

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial
15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física
Maestría en Ciencia de Datos



Sistema de clasificación de ordenes de supresión

TRABAJO RECEPCIONAL que para obtener el **GRADO** de
Maestro en Ciencia de Datos

Presenta:
Thaitiel Jair Durán Valencoa

Director:
Dr. Jaime Emmanuel Alcalá Temores

Tlaquepaque, Jalisco, 22 de mayo de 2024

Sistema de clasificación de ordenes de supresión

Thaitiel Jair Durán Valenco

Resumen

El sistema de órdenes de supresión policial (PSOS, por sus siglas en inglés) comienza como un proyecto Call for Code en colaboración con la Asociación Nacional de Abogados de Defensa Criminal (NACDL) para construir una base de datos utilizando inteligencia artificial para ayudar a rastrear las órdenes de supresión judiciales. El alcance del proyecto se ha ampliado para rastrear cualquier mala conducta policial. El resultado final utilizaría la inteligencia artificial, en específico Procesamiento de Lenguaje Natural (*Natural Language Processing*, NLP por sus siglas en inglés) para eliminar cientos de miles de horas de trabajo manual en el intento de crear y mantener una base de datos que identifique el comportamiento problemático de la policía. Este trabajo se centró en un problema de clasificación de texto, cuyo objetivo principal fue explorar cuáles son los métodos de aprendizaje automático supervisado más efectivos para abordarlo.

Tabla de Contenidos

	Página
1 Introducción	13
1.1. Contexto	13
1.2. Justificación	13
1.3. Problema	14
1.4. Objetivos	15
1.4.1. Objetivo general	15
1.4.2. Objetivos específicos	15
2 Metodología	17
2.1. Descripción de los datos	17
2.2. Análisis exploratorio	18
2.2.1. Limpieza de datos	18
2.2.2. Exploración de los datos	18
2.2.3. Ingeniería de características	18
2.3. Descripción de los modelos	19
2.3.1. Modelos de NLP	19
2.3.2. Modelos de Clasificación	22
2.4. Descripción de las métricas	23
2.5. Descripción de los experimentos o simulaciones	24
3 Resultados y discusión.	27
3.1. Resultados	27
3.2. Discusión	27
4 Conclusiones y trabajo futuro.	29
4.1. Conclusiones	29
4.2. Trabajo futuro	29
4.2.1. Modelos de clasificación convencionales	29
4.2.2. Modelo de clasificación basado en arquitectura <i>Transformers</i>	30
Bibliografía	32

Índice de figuras

	Página
2.1. Histograma de la variable <code>general_cap_classification</code> . . .	19
2.2. Arquitectura de <i>Word2Vec</i> . (tomado de [1]).	20
2.3. Arquitectura de <i>back propagation</i> . (tomado de [2]).	20
2.4. Arquitectura de <i>Transformers</i> . (tomado de [3]).	21
2.5. Ejemplo ilustrativo de <i>Random Forest</i> . (tomado de [4]). . .	22
2.6. Ejemplo ilustrativo de <i>Xgboost</i> . (tomado de [5]).	23
2.7. Ejemplo de Matriz de Confusión. (tomado de [6]).	24
2.8. Diagrama de flujo.	25

Índice de tablas

	Página
3.1. Resultados obtenidos.	27

Dedicado a mi familia, por ser mi constante fuente de apoyo y motivación. A mi pareja, por caminar a mi lado con paciencia, comprensión y amor. A mis amigos, por las risas, los consejos y los momentos de respiro en medio del caos. A mis compañeros, por acompañarme en este viaje académico. Finalmente, a mi mismo, por la tenacidad y continuo deseo de aprender y crecer.

1 Introducción

En este capítulo se presenta el contexto del objeto de estudio, la justificación del objeto de estudio, la definición del problema y los objetivos generales y específicos.

1.1 Contexto

En los Estados Unidos, la mala conducta policial representa un problema persistente y corrosivo. La ausencia de datos transparentes y precisos dificulta enormemente la evaluación y la identificación de patrones en los problemas de comportamiento policial.

En las raras ocasiones en que los oficiales son despedidos por mala conducta, el veinte por ciento son recontratados por otra agencia en el mismo estado en los tres años posteriores [7].

Incluso los tipos más graves de mala conducta, como los homicidios policiales, se ha demostrado que están subreportados en más del 50% [8].

El público tiene limitaciones o se le impide acceder a los archivos disciplinarios policiales en 36 estados, por lo que la magnitud del problema es desconocida, ya que los departamentos pueden proteger el acceso a sus datos, lo que hace imposible evaluar sus mecanismos actuales de rendición de cuentas.

En este estudio, utilizaremos ejemplos concretos de reportes de abuso policial como datos de entrada para clasificar de una manera que permita dar un seguimiento adecuado al reporte. Así, mantendremos una base de datos actualizada que permitirá, a futuro, identificar patrones de conducta y determinar las necesidades y áreas de mejora en los procesos.

1.2 Justificación

El uso de inteligencia artificial (IA) en las firmas legales se ha vuelto cada vez más común. Según el estudio anual "*Impact of Legal Analytics*" realizado por Lex Machina y LexisNexis, *Legal Analytics* se ha consolidado como un componente esencial y perdurable en la práctica

legal, y su adopción puede fomentar un entorno legal más informado y estratégico [9].

El uso de IA en la clasificación de documentos legales (eDiscovery) ha demostrado un incremento significativo en la eficiencia de tareas que tradicionalmente consumen mucho tiempo, como la revisión de documentos. Esto permite que los profesionales legales se concentren más en aspectos estratégicos de su trabajo, al tiempo que aumenta la eficiencia y reduce los costos operativos, mejorando los procesos y la experiencia del cliente [10].

En la industria legal, se ha comprobado que los algoritmos de IA pueden analizar grandes cantidades de datos con un alto grado de precisión, reduciendo así el riesgo de errores humanos en tareas como la revisión de documentos y la extracción de datos [10].

Considerando los beneficios de la IA y el análisis de texto para la clasificación de delitos de autoridad, la implementación de estas tecnologías podría ayudar a optimizar los procesos y mejorar el seguimiento de estos casos, proporcionando una perspectiva más estratégica para el reconocimiento de patrones y la implementación de acciones preventivas así como agilizar los procesos para traer justicia expédita y acompañamiento a las víctimas.

1.3 Problema

Existen deficiencias en la disponibilidad y precisión de datos derivado de una clasificación manual de denuncias de abuso de autoridad policial en los Estados Unidos, estas deficiencias impide identificar y abordar patrones de comportamiento inapropiado dentro de las instituciones pertinentes.

La problemática se subraya aún más por el hecho de que los actos más serios de mala conducta, como los homicidios cometidos por la policía, suelen reportarse un 50 % menos de lo que realmente ocurren. Además, el público tiene un acceso limitado o nulo a los archivos disciplinarios de la policía en 36 estados, lo que restringe aún más la capacidad de los ciudadanos para realizar un escrutinio efectivo y abogar por la rendición de cuentas.

A pesar de las graves consecuencias de la mala conducta policial, incluso cuando los oficiales son despedidos por tales acciones, existe un 20 % de probabilidad de que sean recontratados por otra agencia en el mismo estado dentro de tres años [7].

El alcance completo del problema sigue siendo desconocido debido a que muchos departamentos de policía pueden ocultar el acceso a sus datos, lo que impide una evaluación adecuada de los mecanismos de rendición de cuentas existentes. Esta opacidad no solo protege a los oficiales que incurrir en conductas indebidas, sino que también

compromete la confianza pública en el sistema de justicia penal y obstaculiza los esfuerzos significativos hacia reformas estructurales y mejoras en las prácticas policiales.

1.4 *Objetivos*

1.4.1 *Objetivo general*

Realizar un modelo para categorizar el tipo de abuso policial utilizando reportes del estado de Philadelphia en Estados Unidos, usando métodos de NPL y Aprendizaje Automático (ML).

1.4.2 *Objetivos específicos*

1. Crear una solución asistida por inteligencia artificial combina métodos de NPL y ML para analizar texto y asignar un conjunto de etiquetas o categorías predefinidas basadas en su contenido.
2. Ayudar a extraer patrones de comportamiento de un modelo de clasificación para comprender los patrones de mala conducta.
3. Comparar modelos de clasificación que permita clasificar el tipo de abuso policial de la manera más efectiva.

2 Metodología

En este capítulo se presenta en detalle el desarrollo metodológico que incluye descripción de los datos, análisis exploratorio, descripción de los modelos, descripción de las métricas y descripción de los experimentos.

2.1 Descripción de los datos

Los conjuntos de datos que se utilizará consisten en un archivo en formato CSV con 3400 registros de ordenes de abuso policial proporcionadas por *National Association of Criminal Defense Lawyers* (NACDL) [11] que se utilizará para el entrenamiento del modelo.

Cada registro contiene las siguientes variables:

- `complain_id`: El identificador único del registro.
- `date_received`: La fecha en la que la denuncia fue recibida.
- `district_occurrence`: Variable categórica con el número del distrito en donde ocurrió el evento.
- `general_cap_classification`: Variable categórica que contiene tipo de abuso denunciado.
- `summary`: El resumen de los hechos reportados.

La variable `general_cap_classifications` pueden ser una de las siguientes categorías:

- UNPROFESSIONAL CONDUCT
- LACK OF SERVICE
- PHYSICAL ABUSE
- DOMESTIC
- VERBAL ABUSE
- DEPARTMENTAL VIOLATIONS
- NON-INVESTIGATORY INCIDENT
- HARASSMENT
- CRIMINAL ALLEGATION
- SEXUAL CRIME/MISCONDUCT
- CIVIL RIGHTS COMPLAINT
- OTHER MISCONDUCT

■ FALSIFICACION

Como conjunto de datos de prueba se tienen 57 archivos en formato PDF los cuales se utilizarán para obtener el `summary` y predecir el `general_cap_classification`.

2.2 *Análisis exploratorio*

En esta sección se incluyen un análisis de los datos utilizando tablas y gráficos para justificar la toma de decisiones. Incluye también las transformaciones o cualquier ingeniería de características aplicada a los datos.

2.2.1 *Limpieza de datos*

Se identifica si hay datos faltantes en `general_cap_classification` y `summary`, ya que estas columnas son las de interés para el modelo de clasificación a realizar.

Se localizaron dos registros con datos faltantes (ID 2809 y 2970), siendo una cantidad despreciable, estos registros serán eliminados.

2.2.2 *Exploración de los datos*

Se ha identificado que la distribución de registros por cada categoría de `general_cap_classification` muestra un desbalance significativo en la base de datos. 2.1 En respuesta a esto, se procederá a entrenar el modelo utilizando una muestra balanceada de los datos para asegurar una evaluación más equitativa y representativa del modelo.

Actualmente, no es viable agrupar los valores de menor frecuencia debido a la diversidad en la naturaleza de estos y la gravedad de los delitos que las categorías representan. Esta decisión busca preservar la integridad y la precisión en el análisis y las predicciones del modelo.

2.2.3 *Ingeniería de características*

1. Se utiliza la función `Categorical` de `pandas` para transformar la variable `general_cap_classification`, con el objetivo de tener una variable categórica numérica a predecir.
2. Se utilizan modelos de NLP como `word2vec` [1] y `Transformers` [3] para la variable predictora `summary`.
3. Se utilizará Análisis de Componentes Principales (*Principal Component Analysis*, PCA por sus siglas en inglés) como técnica de reducción de componentes.

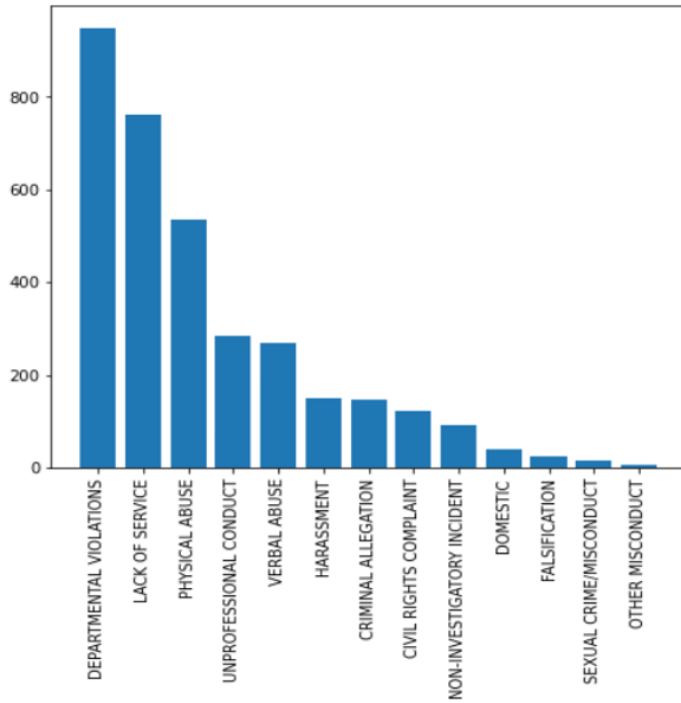


Figura 2.1: Histograma de la variable `general_cap_classification`.

2.3 Descripción de los modelos

Con el objetivo de lograr la mejor predicción se proponen dos modelos de NLP para el procesamiento de los predictores, 1 modelo opcional de reducción de dimensionalidad y 3 modelos de clasificación para la variable de predicción, sumando un total de 12 experimentos distintos para resolver este problema.

2.3.1 Modelos de NLP

Word2Vec es una técnica de NLP que propone transformar las palabras en vectores que tienen significado semántico y sintáctico. Fue desarrollado por Google en Septiembre del 2013 y consiste en la integración de 2 arquitecturas, *Continuous Bag of Words (CBOW)* y *Skip-Gram* [1].

Las palabras son transformadas a vectores numéricos los cuales posteriormente serán las entradas de las redes neuronales de la arquitectura.

Continuous bag of words consiste en una red neuronal que usa el contexto para predecir la palabra objetivo mientras *Skip-gram* usa una palabra para obtener el contexto [2].

Los vectores son ajustados de acuerdo al de ambas redes neuronales

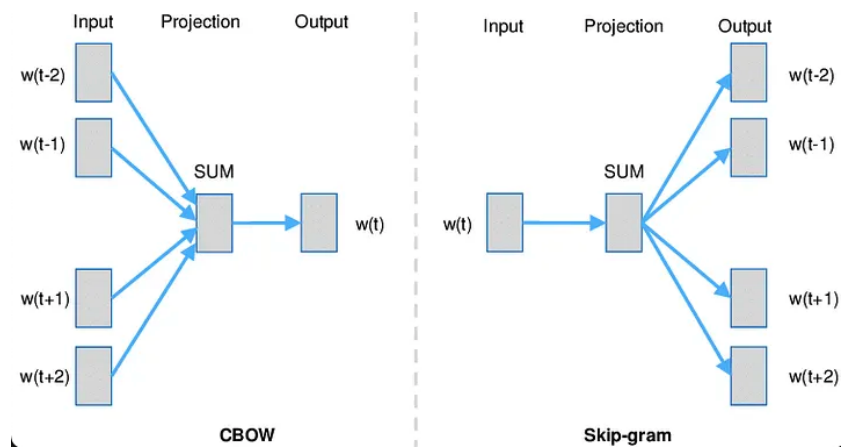


Figura 2.2: Arquitectura de *Word2Vec*. (tomado de [1]).

usando *back propagation*.

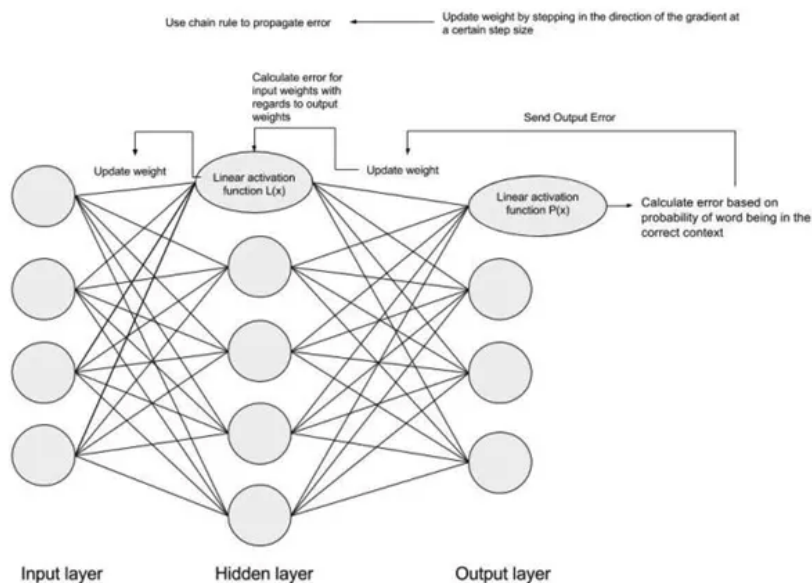


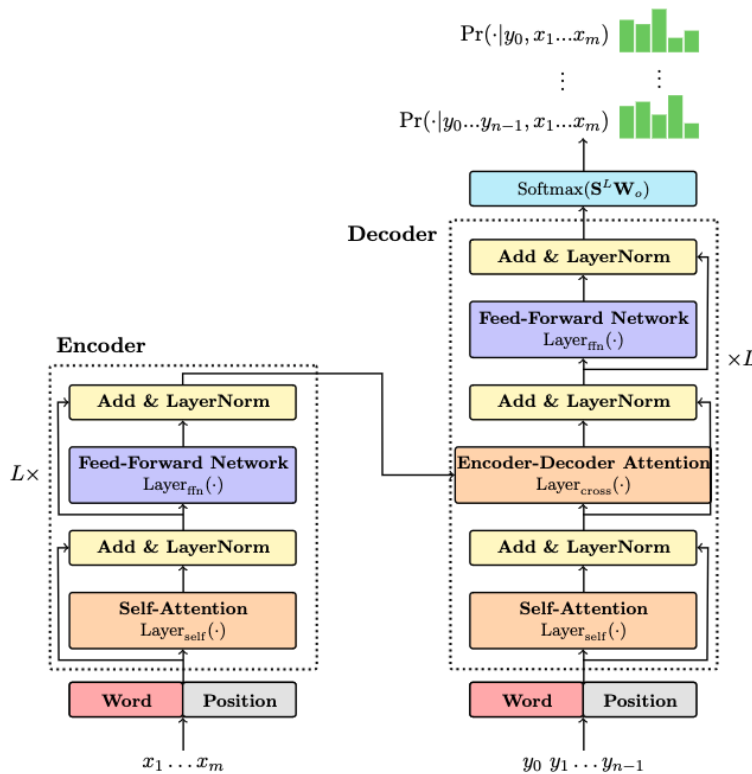
Figura 2.3: Arquitectura de *back propagation*. (tomado de [2]).

Transformers: su arquitectura sigue un marco general de codificador-decodificador, consistiendo en varias capas apiladas tanto en el codificador como en el decodificador. Cada capa del codificador contiene principalmente dos subcapas: una subcapa de auto-atención y una subcapa de red neuronal *feed-forward* (FFN) [3].

Codificador

- **Subcapa de Auto-Atención:** Permite que el modelo pese la importancia de diferentes palabras en la secuencia de entrada para

Figura 2.4: Arquitectura de *Transformers*. (tomado de [3]).



una palabra específica. Esencialmente, calcula una representación de cada palabra que es una mezcla de todas las palabras en la secuencia, ponderada por su relevancia [3].

- **Red Neuronal Feed-Forward (FFN):** Transforma la representación obtenida después de la auto-atención aplicando operaciones lineales seguidas de una función de activación no lineal [3].

Cada una de estas subcapas está seguida por una conexión residual y una normalización de capa, lo que ayuda a evitar el problema del desvanecimiento del gradiente y facilita el entrenamiento de modelos de redes neuronales [3].

Decodificador El decodificador sigue una estructura similar a la del codificador pero con una subcapa adicional [3]:

- **Auto-Atención:** Similar a la del codificador, pero adaptada para evitar que la posición actual tenga en cuenta las futuras posiciones durante el entrenamiento (esto se conoce como enmascaramiento).
- **Atención Codificador-Decoder:** Ayuda al decodificador a

enfocarse en partes relevantes de la entrada al codificador al generar cada palabra.

- **FFN**: Igual que en el codificador.

2.3.2 *Módelos de Clasificación*

Random Forest es la combinación de los predictores árboles de decisión de tal manera que cada árbol depende de los valores de un vector aleatorio muestreado de forma independiente y con la misma distribución para todos los árboles del "bosque" [4].

La precisión en la clasificación mejora significativamente al incrementar el número de árboles ensamblados y permitiendo que estos ensambles voten por la clase más popular de la variable objetivo [4].

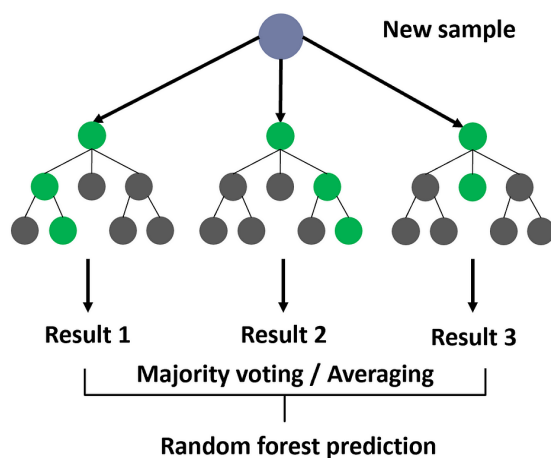


Figura 2.5: Ejemplo ilustrativo de *Random Forest*. (tomado de [4]).

Extreme gradient boosting (XGBoost) es un modelo de ensamble que crea una serie de árboles de decisión de manera iterativa, de tal manera que cada árbol busca corregir los errores del anterior [5].

La optimización del gradiente busca encontrar en que dirección agregar el nuevo árbol para minimizar el error.

Support Vector Classifier (SVC) es un tipo de clasificador que funciona encontrando el hiperplano óptimo separando las clases de datos maximizando el margen entre los hiperplanos y los vectores de soporte más cercanos [12].

Hiperplano: Es el plano en el que las clases serán separadas, este puede ser de dos o más dimensiones, dependiendo de las clases de la variable objetivo [12].

Vectores de soporte: Estos son los puntos de datos más cercanos al

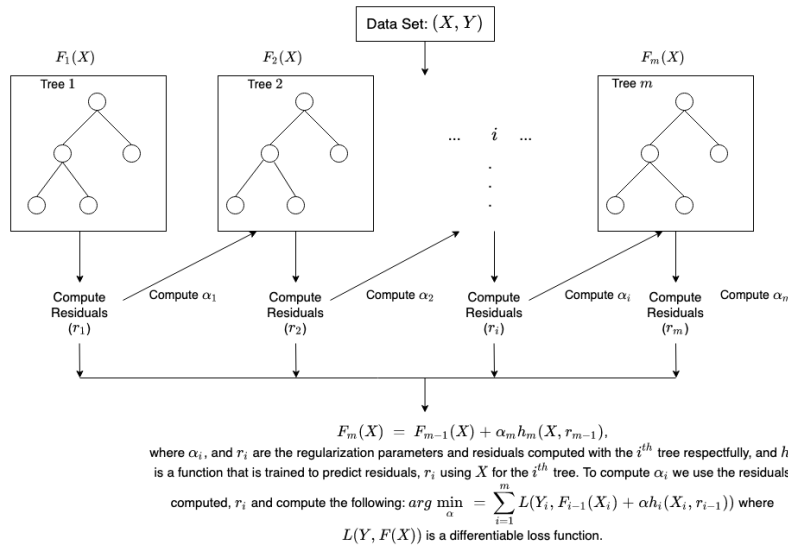


Figura 2.6: Ejemplo ilustrativo de Xgboost. (tomado de [5])

hiperplano, el objetivo es maximizar la distancia entre estos vectores y el hiperplano de decisión, mejorando los predictores [12].

Kernel: Para los datos que no son linealmente separables, SVC puede utilizar una función kernel para transformar el espacio con el objetivo de buscar [12].

2.4 Descripción de las métricas

En esta sección se incluye una descripción de las métricas utilizadas para la comparación de los modelos y el proceso de selección de estas métricas para justificar su pertinencia.[13]

Siendo este un problema de clasificación con clases desbalanceadas, el métrico al que se le da mayor importancia fue **F1 Score** ya que este métrico equilibra el rendimiento del modelo en ambas clases.[13]

Para calcular el F1-Score primero se tiene que observar la matriz de confusión.

La matriz de confusión representa el desempeño de la predicción de un modelo de clasificación [13]. Por ejemplo, en una matriz de confusión para una clase binaria se observaría como en la 2.7

$$F1-Score = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (2.1)$$

Donde Recall y Precisión son calculadas con las fórmulas (2.2) y (2.3)

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.2)$$



Figura 2.7: Ejemplo de Matriz de Confusión. (tomado de [6]).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.3)$$

Para la evaluación de muestras multi clases desbalanceadas es más conveniente utilizar el *F1-Score weighted* el cual considera el desbalance de las clases y nos puede dar una mejor idea de como se comportará el modelo al obtener más datos de las clases minoritarias

$$\text{Precision} = F_{1,\text{weighted}} = \sum_{i=1}^N w_i \cdot F_{1,i} = \sum_{i=1}^N \left(\frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \right) \cdot w_i \quad (2.4)$$

Donde w_i es el peso de la clase i , usualmente el número de instancias verdaderas para la clase i dividido por el total de instancias, Precision_i es la precisión para la clase i , y Recall_i es la sensibilidad para la clase i . N es el número total de clases.

2.5 Descripción de los experimentos o simulaciones

Con base en los modelos explicados en la sección 2.4, se procede a utilizar el modelo de NLP *Word2Vec* para el procesamiento de los datos

y 3 modelos de clasificación *Random Forest*, *XgBoost* y *SVM*, utilizando la librería *scikit-learn* de *Python* para los 3 modelos.

Adicionalmente se utilizará el modelo de NLP *Transformers* y el modulo *TFAutoModelForSequenceClassification* el cual clasificará utilizando las redes neuronales de entrenamiento de la librería *Transformers* de *Python* y la librería *tensorflow* para mejorar el tiempo de ejecución y desempeño de los modelos [14].

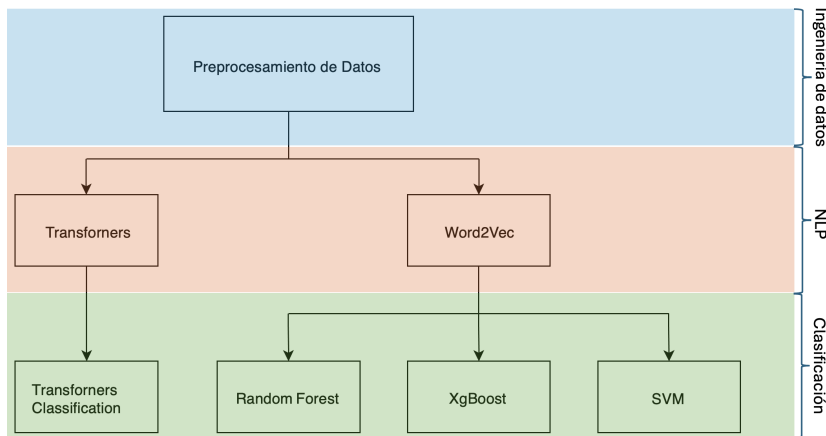


Figura 2.8: Diagrama de flujo.

Ingeniería de Datos: Se eliminan las columnas dejando únicamente la columna de `summary` y `general_cap_classification`. Adicionalmente se eliminan las clases de menor ocurrencia (0thr, Sexual Crime misconduct, Falsification, Domestic y Non-Investigatory incident).

NLP: Se realiza la transformación de los datos de la variable `summary`, para transformar la semántica en vectores numéricos los cuales podrán ser utilizados en los modelos de clasificación.

Clasificación: Se realiza el entrenamiento de los modelos de clasificación con una muestra balanceada de los datos, utilizando `RandomOverSampler`, los modelos de clasificación son ejecutados y se obtienen las métricas para su comparación.

3 Resultados y discusión

En este capítulo se presentan los resultados obtenidos del desarrollo de este trabajo y una discusión sobre el objeto de estudio.

3.1 Resultados

Al realizar la ejecución de los distintos modelos propuestos se obtuvieron los resultados de la *Tabla 3.1*.

Modelo	<i>Precision</i>	<i>Recall</i>	<i>Weighted F1-Score</i>
<i>Random Forest</i>	83 %	45 %	54 %
<i>XgBoost</i>	75 %	43 %	51 %
<i>SVM</i>	100 %	46 %	61 %
<i>Transformers</i>	100 %	94 %	96 %

Tabla 3.1: Resultados obtenidos.

Como se puede observar, en cuanto a la métrica de *Precision*, los modelos de *SVM* y *Transformers*, prácticamente un empate. En cuanto a la métrica de *Recall*, el modelo de *Transformers* es el que mejor desempeño tuvo, el resto de modelos tuvieron un desempeño similar. En cuanto a la métrica de *Weighted F1-Score*, el modelo de *Transformers* fue el que mejor desempeño tuvo, seguido por el modelo de *SVM*, y los modelos de *Random Forest* y *XgBoost* tuvieron un desempeño similar.

3.2 Discusión

- Se puede apreciar un desempeño muy superior en el modelo de *Transformers* en todas las métricas, esto sucede porque la misma red neuronal que se utiliza para decodificar los datos de entrada, se utiliza para codificar y clasificar las observaciones, este modelo resulta ser muy efectivo para resolver el problema de clasificación utilizando un modelo de NLP.
- En contraste los modelos que utilizaron un algoritmo de NLP para la decodificación y un modelo adicional para clasificación, no fueron muy efectivos en el *Weighted F1-Score*, esto debido al gran desbalance de las clases que existe en el dataset, y a pesar de ser entrenado con

una muestra balanceada generando datos similares a los de la clase minoritaria, no fue muy bueno en el *Recall*, esto debido a las pocas observaciones de la clase minoritaria.

- En general los modelos presentan una buena precisión, esto también puede ser atribuido a la buena efectividad que tiene para predecir operaciones de la clase mayoritaria.

4 Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones obtenidas del desarrollo de este trabajo y se proponen posibles líneas de trabajo futuro.

4.1 Conclusiones

De acuerdo con los resultados obtenidos se puede señalar la superioridad en la obtención del contexto y semántica con la arquitectura *Transformers*, puesto que al usar el clasificador de esta arquitectura, se utiliza la misma red neuronal de decodificación para codificar los nuevos valores y clasificarlos.

Si bien el modelo *Transformers* presenta un mejor desempeño al clasificar los datos, el tiempo de ejecución de este fue muy superior a los modelos de clasificación utilizados en los experimentos, siendo que en este proyecto se busca tener la mayor *Precision* posible, el tiempo de ejecución no es relevante para esta aplicación.

El desempeño de 96 % de F1 -Score es significativamente superior al de los modelos de clasificación convencionales, lo cual abre el panorama para la solución de problemas de NLP y como poder utilizarlos para realizar clasificaciones o clustering.

4.2 Trabajo futuro

4.2.1 Modelos de clasificación convencionales

Con el objetivo de mejorar la predicción utilizando modelos convencionales se podrían tomar en consideración los siguientes puntos:

- Realizar la transformación de los datos: Durante la realización de los experimentos, se aplicó PCA sin una mejora significativa en el desempeño de los modelos de clasificación. En conjuntos de datos desbalanceados, tanto el Análisis Discriminante Lineal (LDA) como el Análisis Discriminante Cuadrático (QDA) han demostrado ser efectivos para mejorar la capacidad de predicción en las clases minoritarias."

- Utilizar redes neuronales para clasificación: Word2Vec utiliza redes neuronales para transformar el texto en vectores, una opción para trabajo futuro podría utilizar esa red neuronal para realizar la clasificación.
- Ensamble de modelos: Se observa que SVM tiene mejor precisión que los demás modelos, un ensamble de SVM con XgBoost o *Random Forest*, podría reducir los errores de SVC mejorando el *Weighted F1-Score* que es el métrico de referencia para este tipo de problemas.

4.2.2 *Modelo de clasificación basado en arquitectura Transformers*

- Optimización de hiperparámetros: Se pueden realizar experimentos modificando los hiperparámetros de entrenamiento para obtener mejores resultados o mejorar los tiempos de ejecución.
- Utilizar Vectorización: Con el objetivo de reducir el tiempo de ejecución se puede realizar la vectorización con *Transformers* y clasificación con algún otro modelo de la misma manera que se realizó con Word2Vec en los experimentos.
- Incrementar las observaciones: Se realizó el entrenamiento con el 60% de los datos y se obtuvo el F1-Score de 96%, es importante validar con un dataset mayor para asegurar que el modelo no cae en over-fitting.

Bibliografía

- [1] O. Ogundepo, "Understanding word2vec." <https://medium.com/analytics-vidhya/understanding-word2vec-39fabe660705>, 2021.
- [2] S. Prasoon, "Deep dive into word2vec." <https://medium.com/analytics-vidhya/deep-dive-into-word2vec-7fcefa765c17>, 2019.
- [3] J. Z. Tong Xiao, "Introduction to *Transformers*: an nlp perspective." <https://arxiv.org/pdf/2311.17633.pdf>, 2023.
- [4] N. MArloff, *The Art of Machine Learning*. Kluwer Academic Publishers: No Starch Press, 2024.
- [5] C. G. Tianqi Chen, "Xgboost: A scalable tree boosting system." <https://arxiv.org/pdf/1603.02754.pdf>, 2016.
- [6] J. I. B. Arce, "La matriz de confusión y sus métricas." <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>, 2019.
- [7] J. R. Ben Grunwald, "The wandering officer." <https://www.yalelawjournal.org/article/the-wandering-officer>, 2020.
- [8] F. Sharara, "Fatal police violence by race and state in the usa, 1980-2019: a network meta-regression." <https://www.thelancet.com/action/showPdf?pii=S0140-6736%2821%2901609-3>, 2019.
- [9] LexisNexis, "Annual legal analytics survey reveals: Using legal analytics is table stakes for litigation." <https://www.lexisnexis.com/community/insights/legal/b/thought-leadership/posts/taking-analytics-to-court>, 2023.
- [10] J. Hill, "How is ai used in legal technology?." <https://natlawreview.com/article/how-ai-used-legal-technology>, 2023.

- [11] NACDL, "Dataset full disclosure project." <https://www.nacdl.org/Landing/FullDisclosureProject>, 2023.
- [12] J. A. Rodrigo, "Máquinas de vector soporte (support vector machines, svms)." https://cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines, 2017.
- [13] R. Kundu, "F1 score in machine learning: Intro & calculation)." <https://www.v7labs.com/blog/f1-score-guide>, 2022.
- [14] L. González, "¿qué es tensorflow? ¿cómo funciona?." <https://aprendeia.com/que-es-tensorflow-como-funciona/>, 2024.