# FAKE JOB POSTING PREDICTION

**POR**

María Isabel Rúa Velez

**MATERIA**

Introducción a la Inteligencia Artificial

**PROFESOR**

Raul Ramos Pollan


**UNIVERSIDAD DE ANTIOQUIA**
FACULTAD DE INGENIERÍA
MEDELLÍN
2023

# Contenido

# 1. Introducción

En el ámbito de la detección de fraudes en publicaciones de trabajo, es fundamental contar con herramientas efectivas que puedan identificar y clasificar las descripciones de trabajo fraudulentas. Estas publicaciones falsas no solo pueden causar daños económicos a las personas en busca de empleo, sino que también pueden afectar negativamente la reputación de las empresas y la confianza en los sitios de empleo en línea.

En este contexto, se plantea el presente informe científico con el objetivo de desarrollar un modelo de predicción que aprenda de las descripciones de trabajo y pueda identificar aquellas que son fraudulentas. Para lograr este objetivo, se utiliza un conjunto de datos de propuestas de trabajo recopilado de la plataforma Kaggle, conocido como "Real / Fake Job Posting Prediction". Este dataset proporciona una amplia variedad de características sobre cada publicación de trabajo, incluyendo título, ubicación, departamento, salario, perfil de la compañía, descripción del trabajo, requisitos, beneficios, entre otros.

El enfoque principal de este informe es utilizar técnicas de procesamiento de lenguaje natural para analizar y preprocesar las descripciones de trabajo contenidas en el dataset. Mediante la aplicación de métodos de limpieza y transformación de texto, se busca reducir el ruido y las características irrelevantes, y así obtener un conjunto de datos adecuado para el posterior análisis y modelado.

En las secciones siguientes se describe el proceso de preprocesamiento aplicado al dataset, incluyendo la exploración y eliminación de características con valores nulos, la combinación de características relevantes en una columna, y la limpieza de caracteres especiales y números. Además, se discutirán los avances alcanzados hasta el momento, detallando las operaciones realizadas y su impacto en la calidad de los datos.

# 2. Planteamiento del problema

## 2.1. Dataset

Vamos a usar el dataset de kaggle **Real / Fake Job Posting Prediction** (https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction), que tiene 18.000 numero de muestras, con las siguientes características:

- **Job_id**
- **title -** Título de la publicación
- **location -** Ubicación del lugar de trabajo
- **department -** Departamento de trabajo
- **salary_range -** Rango de salario
- **company_profile -** Perfil de la compañía
- **description -** Descripción del trabajo
- **requirements -** Requerimientos para el  trabajo
- **benefits -** Beneficios con el empleo
- **telecommuting -** 0 o 1 si se realiza teletrabajo
- **has_company_logo -** Si tiene logo de la empresa ( 0 o 1)
- **has_questions -** Si tiene preguntas (0 o 1)
- **employment_type -** Tipo de empleo (tiempo completo, medio tiempo, entre otros)
- **required_experience -** Experiencia requerida
- **required education -** Educación requerida
- **industry -** industria en donde se desarrolla el trabajo
- **function -** Función del trabajo
- **fraudulent -** Característica de salida ( 0 o 1)

## 2.2. Métricas

Las métricas seleccionadas para evaluar el desempeño del modelo son:

Matriz de confusión: La matriz de confusión es una herramienta útil para evaluar el rendimiento de un modelo de clasificación, incluido el caso de la detección de publicaciones de trabajo fraudulentas. Esta matriz muestra de manera resumida las predicciones del modelo en comparación con las clases reales.

**True Positives (tp):** Esta métrica cuenta el número de casos en los que el modelo predijo correctamente una publicación de trabajo fraudulenta como fraudulenta. Es decir, es el número de casos positivos correctamente clasificados.

**False Positives (fp):** Esta métrica cuenta el número de casos en los que el modelo predijo incorrectamente una publicación de trabajo genuina como fraudulenta. Representa los casos negativos incorrectamente clasificados.

**True Negatives (tn):** Esta métrica cuenta el número de casos en los que el modelo predijo correctamente una publicación de trabajo genuina como genuina. Es el número de casos negativos correctamente clasificados.

**False Negatives (fn):** Esta métrica cuenta el número de casos en los que el modelo predijo incorrectamente una publicación de trabajo fraudulenta como genuina. Representa los casos positivos incorrectamente clasificados.

**Binary Accuracy:** Esta métrica calcula la precisión global del modelo, es decir, la proporción de predicciones correctas en relación con el total de predicciones realizadas.

**Precisión:** Esta métrica calcula la proporción de verdaderos positivos en relación con el total de predicciones positivas realizadas. Mide la capacidad del modelo para evitar falsos positivos.

Recall: También conocido como Sensibilidad o Tasa de Verdaderos Positivos (TPR), esta métrica calcula la proporción de verdaderos positivos en relación con el total de casos positivos reales. Mide la capacidad del modelo para detectar correctamente los casos positivos.

**AUC (Area Under the Curve):** Esta métrica evalúa la capacidad de discriminación del modelo al calcular el área bajo la curva ROC (Receiver Operating Characteristic). Cuanto mayor sea el valor de AUC, mejor será la capacidad del modelo para distinguir entre clases positivas y negativas.

**AUC (Area Under the Curve) - PRC (Precision-Recall Curve):** Esta métrica 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.

Estas métricas proporcionan una evaluación exhaustiva del rendimiento del modelo, considerando tanto la capacidad de clasificación correcta de casos positivos y negativos como la precisión y el equilibrio entre precisión y recall. Al utilizar estas métricas, se obtendrá una comprensión más completa y precisa del desempeño del modelo en la detección de publicaciones de trabajo fraudulentas.

## 2.3. Análisis de variable objetivo

La variable objetivo es "**fraudulent**". Esta variable representa la categoría o etiqueta que indica si una publicación de trabajo es fraudulenta (1) o genuina (0). El objetivo principal del análisis y modelado es poder identificar y clasificar correctamente las

publicaciones de trabajo fraudulentas, ya que estas representan un problema significativo en el ámbito laboral.

Por lo tanto, al establecer la variable objetivo como "**fraudulent**", se busca construir un modelo de predicción capaz de identificar y clasificar de manera precisa las publicaciones de trabajo que contienen información fraudulenta. Esto permitirá a las empresas y a los solicitantes de empleo tomar decisiones informadas y evitar situaciones perjudiciales.

## 2.4. Datos faltantes

Al realizar la prelimpieza del dataset nos dimos cuenta que tenemos ciertas características deberían ser eliminadas por la cantidad de información faltante, las cuales son 'department', 'salary_range', 'benefits' y 'required_education'. El código presentado aborda esta situación al realizar operaciones de limpieza y preprocesamiento de los datos.

En primer lugar, se realiza una exploración de los datos para identificar las características que tienen más del 60% de valores nulos. En este caso, las variables mencionadas cumplen con este criterio y se eliminan del conjunto de datos original. Esta eliminación se realiza con el objetivo de reducir el tamaño del conjunto de datos y centrarse en las características que tienen una cantidad significativa de datos disponibles.

Posteriormente, para aquellas características que permanecen en el dataset se procede a rellenar los valores nulos en el conjunto de datos con una cadena vacía utilizando la función fillna(). Esta estrategia se aplica para evitar que los valores nulos afecten el análisis posterior del conjunto de datos. Al rellenarlos con una cadena vacía, se permite que las operaciones de limpieza y preprocesamiento se realicen de manera adecuada y se evita la interrupción de los flujos de análisis.


*Figura 1. Distribución de datos faltantes en el dataset*

# 3. Tratamiento de datos

Se realiza una exploración de los datos para identificar aquellas características que tengan más del 60% de valores nulos y se eliminan del conjunto de datos original. Esto se realiza para reducir el tamaño del conjunto de datos y centrarse en las características que tienen una cantidad significativa de datos disponibles.

Después, se rellenan los valores nulos con una cadena vacía utilizando la función fillna(). Esto se hace para evitar que los valores nulos afecten el análisis posterior del conjunto de datos.

Luego, se eliminan algunas columnas que no son útiles para el análisis posterior utilizando la función drop(). En este caso, las columnas que se eliminan son 'telecommuting', 'has_company_logo', 'has_questions' y 'job_id'.

A continuación, se combinan varias columnas del conjunto de datos en una sola columna llamada 'text'. Esto se hace para tener todas las características relevantes en un solo lugar y facilitar el preprocesamiento posterior del texto.

Después, se realizan una serie de operaciones de limpieza en el texto. En primer lugar, se reemplazan los saltos de línea y espacios que son tabs con espacios en blanco. Luego, se eliminan los números y los caracteres especiales utilizando expresiones regulares y la función apply(). Finalmente, se convierte todo el texto en minúsculas utilizando la función apply().

Con este tratamiento de datos se intenta limpiar y preprocesar los datos antes de analizarlos y modelarlos. La eliminación de características con muchos valores nulos, la combinación de características relevantes en una sola columna y la eliminación de caracteres especiales y números son técnicas comunes para limpiar el texto antes del análisis posterior.

Después, se aplica el tratamiento de datos a la columna 'text' del conjunto de datos. Se utiliza la función apply() para iterar sobre cada descripción de trabajo en la columna 'text'. Dentro de la función, se divide la descripción de trabajo en palabras individuales utilizando el método split(). Luego, se realiza una comprensión de lista para filtrar las palabras y seleccionar solo aquellas que no se encuentran en el conjunto de "stop words". 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'.

Este proceso de eliminación de "stopwords" es común en el procesamiento de texto para reducir el ruido y mejorar la calidad de los datos antes del análisis posterior. Al eliminar las palabras vacías como "a", "the", "is", entre otras, se puede enfocar el análisis en las palabras más relevantes y significativas para el modelo.

Posterior a esto, se utiliza la función one_hot de Keras para aplicar el "one-hot encoding" a cada descripción de trabajo en la columna 'text' del conjunto de datos 'jobs_DS'. La función one_hot toma dos argumentos: el texto a codificar y el tamaño del vocabulario deseado. En este caso, se establece el tamaño del vocabulario en 5000. La función devuelve una lista de códigos numéricos únicos que representan las palabras en cada descripción de trabajo.

Luego, se utiliza la función pad_sequences para aplicar el relleno de secuencia a los datos codificados con "one-hot". La función toma dos argumentos: la lista de códigos "one-hot" (one_hot_x) y la longitud máxima de secuencia deseada (max_l). El resultado es un array bidimensional en el que todas las secuencias tienen la misma longitud, lograda mediante el relleno o truncado según sea necesario.

El uso de "one-hot encoding" y el relleno de secuencia es común en el procesamiento de texto antes de alimentar los datos a modelos de aprendizaje automático, especialmente a redes neuronales. Esto permite representar las palabras como vectores numéricos y garantizar que todas las secuencias tengan la misma longitud, lo cual es esencial para la entrada de datos en el modelo.

# 4. Métodos supervisados

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

El enfoque elegido fue un modelo 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 conjuntamente 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.

Se procedió a probar el modelo supervisado desarrollado utilizando los datos preprocesados. Durante las pruebas, se realizaron dos predicciones con el objetivo de evaluar el rendimiento del modelo. En la primera prueba, se utilizó una descripción de trabajo genuina y, en la segunda prueba, se empleó una descripción de trabajo falsa.

En la primera prueba, el modelo realizó una predicción positiva, indicando que la descripción de trabajo era auténtica. Esto sugiere que el modelo pudo capturar las características relevantes y discriminatorias asociadas con descripciones de trabajo legítimas. Esta predicción alentadora respalda la efectividad del modelo en la detección de descripciones de trabajo genuinas.

La prueba que usamos fue:

> *´Software Developer*
>
> *Job Description: We are seeking a highly skilled software developer to join our development team. You will be responsible for designing, developing, and maintaining high-quality software applications. Strong knowledge of programming languages such as Python and experience in web application development are required. We also value database skills and familiarity with development frameworks such as Django. You will work closely with our team of engineers to create innovative and scalable technology solutions. If you are passionate about coding and enjoy tackling technical challenges, this position is for you!*
>
> *Job Requirements:*
>
> *Demonstrable experience in software development using Python. Strong knowledge of programming languages such as Java, C++, or Ruby. Experience in web application development using frameworks like Django or Flask. Proficiency in relational databases such as MySQL or PostgreSQL. Problem-solving skills and ability to work in a team. Ability to quickly learn new technologies and adapt to changing environments. We offer a dynamic and challenging work environment, professional growth opportunities, and a highly collaborative team. If you are seeking a new challenge in the field of software development, we look forward to receiving your application!"*

*You can use this job description as input in your model to evaluate whether it is classified as true or false. Remember that the model should have been trained on a labeled dataset in order to make accurate predictions.´*

El modelo nos entregó esta predicción:

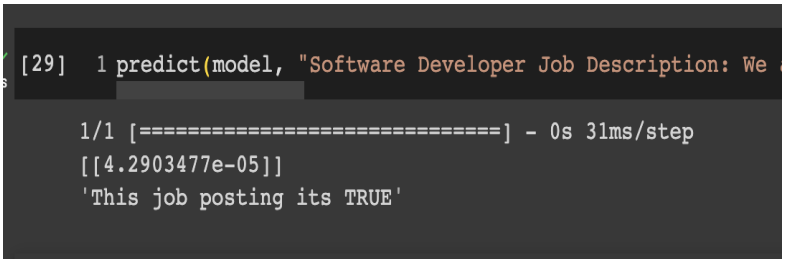


*Figura 2: predicción verdadera*

En la segunda prueba, el modelo emitió una predicción negativa, lo que implica que la descripción de trabajo fue considerada falsa. Este resultado es muy prometedor, ya que indica que el modelo también puede identificar patrones y señales que sugieren la presencia de descripciones de trabajo fraudulentas. Esto demuestra la capacidad del modelo supervisado para discernir entre descripciones de trabajo genuinas y falsas.

La prueba que usamos fue:

*´As coronavirus_are increasing, so have the number of companies asking their employees to stay at home As travelers cancel flights and stocks fall, a global health pandemic now has become a global economic crisis In any health pandemic, our first concern us what the health of those affected, COVID-19 has brought about many more death worldwide and more and more cases are being confirmed daily counties the World But unfortunately, the economic impacts also have dramatic effects on the wellbeing of families and communities Although traditional forms of tutoring, including face.to.face lessons and residential placements remain as popular as ever, has also been gaming traction over the last few years With a distinct use in online tuition websites, many tutors have begun to work exclusively online and some schools have even started offering online programs As the world comes together to solve this coronavirus pandemic, the demand for online tuition has also become more and more In demand Click here and find out how to work from home as an online tutor Here Best Regards Emmanuel´*

El modelo nos entregó esta predicción:

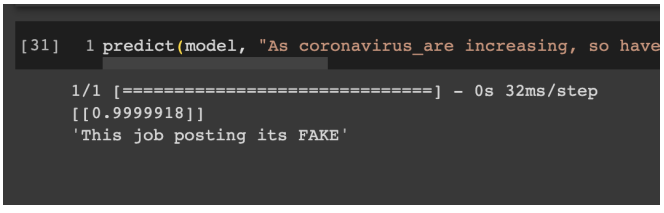


*Figura 3: predicción falsa*

## 5. Modelos no supervisados

No se exploraron modelos no supervisados en este informe, ya que el enfoque se centró en el aprendizaje supervisado para clasificar las ofertas de trabajo como falsas o verdaderas. Sin embargo, en futuras iteraciones, se podrían explorar técnicas no supervisadas como la agrupación o la detección de anomalías para identificar patrones o comportamientos atípicos en los datos.

## 6. Retos y Consideraciones de Despliegue

Al considerar el despliegue del modelo, es importante tener en cuenta los siguientes desafíos y consideraciones:

Interpretabilidad: Los modelos basados en aprendizaje profundo pueden ser difíciles de interpretar debido a su complejidad. Es importante evaluar la transparencia del modelo y garantizar que las decisiones tomadas sean comprensibles y justificables.

Recolección de datos: La calidad de los datos utilizados para entrenar el modelo es fundamental. Se debe garantizar que el conjunto de datos esté equilibrado y represente adecuadamente las características de las ofertas de trabajo falsas y verdaderas.

Escalabilidad: El modelo debe ser capaz de manejar grandes volúmenes de datos y realizar predicciones en tiempo real, al tener un gran volumen de datos suele ser demasiado pesado para procesar y entrenar el modelo. Se deben considerar técnicas de escalabilidad, como el uso de GPU o distribución del modelo en clústeres

## 7. Conclusiones

El uso de modelos supervisados basados en redes neuronales recurrentes, como el modelo de LSTM bidireccional utilizado en este proyecto, ha demostrado ser efectivo en la clasificación de ofertas de trabajo como falsas o verdaderas utilizando información de texto.

El preprocesamiento de datos desempeña un papel crucial en el rendimiento del modelo. El adecuado tratamiento de los datos faltantes, limpieza del texto, eliminación de caracteres especiales y números, y la tokenización y codificación adecuada son etapas fundamentales para lograr buenos resultados.

Las métricas de evaluación, como precisión, recall, exactitud, área bajo la curva (AUC) y la curva de precisión-recall (PRC), brindan una visión completa del rendimiento del modelo y su capacidad para distinguir entre ofertas de trabajo falsas y verdaderas.

Las curvas de aprendizaje son herramientas útiles para identificar problemas de sobreajuste o desajuste en el modelo. Estas curvas permiten ajustar los hiper parámetros del modelo y optimizar su rendimiento.

El despliegue de modelos supervisados basados en lenguaje natural presenta desafíos, como la interpretabilidad del modelo y la calidad de los datos de entrenamiento. Es importante garantizar la transparencia del modelo y utilizar conjuntos de datos equilibrados y representativos.

La exploración de modelos no supervisados, como la agrupación o la detección de anomalías, puede complementar los modelos supervisados y brindar una perspectiva adicional en la identificación de ofertas de trabajo falsas.

En general, el uso de modelos supervisados, junto con un adecuado preprocesamiento de datos, métricas de evaluación y técnicas de ajuste de hiper parámetros, proporciona un enfoque prometedor para la detección de ofertas de trabajo falsas. Sin embargo, es importante considerar los desafíos y consideraciones asociados con el despliegue de estos modelos para garantizar su eficacia en escenarios del mundo real.