DOCUMENTACIÓN

1. Carga y limpieza de datos

El dataset se carga desde un archivo CSV utilizando pandas. Luego, se eliminan valores nulos y duplicados para asegurar la calidad de los datos. También se expanden contracciones en el texto para mejorar la normalización.

2. Preprocesamiento del texto

Se aplican varias técnicas de limpieza y normalización del texto:

- Eliminación de caracteres no ASCII.
- Conversión a minúsculas.
- Eliminación de signos de puntuación.
- Sustitución de números por su representación textual.
- Eliminación de stopwords en español.
- Tokenización de palabras.
- Lematización y stemming para reducir palabras a su forma base.

3. Vectorización del texto

Las palabras procesadas se convierten en representaciones numéricas mediante TfidfVectorizer, seleccionando las 1000 palabras más relevantes y eliminando stopwords. Luego, los datos se dividen en conjuntos de entrenamiento y prueba.

4. Algoritmos utilizados

- Árbol de Decisión (Decision Tree): Es un modelo de clasificación que divide los datos en ramas basadas en condiciones lógicas, facilitando la interpretación. Se utilizó el criterio de entropía para medir la pureza de los nodos y una profundidad máxima de 4 para evitar sobreajuste.
- K-Nearest Neighbors (KNN): Clasifica los datos basándose en la similitud con sus vecinos más cercanos. En este caso, se usaron 3 vecinos para determinar la clase de cada observación.
- Random Forest: Un conjunto de múltiples árboles de decisión entrenados con diferentes subconjuntos de datos para mejorar la precisión y reducir la varianza. Se usaron 100 árboles para lograr un balance entre rendimiento y velocidad.

Estos modelos fueron evaluados utilizando métricas como exactitud, recall, precisión y puntuación F1 para comparar su desempeño.

5. Entrenamiento del modelo Decision Tree

Se entrena un clasificador de árbol de decisión con criterio de entropía y una profundidad máxima de 4. Luego, se evalúa el modelo calculando métricas como exactitud, recall, precisión y puntuación F1. También se muestra la matriz de confusión.

6. Entrenamiento del modelo KNN

Se entrena un clasificador K-Nearest Neighbors (KNN) con 3 vecinos. Posteriormente, se realizan predicciones sobre el conjunto de prueba y se calcula la matriz de confusión. Se evalúa el rendimiento del modelo utilizando métricas como exactitud, recall, precisión y puntuación F1.

7. Entrenamiento del modelo Random Forest

Se implementa un clasificador Random Forest con 100 árboles de decisión y un estado aleatorio de 42. Luego, se evalúa con métricas de exactitud, recall, precisión y F1-score, además de visualizar la matriz de confusión.

8. Exportación del mejor modelo

Se selecciona el mejor modelo basado en las métricas obtenidas y se extraen las palabras clave más relevantes de cada categoría de noticias.

Análisis palabras

```
# Actualizar el feature array
feature_array = vectorizer.get_feature_names_out()

# Top 10 palabras de cada grupo
top_n = 10  # Número de palabras más importantes

for tipo in df_news['Label'].unique():
    tipo_texts = df_news[df_news['Label'] == tipo]['words']
    tipo_tfidf = vectorizer.transform(tipo_texts)
    avg_tfidf_weights = tipo_tfidf.mean(axis=0).A1
    top_features_idx = np.argsort(avg_tfidf_weights)[-top_n:]
    top_features = feature_array[top_features_idx]
    print(f"Palabras clave para tipo {tipo}: {top_features}")
```

```
Palabras clave para tipo 1: ['años' 'com' 'si' 'tras' 'president' 'part' 'gobierno' 'gobiern' 'par'
 'pp']
Palabras clave para tipo 0: ['catalunya' 'pnv' 'bng' 'part' 'president' 'gobierno' 'gobiern' 'par'
 'vers' 'per']
```

Para tipo 1 (noticias falsas), las palabras clave incluyen: "años", "com", "tras", "president", "part", "gobierno", "gobiern", "par", "pp". Estas palabras sugieren un énfasis en temas políticos y gubernamentales, con términos como "president", "gobierno" y "pp", que podrían indicar una tendencia a la desinformación en temas políticos.

Para tipo 0 (noticias verdaderas), las palabras clave incluyen: "catalunya", "pnv", "bng", "part", "president", "gobierno", "gobiern", "par", "vers", "per". Aquí, aparecen términos similares a los del otro grupo, pero con un énfasis más localizado en partidos políticos regionales ("pnv", "bng", "catalunya"), lo que sugiere que los textos verídicos pueden estar más relacionados con temas políticos específicos en lugar de generalizaciones.