PROYECTO 1 - ETAPA 1

GRUPO 22
Juan Manuel Ramírez Tamayo
Santiago Celis Rengifo
Luis Alfredo Borbón Holguín

Construcción de modelos de analítica de textos

Sección 1. Documentación del proceso de aprendizaje automático

TAREA DE APRENDIZAJE	DECISIONES	PROPUESTA DE VALOR	RECOLECCIÓN DE DATOS – NO SE DEBE DILIGENCIAR	FUENTES DE DATOS
¿Cuál es el tipo de aprendizaje? ¿Si es aprendizaje supervisado, indicar qué se predice? ¿Cuáles son los posibles resultados de la tarea de aprendizaje? ¿Cuándo se observan los resultados de esta tarea? Por ejemplo, si es un modelo predictivo indicar si el resultado se obtiene unas horas, días, semanas, o meses antes. En este proyecto manejamos un caso de aprendizaje supervisado. Los datos iniciales poseían una columna adicional que mencionaba nuestra variable objetivo. En este caso, esta variable objetivo representaba si una noticia es falsa o verdadera, y eso es lo que se desea predecir. Se tiene una clasificación binaria	¿Cómo se convierten los resultados del modelo en recomendaciones o decisiones procesables para el usuario final? Los resultados del modelo se convierten en recomendaciones o decisiones procesables al clasificar automáticamente las noticias como verdaderas o falsas, permitiendo al usuario final enfocarse en validar las noticias marcadas como falsas (fake news). Por ejemplo, si un periodico digital desea publicar contenido nuevo, puede revisar antes de	¿Quién es el beneficiario final? ¿De qué empresa es? ¿Qué problemas específicos se abordan? ¿Qué riesgo puede tener para ese beneficiario el uso de este modelo? El beneficiario final pueden ser las empresas de comunicación, redes sociales o verificadores de noticias, cuyo objetivo es identificar y combatir la desinformación (fake news). El modelo ayuda a filtrar noticias falsas de manera eficiente,	¿Cómo se obtiene el conjunto inicial de entidades y resultados (por ejemplo, extractos de bases de datos, extracciones de API, etiquetado manual)? ¿Qué estrategias se aplican para actualizar los datos continuamente, controlando los costos y manteniendo la vigencia?	¿Qué fuentes de datos se utilizan? (Menciona tablas de bases de datos internas y externas o métodos API). ¿De dónde se toman los datos? ¿Se pueden utilizar para realizar el objetivo del análisis? Las fuentes de datos utilizadas son provenientes de un archivo .csv. En este archivo se tienen múltiples registros de varias noticias, tanto falsas como verdaderas. Estos datos tienen etiquetas que nos ayudan a aplicar distintos algoritmos. Los datos se obtuvieron de recopilaciones previas de noticias verificadas por

en la que si una noticia es verdadera se le asigna un valor de 1 y luego un valor de 0 si es falsa. Los posibles resultados de la tarea incluyen resultados **verdaderos** positivos y negativos; y **falsos** positivos y negativos. Estos resultados se pueden evaluar con métricas como el recall y la precisión. Los modelos que se aplicarán son predictivos, es decir, diseñados para clasificar nuevas noticias rápidamente obteniendo resultados en tiempo real o en cuestión de segundos tras introducir una nueva noticia, dependiendo del algoritmo usado y el tamaño del conjunto de datos.	compartir o publicar estas noticias. Esto ayuda a tomar decisiones informadas.	abordando el problema de la propagación de fake news, que afecta la confianza pública y la credibilidad. Sin embargo, existe el riesgo de falsos positivos o negativos, lo que podría llevar a clasificar erróneamente noticias legítimas como falsas o viceversa, afectando la confiabilidad del sistema. Esto puede variar dependiendo del algoritmo usado.		expertos. Estos datos son adecuados para el objetivo del análisis debido a que proporcionan la suficiente información para entrenar modelos predictivos y evaluar los textos. Las principales columnas que nos ayudan en este proceso, son las de descripción y label.

SIMULACIÓN DE IMPACTO

¿Cuáles son los valores de costo/beneficio de las decisiones (in)correctas? ¿Cuáles son los criterios de éxito del modelo para su posterior despliegue? ¿Existen restricciones de equidad?

El costo de decisiones incorrectas incluye la propagación de noticias falsas (falsos negativos) que afecten la credibilidad y confianza en el modelo y por otro lado, la censura de noticias válidas (falsos positivos). El beneficio de las decisiones correctas es mayor precisión en la detección, lo que fortalece los resultados. Los criterios de éxito del modelo incluyen una alta precisión, un recall equilibrado y un bajo número de falsos positivos/negativos antes del despliegue. Las principales restricciones de equidad son evitar sesgos en los datos de entrada que puedan favorecer ciertas fuentes o temas, garantizando que el modelo sea imparcial.

APRENDIZAJE (USO DEL MODELO)

¿El uso del modelo es por lotes o en tiempo real? ¿Con qué frecuencia se usa?

El principal uso del modelo es en tiempo real, ya que está diseñado para clasificar noticias nuevas inmediatamente después de recibirlas. La frecuencia de uso dependerá del flujo de publicaciones o consultas, pero en una implementación común, podría utilizarse constantemente para procesar grandes cantidades de noticias conforme se generan.

CONSTRUCCIÓN DE MODELOS

¿Cuántos modelos se necesitan? ¿Cuándo deben actualizarse? ¿De cuánto tiempo se dispone para generar el modelo (incluido el proceso de ingeniería de características y el análisis o evaluación del mismo)?

Se necesita al menos un modelo para poder realizar el análisis, sin embargo, en este proyecto se usarán 3 modelos. El tiempo necesario para generar el modelo depende del tamaño de los datos, pero incluye el proceso de ingeniería de características, entrenamiento y evaluación. En promedio, este proceso podría oscilar entre horas o pocos días en un entorno de desarrollo bien optimizado si el tamaño de los datos es significativamente grande y si se está aplicando más de un modelo.

INGENIERÍA DE CARACTERÍSTICAS

¿Qué variables/características se utilizan en el modelo? ¿Qué agregaciones o transformaciones se aplican a las fuentes de datos originales – incluir las más importantes--?

Las características utilizadas en el modelo provienen de la columna ‘Descripción’ del conjunto de datos, que se transforma mediante vectorización TF-IDF para convertir el texto en una representación numérica basada en la relevancia de las palabras dentro de cada noticia. Además, el texto pasa por procesos de limpieza y normalización, como eliminación de caracteres especiales, conversión a minúsculas, eliminación de stopwords, tokenización y lematización. Estas transformaciones son esenciales para reducir el ruido y garantizar que las entradas sean estandarizadas y útiles para el modelo de aprendizaje. También es importante resaltar que la columna ‘Label’ nos permite saber si una noticia es falsa o verdadera y esto ayuda en el proceso de modelado y predicción de manera fundamental.

Por otro lado, puede que cada algoritmo necesite de pasos adicionales para preparar los datos.

	MONITOREO NO SE DEBE DILIGENCIAR ¿Qué métricas y KPI se utilizan para hacer un seguimiento del impacto de la solución de ML una vez desplegada, tanto para los usuarios finales como para la empresa? ¿Con qué frecuencia deben revisarse?	

Sección 2. Entendimiento y preparación de los datos (a nivel de código y análisis).

En esta sección, se detalla el proceso de perfilamiento, análisis de calidad y transformación de los datos, con el fin de garantizar un conjunto de datos limpio, estructurado y adecuado para la tarea de clasificación binaria (verdadero o falso). Este análisis consideró tanto la estructura general del dataset como aspectos específicos que impactan directamente en la resolución del problema planteado (detección de noticias falsas a través del texto). Este análisis permitió poder hacer las respectivas transformaciones a los datos, es decir, remover columnas que no se necesitaban, eliminar duplicados y demás.

2.1. Entendimiento de los datos

Descripción de los datos

El archivo utilizado incluye un total de **60215 registros**, con las siguientes columnas:

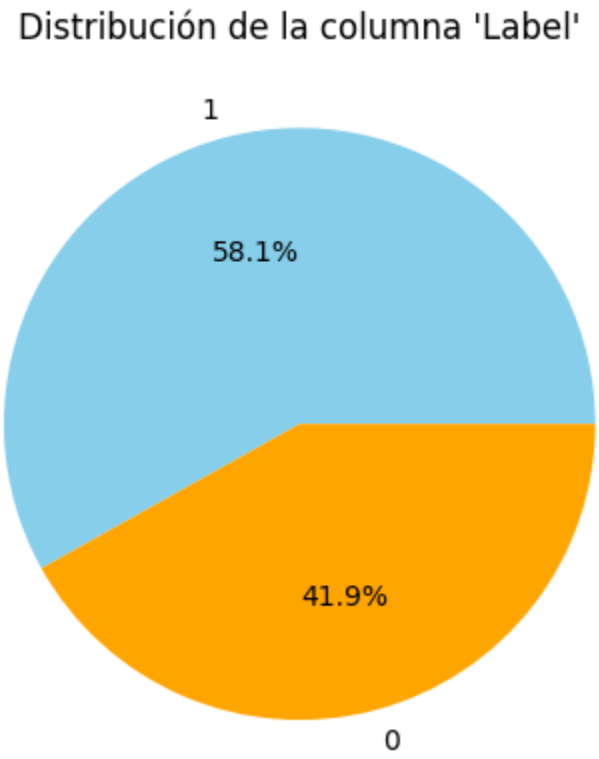
- **ID (int):** Identificador único de cada noticia.
- **Label (int):** Clase binaria que indica si la noticia es `1` (verdadera) o `0` (falsa).
- **Título (char):** Encabezado de la noticia.
- **Descripción (char):** Texto descriptivo que detalla el contenido de la noticia.
- **Fecha (date):** Fecha en la que se publicó la noticia.

Perfilamiento inicial

El perfilamiento incluyó:

- **Estadísticas descriptivas:** Se evaluaron métricas básicas como cantidad de registros, valores nulos y duplicados:
    - Se identificaron **16 títulos sin descripción** y 7425 duplicados en el campo `Descripción`, que fueron gestionados en la etapa de limpieza.

- El dataset tiene un **58% de noticias verdaderas** y **42% de noticias falsas**, como se aprecia en el siguiente gráfico circular:



Observación: La distribución relativamente balanceada de etiquetas es favorable para un modelo supervisado, ya que reduce el riesgo de sesgo hacia una de las clases.

- **Tipos de datos:** Se analizaron los tipos de datos, confirmando que las variables son adecuadas:
  - **Numéricas:** `ID`, `Label`.
  - **Categóricas:** Columnas como `Descripción`, que posteriormente se transformaron en datos numéricos mediante **TF-IDF**.
  - **Fecha:** `Fecha` no es relevante para el análisis directo.
- **Valores atípicos:** En la **columna `Descripción`** se calcularon longitudes de texto, identificando máximos, mínimos y distribuciones de palabras. Esto permitió verificar que los datos estaban dentro de rangos razonables para análisis de texto.

Justificación de decisiones iniciales

1. **Duplicados:** Se eliminaron descripciones duplicadas ya que un mismo texto no aporta valor adicional al análisis.
2. **Valores nulos:** Dado que faltaban 16 descripciones, se decidió eliminarlas para evitar inconsistencias, ya que este dato es clave para el análisis textual.

3. **Transformación de datos no numéricos:** La `Descripción` se convirtió en datos procesables (representación vectorial) para los algoritmos de aprendizaje.

2.2. Preparación de los datos

1. Limpieza de los datos

Se aplicaron las siguientes técnicas para garantizar consistencia en el dataset:

- **Eliminación de duplicados:** Eliminamos los duplicados basándonos en el campo `Descripción`, reduciendo los registros totales de **60215 a 49638**. Esto mejora la unicidad y evita que patrones redundantes sesguen el modelo.
- **Eliminación de columnas irrelevantes:** `ID` y `Fecha` fueron eliminadas, ya que no aportan información relevante para la tarea de clasificación.
- **Limpieza del texto:** Se realizó preprocesamiento textual eliminando símbolos especiales, stopwords, y aplicando lematización para reducir las palabras a sus formas base.

```python
class DataFrameCleaner(BaseEstimator, TransformerMixin):
    def __init__(self, columns_to_remove=None, subset_for_duplicates=None):
        self.columns_to_remove = columns_to_remove  # Columnas a eliminar
        self.subset_for_duplicates = subset_for_duplicates  # Columnas para identificar duplicados

    def fit(self, X, y=None):
        return self

    def transform(self, X, y=None):
        print(f"Tamaño inicial del DataFrame: {X.shape}")

        # Eliminar columnas que no se quieren
        if self.columns_to_remove:
            X = X.drop(columns=self.columns_to_remove, errors="ignore")
            print(f"Columnas eliminadas: {self.columns_to_remove}")

        # Eliminar registros duplicados
        if self.subset_for_duplicates:
            # Basarse en un subconjunto específico
            num_duplicados = X.duplicated(subset=self.subset_for_duplicates).sum()
        else:
            # Basarse en todas las columnas (por defecto)
            num_duplicados = X.duplicated().sum()

        print(f"Número de duplicados antes de limpiar: {num_duplicados}")

        # Eliminar duplicados
        if self.subset_for_duplicates:
            X = X.drop_duplicates(subset=self.subset_for_duplicates)
        else:
            X = X.drop_duplicates()

        # Recuento final de duplicados
        print(f"Número de duplicados después de limpiar: {X.duplicated().sum()}")

        # Imprimir tamaño final del DataFrame
        print(f"Tamaño final del DataFrame: {X.shape}")
```

2. Transformaciones

El texto en la columna Descripción se transformó utilizando **TF-IDF**, una técnica estándar para análisis de texto vista en clase pero aplicada en muchos campos en Machine Learning. Esto creó vectores que asignan mayor peso a las palabras más relevantes en el contexto de la noticia.

```
X_transformed = pipeline.named_steps['tfidf'].fit_transform(news_normalized)
print(X_transformed[:5])
print(X_transformed.shape)
```

3. Divisiones del conjunto

Se dividió el conjunto de datos en entrenamiento (80%) y prueba (20%) para garantizar un correcto entrenamiento y evaluación de los algoritmos. Esto asegura que los modelos sean evaluados con datos que no han visto antes.

```
X = news_df['Descripcion']
y = news_df['Label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

pipeline_neural_net.fit(X_train, y_train)

y_pred = pipeline_neural_net.predict(X_test)

results_df = pd.DataFrame({
    'Opinión': X_test,
    'Predicción': y_pred
})
results_df
```

Justificación de Transformaciones

Preprocesamiento textual: La limpieza y normalización son fundamentales para poder eliminar ruido y ambigüedades del texto, lo que mejora el desempeño del modelo.

TF-IDF: Es una técnica efectiva para capturar relaciones entre palabras en un contexto, convirtiendo datos cualitativos en cuantitativos.

División de datos: Un conjunto de prueba adecuado es crucial para garantizar que el modelo generaliza y no se sobreajusta.

Sección 3. Modelado y evaluación.

En esta sección, se implementaron y evaluaron tres modelos analíticos distintos para abordar la tarea de aprendizaje supervisado de clasificación. Los modelos seleccionados fueron: **Árboles de decisión, Neuronales**, **Naive Bayes** y **KNN**. A continuación, se describen brevemente los algoritmos utilizados, las métricas de evaluación aplicadas y los resultados obtenidos, justificando la selección del modelo más adecuado para la tarea. Cada modelo fue implementado y documentado por estudiantes específicos, según se detalla.

Modelo 1: Árboles de Decisión (Luis Borbón Holguín)
Este modelo crea reglas de decisión en forma de árbol, dividiendo los datos en diferentes ramas basadas en características importantes.

Modelo 2: Naive Bayes (Juan Manuel Ramírez Tamayo)

Modelo basado en probabilidades que asume independencia entre las palabras del texto. Es rápido y eficiente para tareas de clasificación de texto.

3. KNN (Santiago Celis)

Clasifica los datos basándose en las categorías de los vecinos más cercanos utilizando la distancia TF-IDF. Se determinó que el mejor valor de vecinos es k=9 y se ajustó un umbral de 0.6.

Sección 4. Resultados.

Se implementaron los algoritmos de KNN, Árboles de decisión y Naive Bayes

En los tres algoritmos se implementaron matrices de confusión con 4 cuadros para definir en cuando se generaron las predicciones correctas (verdaderos positivos y negativos) y los errores cometidos (falsos positivos y negativos).

Se evaluaron otras métricas de rendimiento como la precisión, el recall y la puntuación F1 que brindó la posibilidad de facilitar una comparación cuantitativa del desempeño de cada modelo. De la misma forma que se analizaron las curvas ROC y Precision-Recall para observar cómo varían estas métricas al modificar el umbral de clasificación.

- KNN fue más eficiente en las agrupaciones del parámetro K
- Arboles de Desición fue más eficiente en las reglas de decisión
- Naive Bayes fue más eficiente en la clasificación de los datos en el enfoque probabilístico.

A partir de las tres matrices de confusión mostradas (KNN, Árboles de Decisión y Naive Bayes), podemos comparar en detalle tanto la exactitud general como el balance entre falsos positivos y falsos negativos. A continuación, se presenta un análisis punto por punto:

KNN (k=9)

- Verdaderos Negativos: 1870
- Falsos Positivos: 2938
- Falsos Negativos: 1954
- Verdaderos Positivos: 4651

La cantidad de falsos positivos y negativos es un poco equilibrada, por lo tanto KNN comete errores de ambos tipos de una forma similar.

- Datos totales = 11413
- Exactitud aproximada -> (1870+4561)/11413 ≈ 57,1%

Árbol de decisión

- Verdaderos Negativos: 132
- Falsos Positivos: 4676
- Falsos Negativos: 3
- Verdaderos Positivos: 6602

La cantidad de falsos positivos y negativos es extremadamente dispar, por lo tanto los árboles de decisión cometen muchos más errores de un tipo que de otro.

- Datos totales = 11413
- Exactitud aproximada -> (132+6602)/11413 ≈ 59,0%

Naive Bayes

- Verdaderos Negativos: 1296
- Falsos Positivos: 3512
- Falsos Negativos: 298
- Verdaderos Positivos: 6307

La cantidad de falsos positivos y negativos es extremadamente dispar, por lo tanto naive bayes comete muchos más errores de un tipo que de otro.

Datos totales = 11413
Exactitud aproximada -> (1296+6307)/11413 ≈ 66,6%

Recall

KNN (k=9)

- Falsos Negativos: 1954
- Verdaderos Positivos: 4651
- Recall -> 4651/(4651+1954) ≈ 70,4%

Árbol de decisión

- Falsos Negativos: 3

- Verdaderos Positivos: 6602
- Recall -> 6602/(6602+3) ≈ 99,9%

Naive Bayes

- Falsos Negativos: 298
- Verdaderos Positivos: 6307
- Recall -> 6307/(6307+298) ≈ 95,5%

El recall es muy importante en los contextos donde es crítico detectar la mayor cantidad de noticias falsas, aunque debe evaluar el balance con precisión entre detectar correctamente las noticias que son falsas y evitar que se clasifican erróneamente.

Sección 5. Trabajo en equipo

Para la elaboración del proyecto se realizaron múltiples reuniones con los integrantes del grupo. La cantidad de tiempo que duraban las reuniones variaba con base al tipo de reunión. La inicial duró dos horas y media debido que fue el primer acercamiento del grupo y del proyecto. El resto de reuniones eran de planeación y de seguimiento. En esas reuniones se analizaron las tareas pendientes y se establecieron tiempos de entrega. Cada estudiante comentaba de manera activa que avances tenía y que tareas iba a realizar después dependiendo del tema.

Como mencionaba el enunciado cada estudiante realizó un algoritmo distinto. Estos fueron escogidos personalmente por cada integrante y por ende, cada uno fue completamente responsable de su implementación. Sin embargo, en el caso en el que surgiera una duda, esta se resolvía de manera grupal con el fin de mantener un ambiente colaborativo. Los algoritmos implementados fueron los siguientes:

1. Santiago Celis Rengifo - Algoritmo KNN (K-N vecinos)
2. Luis Borbon Holguin - Algoritmo de Árboles de Decisión
3. Juan Manuel Ramírez Tamayo - Algoritmo de Naive Bayes

Principales retos:

Los principales retos en el proyecto fue el trabajo colaborativo, esto debido a que en nuestro proceso de desarrollo utilizamos el entorno de Google Colab, y por ende, dos integrantes del grupo no podían editar el notebook al mismo tiempo. Para solventar esto se realizó una repartición de tareas y se comunicaron los tiempos de trabajo para saber cuando alguien iba a realizar cambios en el código. Otro reto fueron las dificultades técnicas y de conocimiento del proyecto, para esto nos apoyamos en la documentación de SciKit-learn, los notebooks de la materia, el conocimiento de cada integrante y el uso de IA para algunas dudas.

Distribución de puntos:

Nosotros nos repartimos los 100 puntos de la siguiente manera debido a que consideramos que cada estudiante aportó de manera equitativa a las tareas del proyecto:

1. Santiago Celis Rengifo - 33.3 puntos
2. Luis Borbon Holguin - 33.3 puntos
3. Juan Manuel Ramírez Tamayo - 33.3 puntos

ROLES:

Se definieron los siguientes roles para poder gestionar las tareas y los distintos entregables que se necesitaban (cada líder organizó la división de trabajo en su área específica de tal manera que fuera lo más balanceado posible):

- **Líder de Proyecto:** Santiago Celis Rengifo
- **Líder de datos:** Juan Manuel Ramírez Tamayo
- **Líder de analítica:** Luis Borbon Holguin

Tareas realizadas:

Además de que cada estudiante realizará su respectivo algoritmo cada uno hizo las siguientes tareas:

1. Santiago Celis Rengifo:
    a. Organización del proyecto
    b. Parte 1. de la limpieza de datos
    c. Realización del documento
    d. Realización de las diapositivas a presentar
    e. Grabación del video
    f. Realización del algoritmo respectivo

2. Luis Borbon Holguin:
    a. Análisis del caso: Entendimiento del problema (revisión de variables, características, aspectos importantes).
    b. Realización del algoritmo respectivo.
    c. Coordinación del entendimiento del problema y reuniones afines.
    d. Preparación del ambiente de trabajo.
    e. Carga de datos iniciales.

3. Juan Manuel Ramírez Tamayo:
    a. Realización del algoritmo respectivo.
    b. Coordinación de las tareas de limpieza de datos.
    c. Parte 2. de la limpieza de datos.
    d. Realización de resultados en el Notebook.
    e. Realización de las diapositivas a presentar
    f. Edición del video

Sección 6. Otros entregables y espacios de evaluación

Link del repositorio: https://github.com/NEXAF7/Grupo22-BI-20251
Los demás archivos están cargados en el repositorio.