Ciencia de datos

Big data

John Matthew Espinosa Rojas

Profesor: Sebastián Perdomo

Escuela Tecnológica Instituto Técnico Central

Bogotá D.C 2024

John Matthew Espinosa. Escuela Tecnológica Instituto Técnico Central. Semestre 7.

Correspondencia: jmespinosar@itc.edu.co Tel. 3157118023

Tabla de contenidos

1. **Introducción**

La ciencia de datos es un campo que combina la estadística, matemática y programación para analizar grandes volúmenes de datos y extraer información útil. Su propósito es extraer información valiosa y relevante a partir de datos, por lo tanto, ayuda a tomar decisiones más informadas y a resolver problemas en distintos sectores, desde negocios hasta salud. En resumen, la ciencia de datos transforma los datos, lo cual impulsa el avance y la innovación.

2. Desarrollo

2.1. El Impacto de la ciencia de datos en la industria financiera

La ciencia de datos ha transformado significativamente la industria financiera, impulsando mejoras en la toma de decisiones, la gestión de riesgos y la personalización de productos financieros. A medida que el volumen de datos financieros ha crecido exponencialmente, los bancos y otras instituciones financieras han adoptado herramientas avanzadas de análisis de datos para procesar y extraer valor de esta enorme cantidad de información.

Una de las principales áreas donde la ciencia de datos ha marcado una diferencia es en la gestión de riesgos. Antes, los bancos dependían de modelos tradicionales para evaluar la capacidad crediticia de los clientes, basándose en información limitada como ingresos y historial crediticio. Hoy en día, los algoritmos de machine learning permiten a las instituciones financieras analizar grandes cantidades de datos no estructurados, como el comportamiento en redes sociales o patrones de consumo, lo que lleva a decisiones de crédito más informadas y precisas.

Además, la detección de fraudes ha mejorado considerablemente con el uso de la ciencia de datos. Algoritmos sofisticados analizan transacciones en tiempo real y detectan patrones anómalos que podrían indicar actividades fraudulentas. Gracias a estas herramientas, los bancos pueden reaccionar más rápido y prevenir pérdidas significativas.

El aumento de los datos también ha permitido la personalización de servicios financieros. A través del análisis de datos de comportamiento, las instituciones pueden ofrecer productos financieros más personalizados, como préstamos a medida o recomendaciones de inversión basadas en el perfil de riesgo y objetivos financieros de cada cliente.

2.2. **Informe del análisis descriptivo del dataset Iris**

El Iris Dataset es uno de los conjuntos de datos más conocidos en el campo de la Ciencia de Datos, utilizado frecuentemente para realizar análisis básicos y pruebas de modelos de clasificación. En este informe, se presentan los resultados del análisis descriptivo de las variables numéricas del conjunto de datos, incluyendo la longitud y anchura de sépalos y pétalos para tres especies de flores Iris.

2.2.1. **Resultados obtenidos**

Los resultados muestran que la longitud promedio de los sépalos es de 5.84 cm, mientras que su anchura promedio es de 3.05 cm. La longitud de los pétalos tiene una media de 3.75 cm, y el ancho promedio es de 1.19 cm. Estos valores sugieren una diferencia significativa en las dimensiones entre los sépalos y los pétalos de las flores, lo cual es consistente con las observaciones en la naturaleza, donde los pétalos suelen ser más largos y estrechos que los sépalos.

2.2.2. **Interpretación de resultados**

- **Media:**
  - **Longitud de sépalo:** 5.84 cm
  - **Ancho de sépalo:** 3.05 cm
  - **Longitud de pétalo:** 3.75 cm
  - **Ancho de pétalo:** 1.19 cm

- **Mediana:**
  - **Longitud de sépalo:** 5.80 cm
  - **Ancho de sépalo:** 3.00 cm
  - **Longitud de pétalo:** 4.35 cm
  - **Ancho de pétalo:** 1.30 cm

- **Desviación estándar:**
  - **Longitud de sépalo:** 0.83 cm
  - **Ancho de sépalo:** 0.43 cm
  - **Longitud de pétalo:** 1.76 cm
  - **Ancho de pétalo:** 0.76 cm

```
Características (X):
   sepal length  sepal width  petal length  petal width
0           5.1          3.5           1.4          0.2
1           4.9          3.0           1.4          0.2
2           4.7          3.2           1.3          0.2
3           4.6          3.1           1.5          0.2
4           5.0          3.6           1.4          0.2
5           5.4          3.9           1.7          0.4
6           4.6          3.4           1.4          0.3
7           5.0          3.4           1.5          0.2
8           4.4          2.9           1.4          0.2
9           4.9          3.1           1.5          0.1

Objetivo (Y):
         class
0  Iris-setosa
1  Iris-setosa
2  Iris-setosa
3  Iris-setosa
4  Iris-setosa
5  Iris-setosa
6  Iris-setosa
7  Iris-setosa
8  Iris-setosa
9  Iris-setosa

Media:
sepal length    5.843333
sepal width     3.054000
petal length    3.758667
petal width     1.198667
dtype: float64

Mediana:
sepal length    5.80
sepal width     3.00
petal length    4.35
petal width     1.30
dtype: float64

Desviación estándar:
sepal length    0.828066
sepal width     0.433594
petal length    1.764420
petal width     0.763161
dtype: float64
```

2.3. **Informe algoritmo de optimización simple del dataset Boston Housing**

En este ejercicio se implementó el algoritmo de gradiente descendente en el lenguaje de programación Python, haciendo uso de las librerías Pandas, NumPy y Matplotlib para resolver un problema de regresión lineal por medio del dataset de Boston Housing Dataset. El objetivo era predecir los precios de las viviendas en función de diversas características, como el número de habitaciones, la antigüedad, entre otras.

2.3.1. **Implementación**

El código hace uso de las librerias NumPy, Pandas, y Matplotlib. El algoritmo de gradiente descendente se utiliza para ajustar una regresión lineal simple, que predice el valor de una vivienda (objetivo) en función del número de habitaciones (característica independiente).

- **Carga de datos:** El primer paso fue cargar el dataset con la función fetch_openml de Scikit-learn y preparar las matrices de características (X) y la variable objetivo (Y).
- **Inicialización de parámetros:** Se definieron los parámetros del modelo, donde m representa la pendiente de la recta (coeficiente del modelo) y b el intercepto (término independiente). Estos se inicializan en cero, y serán ajustados iterativamente por el gradiente descendente.
- **Ejecución del gradiente descendente**: En cada iteración, el algoritmo actualiza los valores de m y b minimizando el error (diferencia entre el valor predicho y el valor real). La tasa de aprendizaje (L) controla el tamaño del paso en cada iteración y está configurada en 0.01 para evitar convergencia lenta o divergencia.
- **Visualización de resultados**: Se utiliza **Matplotlib** para generar una gráfica que muestra la recta de regresión ajustada sobre los datos reales en el gráfico de dispersión.

2.3.2. **Resultados**

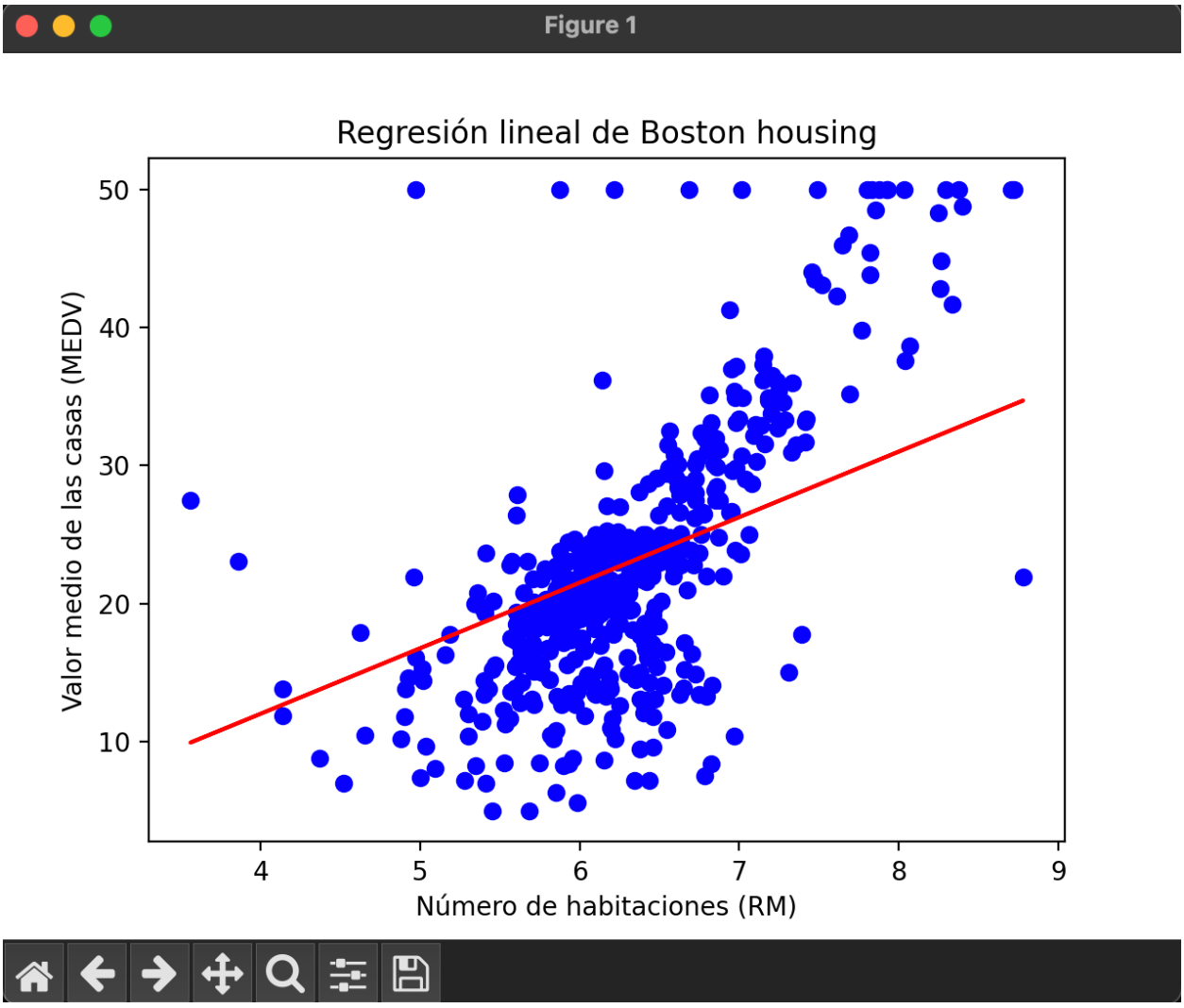
Tras ejecutar el algoritmo, la regresión lineal ajusta la pendiente y el intercepto de manera que minimiza el error cuadrático medio. En este caso, la pendiente m refleja cómo varía el precio de la vivienda con respecto al número de habitaciones. Al graficar los resultados, se puede observar

cómo la línea de regresión (en rojo) se ajusta a la tendencia general de los datos reales (puntos en azul).

En resumen, el algoritmo de gradiente descendente aplicado en este ejercicio es efectivo para resolver problemas de regresión lineal y proporciona una forma eficiente de ajustar los parámetros del modelo a los datos disponibles.

2.3.3. **Adecuación del algoritmo**

El gradiente descendente es un algoritmo adecuado para este tipo de problemas, ya que permite ajustar los parámetros de forma eficiente incluso cuando se trabaja con conjuntos de datos grandes. En el caso del dataset de Boston Housing, el gradiente descendente convergió rápidamente a un mínimo, lo que muestra que el modelo es capaz de aprender de los datos y realizar predicciones precisas. Sin embargo, el rendimiento del algoritmo depende de la elección de la tasa de aprendizaje; una tasa demasiado alta puede causar que el algoritmo no converja, mientras que una tasa demasiado baja puede hacer que la convergencia sea muy lenta.



2.4. **Informe algoritmo de optimización avanzado del dataset TSP**

2.4.1. **Descripción del código**

El código en Python implementa un algoritmo genético (AG) para resolver el problema del Viajante de Comercio (TSP).

- **Parámetros y configuración:** Define el número de ciudades (numCiudades), el tamaño de la población (tamañoPoblacion), el número de generaciones (numGeneraciones), y las probabilidades de cruce y mutación (probCruce, probMutacion).

- **Generación de matriz de distancias:** generarMatrizDist(n) genera una matriz simétrica que representa las distancias entre las ciudades. Esta matriz es crucial para evaluar la eficacia de cada ruta.
- **Definición de individuo y evaluación:** Utiliza DEAP (Distributed Evolutionary Algorithms in Python) para crear individuos (rutas) y evaluar sus fitness mediante la función evaluarTSP, que suma las distancias de una ruta completa.
- **Ejecución del algoritmo genético:** Se registra la creación de individuos, la población, las funciones de cruce, mutación, selección, y evaluación. Luego se ejecuta el algoritmo, seleccionando el mejor recorrido y graficando los resultados.

2.4.2. **Implementación del algoritmo genético:**

- **Mejorar la función de evaluación:** Se considera usar heurísticas para mejorar el cálculo de la distancia total, o aplicar técnicas de "local search" después de que el algoritmo genético haya encontrado una solución.
- **Ajustar parámetros:** Se experimenta con diferentes tamaños de población, tasas de cruce y mutación para ver cómo afectan la calidad de la solución y la convergencia.
- **Almacenamiento de resultados:** Se agrega un sistema para guardar las mejores soluciones en cada generación y observar cómo evoluciona la solución a lo largo del tiempo.

2.4.3. **Análisis critico**

El uso de algoritmos genéticos (AG) para resolver el problema del Viajante de Comercio (TSP) es una elección sólida, dado que el TSP es un problema NP-duro. Esto significa que las soluciones óptimas son difíciles de encontrar en un tiempo razonable, especialmente a medida que aumenta el número de ciudades.

Los algoritmos genéticos son una técnica de búsqueda basada en la evolución natural, lo que les permite explorar una gran cantidad de soluciones posibles de manera efectiva. A través de operaciones como cruce y mutación, el AG puede evitar caer en óptimos locales, a menudo encontrando soluciones que son cercanas a la óptima global.

La implementación presentada genera una matriz de distancias aleatoria, lo que puede limitar la reproducibilidad de los resultados. Sería útil utilizar un conjunto de datos de distancias más realistas, como un dataset del TSP, para evaluar la eficacia del algoritmo en situaciones más aplicables.

Además, el algoritmo podría beneficiarse de la incorporación de técnicas de ajuste de parámetros, como el uso de una tasa de mutación adaptativa. Esto permitiría al AG adaptarse a la complejidad del problema, potencialmente mejorando la calidad de las soluciones encontradas.

En general, el AG ha demostrado ser efectivo en la búsqueda de soluciones al TSP en este contexto, pero siempre hay margen para optimizar su rendimiento y aplicabilidad a conjuntos de datos más desafiantes. La visualización de la ruta final también proporciona una representación clara de los resultados, haciendo que el análisis y la interpretación sean más accesibles.

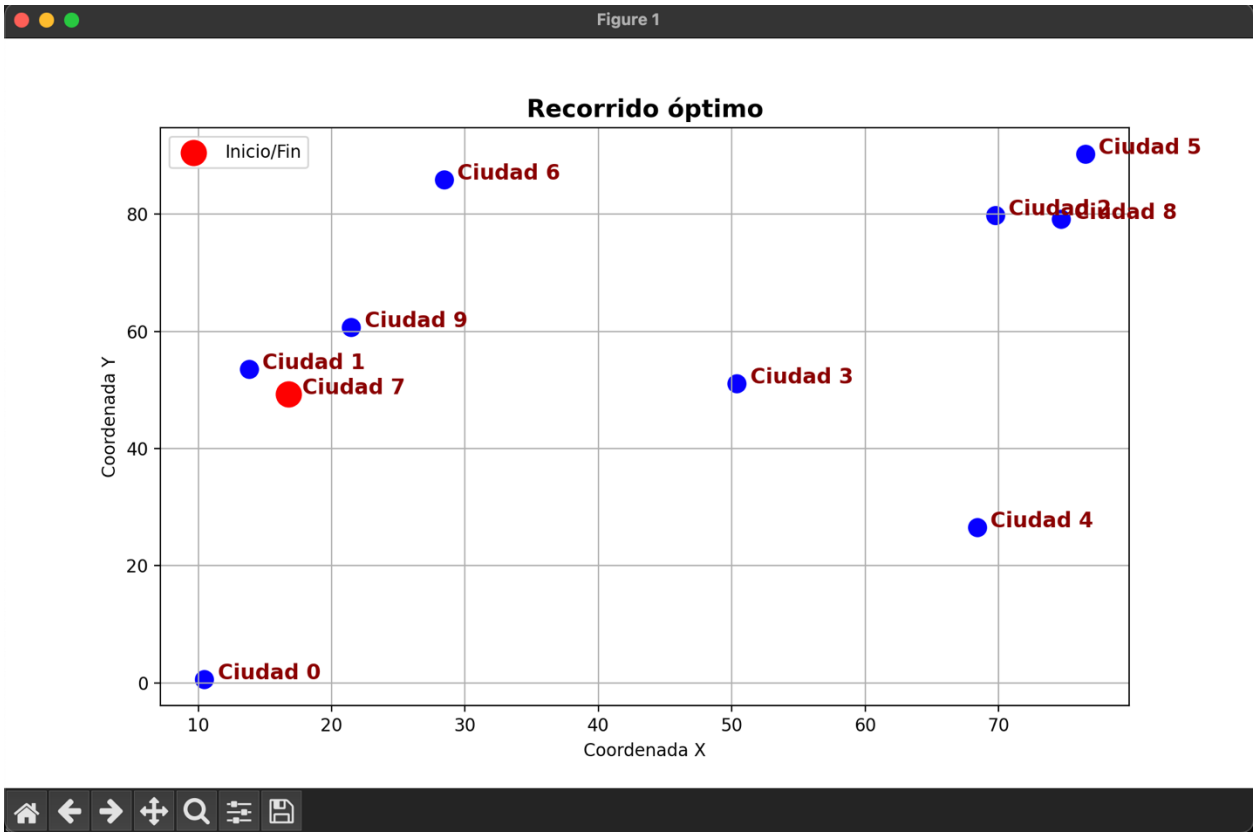


Figure 1
Recorrido óptimo
Inicio/Fin
Ciudad 6
Ciudad 5
Ciudad 2
Ciudad 8
Ciudad 9
Ciudad 1
Ciudad 7
Ciudad 3
Ciudad 4
Ciudad 0
Coordenada Y
Coordenada X
0
20
40
60
80
10
20
30
40
50
60
70

3. **Conclusiones**

En conclusión, la evolución de la ciencia de datos ha permitido a la industria financiera optimizar sus operaciones, mejorar la experiencia del cliente y gestionar riesgos de manera más efectiva. Con el continuo crecimiento de los datos y el avance tecnológico, es probable que la ciencia de datos siga desempeñando un papel crucial en el futuro de las finanzas, además ha demostrado ser un factor clave en la transformación y modernización de la industria financiera.

4. **Bibliografía**

Iuvity. (s. f.). Ciencia de datos: ¿conoces su aporte al sector financiero? *iuvity — TODO1 Services Inc. DBA iuvity*. https://www.iuvity.com/es/blog/ciencia-de-datos-conoces-su-aporte-al-sector-financiero#:~:text=La%20ciencia%20de%20datos%20tiene,comprensi%C3%B3n%20de%20los%20procesos%20econ%C3%B3micos.

Gabayet, C. (2024, 3 junio). *Ciencia de datos: ¿Cómo los datos están revolucionando el sector financiero?* DigDash. https://www.digdash.com/es/news-articles-es/business-intelligence-es/ciencia-de-datos-como-los-datos-estan-revolucionando-el-sector-financiero/

Joeportilla. (2023, 5 abril). *Analisis Exploratorio de Datos dataset Iris*. Kaggle. https://www.kaggle.com/code/joeportilla/analisis-exploratorio-de-datos-dataset-iris

Torres, A. (2023, 19 mayo). *Descenso de gradiente: ejemplo de algoritmo de aprendizaje automático*. freeCodeCamp.org. https://www.freecodecamp.org/espanol/news/descenso-de-gradiente-ejemplo-de-algoritmo-de-aprendizaje-automaticod/