ENTREGA FINAL:
PREDICCIÓN DE LA VERACIDAD DE PROPUESTAS LABORALES

UNIVERSIDAD
DE ANTIOQUIA
1 8 0 3

EQUIPO DE TRABAJO
SAMUEL A. ARISTIZÁBAL GONZÁLEZ
JULITZA DAZA ZAPATA
JUAN DAVID LOPEZ AGUIRRE

PROFESOR ENCARGADO:
RAUL RAMOS POLLAN

UNIVERSIDAD DE ANTIOQUIA
FACULTAD DE INGENIERÍA
INTELIGENCIA ARTIFICIAL PARA LAS CIENCIAS Y LAS INGENIERÍAS
MEDELLÍN
2023

I. Introducción

La creciente digitalización de los procesos de reclutamiento ha dado lugar a un aumento en la proliferación de ofertas de trabajo fraudulentas, lo que destaca la necesidad urgente de soluciones automatizadas y eficientes.

Para ello, se realizó un modelo de inteligencia artificial diseñado para detectar ofertas de trabajo falsas frente a las genuinas representa una herramienta crucial en la lucha contra el fraude laboral y la protección de los candidatos. Este modelo puede aprovechar técnicas avanzadas de procesamiento de lenguaje natural (NLP) y aprendizaje automático para analizar el contenido de las ofertas de trabajo y detectar patrones que sugieran fraudes potenciales.

La capacidad de distinguir entre elementos legítimos y señales de advertencia en el lenguaje empleado, los requisitos poco realistas o la falta de información sobre la empresa puede ayudar a prevenir que los candidatos caigan en trampas laborales.

La implementación de un modelo de este tipo no solo contribuiría a la seguridad y la confianza de los candidatos, sino que también aliviaría la carga sobre los departamentos de recursos humanos y las plataformas de búsqueda de empleo, permitiéndoles identificar y abordar rápidamente ofertas engañosas. Además, al adaptarse a las tendencias cambiantes de los fraudes laborales, un modelo de inteligencia artificial puede evolucionar y actualizarse para hacer frente a nuevas estrategias utilizadas por los estafadores.

En última instancia, un modelo de este tipo no solo protegerá a los candidatos individuales, sino que también contribuirá a la integridad del mercado laboral digital, fomentando un entorno más seguro y confiable para la búsqueda de empleo.

II. Exploración de datos

De un dataset con dieciocho mil descripciones laborales de las cuales ochocientas son falsas se espera realizar un modelo capaz de clasificar las propuestas laborales falsas de las auténticas. El dataset escogido fue tomado de la plataforma Kaggle con el nombre de “Real / Fake Job Posting Prediction”. Las muestras tienen como columnas/variables las siguientes características:

1. **Job_id**
2. **title -** Título de la publicación

3. **location** - Ubicación del lugar de trabajo
4. **department** - Departamento de trabajo
5. **salary_range** - Rango de salario
6. **company_profile** - Perfil de la compañía
7. **description** - Descripción del trabajo
8. **requirements** - Requerimientos para el trabajo
9. **benefits** - Beneficios con el empleo
10. **telecommuting** - 0 o 1 si se realiza teletrabajo
11. **has_company_logo** - Si tiene logo de la empresa (0 o 1)
12. **has_questions** - Si tiene preguntas (0 o 1)
13. **employment_type** - Tipo de empleo (tiempo completo, medio tiempo, entre otros)
14. **required_experience** - Experiencia requerida
15. **required education** - Educación requerida
16. **industry** - industria en donde se desarrolla el trabajo
17. **function** - Función del trabajo
18. **fraudulent** - Característica de salida (0 o 1)

Con el objetivo de llenar las 30 columnas mínimas pedidas en el enunciado se llenaron 12 variables más conlas librerías pandas y random:

19. **country –** País de origen
20. **special_characters –** (0 o 1) si tiene caracteresespeciales
21. **phone_number –** (0 o 1) si tiene número decontacto
22. **website –** (0 o 1) si tiene pagina web
23. **longevity** – Cuantos años lleva la empresa registradaen la plataforma
24. **vacants –** Numero de vacantes
25. **contract_type -** Definido o indefinido
26. **comercial_id -** (0 a 1) Existe un número de registro de la empresa en base de datos gubernamentales o no
27. **multinational –** (0 o 1) Es multinacional o no
28. **number_employees** – (0 a 1) Informa sobre elnúmero de empleados actuales o no
29. **calification** – (0 a 1) Existen reseñas de los clientesen otras plataformas virtuales o no

30. **deadline –** (0 a 1) Hay fecha límite de la inscripciónpara la oferta

METRICAS

Las métricas seleccionadas para evaluar el desempeño del modelo son:

1. **Matriz de confusión:** La matriz de confusión es una herramienta valiosa para evaluar el rendimiento de un modelo de clasificación, como en el caso de la detección de publicaciones de trabajo fraudulentas. Esta matriz resume de manera concisa las predicciones del modelo en comparación con las clases reales.

2. **True Positives (TP)** es una medida que indica el número de casos en los que el modelo de clasificación predijo correctamente una publicación de trabajo fraudulenta como fraudulenta. En otras palabras, TP representa el número de casos positivos correctamente clasificados por el modelo. Esta métrica es importante para evaluar el rendimiento de un modelo de clasificación, ya que indica cuántos de los casos realmente fraudulentos han sido identificados correctamente por el modelo.
3. **False Positives (FP):** cuenta el número de casos en los que el modelo clasificó incorrectamente una publicación de trabajo genuina como fraudulenta. Representa los casos negativos incorrectamente clasificados por el modelo. En otras palabras, FP indica cuántas publicaciones de trabajo genuinas fueron erróneamente identificadas como fraudulentas por el modelo de clasificación. Esta métrica es fundamental para comprender los errores de clasificación del modelo y evaluar su desempeño en la detección de publicaciones de trabajo fraudulentas.
4. **True Negatives (TN):** cuenta el número de casos en los que el modelo de clasificación predijo correctamente una publicación de trabajo genuina como genuina. En otras palabras, TN representa el número de casos negativos correctamente clasificados por el modelo. Esta métrica es importante para evaluar el desempeño del modelo de clasificación, ya que indica cuántas publicaciones de trabajo genuinas fueron identificadas correctamente por el modelo.

La matriz de confusión es una herramienta útil para visualizar estas métricas y evaluar el rendimiento del modelo en la detección de publicaciones de trabajo fraudulentas.

5. **False Negatives (FN):** cuenta el número de casos en los que el modelo clasificó incorrectamente una publicación de trabajo fraudulenta como genuina. En otras palabras, FN representa los casos positivos incorrectamente clasificados por el modelo.
6. **Binary Accuracy:** calcula la precisión global del modelo, es decir, la proporción de predicciones correctas en relación con el total de predicciones realizadas. Esta métrica es especialmente útil en el contexto de clasificación binaria, donde se evalúa la coincidencia entre las predicciones y los valores reales para etiquetas binarias. Por ejemplo, en un problema de clasificación binaria, la métrica Binary Accuracy proporciona una medida de cuántas predicciones coinciden con las etiquetas reales, lo que permite evaluar el rendimiento general del modelo en la tarea de clasificación binaria.
7. **Precisión:** calcula la proporción de verdaderos positivos en relación con el total de predicciones positivas realizadas. Esta métrica mide la capacidad del modelo para evitar falsos positivos. La precisión es especialmente relevante en situaciones donde minimizar los falsos positivos es crucial, ya que proporciona información sobre la confiabilidad del modelo al clasificar muestras como positivas. Por ejemplo, en la detección de publicaciones de trabajo fraudulentas, la precisión es fundamental para evaluar cuán confiable es el modelo al identificar correctamente las publicaciones fraudulentas, evitando clasificar erróneamente publicaciones genuinas como fraudulentas.
8. **Recall:** también conocida como Sensibilidad o Tasa de Verdaderos Positivos (TPR), calcula la proporción de verdaderos positivos en relación con el total de casos positivos reales. Esta métrica mide la capacidad del modelo para detectar correctamente los casos positivos. En resumen, el recall proporciona información sobre el rendimiento del clasificador en términos de falsos negativos, es decir, cuántos casos positivos reales no fueron identificados por el modelo. Esta métrica es fundamental para evaluar la capacidad del modelo de detectar de manera efectiva los casos positivos en aplicaciones como la detección de publicaciones de trabajo fraudulentas.
9. **AUC (Área Bajo la Curva)**: evalúa la capacidad de discriminación del modelo al calcular el área bajo la curva ROC (Característica Operativa del Receptor). Cuanto mayor sea el valor de AUC, mejor será la capacidad del modelo para distinguir entre clases positivas y negativas. Esta métrica es particularmente útil en la evaluación del rendimiento de modelos de clasificación, ya que proporciona una medida agregada de la capacidad del modelo para

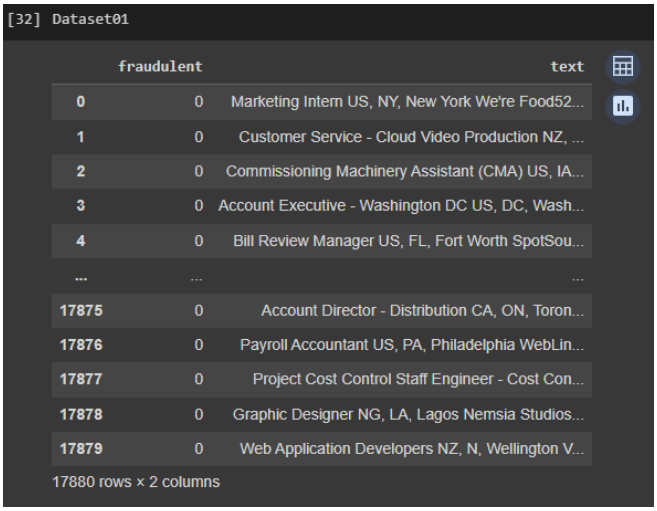
clasificar correctamente las muestras positivas y negativas en diferentes umbrales de clasificación. En resumen, el AUC ofrece una visión general de la capacidad discriminativa del modelo, lo que lo hace útil para comparar y evaluar modelos en una variedad de escenarios de clasificación.

10. **PRC (Precision-Recall Curve)** calcula el área bajo la curva de precisión y recall. La curva de precisión-recall muestra la relación entre la precisión y el recall para diferentes umbrales de clasificación. Un valor alto de AUC-PRC indica un buen equilibrio entre precisión y recall. La métrica PRC-AUC es útil para comparar y evaluar diferentes modelos de clasificación, especialmente en situaciones donde se trabaja con datos imbalanciados, ya que proporciona una medida agregada de la capacidad del modelo para clasificar correctamente las muestras positivas y negativas en diferentes umbrales de clasificación. Un modelo con un buen AUC-PRC tendrá un equilibrio adecuado entre precisión y recall, lo que indica que el modelo está clasificando eficientemente tanto las muestras positivas como las negativas.

III. Tratamiento de Datos

Luego de cargar las librerías y datos que fueron montados en el github personal, se procede a identificar aquellas características que tengan más del 60% de valores nulos y se remueven del dataset para centrarse en las características que tienen una cantidad significativa de datos disponibles y disminuir uso de memoria en Colab.

Con fillna() se rellenan los valores nulos de las columnas restantes con un string vacio para evitar que afecten en el análisis de las demás variables. Con este mismo objetivo, se eliminan algunas columnas que no son útiles para el análisis con la función drop(). Las otras columnas se combinan en una sola columna llamada ‘text’ para facilitar el procesamiento de todas las columnas en un solo lugar.



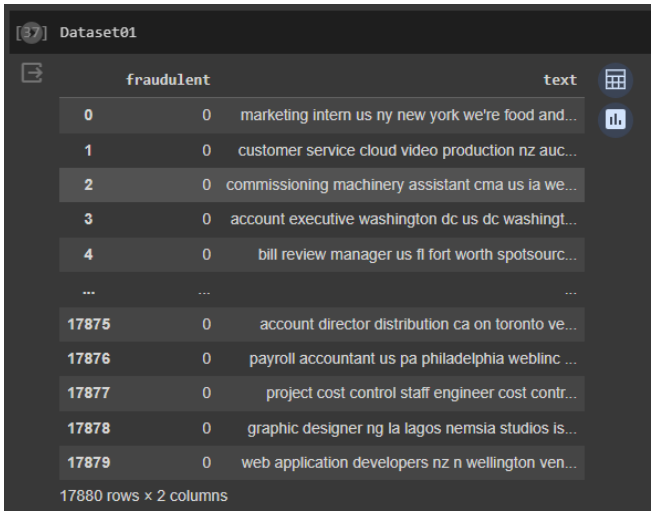
[32] Dataset01

	fraudulent	text
0	0	Marketing Intern US, NY, New York We're Food52...
1	0	Customer Service - Cloud Video Production NZ, ...
2	0	Commissioning Machinery Assistant (CMA) US, IA...
3	0	Account Executive - Washington DC US, DC, Wash...
4	0	Bill Review Manager US, FL, Fort Worth SpotSou...
...	...	...
17875	0	Account Director - Distribution CA, ON, Toron...
17876	0	Payroll Accountant US, PA, Philadelphia WebLin...
17877	0	Project Cost Control Staff Engineer - Cost Con...
17878	0	Graphic Designer NG, LA, Lagos Nemsia Studios...
17879	0	Web Application Developers NZ, N, Wellington V...

17880 rows × 2 columns

Fig 1. Dataset después de la sinterización de columnas.

Continuando, se realiza una limpieza de los datos: los saltos de línea, tabs, los números y los caracteres especiales son remplazados por string vacíos utilizando la función apply() para luego convertir todo en minúsculas con ayuda de métodos de string: lower().



[37] Dataset01

	fraudulent	text
0	0	marketing intern us ny new york we're food and...
1	0	customer service cloud video production nz auc...
2	0	commissioning machinery assistant cma us ia we...
3	0	account executive washington dc us dc washingt...
4	0	bill review manager us fl fort worth spotsourc...
...	...	...
17875	0	account director distribution ca on toronto ve...
17876	0	payroll accountant us pa philadelphia weblinc ...
17877	0	project cost control staff engineer cost contr...
17878	0	graphic designer ng la lagos nemsia studios is...
17879	0	web application developers nz n wellington ven...

17880 rows × 2 columns

Fig 2. Dataset después de la limpieza de caracteres especiales.

Utilizando apply(), se divide la descripción de trabajo en palabras individuales almacenadas en una lista utilizando el método split(). Luego, se realiza una comprensión de lista para filtrar las palabras y seleccionar solo aquellas que no son "stop words".

Las **"stopwords"** o palabras vacías son aquellas palabras sin significado como artículos, pronombres, preposiciones, etc. que son filtradas antes o después del procesamiento de datos en lenguaje natural (texto). Este filtro se realiza con las stopwords de nltk ya que son palabras que no

agregan valor al modelo para enfocar el análisis en las palabras más relevantes y significativas para el modelo.

Por último, se utiliza el método join() para unir las palabras filtradas nuevamente en una sola cadena y se asigna el resultado de vuelta a la columna ‘text’.

Posterior a esto, se comenzó la estandarización de los datos que se le van a entregar del modelo. Con la función one_hot de Keras se aplicó el "one-hot encoding" a cada descripción de trabajo en ‘text’. El **"one-hot encoding"** consiste en convertir esas cadenas de caracteres en bits de 1 y 0 para procesar numéricamente una variable categórica como lo son los caracteres de los strings de la variable ‘text’.

La función one_hot tiene de argumentos el texto a codificar y el tamaño del vocabulario deseado (en este caso 5000) y entregara una secuencia de 1 y 0 que representaran las palabras contenidas en las cadenas de texto.

Con ello, la función **pad_sequences** procede a rellenar la secuencia de "one-hot" (one_hot_x) hasta llegar a una longitud máxima (max_l) y así normalizar todas las secuencias a la misma longitud, logrado mediante el relleno o truncado según sea necesario.

Todo lo anterior con el fin de normalizar vectores numéricos que representan palabras y garantizar que todas las secuencias tengan la misma longitud, lo cual es esencial para la entrada de datos en el modelo.

IV. Métodos supervisados

Los modelos supervisados se emplearon para clasificar las ofertas de trabajo como falsas o verdaderas. Estos modelos se entrenan utilizando un conjunto de datos de entrenamiento que incluye ejemplos etiquetados, donde cada ejemplo consta de una descripción de trabajo y su correspondiente etiqueta de autenticidad.

Se optó por un enfoque de aprendizaje secuencial utilizando redes neuronales recurrentes (RNN) con una capa de LSTM bidireccional. Las RNN son adecuadas para modelar secuencias de datos, como el texto, ya que tienen la capacidad de capturar la dependencia a largo plazo de los datos. La capa LSTM bidireccional permite que el modelo capture la información contextual tanto hacia adelante como hacia atrás en la secuencia de texto, lo que ayuda a capturar relaciones complejas en el lenguaje natural.

Además de la capa de LSTM bidireccional, se incluyeron otras capas en el modelo para mejorar su rendimiento. Se utilizó una capa de embedding para representar las palabras en forma de vectores numéricos densos. Esta capa convierte las palabras en puntos en un espacio vectorial, lo que ayuda al modelo a capturar relaciones semánticas entre las palabras. La capa de embedding se inicializó con pesos aleatorios y se entrenó junto con el resto del modelo durante el proceso de entrenamiento.

Para evitar el sobreajuste, se agregó una capa de Dropout después de la capa LSTM. El Dropout consiste en desactivar aleatoriamente un porcentaje de las neuronas durante el entrenamiento, lo que ayuda a evitar que el modelo se vuelva demasiado dependiente de características específicas y mejora su generalización.

Finalmente, se agregó una capa densa con función de activación sigmoide para realizar la clasificación binaria. La función de activación sigmoide comprime la salida del modelo entre 0 y 1, lo que se interpreta como la probabilidad de que una oferta de trabajo sea falsa. Si la probabilidad supera un umbral determinado (generalmente 0.5), se clasifica como falsa; de lo contrario, se clasifica como verdadera.

Durante el entrenamiento del modelo, se utilizó la pérdida de entropía cruzada binaria como función de pérdida y el optimizador Adam para ajustar los pesos del modelo. El conjunto de datos se dividió en conjuntos de entrenamiento y prueba utilizando la técnica de validación cruzada k-fold para evaluar el rendimiento del modelo en diferentes divisiones de los datos.

La evaluación del modelo se realizó utilizando métricas como precisión, recall, exactitud, área bajo la curva (AUC) y la curva de precisión-recall (PRC). Estas métricas proporcionan una medida completa del rendimiento del modelo, teniendo en cuenta tanto los verdaderos positivos y negativos como los falsos positivos y negativos

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 40, 40)            200000

 bidirectional (Bidirection  (None, 200)               112800
 al)

 dropout (Dropout)           (None, 200)               0

 dense (Dense)               (None, 1)                 201

=================================================================
Total params: 313001 (1.19 MB)
Trainable params: 313001 (1.19 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

Fig 3. Resumen de las características del modelo.

Por último, se probó el modelo supervisado desarrollado utilizando los datos preprocesados. Para ello, se filtró el dataset respecto a la columna 'fraudulent', así aquellos que tenían un 0 en esa columna eran verdaderas y un 1 para las fraudulentos.

Después del filtro, con la librería random se eligió aleatoriamente uno de los posts para aplicarles la funcion predict que recibe como argumentos el modelo creado y la propuesta de trabajo ya filtrada y limpiada de ruido.

```
def predict(m,fake_job_post):
  one_hot_input = one_hot(fake_job_post,5000)
  embedded = pad_sequences([one_hot_input],maxlen=max_l)

  pred = m.predict(embedded)
  #print(pred)

  if(pred > 0.5):
    return "FAKE"
  else:
    return "TRUE"
```

Fig 4. Función para la prueba de las ofertas de trabajo.

Se puede ver, entonces, que la función entregara ‘FALSE’ para aquellos que cumplan la condición para ser falsos y ‘TRUE’ de lo contrario.

En la primera prueba, el modelo realizó una predicción positiva indicando que la descripción de trabajo era auténtica. De la misma forma, predijo que la segunda era falsa y estas conclusiones se mantuvieron a medida que la librería random cambiaba de post.

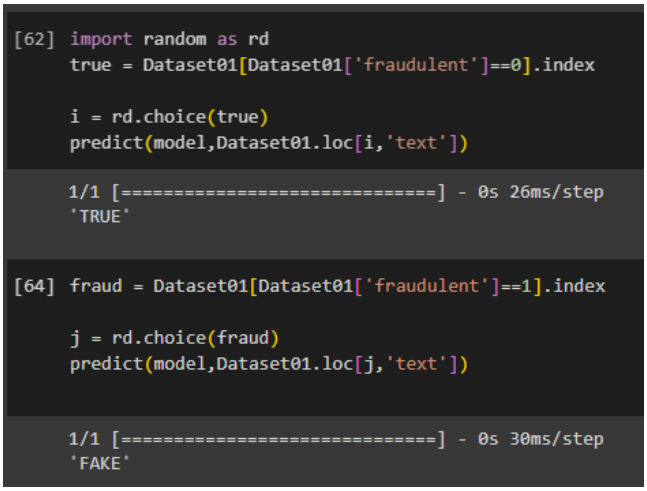


Fig 5. Prueba del modelo.

Esto sugiere que el modelo pudo capturar las características relevantes y discriminatorias asociadas con descripciones de trabajo legítimas lo que respalda la efectividad del modelo en la detección de descripciones de trabajo genuinas.

V. Métodos no supervisados

La exploración de modelos no supervisados fue excluida, ya que la atención se centró en el aprendizaje supervisado para la clasificación de ofertas de empleo como auténticas o fraudulentas. Sin embargo, en fases posteriores de desarrollo, sería beneficioso considerar la implementación de enfoques no supervisados, como la aplicación de técnicas de agrupación o la detección de anomalías, para identificar posibles patrones o comportamientos atípicos en los datos y realizar un modelo respecto a ellas. Esta perspectiva adicional podría ofrecer conclusiones que complementen el análisis supervisado, mejorando así la capacidad del sistema para distinguir de manera más efectiva entre ofertas de trabajo legítimas y las fraudulentas.

VI. Retos y consideraciones de despliegue

Cuando se considera el despliegue del modelo, es importante tener en cuenta los siguientes desafíos y consideraciones:

Interpretabilidad: Es importante tener en cuenta los desafíos que implica en términos de equilibrio entre precisión e interpretabilidad para modelos de aprendizaje profundos como este, así como en la definición de estándares para evaluar la interpretabilidad de diferentes modelos. Así, la búsqueda de

métodos y enfoques que equilibren la complejidad inherente de estos modelos con la necesidad de comprensión humana sigue siendo un campo de investigación esencial para avanzar en la adopción segura y efectiva de este modelo

Recolección de datos: La calidad de los datos tiene un impacto directo en el rendimiento y la eficacia de los modelos de NLP, ya que estos modelos dependen en gran medida de patrones y estructuras lingüísticas aprendidas a partir de ejemplos específicos. La diversidad en los datos garantiza que el modelo se exponga a una amplia variedad de contextos lingüísticos, dialectos, registros y temas. Esto permite que el modelo aprenda patrones más generales y aplicables en lugar de depender en exceso de características específicas de un conjunto de datos limitado para así poder acertar en las ofertas de trabajo fraudulentas.

Actualización temporal del modelo: La relevancia temporal también es un factor crucial en la recolección de datos para NLP, ya que el lenguaje natural y los fraudes en internet están en constante evolución. Los conjuntos de datos actualizados reflejan las tendencias y cambios en el uso del lenguaje, lo que es esencial para mantener la relevancia y la eficacia de los modelos de NLP a lo largo del tiempo.

Escalabilidad: Es esencial para abordar conjuntos de datos masivos y variados. A medida que las aplicaciones de lenguaje natural se expanden, los modelos deben ser capaces de manejar grandes cantidades de información de manera eficiente. Los modelos deben procesar textos extensos, conjuntos de datos diversificados y lidiar con la complejidad inherente al lenguaje natural en tiempo real. Para lograr esto, se pueden considerar técnicas de escalabilidad, como el uso de GPU o distribución del modelo en clústeres.

VII. Conclusiones

En resumen, el uso de modelos supervisados basados en redes neuronales recurrentes, como el modelo de LSTM bidireccional implementado en este proyecto, ha demostrado ser altamente efectivo en la tarea de clasificación de ofertas de trabajo como falsas o verdaderas mediante el análisis de información textual.

El preprocesamiento de datos juega un papel crítico en el éxito del modelo, destacando la importancia de abordar aspectos como la gestión de datos faltantes, la limpieza del texto, la eliminación de

caracteres especiales y numéricos, así como la adecuada tokenización y codificación. Estas etapas son esenciales para garantizar la calidad de los datos de entrada y, por ende, la precisión del modelo.

La evaluación del rendimiento a través de métricas como precisión, recall, exactitud, AUC y la curva PRC proporciona una visión completa y detallada de la capacidad del modelo para distinguir entre ofertas de trabajo falsas y verdaderas. Además, las curvas de aprendizaje emergen como herramientas valiosas para identificar problemas de sobreajuste o desajuste, permitiendo ajustar los hiperparámetros y optimizar el rendimiento del modelo.

El despliegue de modelos supervisados basados en lenguaje natural plantea desafíos importantes, entre ellos la interpretabilidad del modelo y la calidad de los datos de entrenamiento. La transparencia y el uso de conjuntos de datos equilibrados y representativos son fundamentales para garantizar la confiabilidad del modelo en situaciones del mundo real.

La exploración de modelos no supervisados, como técnicas de agrupación o detección de anomalías, surge como una estrategia complementaria valiosa que puede ofrecer perspectivas adicionales en la identificación de ofertas de trabajo falsas.

En conclusión, la combinación de modelos supervisados, un riguroso preprocesamiento de datos, métricas de evaluación sólidas y técnicas de ajuste de hiperparámetros proporciona un enfoque prometedor y holístico para abordar la detección de ofertas de trabajo falsas. No obstante, se destaca la necesidad de considerar cuidadosamente los desafíos asociados con el despliegue de estos modelos, asegurando así su eficacia y utilidad en entornos del mundo real.