usuarios.

Finalmente, el análisis sugiere que **las estrategias de renovación del parque** deben tener en cuenta las características específicas de los vehículos, como el kilometraje y el tipo de carrocería, para maximizar el retorno de inversión y garantizar un uso más prolongado y efectivo de los mismos.

MODELADO PREDICTIVO

Metodología de modelado

La metodología de modelado utilizada siguió un enfoque típico y estructurado dentro de la ciencia de datos y machine learning, conocido como el ciclo de vida de un proyecto de modelado predictivo. Este proceso incluyó varias etapas que son importantes para garantizar que el modelo que se construye sea robusto, interpretable y útil para los objetivos del negocio.

Dichas etapas fueron:

- Definir el problema: Predecir el precio de los autos en base a varias características.
- Preprocesamiento de datos: Transformar variables categóricas y numéricas, limpieza de datos, división en conjuntos de entrenamiento y prueba.
- Selección de modelos: Probar diferentes algoritmos de aprendizaje (Regresión lineal múltiple, Gradient Boosting, Random Forest).
- Entrenamiento del modelo: Ajustar cada modelo utilizando los datos de entrenamiento.
- Evaluación del rendimiento: Utilizar métricas como MSE, MAE, y R^2 para comparar modelos y seleccionar el mejor.
- Optimización: Ajustar hiperparámetros (si es necesario) para mejorar el rendimiento del modelo seleccionado.
- Interpretación y conclusiones: Aplicar los resultados obtenidos al contexto del negocio.

Resultados de entrenamiento y evaluación

Se evaluó el rendimiento de cada modelo comparando las métricas obtenidas:

	MSE	MAE	R^2
Regresión lineal múltiple - Primera prueba	1424760535.80	10918.03	0.36
Regresión lineal múltiple - Tercera prueba	786232977.76	8405.88	0.65
Gradient Boosting	657591633.06	6342.08	0.71
Random Forest	956274644.38	7212.61	0.57

Selección del mejor modelo

Después de evaluar varios modelos predictivos, se optó por el que logró el mejor equilibrio entre precisión y robustez: el **Gradient Boosting**. Este modelo destacó por su rendimiento, alcanzando los valores más bajos de error cuadrático medio (MSE) y los más altos de coeficiente de determinación (R^2).

Si bien la tercera implementación de la regresión lineal múltiple arrojó resultados aceptables, el Gradient Boosting no solo capturó de manera más efectiva las relaciones no lineales entre las variables, sino que también mostró una menor propensión al overfitting, lo que lo convierte en la opción más sólida para la predicción de precios de autos usados.

INFORME Y DOCUMENTACIÓN TÉCNICA

Resumen

En esta parte se presenta el informe técnico que sintetiza el trabajo realizado en el curso de **Práctica Profesionalizante I**, documentando el proceso de carga y revisión de datos sobre ventas de vehículos automotores, así como la evaluación de tres modelos de aprendizaje automático y el análisis de los resultados obtenidos.

Introducción

Este trabajo se desarrolla en el marco de un espacio curricular que simula un entorno empresarial dedicado al análisis de datos, la empresa **DataVista Analytics**, con el objetivo de abordar la carencia de análisis de información en tiempo real para la toma de decisiones en la industria automotriz.

La estructura del trabajo incluyó la carga de librerías y configuración del entorno, una preparación básica del conjunto de datos, la evaluación de tres modelos de aprendizaje automático, un análisis comparativo de sus resultados y un ajuste detallado de los parámetros de uno de los modelos.

Metodología

La metodología de trabajo utilizada fue un entorno tipo Notebooks de **Jupyter/Colab**, ya que se basa en un enfoque interactivo que permite combinar celdas de código y texto en un solo documento. Por tanto, se pueden escribir y ejecutar bloques de código de manera

independiente, lo que facilita la experimentación y la depuración, mientras que las celdas de texto en formato Markdown permiten documentar el proceso y explicar los resultados. Esta estructura favoreció la ejecución y la colaboración grupal en tiempo real, mostrando resultados inmediatos y permitiendo ajustes dinámicos en los métodos utilizados.

Desarrollo

En esta sección, se presenta un resumen de las diversas partes del código implementadas y ejecutadas en etapas sucesivas. Para obtener información más detallada, se sugiere consultar el trabajo completo, donde se proporciona una descripción más exhaustiva y comentarios sobre el código.

1. Configuramos un entorno de trabajo en Google Colab para el análisis de datos utilizando la biblioteca Pandas. Primero, se habilita el formato de visualización de tablas para los DataFrames mediante `data_table.enable_dataframe_formatter()`. Luego, se importa la biblioteca Pandas y se establece una opción para mostrar números flotantes con dos decimales, utilizando `pd.set_option('display.float_format', lambda x: '%.2f' % x)`. Esto permite que todos los números flotantes en las salidas de Pandas se presenten con un formato más legible y estandarizado, mejorando la claridad en la visualización de datos.
2. Cargamos el conjunto de datos en un DataFrame de Pandas llamado `data`. Se proporcionan dos opciones para la carga del dataset: la primera permite cargarlo desde el entorno local mediante un archivo CSV, que debe ser activada al comentar la línea correspondiente; la segunda opción, que está activa, lee el archivo directamente desde una URL en GitHub. Finalmente, incluimos una línea para verificar el contenido del DataFrame 'data', mostrando así los datos cargados.
3. Establecemos una tasa de cambio de 1 USD a 380 ARS y definimos una función llamada `convertir_a_dolar` que convierte precios de pesos a dólares, dependiendo del valor de la columna `Moneda` en cada fila del DataFrame. Si la moneda es `pesos`, el precio se divide por la tasa de cambio; de lo contrario, se mantiene sin cambios. A continuación, se aplica esta función a cada fila del DataFrame `data`, redondeando los resultados a dos decimales. Posteriormente, se eliminan las columnas `Moneda` y `Año_zscore` del DataFrame, y finalmente, verificamos el contenido del DataFrame transformado.

4. Utilizamos la función `train_test_split` de la biblioteca `sklearn.model_selection` para dividir el conjunto de datos en conjuntos de entrenamiento y prueba. Primero, se seleccionan las características (X) excluyendo la columna `Precio`, que se define como la variable objetivo (y). Luego, se realiza la división del dataset, asignando el 80% de los datos a los conjuntos de entrenamiento (`X_train` y `y_train`) y el 20% restante a los conjuntos de prueba (`X_test` y `y_test`), utilizando un parámetro `random_state` para asegurar la reproducibilidad del proceso. Esto permite evaluar el rendimiento del modelo de manera efectiva al entrenarlo con un conjunto de datos y probarlo con otro.
5. **Regresión lineal múltiple:** implementamos un modelo de regresión lineal utilizando la biblioteca `scikit-learn`, comenzando por la definición de las columnas predictoras y la variable objetivo a partir del conjunto de datos previamente cargado. Se identifican las características categóricas y numéricas, y se aplica un proceso de codificación para las variables categóricas mediante `LabelEncoder`, que incluye una función personalizada para manejar valores desconocidos durante la transformación. A continuación, se crea un preprocesador utilizando `ColumnTransformer` para mantener las características numéricas sin cambios. Posteriormente, se establece un `Pipeline` que integra el preprocesador y el modelo de regresión lineal. El modelo se entrena con los datos de entrenamiento y se realizan predicciones sobre el conjunto de prueba. Finalmente, se evalúa el rendimiento del modelo utilizando métricas como el error cuadrático medio (MSE), el error absoluto medio (MAE) y el coeficiente de determinación (R^2), imprimiendo los resultados obtenidos.
6. En otra implementación de **regresión lineal múltiple** aplicamos One-Hot Encoding a las columnas categóricas del conjunto de datos, identificando previamente las características categóricas y numéricas. Se definen las columnas categóricas así como las columnas numéricas. A continuación, se crea un preprocesador utilizando `ColumnTransformer`, que aplica el codificador One-Hot a las variables categóricas mientras mantiene las columnas numéricas sin cambios mediante el parámetro `remainder='passthrough'`. Esto nos permitiò transformar adecuadamente los datos para su posterior uso en el modelo de aprendizaje automático, asegurando que las variables categóricas sean representadas de manera adecuada.
7. **Gradient Boosting:** implementamos un modelo de regresión utilizando el algoritmo de Gradient Boosting para predecir precios a partir del conjunto de datos. Primero, definimos las características (X) excluyendo la columna `Precio`, que se establece como la variable objetivo (y). Se identifican las columnas categóricas y numéricas, y se crea un preprocesador mediante `ColumnTransformer` que aplica One-Hot Encoding a las

variables categóricas, ignorando categorías no vistas, mientras mantiene las columnas numéricas sin cambios. A continuación, se construye un `Pipeline` que integra el preprocesador y el modelo de Gradient Boosting, configurado con 100 estimadores, una tasa de aprendizaje de 0.1 y una profundidad máxima de 3. El conjunto de datos se divide en entrenamiento y prueba (80% y 20%, respectivamente), y el modelo se entrena con los datos de entrenamiento. Posteriormente, se realizan predicciones sobre el conjunto de prueba y se evalúa el rendimiento del modelo utilizando métricas como el error cuadrático medio (MSE), el error absoluto medio (MAE) y el coeficiente de determinación (R^2), imprimiendo los resultados obtenidos.

8. **Random Forest:** el último modelo que implementamos es un modelo de regresión utilizando el algoritmo de Random Forest para predecir precios a partir del conjunto de datos. Definimos las características (X) excluyendo la columna `Precio`, que se establece como la variable objetivo (y). Se identificaron las columnas categóricas y numéricas, y se creó un preprocesador mediante `ColumnTransformer`, que aplica One-Hot Encoding a las variables categóricas, ignorando categorías no vistas y manteniendo las columnas numéricas sin cambios. A continuación, se construyó un `Pipeline` que integra el preprocesador y el modelo de Random Forest, configurado con 100 estimadores y un estado aleatorio para asegurar la reproducibilidad. El conjunto de datos se dividió en entrenamiento y prueba (80% y 20%, respectivamente), y el modelo se entrenó con los datos de entrenamiento. Posteriormente, se realizaron predicciones sobre el conjunto de prueba y se evaluó el rendimiento del modelo utilizando métricas como el error cuadrático medio (MSE), el error absoluto medio (MAE) y el coeficiente de determinación (R^2), imprimiendo los resultados obtenidos.

CONCLUSIONES

Resultados y análisis

Aquí se presenta el rendimiento de cada modelo comparando las métricas obtenidas:

	MSE	MAE	R^2
Regresión lineal múltiple - Primera prueba	1424760535.80	10918.03	0.36
Regresión lineal múltiple - Tercera prueba	786232977.76	8405.88	0.65
Gradient Boosting	657591633.06	6342.08	0.71
Random Forest	956274644.38	7212.61	0.57

Por último, interpretamos los resultados del modelo en el contexto del negocio automotriz, analizando su utilidad para la toma de decisiones y sus limitaciones. Las predicciones de precios de vehículos usados son cruciales para concesionarios y plataformas de venta, ya que permiten ajustar precios, formular estrategias de compra y establecer precios competitivos.

1. **MSE (Error Cuadrático Medio):** El modelo de Gradient Boosting presenta el menor MSE (657 millones), lo que indica que sus predicciones son más precisas, ayudando a evitar errores en la valoración de automóviles. La regresión lineal muestra mejoras, pero sigue siendo menos precisa.
2. **MAE (Error Absoluto Medio):** Con un MAE de 6342,08, Gradient Boosting también es el más preciso. Este error puede ser aceptable o no, dependiendo del precio promedio de los vehículos; es más significativo en autos de menor valor. Comparado con la regresión lineal (8405,88) y Random Forest (7212,61), el Gradient Boosting se destaca.

3. **R^2 (Coeficiente de determinación):** Gradient Boosting explica el 71% de la variabilidad en los precios, lo que indica un buen ajuste del modelo. La regresión lineal tiene un R^2 de 0.65 y Random Forest un 0.57, sugiriendo que este último es menos confiable para decisiones críticas.

En resumen, el modelo de Gradient Boosting se muestra como la mejor opción para predecir precios en el negocio automotriz, aunque cada modelo tiene sus propias limitaciones y aplicaciones.

Conclusiones finales

Gradient Boosting se destacó como el mejor modelo, presentando el menor MSE, MAE y el mayor R^2. En el ámbito empresarial, su uso podría resultar en predicciones de precios más precisas para automóviles, optimizando ganancias, ajustando estrategias de precios y minimizando errores significativos.

Impacto en el negocio: La precisión en la predicción de precios ayuda a evitar tanto la sobrevaloración, que puede impedir ventas, como la subvaloración, que puede resultar en pérdidas. Un menor error promedio con Gradient Boosting permite a la empresa establecer precios más competitivos, lo que podría aumentar las ventas y reducir riesgos financieros.

Posibles mejoras o consideraciones: Si los errores de predicción son aún relevantes, se podrían ajustar los hiperparámetros del modelo o explorar técnicas como el stacking de modelos y el uso de más datos disponibles. Además, la calidad de los datos es crucial; contar con información adicional sobre ventas y demanda podría mejorar aún más la precisión del modelo.

Por último, se logró cumplir con todos los objetivos de análisis, incluyendo la exploración y limpieza de datos, así como el desarrollo de herramientas específicas para el análisis. No obstante, debido a limitaciones temporales, no fue posible finalizar el desarrollo de las plantillas de informes ni la automatización en la generación de los mismos. Así, la interfaz final destinada al usuario aún se encuentra en proceso de implementación.

REFERENCIAS

Referencias bibliográficas:

Géron, A. (2020). *Aprende Machine Learning con Scikit-Learn, Keras y TensorFlow: conceptos, herramientas y técnicas para construir sistemas inteligentes.* Anaya Multimedia.

Ciencia de Datos. (2024). *Material formativo sobre estadística, algoritmos de machine learning, ciencia de datos y programación en R y Python.* https://cienciadedatos.net/

Ciencia de Datos. (2024). *Machine learning con Python y scikit-learn.* https://cienciadedatos.net/documentos/py06_machine_learning_python_scikitlearn

Ciencia de Datos. (2024). *Regresión lineal con Python.* https://cienciadedatos.net/documentos/py10-regresion-lineal-python

Zapata, J. (2023). *Clasificación en Python para Ciencia de Datos.* https://joserzapata.github.io/courses/python-ciencia-datos/clasificacion

Apuntes de la materia

Bibliotecas externas utilizadas (por orden de aparición):

Google. (s.f.). Google Colaboratory (s.f.) [Software]. Google. https://colab.research.google.com

McKinney, W. (2010). pandas: a foundational Python library for data analysis (2024) [Software]. GitHub. https://pandas.pydata.org/

Cournapeau, D. (2011). scikit-learn: Machine Learning in Python (Versión 0.24.2) [Software]. Journal of Machine Learning Research. https://scikit-learn.org/

Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585, 357–362. https://numpy.org/

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95. https://matplotlib.org/

Waskom, M. L. (2021). seaborn: statistical data visualization (Versión 0.11.1) [Software]. GitHub. https://seaborn.pydata.org/