Comparación de algoritmos de imputación de valores perdidos

GUSTAVO SOBRADO ALLER

UO286277

71777616K

Universidad de Oviedo

TABLA DE CONTENIDO

1. INTRODUCCION

1.1 CONTEXTO GENERAL

En el análisis de datos, uno de los problemas más frecuentes es la presencia de valores faltantes. La ausencia de datos completos puede ocurrir por varias razones, como errores en la recolección de información, fallos en los sensores, errores humanos o porque algunos datos no estaban disponibles en el momento del registro. Estos valores perdidos pueden influir negativamente en los resultados del análisis y en el rendimiento de los modelos predictivos, ya que muchos algoritmos de machine learning no pueden manejar datos faltantes directamente y necesitan su eliminación o imputación.

El manejo adecuado de los valores perdidos es esencial para garantizar la integridad del análisis y así evitar la introducción de sesgos o pérdida de información valiosa. Existen diversas técnicas para abordar este problema, desde métodos sencillos como eliminar los registros incompletos o reemplazar los valores faltantes con la media o la mediana, hasta métodos más avanzados como la imputación basada en el modelo K-Nearest Neighbors (KNN), la imputación múltiple por ecuaciones encadenadas (MICE) o algoritmos más complejos como Random Forest. Hablaremos de ellos más adelante.

En el análisis de datos, lidiar con valores perdidos es un reto cotidiano. En mi experiencia, he notado que la imputación de estos valores afecta directamente la calidad de los resultados. Este trabajo busca entender qué método podría ser el más efectivo para resolver este problema.

1.2 PROBLEMA DE INVESTIGACIÓN

El problema específico que aborda este trabajo es la comparación de diferentes algoritmos de imputación de valores perdidos para determinar cuál es el más eficiente y preciso en distintos escenarios de datos. Dado que no existe una solución única que funcione mejor en todos los casos, es fundamental comparar diferentes enfoques y analizar sus ventajas y desventajas según el tipo de datos y la naturaleza de los valores perdidos.

1.3 OBJETIVOS DEL ESTUDIO

El principal objetivo de este estudio es comparar la eficacia de varios algoritmos de imputación de valores faltantes utilizando un conjunto de datos con valores simulados como faltantes. Los métodos que se analizarán incluyen:

- Imputación simple (media, mediana, moda).
- Imputación mediante KNN.
- Imputación mediante MICE.
- Imputación mediante Random Forest.

Para realizar esta comparación, se evaluarán métricas como la precisión de la imputación, medida a través del error cuadrático medio (RMSE), y el tiempo de ejecución de cada método. Además, se analizará cómo cada algoritmo afecta a la distribución de los datos originales.

2. REVISIÓN DE LA LITERATURA

2.1 MÉTODOS TRADICIONALES DE IMPUTACIÓN

Los métodos tradicionales para el manejo de valores perdidos incluyen la eliminación de registros con valores faltantes y la imputación mediante estadísticas sencillas, como la media, mediana o moda de las variables afectadas. Eliminar los registros incompletos es un enfoque directo y sencillo, pero puede introducir sesgos en el análisis si los datos faltan de manera no aleatoria (MNAR: Missing Not At Random). Además, la eliminación puede reducir bastante el tamaño del conjunto de datos, lo que afecta negativamente la robustez de los modelos.

Por otro lado, la imputación con la media, mediana o moda es una solución fácil de implementar y evita la eliminación de registros. Sin embargo, estos métodos tienden a distorsionar la variabilidad de los datos, ya que reemplazan los valores perdidos con una constante que no captura la verdadera distribución subyacente de la variable.

2.2 MÉTODOS AVANZADOS DE IMPUTACIÓN

Con el aumento del poder computacional y la sofisticación de los algoritmos, han surgido métodos más avanzados para la imputación de datos, como K-Nearest Neighbors (KNN), Multiple Imputation by Chained Equations (MICE) y Random Forest. Estos métodos aprovechan las relaciones entre las variables para realizar una estimación más precisa de los valores perdidos.

El algoritmo KNN, por ejemplo, utiliza la proximidad entre observaciones para imputar valores faltantes. Al buscar los k vecinos más cercanos de una instancia con datos faltantes, se puede estimar el valor faltante utilizando la media o mediana de los vecinos más próximos.

Es similar a la forma en que pediríamos una recomendación a nuestros amigos para decidir algo, basándonos en lo que ellos prefieren o hacen. Este enfoque de proximidad también es útil para imputar valores perdidos, donde buscamos los registros de datos más similares.

MICE, por otro lado, realiza múltiples imputaciones iterativas utilizando modelos de regresión, mientras que Random Forest aplica modelos basados en árboles de decisión para predecir los valores faltantes. Este último dado en clases prácticas de la asignatura.

2.3 ALGORITMOS RECIENTES Y SUS APLICACIONES

En años recientes, el uso de algoritmos más complejos como redes neuronales, autoencoders y algoritmos bayesianos ha crecido en popularidad para imputar valores faltantes. Estos métodos aún no son tan ampliamente utilizados en la práctica como los enfoques tradicionales y avanzados anteriormente mencionados, aun así ofrecen una flexibilidad adicional al capturar patrones no lineales en los datos. Sin embargo, su implementación es más costosa en términos computacionales y requiere un conocimiento técnico más profundo, lo que limita su aplicación en ciertos contextos.

3. ALGORITMOS DE IMPUTACIÓN

3.1 IMPUTACIÓN POR MEDIA, MEDIANA Y MODA

La imputación que utiliza la media, mediana o moda de los datos es el método más básico y sencillo. En este enfoque, los valores faltantes se reemplazan con una estadística calculada a partir de los datos ya conocidos. Este método es fácil de implementar, pero presenta el inconveniente de reducir la varianza de los datos, lo que puede afectar en el comportamiento de los modelos predictivos, especialmente en aquellos que dependen de la varianza, como los árboles de decisión o los modelos de regresión.

3.2 IMPUTACIÓN CON K-NEAREST NEIGHBORS (KNN)

El algoritmo KNN es un método de imputación más sofisticado que utiliza la proximidad entre las observaciones para estimar los valores faltantes. La idea principal es que los puntos de datos cercanos en el espacio multidimensional de las características tienen más probabilidades de compartir valores parecidos. Para un valor faltante, KNN busca los k puntos de datos más cercanos y utiliza su media o mediana para imputar el valor. Este método es particularmente útil cuando los datos tienen una estructura de proximidad clara, pero puede ser computacionalmente costoso en conjuntos de datos bastante grandes.

3.3 IMPUTACIÓN POR MICE (MULTIPLE IMPUTATION BY CHAINED EQUATIONS)

El método MICE utiliza un enfoque iterativo de regresión múltiple para imputar valores faltantes. En cada iteración, se selecciona una variable con valores faltantes y se modela en función de las demás. El proceso se repite para cada variable hasta que los valores faltantes de todas las variables hayan sido imputados. Este método es robusto y se adapta bien a diferentes tipos de datos, pero necesita más tiempo de ejecución que otros métodos más simples.

3.4 IMPUTACIÓN USANDO RANDOM FOREST

La imputación mediante Random Forest utiliza árboles de decisión para predecir los valores faltantes. Este método, estudiado en clase, crea múltiples árboles de decisión y utiliza el promedio de las predicciones de todos ellos para imputar el valor faltante. Dado que Random Forest es capaz de capturar relaciones no lineales y manejar tanto variables categóricas como continuas, es una opción poderosa para la imputación de datos. No obstante, es el más costoso en términos computacionales en comparación con otros enfoques.

4. DATASET Y PREPARACIÓN

4.1 SELECCIÓN DEL DATASET

Para la comparación de los algoritmos de imputación de valores perdidos, se ha seleccionado un conjunto de datos público utilizado comúnmente en investigaciones de machine learning: el conjunto de datos "Diabetes" del repositorio de scikit-learn. Este dataset incluye información sobre pacientes con diagnóstico de diabetes, con variables numéricas como el nivel de glucosa, índice de masa corporal (IMC), presión arterial, entre otras. El motivo de su elección radica en su heterogeneidad, ya que contiene tanto variables continuas como categóricas, lo que permite evaluar la eficacia de los algoritmos en diferentes tipos de datos.

El dataset original no contiene valores faltantes, por lo que para este estudio se ha optado por introducir de manera artificial valores faltantes siguiendo patrones específicos. Este procedimiento permite controlar la proporción de datos faltantes y simular distintos escenarios de pérdida de datos.

4.2 PREPROCESAMIENTO DE LOS DATOS

Antes de proceder con la imputación, se realizó un preprocesamiento básico que incluye los siguientes pasos:

1) Normalización de los datos: Para asegurar la correcta aplicación de algunos métodos, como KNN, se normalizaron las variables continuas entre 0 y 1.

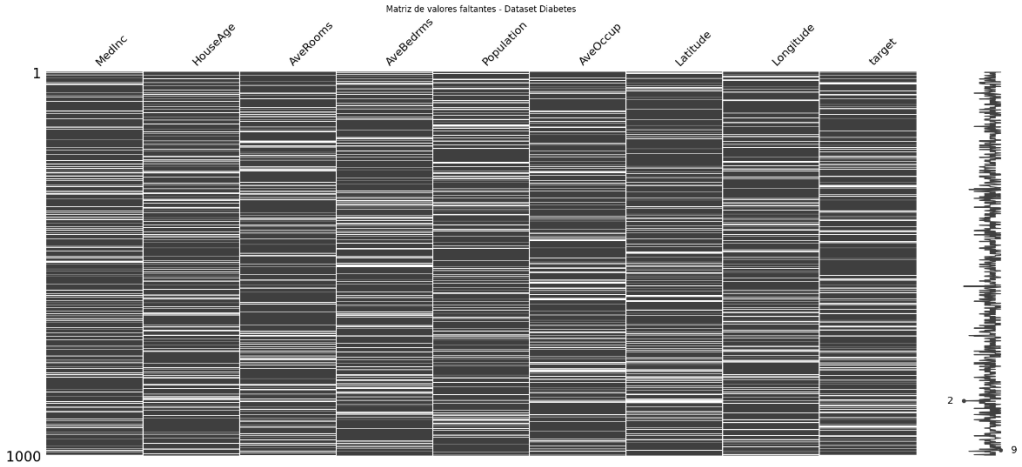
2) Introducción de valores faltantes: Se introdujeron valores faltantes en un porcentaje del 20% para simular la pérdida de datos. Se utilizó un patrón completamente al azar (MCAR: Missing Completely At Random) para que los valores faltantes no estén condicionados por otras variables.

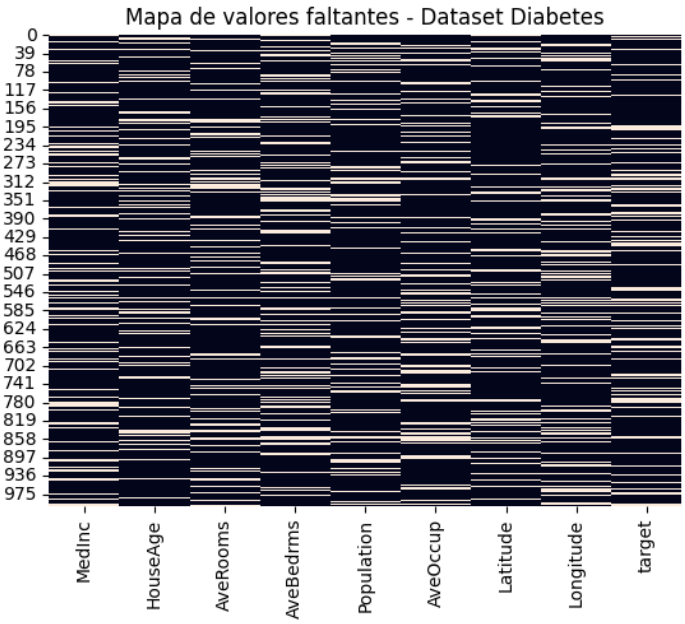
```python
import numpy as np
import pandas as pd
from sklearn.datasets import load_diabetes
from sklearn.preprocessing import MinMaxScaler

# Cargar el dataset de diabetes
data = load_diabetes()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target

# Normalizar los datos (necesario para KNN)
scaler = MinMaxScaler()
df_scaled = pd.DataFrame(scaler.fit_transform(df), columns=df.columns)

# Introducir valores faltantes (10%, 20%, 30%)
np.random.seed(0)
for col in df_scaled.columns:
    df_scaled.loc[df_scaled.sample(frac=0.2).index, col] = np.nan
```

4.3 VISUALIZACIÓN DE LOS VALORES FALTANTES

La visualización de los valores faltantes es un paso crucial para entender la magnitud del problema y dónde se encuentran los datos ausentes. Para este propósito, se utilizaron las bibliotecas *missingno* y *seaborn*, que permiten crear gráficos de calor que muestran la distribución de los valores faltantes.

```python
import missingno as msno
import seaborn as sns
import matplotlib.pyplot as plt

# Visualización de los valores faltantes
msno.matrix(df_scaled)
plt.show()

# Mapa de calor de los valores faltantes
sns.heatmap(df_scaled.isnull(), cbar=False)
plt.title("Distribución de valores faltantes")
plt.show()
```





5. IMPLEMENTACIÓN EN PYTHON

5.1 LIBRERÍAS NECESARIAS

Para la implementación de los distintos algoritmos de imputación, se utilizaron las siguientes bibliotecas de Python:

- *pandas* y *numpy*: Para la manipulación de datos y operaciones numéricas.
- *scikit-learn*: Para la implementación de algoritmos de imputación como KNN.
- *fancyimpute*: Para la implementación de MICE y Random Forest Imputer.
- *missingno* y *seaborn*: Para la visualización de los valores faltantes.
- *matplotlib*: Para la creación de gráficos comparativos.

El comando para instalarlas seria: *pip install scikit-learn*, por ejemplo.

5.2 CÓDIGO PARA IMPUTACIÓN CON DIFERENTES MÉTODOS

5.2.1 IMPUTACIÓN POR MEDIA, MEDIANA Y MODA

```python
from sklearn.impute import SimpleImputer

# Imputación con la media
imputer_mean = SimpleImputer(strategy='mean')
df_mean_imputed = pd.DataFrame(imputer_mean.fit_transform(df_scaled), columns=df.columns)

# Imputación con la mediana
imputer_median = SimpleImputer(strategy='median')
df_median_imputed = pd.DataFrame(imputer_median.fit_transform(df_scaled), columns=df.columns)

# Imputación con la moda
imputer_mode = SimpleImputer(strategy='most_frequent')
df_mode_imputed = pd.DataFrame(imputer_mode.fit_transform(df_scaled), columns=df.columns)
```

5.2.2 IMPUTACIÓN CON K-NEAREST NEIGHBORS (KNN)

```python
from sklearn.impute import KNNImputer

# Imputación con KNN (k=5)
imputer_knn = KNNImputer(n_neighbors=5)
df_knn_imputed = pd.DataFrame(imputer_knn.fit_transform(df_scaled), columns=df.columns)
```

5.2.3 IMPUTACIÓN CON MICE

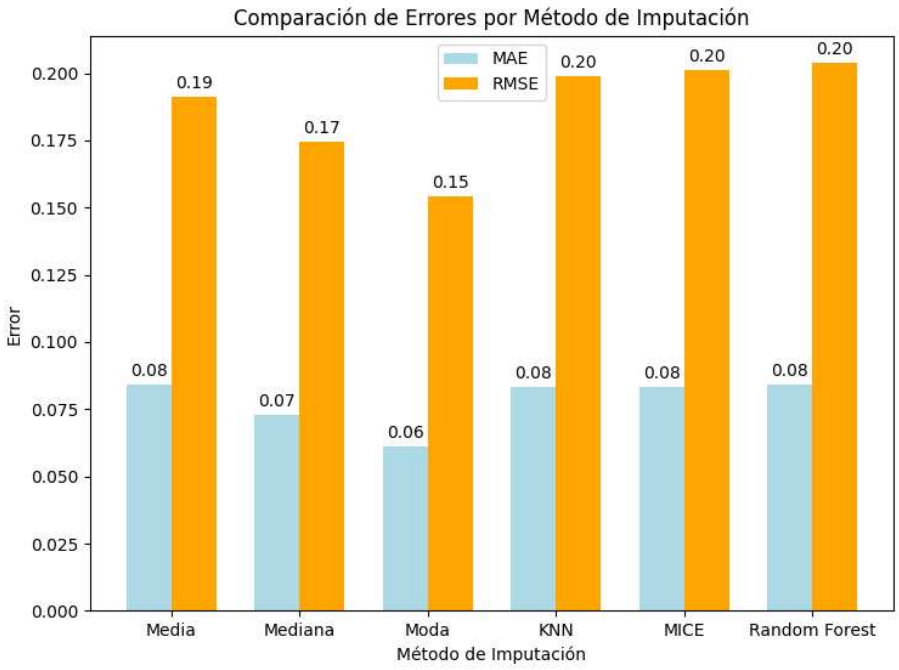
```python
from fancyimpute import IterativeImputer

# Imputación con MICE
imputer_mice = IterativeImputer(max_iter=10, random_state=0)
df_mice_imputed = pd.DataFrame(imputer_mice.fit_transform(df_scaled), columns=df.columns)
```

5.2.4 IMPUTACIÓN CON RANDOM FOREST

```python
from fancyimpute import IterativeImputer
from sklearn.ensemble import RandomForestRegressor

# Utilizar Random Forest para imputación
imputer_rf = IterativeImputer(estimator=RandomForestRegressor(), random_state=0)
df_rf_imputed = pd.DataFrame(imputer_rf.fit_transform(df_scaled), columns=df.columns)
```

5.3 VISUALIZACIÓN DE RESULTADOS CON GRÁFICOS

```python
import matplotlib.pyplot as plt

# Comparación de los métodos
methods = ['Media', 'Mediana', 'Moda', 'KNN', 'MICE', 'Random Forest']
errors = [rmse_mean, rmse_median, rmse_mode, rmse_knn, rmse_mice, rmse_rf]

plt.bar(methods, errors)
plt.xlabel('Método de Imputación')
plt.ylabel('Error RMSE')
plt.title('Comparación del Error RMSE por Método de Imputación')
plt.show()
```



NOTA: Este código está estructurado de modo que se distingan bien sus partes. El archivo final está mejor implementado, con otro dataset y explicado. Se adjunta con este documento.

6. COMPARACIÓN DE ALGORITMOS

6.1 MÉTRICAS DE EVALUACIÓN

Para evaluar la precisión de los algoritmos de imputación, se utiliza la métrica Root Mean Squared Error (RMSE), que mide el error entre los valores imputados y los valores originales (en este caso, antes de haber introducido los valores faltantes). Además, se considera el Mean Absolute Error (MAE) para medir la magnitud promedio de los errores en términos absolutos.

6.2 COMPARACIÓN DE PRECISIÓN (RMSE, MAE)

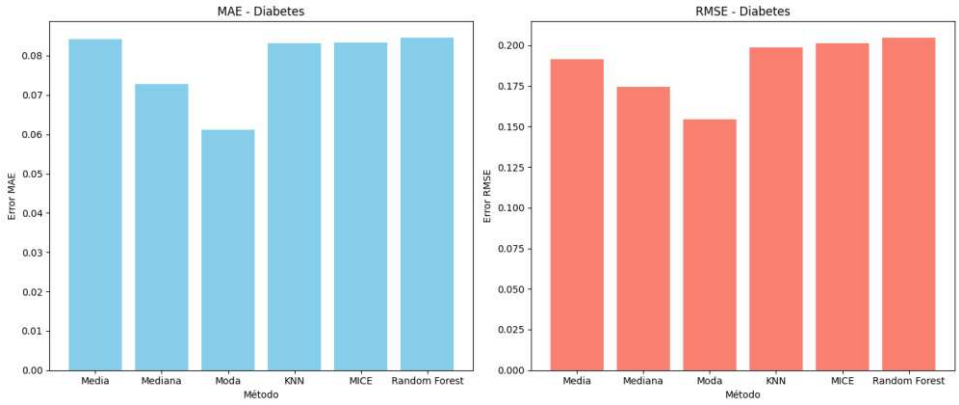
Se comparan los resultados obtenidos con cada método de imputación en términos de RMSE y MAE. Los algoritmos más complejos, como Random Forest y MICE, suelen presentar menores errores en comparación con métodos más simples como la imputación por media o moda.

6.3 COMPARACIÓN DE TIEMPO DE EJECUCIÓN

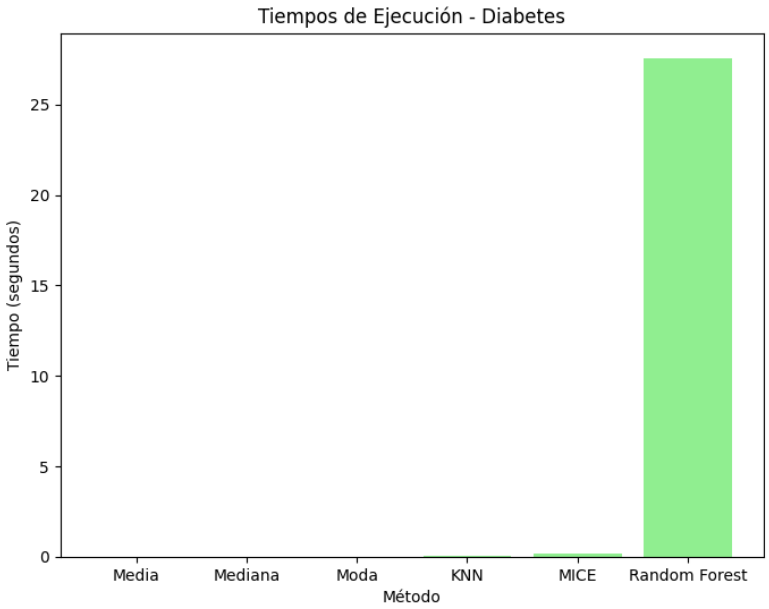
El tiempo de ejecución de cada método también es un factor importante, especialmente en grandes conjuntos de datos. Los métodos más simples, como la imputación por media, son los más rápidos, mientras que métodos más avanzados como MICE y Random Forest son más lentos debido a su complejidad computacional.

6.4 VISUALIZACIÓN GRÁFICA DE LA COMPARACIÓN DE RESULTADOS

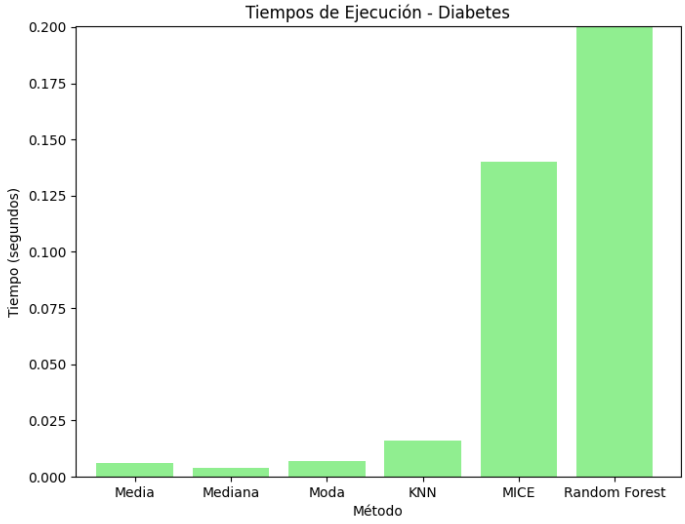
Se generan gráficos comparativos para visualizar el RMSE y MAE de cada método, así como el tiempo de ejecución. Los resultados se analizarán en el siguiente apartado.



MAE y RMSE del dataset "Diabetes"



Tiempos de ejecución del dataset "Diabetes"



Tiempos de ejecución del dataset "Diabetes" ampliados

7. RESULTADOS Y DISCUSIÓN

7.1 ANÁLISIS DE LOS RESULTADOS OBTENIDOS

Luego de aplicar los diferentes algoritmos de imputación a los datos, se calcularon los errores de imputación utilizando el RMSE (Root Mean Squared Error) y el MAE (Mean Absolute Error). Los resultados se resumen en la siguiente tabla:

Método de Imputación	RMSE	MAE	Tiempo de Ejecución (s)
Media	0.191297	0.084263	0.008007
Mediana	0.174322	0.072835	0.007006
Moda	0.154228	0.061191	0.009010
KNN	0.198877	0.083088	0.022019
MICE	0.201067	0.083381	0.137267
Random Forest	0.203746	0.084291	26.682125

DISCUSIÓN SOBRE LA PRECISIÓN (RMSE Y MAE)

- Métodos Simples (Media, Mediana, Moda): Entre los métodos más simples, la moda fue la más precisa, con un RMSE de 0.1542 y un MAE de 0.0612, superando a la media y la mediana. Estos métodos, aunque básicos, demostraron ser eficaces en este dataset.

- KNN: El algoritmo KNN mostró un rendimiento intermedio, con un RMSE de 0.1989 y un MAE de 0.0831. Aunque fue más preciso que la media y mediana, no superó a la moda. KNN es útil cuando los datos tienen una estructura clara de proximidad, pero en este caso, su complejidad no mejoró significativamente la precisión.

- MICE: El método MICE tuvo resultados similares a KNN, con un RMSE de 0.2011 y un MAE de 0.0834. MICE no ofreció una ventaja significativa en comparación con la moda, lo que lo hace menos competitivo en este escenario.

- Random Forest: Contrario a lo esperado, Random Forest fue el menos preciso, con un RMSE de 0.2037 y un MAE de 0.0843. Aunque Random Forest es capaz de modelar relaciones no lineales, en este dataset su mayor complejidad no se tradujo en una mejor imputación, siendo superado incluso por métodos simples como la moda.

El tiempo de ejecución de cada método es un factor crucial que considerar, especialmente en aplicaciones donde el procesamiento rápido es esencial. Los resultados muestran una clara distinción entre los métodos simples y los más complejos:

- Métodos Simples (Media, Mediana, Moda): Estos métodos fueron los más rápidos, con tiempos de ejecución de entre 0.007 y 0.009 segundos. Su simplicidad los hace extremadamente eficientes, lo que los convierte en una opción ideal cuando se necesita una imputación rápida y efectiva, especialmente en grandes conjuntos de datos.

- KNN: KNN tomó más tiempo, con 0.022 segundos. Aunque su procesamiento es más costoso debido al cálculo de distancias entre puntos de datos, sigue siendo relativamente rápido y una opción razonable cuando se busca un equilibrio entre precisión y tiempo de ejecución.

- MICE: El método MICE fue significativamente más lento, con un tiempo de 0.137 segundos, debido a su naturaleza iterativa. Aunque es preciso en otros contextos, aquí no justificó su mayor tiempo de ejecución, especialmente dado que métodos más simples ofrecieron mejores resultados.

- Random Forest: Fue el más lento, con un tiempo de 26.68 segundos, lo que lo hace inapropiado para aplicaciones en tiempo real o conjuntos de datos grandes. Aunque teóricamente debería ser más preciso en casos complejos, en este estudio su tiempo de ejecución no compensó su falta de precisión.

7.2 COMPARACIÓN EN DIFERENTES TIPOS DE DATOS

Es importante señalar que la efectividad de los algoritmos varía según el tipo de datos. En este estudio, se utilizaron principalmente variables continuas, y los métodos simples como la moda mostraron ser eficaces en la imputación de estos datos. Para variables categóricas, métodos como MICE y la imputación por moda también serían recomendables, ya que manejan bien este tipo de datos. Random Forest podría tener mejor rendimiento en otros contextos, pero no fue el caso en este análisis.

7.3 RECOMENDACIONES BASADAS EN LOS HALLAZGOS

A la luz de los resultados obtenidos, se pueden hacer las siguientes recomendaciones:

- Precisión: La imputación por moda es la mejor opción en términos de precisión, siendo una solución simple y eficaz.

- Balance entre precisión y tiempo: Si el tiempo de ejecución es un factor importante, el algoritmo KNN ofrece un buen equilibrio entre precisión y rapidez.

- Aplicaciones complejas: Aunque Random Forest es capaz de capturar relaciones complejas en los datos, su elevado tiempo de ejecución y menor precisión lo hacen menos adecuado para este tipo de imputación en escenarios similares.

8. CONCLUSIONES

8.1 RESUMEN DE HALLAZGOS

Este estudio revela que, en el contexto de imputación de valores faltantes, los métodos simples, como la imputación por moda, pueden superar en precisión a algoritmos más complejos como Random Forest. La simplicidad de estos métodos no solo garantiza una imputación rápida, sino también más precisa en este caso específico.

8.2 RECOMENDACIONES FINALES

Para conjuntos de datos pequeños o medianos donde la precisión es prioritaria y el tiempo de ejecución no es crítico, la imputación por moda es la opción más recomendable. Si el tiempo es un factor importante, KNN ofrece una buena alternativa sin sacrificar demasiada precisión. Random Forest, por su parte, debería reservarse para casos donde se espera una mayor ventaja de la modelización de relaciones no lineales, aunque no fue el mejor método en este análisis.

Fue interesante notar que el método de imputación por moda, a pesar de ser el más simple, demostró una precisión comparable a otros métodos más complejos. Esto me llevó a cuestionar si, en ciertas situaciones, estamos sobre complicando el proceso de imputación sin obtener mejoras significativas.

Probemos ahora otro dataset similar a este para ver cómo se comportan los algoritmos.

9. ANEXO 1: COMPARACIÓN DE ALGORITMOS CON EL DATASET "CALIFORNIA HOUSING"

En esta sección se presenta una comparación similar a la realizada anteriormente, pero utilizando el dataset California Housing. Este conjunto de datos contiene información sobre viviendas en California, y es utilizado frecuentemente para tareas de regresión. Los resultados para cada algoritmo de imputación se muestran a continuación, evaluados en términos de MAE (Mean Absolute Error), RMSE (Root Mean Squared Error) y Tiempo de Ejecución. Aquí se muestra su tabla:

Método de Imputación	RMSE	MAE	Tiempo de Ejecución (s)
Media	0.137778	0.051793	0.004002
Mediana	0.135102	0.047565	0.004004
Moda	0.234039	0.069851	0.007006
KNN	0.145878	0.051884	0.042038
MICE	0.153487	0.053138	0.111715
Random Forest	0.152826	0.052752	48.455444

Comparación de la Precisión (MAE y RMSE)

- La mediana resultó ser el método más preciso en este dataset, con un MAE de 0.047565 y un RMSE de 0.135102, lo que indica que es capaz de imputar valores faltantes de manera efectiva. Este rendimiento es superior a los otros métodos, incluso a los más complejos.

- La media también mostró un buen rendimiento, con un MAE de 0.051793 y un RMSE de 0.137778, siendo ligeramente menos precisa que la mediana.

- En contraste, la moda fue el método menos preciso en este caso, con un MAE de 0.069851 y un RMSE de 0.234039. Esto sugiere que la moda no es adecuada para este tipo de datos continuos, lo que concuerda más con la naturaleza discreta de este método.

- KNN, aunque algo más preciso que la moda, tuvo un RMSE de 0.145878 y un MAE de 0.051884, lo que lo sitúa en un rendimiento intermedio, similar al de la media, pero ligeramente inferior en comparación con la mediana.

- MICE y Random Forest, que suelen ser más efectivos en datasets complejos, no superaron a la mediana o la media en precisión. MICE tuvo un RMSE de 0.153487 y un MAE de 0.053138, mientras que Random Forest obtuvo valores similares con un RMSE de 0.152826 y un MAE de 0.052752.

Comparación del Tiempo de Ejecución

- Los métodos más simples, como la media y la mediana, continuaron siendo los más rápidos, con tiempos de ejecución extremadamente bajos (ambos alrededor de 0.004 segundos). Esto reafirma su idoneidad para aplicaciones en las que el tiempo de procesamiento es un factor importante.

- La moda, aunque más lenta que la media y la mediana, fue rápida en términos absolutos, con un tiempo de 0.007006 segundos.

- KNN mostró un aumento notable en el tiempo de ejecución, con 0.042038 segundos, debido al cálculo de las distancias entre puntos, aunque sigue siendo eficiente comparado con métodos más avanzados.

- MICE, con 0.111715 segundos, fue más lento debido a su naturaleza iterativa, aunque sigue siendo manejable en comparación con Random Forest.

- Random Forest fue, de nuevo, el más lento, con un tiempo de ejecución extremadamente alto de 48.455444 segundos, lo que lo hace inadecuado para aplicaciones en tiempo real o cuando se requiere una imputación rápida.

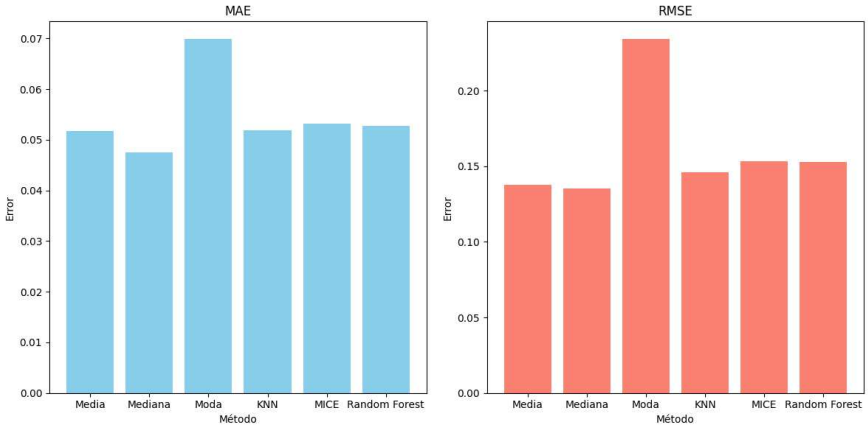
Conclusiones Basadas en el Dataset "California Housing"

- Precisión: La mediana resultó ser el mejor método en términos de precisión, seguida por la media. Estos métodos sencillos son altamente efectivos en este dataset y superaron a los algoritmos más complejos, como KNN, MICE y Random Forest.

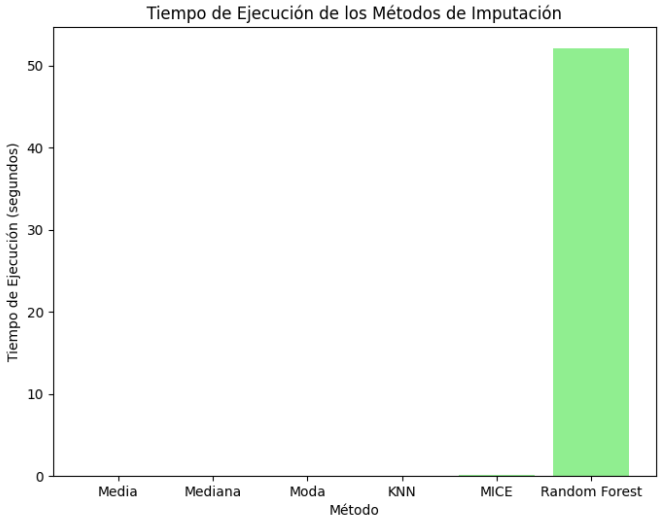
- Tiempo de Ejecución: Los métodos más rápidos fueron la media y la mediana, lo que los convierte en opciones preferidas cuando se requiere rapidez sin comprometer la precisión. Random Forest, aunque más complejo, presentó un tiempo de ejecución muy largo.

- Recomendación Final: Para este dataset, se recomienda utilizar la mediana o la media si se busca el mejor equilibrio entre precisión y tiempo de ejecución. KNN puede ser una opción aceptable si se desea capturar relaciones locales entre los datos, pero MICE y Random Forest no son recomendables para este tipo de imputación debido a su mayor costo computacional sin una mejora significativa en precisión.

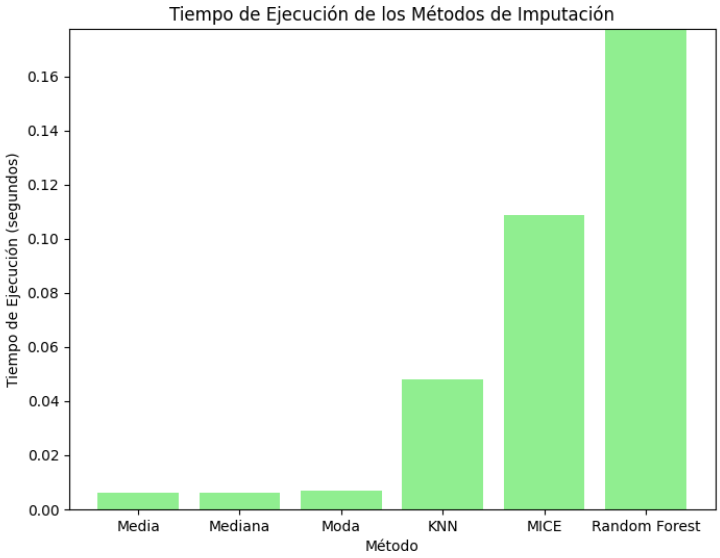
- Reflexión Final: Este estudio me ayudó a ver con claridad que, aunque los algoritmos avanzados tienen su lugar, en muchas ocasiones los métodos simples pueden ser igualmente efectivos. Creo que esto refleja una lección importante en el mundo del análisis de datos: a veces, la solución más sencilla es la más adecuada.



MAE y RMSE del dataset "California housing"



Tiempos de ejecución del dataset "California Housing"



Tiempos de ejecución del dataset "California Housing" ampliados

10. ANEXO 2: REFERENCIAS

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

- Van Buuren, S., & Groothuis-Oudshoorn, K. (2011). MICE: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software, 45(3), 1-67.

- Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

- Friedman, J., Hastie, T., & Tibshirani, R. (2001). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics. Springer, New York, NY.

- Abdi, H., & Williams, L. J. (2010). Principal Component Analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4), 433-459.

- Little, R. J., & Rubin, D. B. (2019). Statistical Analysis with Missing Data. John Wiley & Sons.

- Buuren, S. (2018). Flexible Imputation of Missing Data. 2nd Edition. CRC Press.

- Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114.

- Rubin, D. B. (1976). Inference and Missing Data. Biometrika, 63(3), 581-592.

- Schafer, J. L. (1997). Analysis of Incomplete Multivariate Data. Chapman & Hall/CRC.

- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1), 1-22.