Trabajo de fin de master: Votaciones al Parlamento Europeo junio de 2024

Fuente: CIS

Walter Daniel Aranda

2025

Contents

Memoria Académica: Análisis y Evaluación del Hito 1

Introducción

El presente documento constituye una memoria académica que analiza y evalúa el trabajo realizado en el marco del Hito 1 de la práctica final del Máster en Inteligencia Artificial. El objetivo principal del proyecto es el desarrollo de un modelo de inteligencia artificial capaz de predecir la intención de voto en las elecciones al Parlamento Europeo de 2024, a partir de un conjunto de datos sociodemográficos y de opinión extraídos del estudio Nž 3452 del Centro de Investigaciones Sociológicas (CIS).

Esta memoria se centra en la evaluación de las dos tareas fundamentales del Hito 1: la selección y preparación del modelo y el preprocesamiento de datos. Se analizará la idoneidad de las decisiones tomadas, la rigurosidad de los procedimientos aplicados y el cumplimiento de las consignas académicas establecidas, todo ello en un tono ftécnico y análitico.

1. Preprocesamiento y Preparación de los Datos

La fase de preprocesamiento de datos es un pilar fundamental en cualquier proyecto de aprendizaje automático, ya que de la calidad y adecuación de los datos de entrada depende en gran medida el rendimiento del modelo. En este proyecto, se ha llevado a cabo un exhaustivo proceso de transformación y limpieza del dataset original, con el objetivo de prepararlo para los algoritmos de clasificación seleccionados. A continuación, se detallan las principales técnicas de preprocesamiento aplicadas, justificando su relevancia en el contexto del problema.

1.1. Codificación de Variables Categóricas

Una parte significativa de las variables del dataset original son de naturaleza categórica. Para que los modelos de machine learning puedan procesar esta información, ha sido necesario convertirlas a un formato numérico. Se han empleado dos estrategias principales:

- Codificación Ordinal: Para variables con una relación de orden intrínseca, como `ingreso_hogar`, `Habitantes_municipio` o `nivel_educacion_encoded`, se ha utilizado una codificación ordinal. Esta técnica asigna un número entero a cada categoría, preservando la jerarquía entre ellas. Por ejemplo, en la variable `ingreso_hogar`, se asigna un valor de `1` a Menos de 1.100 y un valor de `6` a Más de 5.000 , reflejando el orden creciente de los niveles de ingreso.
- Codificación One-Hot y Reducción de Dimensionalidad con PCA: Para variables categóricas nominales sin un orden inherente, como `provincia`, la codificación ordinal no es apropiada. En su lugar, se ha optado por una estrategia más sofisticada. Primero, se aplicaría una codificación one-hot para crear variables dummy para cada provincia. Sin embargo, para evitar la maldición de la dimensionalidad que esto generaría, se ha aplicado un Análisis de Componentes Principales (PCA) sobre estas variables dummy. De esta forma, se ha reducido la dimensionalidad a 10 componentes principales

(provincia_pca_0 a provincia_pca_9), que captan la mayor parte de la varianza de la variable original en un espacio de características mucho más reducido.

1.2. Transformación de Variables Numéricas

Las variables numéricas también han sido objeto de transformaciones para optimizar su uso en los modelos:

- Normalización y Estandarización: La variable probabilidad_voto_generales se ha normalizado a una escala de 0 a 1, mientras que Renta_Per_Capita_2023_miles_euros ha sido estandarizada (media 0 y desviación estándar 1) utilizando StandardScaler. Estas transformaciones son determinantes para algoritmos sensibles a la escala de las características, como la Regresión Logística o el Perceptrón Multicapa, asegurando que todas las variables contribuyan de manera equitativa al aprendizaje del modelo.

1.3. Tratamiento de la Variable Objetivo

La variable objetivo, Intención_voto_encoded, ha sido cuidadosamente procesada. Se ha realizado una codificación ordinal, agrupando las respuestas de No sabe, No contesta, En Blanco y Voto nulo en una única categoría (0), y asignando un identificador numérico único a cada partido político. Además, se ha creado una agrupación por bloques ideológicos (mapa_bloques_ideologicos), lo que podría permitir en futuras fases del proyecto un análisis a un nivel más agregado.

En resumen, el preprocesamiento de datos ha sido estudiado desde un enfoque meticuloso y bien justificado. Se han seleccionado técnicas apropiadas para cada tipo de variable, abordando problemas comunes como la alta dimensionalidad y la escala de las características. Esta preparación sienta una base sólida para la siguiente fase del proyecto: la selección y el entrenamiento de los modelos de clasificación.

2. Selección y Justificación de Modelos

La elección de los algoritmos de aprendizaje automático es una decisión crítica que define la capacidad del proyecto para resolver el problema de clasificación planteado. En el Hito 1, se ha propuesto una selección de cuatro modelos que abarcan diferentes paradigmas del aprendizaje automático, permitiendo un análisis comparativo robusto y una evaluación exhaustiva del rendimiento. La justificación académica para la selección de estos modelos se basa en una estrategia metodológica que busca un equilibrio entre interpretabilidad, rendimiento y complejidad.

2.1. Modelos Seleccionados

El notebook Hito_1B.ipynb presenta una justificación detallada para la elección de los siguientes modelos:

1. Regresión Logística (Logistic Regression): Este modelo se ha seleccionado como una línea base (baseline) fundamental. Su simplicidad y alta interpretabilidad permiten establecer un punto de referencia de rendimiento. La Regresión Logística es un modelo lineal que, a pesar de su sencillez, es muy eficaz para problemas de clasificación binaria y multiclase. Su inclusión es funcional desde el punto de vista de la metodología ya que evalúa si los modelos más complejos aportarán o no una mejora significativa que justifique su mayor coste computacional y menor interpretabilidad.
2. Random Forest Classifier (Bosque Aleatorio): Como representante de los métodos de ensamble basados en bagging, el Random Forest se ha elegido por su robustez y su capacidad para manejar relaciones no lineales. Este modelo construye múltiples árboles de decisión durante el entrenamiento y combina sus predicciones para obtener un resultado más preciso y estable. Es menos propenso al sobreajuste que un único árbol de decisión y proporciona una medida de la importancia de las características, lo que añade una capa de interpretabilidad al análisis.
3. Gradient Boosting: Este es otro método de ensamble, pero a diferencia de Random Forest, se basa en la técnica de boosting. Los modelos de Gradient Boosting construyen los árboles de forma secuencial,

donde cada nuevo árbol corrige los errores del anterior. Se ha seleccionado por su alto rendimiento predictivo, siendo a menudo uno de los algoritmos más competitivos en problemas de clasificación con datos tabulares. Su inclusión permite explorar el límite superior del rendimiento alcanzable con los datos disponibles.

4. Perceptrón Multicapa (MLP) / Red Neuronal: La inclusión de un MLP introduce el paradigma del aprendizaje profundo (Deep Learning) en el proyecto. Aunque más complejo de entrenar y optimizar, un MLP tiene la capacidad de aprender patrones muy complejos y no lineales en los datos. Su selección se justifica por la necesidad de explorar si una arquitectura de red neuronal podría captar relaciones sutiles que los modelos más tradicionales dejaríanpasar por alto, ofreciendo potencialmente un rendimiento superior.

2.2. Justificación de la Selección

La selección de estos cuatro modelos es académicamente sólida y metodológicamente coherente. Se ha seguido un enfoque en el que se incrementa la complejidad: comenzando con un modelo lineal simple (Regresión Logística), pasando por modelos de ensamble potentes y robustos (Random Forest y Gradient Boosting), y culminando con un modelo de aprendizaje profundo (MLP). Esta estrategia no solo permite la comparación rigurosa del rendimiento, sino que también facilita la comprensión más profunda de la naturaleza del problema y de las características del dataset.

La justificación proporcionada en el notebook `Hito_1B.ipynb` busca ser exhaustiva y demostrar una comprensión clara de las fortalezas y debilidades de cada modelo, así como de su contribución específica al proyecto. Esta selección estratégica sienta las bases para las fases posteriores de entrenamiento, optimización y evaluación de modelos.

3. Consignas del Hito 1

Una vez analizados el preprocesamiento de los datos y la selección de modelos, es fundamental evaluar en qué medida el trabajo realizado se ajusta a las consignas específicas del Hito 1, tal y como se describen en el documento de la práctica. El Hito 1 se divide en dos componentes principales: la selección del modelo (1 punto) y el preprocesamiento de datos.

3.1. Selección del Modelo

La consigna para este apartado es: Elegir los modelos de machine learning o deep learning que se entrenarán. Justificar la elección en base al tipo de problema (clasificación, regresión, etc.), las características del dataset y el rendimiento esperado de cada algoritmo.

Evaluación:

Se justifica la selección de los cuatro modelos (Regresión Logística, Random Forest, Gradient Boosting y MLP) de manera académicamente rigurosa, detallada y bien fundamentada. Para cada modelo, se explica claramente que:

- El tipo de problema: Se identifica correctamente como un problema de clasificación.
- Las características del dataset: Se consideran implícitamente al justificar la necesidad de modelos que puedan manejar la no linealidad y la complejidad de las interacciones entre variables sociodemográficas.
- El rendimiento esperado: Se establece una expectativa de rendimiento para cada modelo, desde la línea base de la Regresión Logística hasta el potencial de alto rendimiento de Gradient Boosting y MLP.
- La contribución metodológica: Se va más allá de la simple elección de modelos y se explica el rol estratégico de cada uno dentro del proyecto (línea base, robustez, rendimiento, exploración de paradigmas).

A través de la justificación se intenta mostrar una comprensión profunda de la teoría subyacente a cada algoritmo y de su aplicación práctica.

3.2. Preprocesamiento de Datos

La consigna para este apartado es: Realizar el preprocesamiento adecuado para cada modelo, asegurando que los datos están en un formato compatible y optimizado para los algoritmos seleccionados (normalización, estandarización, codificación, etc.).

Evaluación:

Se enfrenta preprocesamiento de datos, documentado en `DOCUMENTACION.md` y ejecutado en el notebook `Hito_1A.ipynb`, desde un punto de vista exhaustivo. Se han aplicado las técnicas que se consideran correctas para cada tipo de variable, teniendo en cuenta la preparación de datos para el modelado.

- Codificación: Se ha utilizado codificación ordinal y una combinación de one-hot con PCA para manejar las variables categóricas.
- Normalización y Estandarización: Se han aplicado transformaciones de escala a las variables numéricas, lo cual es esencial para el correcto funcionamiento de varios de los modelos seleccionados.
- Limpieza: Se han tratado los valores nulos y se han limpiado las variables para asegurar la calidad de los datos.

El resultado es un conjunto de datos (`data_ML.csv`) estructurado y optimizado para ser utilizado por los modelos de machine learning. El trabajo realizado en esta área apunta con diligencia a la necesidad de asegurar una base de datos de alta calidad.

4. Conclusiones

El preprocesamiento de los datos ha sido meticuloso, aplicando técnicas adecuadas para transformar un dataset complejo en un formato optimizado para el modelado. La selección de modelos está justificada, proponiendo un abanico de algoritmos que permitirá una evaluación comparativa exhaustiva en las fases posteriores del proyecto.

Memoria Académica: Hitos 2 y 3

Introducción

Esta memoria describe de manera objetiva y académica el trabajo realizado en los Hitos 2 y 3 del proyecto, centrados en el entrenamiento, evaluación y optimización de los modelos de clasificación para la predicción de la intención de voto. El documento se estructura en dos secciones principales, correspondiendo a cada uno de los hitos y presenta los resultados obtenidos.

Hito 2: Entrenamiento Inicial del Modelo

En esta fase, se realizó el entrenamiento inicial de los cuatro modelos seleccionados (Regresión Logística, Random Forest, Gradient Boosting y Perceptrón Multicapa) utilizando sus parámetros por defecto. El objetivo fue establecer una línea de base de rendimiento para cada uno y realizar una primera comparación.

2.1. Metodología de Entrenamiento

Para todos los modelos, el proceso de entrenamiento siguió una metodología común. En primer lugar, se identificó un desbalance significativo en las clases de la variable objetivo en el conjunto de entrenamiento. Para mitigar el posible sesgo del modelo hacia las clases mayoritarias, se aplicó la técnica de sobremuestreo Over-sampling Simple) únicamente al conjunto de entrenamiento. El conjunto de prueba se mantuvo en su estado original para garantizar una evaluación realista del rendimiento del modelo en datos no vistos.

El entrenamiento se ejecutó de forma individual para cada modelo, como se documenta en sus respectivos notebooks (`RL.ipynb`, `RF.ipynb`, `GB.ipynb`, `MLP.ipynb`). Se registraron las métricas de rendimiento tanto para el conjunto de entrenamiento sobremuestreado como para el conjunto de prueba original.

2.2. Resultados del Entrenamiento Base

Los resultados del entrenamiento inicial con parámetros por defecto permitieron obtener una primera impresión del comportamiento de cada algoritmo. Se evaluaron métricas globales como `accuracy`, `precision`, `recall` y `F1-score`, y se generaron visualizaciones como matrices de confusión y curvas ROC para un análisis más detallado.

Por ejemplo, en el entrenamiento inicial del modelo MLP, se obtuvo una matriz de confusión que muestra el número de predicciones correctas e incorrectas para cada clase en el conjunto de prueba, como se observa en la figura `matriz_confusion_default.png`. De manera similar, las curvas ROC, visibles en `roc_auc_default.png`, ilustraron la capacidad del modelo para discriminar entre las diferentes clases.

Esta fase concluyó con la obtención de un conjunto de métricas y resultados de línea de base para cada uno de los cuatro modelos, lo que sirvió como punto de partida para la fase de optimización. Asimismo se abordó la problematica de la Clase 6.

Hito 3: Optimización del Modelo

El objetivo de este hito fue mejorar el rendimiento de los modelos base mediante el ajuste de hiperparámetros y la aplicación de técnicas de regularización para mejorar la generalización y evitar el sobreajuste.

3.1. Ajuste de Hiperparámetros

Para cada uno de los modelos, se llevó a cabo un proceso de optimización de hiperparámetros. La técnica principal utilizada fue la búsqueda en rejilla con validación cruzada (`GridSearchCV`). Se definió un espacio de búsqueda de hiperparámetros para cada algoritmo, cubriendo aquellos que tienen un mayor impacto en el rendimiento del modelo, como la tasa de aprendizaje, el número de estimadores, la profundidad de los árboles o los parámetros de regularización.

El proceso de `GridSearchCV` se entrenó utilizando el conjunto de entrenamiento sobremuestreado (`X_train_oversampled`, `Y_train_oversampled`) para encontrar la combinación de hiperparámetros que maximizaba una métrica de rendimiento específica (generalmente, el `F1-score` ponderado, debido al desbalance de clases original).

Una vez encontrados los mejores hiperparámetros, se reentrenó el modelo con esta configuración óptima y se evaluó su rendimiento final sobre el conjunto de prueba no modificado. Forzando, también, la presencia de la clase 6.

3.2. Regularización y Generalización

Se prestaron esfuerzos para controlar el sobreajuste (overfitting), que ocurre cuando un modelo aprende el ruido del conjunto de entrenamiento en lugar de los patrones subyacentes, lo que lleva a un bajo rendimiento en datos nuevos. La principal estrategia para identificar el sobreajuste fue comparar las métricas de rendimiento entre el conjunto de entrenamiento y el de prueba. Una brecha significativa entre ambas indicaría un posible sobreajuste.

Se aplicaron diversas técnicas de regularización:

- Regularización L1/L2: En el modelo de Regresión Logística, se exploraron los parámetros de regularización `penalty` y `C` para controlar la complejidad del modelo.
- Parámetros de los árboles: En los modelos basados en árboles como Random Forest y Gradient Boosting, hiperparámetros como `max_depth`, `min_samples_leaf` y `n_estimators` actúan como regularizadores al limitar la complejidad de los árboles de decisión individuales.
- Dropout: En el Perceptrón Multicapa (MLP), se incluyeron capas de `Dropout` en la arquitectura. Esta técnica desactiva aleatoriamente un porcentaje de neuronas durante el entrenamiento, forzando a la red a aprender representaciones más robustas y menos dependientes de neuronas específicas.

Los resultados de los modelos optimizados, como se puede apreciar en las matrices de confusión (`confusion_matrix_hyperparams.png`) y curvas ROC (`roc_curves_hyperparams.png`) del MLP optimizado, reflejan el efecto de este proceso de ajuste y regularización.

Memoria Académica: Hito 4 - Evaluación y Comparación de Modelos

Introducción

Esta memoria documenta de manera descriptiva y académica el proceso y los resultados correspondientes al Hito 4 del proyecto. El objetivo de esta fase fue la evaluación final y la comparación exhaustiva de los cuatro modelos de clasificación (Regresión Logística, Random Forest, Gradient Boosting y Perceptrón Multicapa) después de su optimización en el Hito 3. El análisis se centra en el rendimiento de los modelos sobre el conjunto de datos de prueba, un análisis detallado de los errores y las propuestas de mejora derivadas de dicho análisis, basándose en los resultados consolidados en el notebook `Maestro.ipynb` y los archivos de datos asociados.

4.1. Evaluación y Comparación de Modelos en el Conjunto de Prueba

La evaluación final de los modelos optimizados se realizó sobre el conjunto de prueba, que no fue utilizado durante las fases de entrenamiento u optimización, garantizando así una medida insesgada de su capacidad de generalización. Todos los resultados, métricas y artefactos generados por los modelos individuales fueron centralizados y consolidados en el notebook `Maestro.ipynb` para facilitar un análisis comparativo directo y riguroso.

4.1.1. Comparación de Métricas de Rendimiento

Se realizó la comparación cuantitativa utilizando un conjunto de métricas de rendimiento globales y por clase. El archivo `Master.csv` y `Resultados_Datos_Entrenados.csv` compilan las métricas clave como accuracy, precision, recall y F1-score para cada modelo en los conjuntos de entrenamiento y prueba. Esto permitió no solo comparar los modelos entre sí, sino también evaluar el grado de sobreajuste al observar la diferencia de rendimiento entre ambos conjuntos.

Para una visualización más clara, se generaron tablas y gráficos comparativos. Por ejemplo, el archivo `AUC_Scores_Completo.csv` resume el rendimiento del Área Bajo la Curva (AUC) para cada clase y modelo, mientras que `ROC_Curves_Completo.csv` contiene los datos para graficar las curvas ROC de todos los modelos, permitiendo una comparación visual de su capacidad para discriminar entre clases.

4.1.2. Eficiencia Computacional

Además del rendimiento predictivo, se consideró la eficiencia computacional de cada modelo. El archivo `Tiempos_Computacionales.csv` registra los tiempos de entrenamiento y predicción para cada algoritmo, un factor relevante en entornos de producción donde la velocidad de respuesta puede ser crítica.

4.2. Análisis de Errores

Se llevó a cabo un análisis profundo de los errores de predicción con el fin de identificar patrones y comprender las debilidades de cada modelo. Este proceso es fundamental para proponer mejoras informadas.

4.2.1. Análisis de Matrices de Confusión

El análisis de las matrices de confusión, cuyos datos se consolidaron en el archivo `Matrices_Confusion.csv`, fue la principal herramienta para este propósito. Al examinar qué clases se confundían entre sí con mayor frecuencia, se pudieron identificar las áreas problemáticas específicas. Por ejemplo, se observó si los modelos tendían a confundir partidos políticos con ideologías similares o si las clases minoritarias eran sistemáticamente clasificadas incorrectamente como clases mayoritarias.

4.2.2. Métricas por Clase

Además de las matrices de confusión, se realizó un análisis detallado de las métricas de rendimiento para cada clase individual, como se recoge en `Metricas_Por_Clase.csv`. Esto permitió identificar con precisión qué clases eran las más difíciles de predecir para cada modelo, observando valores bajos de `precision` o `recall` para clases específicas, especialmente las minoritarias.

4.2.3. Propuestas de Mejora

Basándose en el análisis de errores, se propusieron varias vías para mejorar el rendimiento del modelo en futuras iteraciones:

- Mejoras en el Dataset:
  - Ingeniería de Características (Feature Engineering): Creación de nuevas variables que capten mejor las sutilezas entre las clases que se confunden con frecuencia.
  - Tratamiento Avanzado de Clases Minoritarias: Explorar técnicas de sobremuestreo más sofisticadas que el oversampling estándar.
- Mejoras en la Arquitectura del Modelo:
  - Ensamble de Modelos (Model Stacking/Blending): Combinar las predicciones de los modelos más efectivas (por ejemplo, Random Forest y Gradient Boosting) para crear un meta-modelo que podría superar el rendimiento de cualquier modelo individual.
  - Ajuste Fino Específico: Re-optimizar los hiperparámetros de un modelo centrándose específicamente en mejorar el rendimiento de las clases peor clasificadas.

Este análisis de errores no solo sirvió para evaluar el estado final de los modelos, sino que también proporcionó una hoja de ruta clara para el trabajo futuro en el proyecto.

Memoria Académica: Hito 5 - Interpretación y Evaluación de Sesgos

Introducción

Esta memoria se enfoca en la interpretación de los modelos de machine learning y la evaluación de posibles sesgos. La confianza y la transparencia son esenciales en la implementación de sistemas de inteligencia artificial, y este hito aborda directamente estos aspectos. El objetivo es desmitificar la caja negra de los modelos, entender qué características impulsan sus decisiones y analizar si estas decisiones afectan de manera desproporcionada a los diferentes subgrupos de la población. Para ello, se utiliza la técnica SHAP (SHapley Additive exPlanations) como herramienta principal, aplicada a los cuatro modelos optimizados en las fases anteriores.

5.1. Interpretación del Modelo con SHAP

Este punto se abordó utilizando exclusivamente la metodología SHAP (SHapley Additive exPlanations). La elección se fundamenta en su sólida base teórica, derivada de la teoría de juegos cooperativos, que permite asignar a cada característica una contribución específica y justa a la predicción final. A diferencia de otras técnicas locales como LIME, SHAP garantiza consistencia y exactitud, asegurando que la suma de las importancias de las características sea igual a la predicción del modelo.

5.1.1. Cálculo de los Valores SHAP

Se calcularon los valores SHAP para los cuatro modelos sobre un subconjunto del conjunto de datos de prueba para mantener la eficiencia computacional. Se utilizaron los `explainers` apropiados para cada tipo de modelo:

- **`shap.TreeExplainer`** para los modelos basados en árboles (Random Forest y Gradient Boosting).
- **`shap.KernelExplainer`** para el modelo de Regresión Logística.
- **`shap.DeepExplainer`** para el modelo de Perceptrón Multicapa (MLP) desarrollado en PyTorch.

Los resultados de este análisis, que incluyen los `summary plots` y los `dependence plots`, se encuentran detallados en el notebook `hito5.ipynb`.

5.1.2. Análisis de la Importancia de las Características

El análisis de los valores SHAP permitió identificar las características más influyentes para cada modelo. Los `summary plots` generados para cada algoritmo muestran la distribución de los valores SHAP para cada característica, revelando no solo qué variables son más importantes, sino también cómo afectan a la predicción (un valor SHAP positivo indica un aumento en la probabilidad de la clase predicha, y viceversa).

Se observó una consistencia notable entre los modelos en cuanto a las características más relevantes. Variables como la intención de voto previa (**`intencion_voto_encoded`**) y algunas de las componentes principales

derivadas del análisis de PCA (`categorico_pca_0`, `categorico_pca_1`, etc.) aparecieron consistentemente en los primeros puestos de importancia, lo que valida su relevancia en el problema de predicción.

5.2. Evaluación de Sesgos

Una de las aplicaciones más importantes de la interpretabilidad es la capacidad de auditar los modelos en busca de sesgos. Un modelo se considera sesgado si su rendimiento o su comportamiento varía significativamente entre diferentes subgrupos de la población, especialmente aquellos definidos por características sensibles.

5.2.1. Metodología de Análisis de Sesgos

El análisis de sesgos se centró en dos variables sensibles identificadas en el conjunto de datos:

- Género (**`genero_encoded`**)
- Autoubicación Ideológica (**`autoubicacion_ideologica_encoded`**)

La estrategia consistió en segmentar los datos según los valores de estas variables y analizar si existían disparidades en el comportamiento del modelo. Para ello, se comparó la distribución de los valores SHAP para cada característica entre los diferentes subgrupos. Una diferencia sistemática en la magnitud o dirección de los valores SHAP entre, por ejemplo, hombres y mujeres, podría ser un indicador de sesgo.

5.2.2. Hallazgos del Análisis de Sesgos

El análisis, documentado en el notebook `hito5.ipynb`, reveló los siguientes puntos:

- Diferencias de Escala: Se observó que la escala de los valores SHAP variaba considerablemente entre los diferentes tipos de modelos, lo que impidió una comparación numérica directa de las magnitudes de SHAP entre ellos. Por ejemplo, los valores para la Regresión Logística fueron de un orden de magnitud mucho mayor que para los modelos de árboles.
- Indicios de Disparidad: A pesar de las diferencias de escala, se encontraron patrones consistentes. Tanto en el modelo Random Forest como en el MLP, se observaron indicios de que ciertos grupos ideológicos recibían, en promedio, contribuciones SHAP de mayor magnitud. Las diferencias por género, aunque presentes, fueron menos pronunciadas.
- Conclusión Provisional: El análisis sugiere la existencia de disparidades en el tratamiento de los subgrupos por parte de los modelos, especialmente en relación con la autoubicación ideológica. Sin embargo, como se señala en las conclusiones del notebook, para confirmar un sesgo sistemático se requeriría un análisis más profundo, incluyendo pruebas estadísticas y una visualización más detallada de las diferencias por característica.

5.2.3. Recomendaciones y Pasos Futuros

El análisis concluyó con una serie de recomendaciones para futuros trabajos, entre las que se incluyen:

- Mejorar la organización y visualización de los resultados de SHAP para facilitar la comparación entre subgrupos.
- Realizar un análisis estadístico para confirmar la significancia de las disparidades observadas.
- En caso de confirmarse la existencia de sesgos, aplicar técnicas de mitigación, como el reequilibrio de datos, el ajuste de umbrales de decisión o el uso de algoritmos de aprendizaje justo (fairness-aware learning).