Identificación de procesos impactados en la gerencia de mercado de energía a través de redes de procesamiento de lenguaje natural

Camilo Mejía López

Informe de práctica presentado para optar al título de Ingeniero Energético

Tutor
Álvaro Jaramillo Duque, Doctor (PhD) en Ingeniería Eléctrica

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería Energética
El Carmen de Viboral, Antioquia, Colombia
2024

Cita	Mejía López [1]
Referencia Estilo IEEE (2020)	[1] C. Mejía López, “identificación de procesos impactados en la gerencia de mercado de energía a través de redes de procesamiento de lenguaje”, Trabajo de grado profesional, Ingeniería Energética, Universidad de Antioquia, El Carmen de Viboral, Antioquia, Colombia, 2023.

Dedicatoria

Para mi familia, mis amigos y todas las personas que de otra manera me hicieron llegar a donde estoy

Agradecimientos

A XM que confió en un tímido Camilo, a Liliana Pineda por darme la confianza que yo no tenía, a Noé Mesa por apoyar a un confundido estudiante.

TABLA DE CONTENIDO

LISTA DE FIGURAS

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

MEM	Mercado de Energía Mayorista
CND	Centro Nacional de Despacho
NLP	Natural Lenguage Processing
GEME	Gerencia Mercado de Energía
CREG	Comisión de Regulación de Energía y Gas
MME	Ministerio de Minas y Energía
SVM	Support Vector Machine
IA	Inteligencia Artificial
SIC	Sistema de Intercambios Comerciales
LAC	Liquidación y Administración de Cuentas

RESUMEN

En la Gerencia de Mercado de Energía (GEME) uno de los procesos más importantes es la implementación de nuevas resoluciones, decretos o cualquier cambio regulatorio expedido por los entes. En estos procesos uno de los hitos claves es identificar de manera oportuna qué procesos son impactados por estos cambios regulatorios para garantizar y tener una vista integral de los impactos. Además, proponer soluciones que garanticen la aplicación de los cambios regulatorios. Sin embargo, la identificación de los procesos impactados varía dependiendo de la complejidad y dimensión de la resolución. Por tanto, se propone una herramienta que pueda procesar el contenido de la resolución y determinar los posibles equipos impactados de manera mucho más ágil y de forma desatendida. Esta tarea se pretende realizar a través de un modelo de procesamiento de lenguaje natural el cual tomará las resoluciones nuevas para lograr una implementación de un proceso automático.

Este proyecto, por tanto, muestra tres estrategias de manejo de datos, sus resultados y además los procesos de mejora para obtener clasificaciones más reales. Además de presentar las demás actividades en las que también participó.

Palabras clave — **NLP, Clasificación, Cambios Regulatorios, SVM, Scikit – Learn**, **Python**

ABSTRACT

Within the Energy Market Management (GEME by its acronym in Spanish), one of the most important processes is the implementation of new resolutions, decrees, or any regulatory changes issued by authorities. In these processes, one of the key milestones is to timely identify which processes are impacted by these regulatory changes to ensure and have a comprehensive view of the impacts. Additionally, proposing solutions that guarantee the application of regulatory changes is crucial. However, the identification of impacted processes varies depending on the complexity and scope of the resolution. Therefore, a tool is proposed that can process the content of the resolution and determine the possible impacted teams in a much more agile and unattended manner. This task is intended to be conducted through a natural language processing model that will take new resolutions to achieve an implementation of an automatic process.

This project, therefore, highlights three data management strategies, their outcomes, and the improvement processes to achieve more realistic classifications. It also presents other activities in which I also participated.

Keywords — **NLP, Classification, Regulatory Changes, SVM, Scikit – Learn**, **Python**

I. INTRODUCCIÓN

XM S.A E.S.P (XM) cómo administrador del mercado de energía mayorista (MEM) en Colombia, realiza actividades que se encuentran fundamentadas en las aplicaciones de normatividad vigente; por tanto, sus procesos son sensibles a cambios regulatorios que pueden generar grandes cambios al interior de los equipos de trabajo, como liquidación de transacciones del mercado, liquidación de bolsa, agentes, fronteras, contratos, entre otros. Estos aspectos se encuentran en constante cambio y se deben adaptar de manera ágil y eficiente acorde a los cambios regulatorios que trae la transición energética. La dirección de Analítica y Desarrollo de Mercado de Energía se encarga del análisis de impactos, propuestas normativas a los entes regulatorios, análisis de datos de impacto en el mercado y de la implementación regulatoria al interior de XM. Sin embargo, al estar en un ambiente regulatoriamente cambiante, es vital el análisis ágil y oportuno de la regulación con el fin de dar alertas tempranas de los procesos impactados.

En el proceso de la implementación regulaciones que tiene la dirección de Analítica y Desarrollo del Mercado, se identifican entre las principales actividades: determinar los procesos impactados, identificar el equipo de trabajo y estimar los posibles recursos necesarios para la ejecución del proyecto. Para realizar estas tareas es crítico identificar qué procesos afecta la nueva resolución emitida. Una vez identificados los equipos, es cuando la implementación se puede iniciar. El proceso de identificación puede tomar tiempo, que muchas veces la norma no prevé y los procesos internos pueden dilatarse y lo anterior puede generar riesgos de cumplimiento regulatorio o la identificación de un proceso de forma no oportuna.

Por tanto, este proyecto busca realizar una herramienta de procesamiento de lenguaje natural que permita analizar las nuevas resoluciones e identificar los posibles equipos impactados con el fin de agilizar el análisis de impactos, la gestión de recursos y los procesos afectados.

II. OBJETIVOS

A. Objetivo general

Identificar procesos y equipos impactados en la Gerencia del Mercado de Energía por nuevas resoluciones mediante el uso de una herramienta de procesamiento de lenguaje natural que permita el seguimiento y gestión a la implementación regulatoria.

B. Objetivos específicos

- Determinar qué tipo de modelos de procesamiento de lenguaje natural son adecuados para procesar y clasificar la información contenida en las nuevas normativas.
- Construir una base de datos con resoluciones procesadas de las diferentes gerencias de la GEME para realizar el proceso de entrenamiento y validación.
- Realizar pruebas paralelas con el equipo de la GEME y el modelo implementado.
- Documentar los procesos realizados en base a la metodología corporativa.
- Analizar posibles mejoras al proceso, habilitando la mejora continua
- Apoyar en la implementación regulatoria y tareas del equipo.

III. MARCO TEÓRICO

El procesamiento de lenguaje natural (NLP por siglas en inglés) abarca diversas técnicas para analizar y comprender el lenguaje humano. Dentro de este campo, la Tokenización y Lematización son procesos fundamentales. La Tokenización implica dividir el texto en unidades más pequeñas, generalmente palabras, para facilitar el análisis. Por otro lado, la Lematización se centra en reducir las palabras a su forma base, simplificando la interpretación del texto. En este contexto, técnicas *como Keyword Frequency Analysis* (KFA) y *Named Entity Recognition* (NER) se benefician de la Tokenización para extraer información clave y reconocer entidades específicas en el texto.

Estás técnicas pueden ser aplicadas a través de librerías de Python cómo *Scikit – Learn*, proporcionan una manera de eficiente y rápida de implementar algoritmos, cómo Regresión Logística, Máquina de Vectores de Soporte (*Support Vector Machine), Naive Bayes Multinomial* y *Random Forest*. Estos modelos son utilizados en diversas tareas, incluido el análisis de impacto regulatorio, donde se evalúan las implicaciones normativas.

La evaluación de modelos en el NLP es crucial, y métricas como *Accuracy, Precision, Recall y F1 Score* son comúnmente utilizadas. Estas métricas permiten medir la eficacia de los modelos en la clasificación y extracción de información. La exactitud (*Accuracy*) mide la proporción de predicciones correctas, la precisión (*Precision*) evalúa la proporción de predicciones correctas entre las positivas, la sensibilidad (*Recall*) cuantifica la capacidad de un modelo para capturar todos los casos positivos y el *F1 Score* combina precisión y sensibilidad en una sola métrica.

A. Procesamiento de Lenguaje Natural

Acorde a IBM, el procesamiento de lenguaje natural (NLP), es una rama de la Inteligencia Artificial (IA), que busca poder procesar palabras y textos de la manera que los humanos las concebimos. NLP combina reglas computacionales y lingüísticas, junto con conceptos estadísticos para poder procesar textos o audios, de la misma manera que los humanos lo podemos hacer. Este proceso se puede dividir, según Indurkhya et al [1] en 5 etapas: Tokenización, análisis léxico, análisis sintáctico, análisis semántico, análisis pragmático. Esto nos permite transformar sentimientos y contexto en datos analizables y organizados. En general los nuevos procesos de

NLP consisten en metodologías basadas en grandes volúmenes de datos o "*big data*", los cuales pueden aumentar la capacidad de extracción de la información relevante [2].

1. *Keyword Frequency Analysis (KFA)*

Este es uno de los métodos más antiguos y a su vez más simples, el cual busca, a través de un conteo de palabras tratar de predecir resultados. Este proceso es útil para construir aplicaciones de alto nivel, y funciona de base útil para otras aplicaciones más complejas [3].

2. *Named Entity Recognition (NER)*

Está metodología busca extraer los tipos específicos (elementos, cantidad, organización, etc.) de la información. Está metodología puede servir cómo análisis exploratorio de la información además de servir de paso previo a un análisis mucho más riguroso [3].

Además de eso, nuevas metodologías cómo las presentadas en [4][5], permite vectorizar los caracteres en números, los cuales puedan ser procesados por algoritmos de clasificación como LSTM (*Long Term Short Memory*), SVM (*Support Vector Machine*), o CNN (*Convolutional Neural Network*), esto para aumentar la capacidad de procesamiento y el reducir los tiempos de respuesta.

B. Tokenización

La tokenización es un proceso obligatorio en todo análisis de lenguaje natural. Es un proceso el cual busca generar un texto con una forma estándar a cualquier idioma, la cual sea posible de procesar, basado en unidades, las cuales pueden ser palabras o raíces de texto, que además pueden ser diferenciables y analizables [3]. Este proceso puede ser desafiante debido a lo diferente que pueden ser los lenguajes; por ejemplo, en español o en inglés se tiene una diferenciación clara de cada palabra, sin embargo, en idiomas cómo inglés, palabras como "I'm" son en realidad "I" y "am", y en idiomas cómo japones no existe diferenciación entre palabras [1]. En el proceso de tokenizar la meta es poder obtener texto sin ambigüedades lingüísticas y estructurales, las cuales pueden afectar la interpretación de los textos. Trabajos cómo Shanyan Lai et al [6] donde el contexto y la estructura importa, debido a que puede ayudar a diferenciar comportamientos, y se proponen diferentes estructuras de tokenización, las cuales busquen que el contexto sea parte de las predicciones.

C. Lematización

La lematización es el proceso por el cual se busca encontrar si dos o más palabras comparten la misma raíz, [3] palabras cómo “am” “are” “is” comparten la misma raíz “be” lo cual estable una correlación entre palabras similares. Por ejemplo, en un análisis “Camión” y “camión” serían consideradas cómo dos palabras diferentes, o incluso “Comió” y “Comer” también se consideran distintas, esto puede agregar ruido a los análisis, sin embargo, el proceso de lematización puede ser complejo y computacionalmente costoso [7]. Además, si no se tienen los algoritmos correctos, pueden agregar errores y fallos en los procesos. Por tanto, el proceso de lematización puede mejorar la precisión de modelos en los cuales se tenga pleno conocimiento de forma y tamaño; además en los cuales el valor computacional no sobrepasa el valor de oportunidad del proceso.

D. Análisis de Impacto Regulatorio

Según la Organización para la Cooperación y el Desarrollo Económicos (OECD por sus siglas en inglés) el análisis de impacto regulatorio provee a instituciones gubernamentales y legisladores de herramientas y datos, los cuales permiten evaluar de forma consiente las posibles consecuencias de los cambios en cierta regulación [8]. Por tanto, XM en su labor es responsable de realizar el análisis de impacto regulatorio para el análisis consiente de las propuestas de parte de la Comisión de Regulación de Energía y Gas (CREG) el Ministerio de Minas y Energía (MME). Este análisis busca no sólo levantar señales de posibles riesgos, sino también de guiar la aplicación de las resoluciones dentro de los equipos de gerencia MEM.

E. Scikit - Learn

Scikit – Learn es un módulo integrado de Python que es ampliamente utilizado y que hace uso de la mejor tecnología de *machine learning* para mediana y pequeña escala en problemas supervisados y no supervisados. Esta librería busca llevar el aprendizaje automático a personas no especialistas a través de lenguajes de programación de alto nivel. [9]. Gracias a su amplia documentación permite aplicar diferentes tipos de algoritmos de clasificación y evaluación de manera simple y eficiente, lo que hace que sea el módulo elegido para el desarrollo del proceso.

F. Regresión Logística (Logistic Regression)

La Regresión Logística es un modelo estadístico el cual modela la probabilidad de ocurrencia de un evento binario [10]. A diferencia de una regresión lineal que modela variables continuas, la regresión logística utiliza la función logística para transformar la combinación lineal de las variables predictoras en una probabilidad en el rango de 0 a 1. La función logística o función sigmoide se define cómo [11]:

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_0 X_2 + \cdots + \beta_n X_n)}}$$

Donde:

- $P(Y = 1)$ Es la probabilidad de que la variable dependiente Y sea igual a 1.
- $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ Son los coeficientes del modelo.
- $X_1, X_2, \ldots, X_n$ Son las variables predictoras.

El modelo se usa en problemas de clasificación porque, una vez encontrados los coeficientes, pueden predecir la probabilidad de que un nuevo conjunto de variables predictoras pertenezca a la clase Y.

G. Máquina de vectores de soporte (Support Vector Machine)

Una máquina de vectores de soporte o *Support Vector Machine* (SVM) es una serie de algoritmos de aprendizaje supervisado utilizador para tareas de clasificación y regresión, su enfoque principal es encontrar un hiperplano en un espacio multidimensional que mejor separe los puntos de datos de diferentes clases. Figura 1 [12].

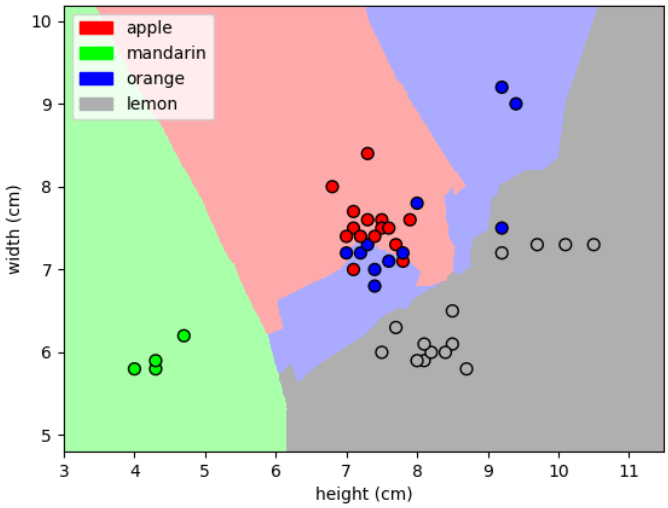

Figura 1. Visualización de regiones de clasificación.

El objetivo fundamental de un SVM en problemas de clasificación es encontrar el hiperplano óptimo que maximice el margen entre las clases. [13]. El margen se define como la distancia perpendicular desde el hiperplano a los puntos más cercanos de cada clase, que son llamados vectores de soporte. Estos vectores de soporte son los puntos de datos más difíciles de clasificar y juegan un papel crucial en la determinación del hiperplano. [14]

La idea es encontrar el hiperplano que maximiza el margen y al mismo tiempo minimiza la clasificación incorrecta. En casos donde los datos no son linealmente separables en el espacio original, los SVM pueden aplicar transformaciones no lineales, utilizando funciones kernel, para proyectar los datos en un espacio de mayor dimensión donde la separación lineal sea posible. [15]

En la formulación matemática, el problema de optimización de un SVM busca encontrar los coeficientes del hiperplano y, posiblemente, la función de kernel, que maximicen la separación entre clases.

H. Naive Bayes Multinomial

El algoritmo de Naive Bayes Multinomial es una variante del clasificador Naive Bayes que se utiliza comúnmente en problemas de clasificación de texto [16], como la categorización de documentos y la detección de spam. Este algoritmo asume que las características utilizadas para la clasificación son independientes entre sí, dada la clase de la instancia. Aunque esta suposición es a menudo simplista e "ingenua", suele funcionar bien en la práctica.

En el caso del Naive Bayes Multinomial, se asume que las características son variables discretas y que siguen una distribución multinomial. Este tipo de distribución es adecuado para modelos de clasificación de texto, donde se cuenta la frecuencia de ocurrencia de las palabras en un documento.[17]. Se basa en la aplicación de la Regla de Bayes para predecir la probabilidad condicional de que un documento pertenezca a una clase $P(c_{ij}|d_j)$ a partir de la probabilidad de los documentos, dada la clase $P(d_j|c_i)$ y la probabilidad de la clase del conjunto de entrenamiento $P(c_i)$.[17].

El clasificador asignará la instancia a la clase con la probabilidad más alta. El Naive Bayes Multinomial es especialmente útil en situaciones donde el vocabulario es grande y las características (palabras) son discretas, como en el análisis de texto [18].

I. Random Forest

Random Forest o Bosques Aleatorios es un poderoso algoritmo de *machine learning* el cual es ampliamente utilizado en problemas de clasificación. Este algoritmo se base en Arboles de decisiones (Figura 2) los cuales son árboles que nacen de manera aleatoria, a través, de subconjuntos de los datos. La clasificación final nace de la votación mayoritaria de las decisiones de individuales de los subconjuntos [19]

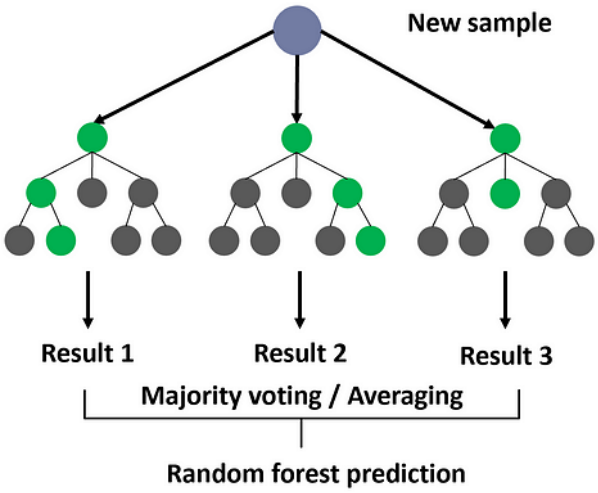

Figura 2. Clasificación usando Random Forest.

J. Métricas de evaluación de modelos

Comúnmente los modelos de clasificación se basan en 4 métricas principales, Exactitud o *Accuracy*, Precisión o *Precision*, Sensibilidad o *Recall*, y el *F1 Score*. [20] Estas métricas se calculan a partir de la matriz de confusión, que es una tabla que resume el rendimiento del modelo en términos de verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN). Las métricas son importantes para evaluar el rendimiento de un modelo de clasificación desde diferentes perspectivas, teniendo en cuenta tanto los errores de tipo I (falsos positivos) como los errores de tipo II (falsos negativos) [20]. La elección de la métrica a utilizar depende del contexto y de la importancia relativa de los diferentes tipos de errores en una aplicación específica.

1. Accuracy (Exactitud)

La exactitud mide la proporción de predicciones correctas realizadas por el modelo sobre el total de predicciones. Es una métrica simple y fácil de entender, pero puede ser engañosa en casos de conjuntos de datos desbalanceados, donde una clase es mucho más frecuente que la otra.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Precision (Precisión)

La precisión se centra en la proporción de instancias positivas predichas correctamente con respecto a todas las instancias predichas como positivas. Es útil cuando el costo de los falsos positivos es alto.

$$Precision = \frac{TP}{TP + FP}$$

3. Recall (Sensibilidad)

El *recall* mide la proporción de instancias positivas que fueron identificadas correctamente por el modelo con respecto al total de instancias positivas. Es útil cuando el costo de los falsos negativos es alto.

$$Recall = \frac{TP}{TP + FN}$$

4. *F1 Score*

El *F1 Score* es la media armónica entre la precisión y el *recall*. Es una métrica que equilibra ambas métricas y es útil cuando hay un desequilibrio entre las clases. El *F1 Score* proporciona un único número que resume el rendimiento del modelo.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

IV. METODOLOGÍA

El generar una base datos lo suficientemente robusta implica una revisión inicial de los procesos y una división eficiente de los mismos para una clasificación acertada. Una vez que se ha recopilado y organizado la información, se pasa a la etapa de tratamiento de los datos. En esta fase, se aplican técnicas para limpiar, transformar y estructurar los datos de manera que sean aptos para el análisis, estableciendo estrategias de procesamiento que permitan encontrar patrones de información eficientes.

Ya con los datos procesados, el elegir y evaluar el método adecuado, es fundamental para garantizar una clasificación confiable, este proceso es necesario para conocer la utilidad del modelo en contextos prácticos. Finalmente, tras evaluar el modelo, se procede a la "Identificación de posibles mejoras". Este último paso implica revisar el rendimiento del modelo y buscar oportunidades para mejorarlos, encontrando, según las evaluaciones, qué aspectos son necesarios para conseguir mejores clasificaciones.

A. Generar base de datos:

Teniendo en cuenta que dentro del proceso no existía un banco de datos inicial, la prioridad era generar estrategias de adquisición de información útil, para entrenar la red de procesamiento de lenguaje natural, por tanto, la primera fase implementación se basó en:

1. Revisión de información actual y división de equipos

Uno de los pasos primordiales se basaba en la identificación de equipos a los cuales los cambios regulatorios podrían impactar, para luego dividir la regulación más importante de cada equipo y que a su vez podría impactar. Para esto, se tuve en cuenta que la GEME se encuentra divida en principalmente 4 equipos:

1. El equipo de liquidación del Sistema de Intercambios Comerciales (SIC)
2. El equipo de Liquidación y Administración de Cuentas (LAC)
3. El equipo de Contratos
4. El equipo de Fronteras

Cada equipo cuenta con una regulación propia, la cual se presentan en los planes de entrenamiento de los nuevos vinculados, y que permite generar un banco inicial de resoluciones que se podrían analizar. Luego de esto, a los lideres de cada equipo se le pidió retroalimentación y sugerencias de posibles resoluciones, que podrían ser relevantes para el análisis. Obteniendo así un banco de resoluciones, que luego será tratado, y clasificado.

B. Tratamiento de los datos:

Una vez generado el banco de resoluciones, se debe analizar cómo podría ser el tratamiento de los datos donde, se entra en consideración si:

Caso 1: Se procesa todo el texto de la resolución incluyendo pronombres, espacios, adjetivos y demás

Caso 2: Procesar los textos palabra por palabra, eliminando pronombres, espacios, adverbios, caracteres especiales y números.

Caso 3: Procesando palabra por palabra, eliminando pronombres, espacios, adverbios, adicionalmente eliminando las palabras que se repitan dentro de otras resoluciones, las cuales no compartan clasificador. Esto para obtener bloques de palabras únicas por tipo de resoluciones.

Por tanto, se requiere una evaluación de como esté procesamiento podría afectar las predicciones y podría dar espacio para mejorar continuas.

C. Selección de modelo

Debido al tipo de datos y la necesidad de evaluar la mejor estrategia, se eligió evaluar de manera simultánea, 4 tipos de modelos de clasificación:

1. *Logistic Regression*
2. *SVM*
3. *Naive Bayes*
4. *Random Forest*

Estos modelos se implementan a través de *sklearn* usando de base el trabajo de Zahidul Islam [21] el cual usó estos modelos para el análisis de banco de datos de BBC el cual se basa de aproximadamente 2225 textos que buscan clasificar si un texto pertenecer a la categoría de:

1. Deportes

2. Negocios
3. Política
4. Tecnología
5. Entretenimiento

En este contexto, se llevaron a cabo ajustes en el código base y se incorporaron componentes adicionales para permitir el procesamiento y la clasificación efectiva de textos específicos de resoluciones. Se hicieron modificaciones para adaptar la solución existente a las necesidades y requisitos particulares relacionados con la clasificación de contenido textual específico.

D. Evaluación del modelo

Se llevó a cabo mediante la aplicación de métodos estándar de evaluación, centrándose en métricas clave como *Precision, Accuracy, Recall* y *F1 Score*. Para cada modelo, se realizó una división adecuada del conjunto de datos, en conjuntos de entrenamiento y prueba. Posteriormente, se ajustaron los modelos a los datos de entrenamiento y se realizaron predicciones sobre el conjunto de prueba. La métrica de *Accuracy* proporcionó una visión general de la precisión global del modelo, mientras que *Precision* medió la proporción de instancias positivas correctamente identificadas. Por otro lado, *Recall* evaluó la capacidad del modelo para identificar todas las instancias positivas. La métrica F1 Score, que combina *Precision* y *Recall*, ofreció una medida equilibrada entre ambas. Estas evaluaciones se hicieron sistemáticamente para cada modelo, permitiendo comparar su desempeño y facilitar la toma de decisiones informadas sobre la elección del modelo apropiado para la clasificación.

E. Identificación de posibles mejoras

Teniendo en cuenta los resultados de la evaluación del modelo se establecieron posibles mejoras en los procesos de adquisición de datos para futuros trabajos, evaluando metodologías de evaluación a nivel de artículos y no de resoluciones completas, además de establecer un flujo de trabajo para la obtención de mejores resultados.

V. RESULTADOS

Dentro de las resoluciones mapeadas, se obtuvo un banco de 52 resoluciones CREG distribuidas cómo se evidencia en la Figura 3. Donde adicionalmente luego se implementaron los 4 tipos de clasificadores, los cuales luego para cada tipo de caso se validaron de manera cruzada con su *Precision, Accuracy, Recall* y *F1 Score* y las matrices de confusión, las cuales nos dan señales visuales del tamaño y calidad de los datos de prueba.

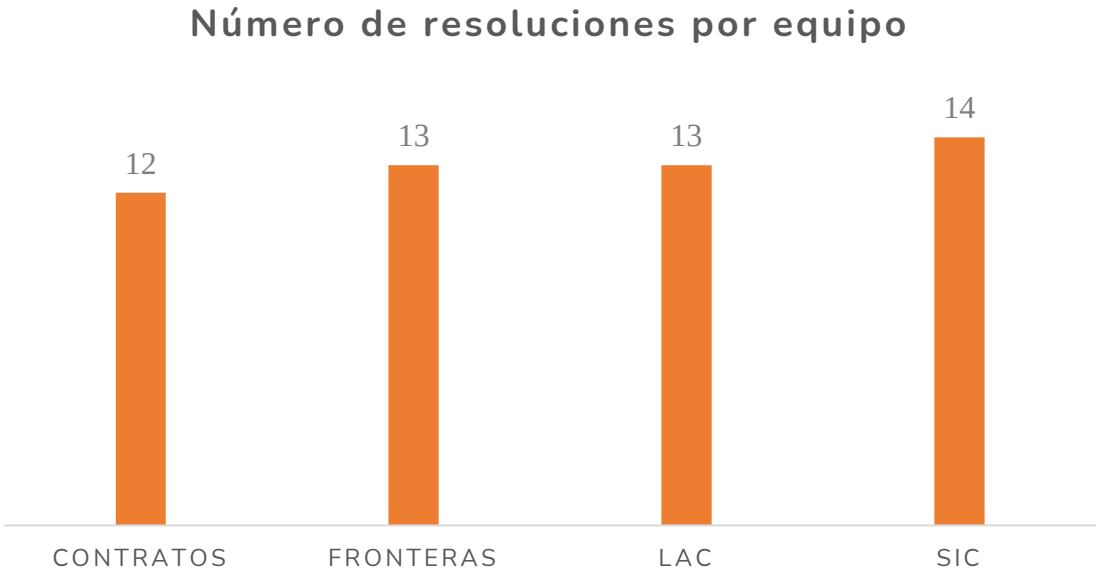

Figura 3. Distribución banco resoluciones

En total este banco se compone de 44641 palabras, de resoluciones que van desde 1995 hasta el 2023, y que son representativas de los procesos seleccionados, y que adicionalmente fueron validadas por los lideres de cada uno de los equipos analizados. Por tanto, realizando el modelado tomando 39 resoluciones de modelado y 13 de prueba, y tomando los casos presentados en la sección *Tratamiento de los datos:* se tiene que:

A. Caso 1

Para el Caso 1, cómo se evidencia en la Figura 4 el modelo que mejor se ajusta es *Random Forest*, sin embargo, los resultados demuestran que el tanto la precisión y *accuracy* no superan el 70%, lo cual demuestra una baja confiabilidad en el modelo y, por tanto, la estrategia del Caso 1 no es la recomendada para la implementación.

Adicionalmente analizando las matrices de confusión para cada algoritmo se tiene que:

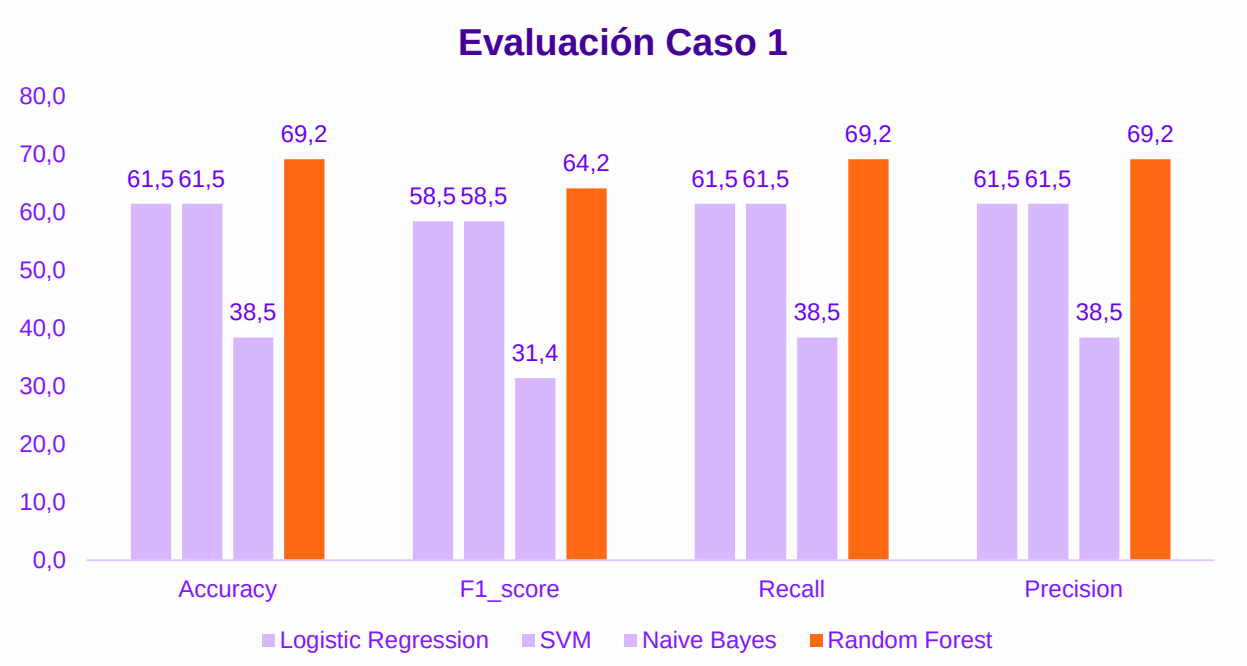

Figura 4. Evaluación modelos Caso 1

1. *Logistic Regression*

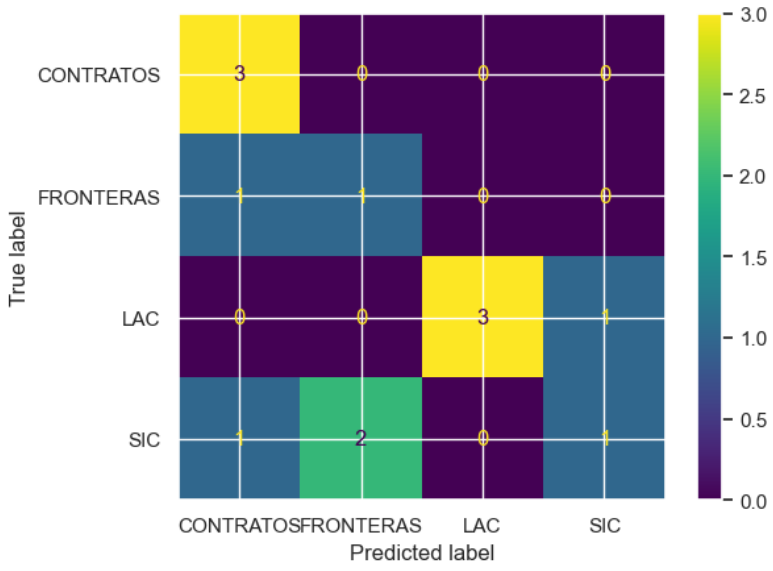

Figura 5. Matriz de confusión Logistic Regression

Según la Figura 5, se puede concluir que el modelo puede predecir de manera más o menos confiable los datos pertenecientes a Contratos y LAC, sin embargo, para las demás categorías no se tiene esa confiabilidad, además se puede visualizar la baja cantidad de datos de prueba.

1. *Support Vector Machine*

Según la Figura 6 para el modelo de SVM tiene un comportamiento similar a la regresión logística, sin embargo, también se nota una baja cantidad de datos de prueba, sin embargo, para el caso de contratos, y LAC el modelo si tiene capacidad de predicción.

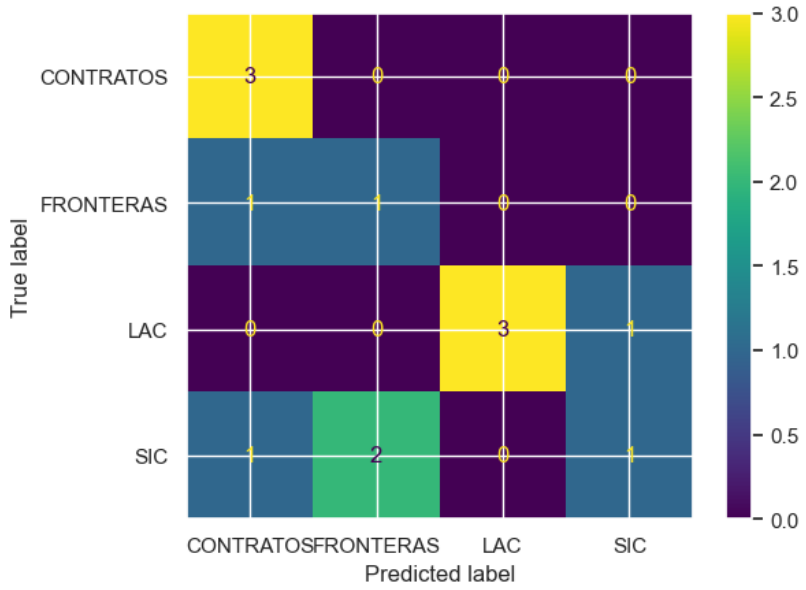

Figura 6. Matriz de confusión SVM

2. *Naive Bayes*

Para *Naive Bayes* la predicción falla mucho más (Ver Figura 7), sólo para LAC y para algunos casos de Fronteras, se notan valores *True Positive Values.*

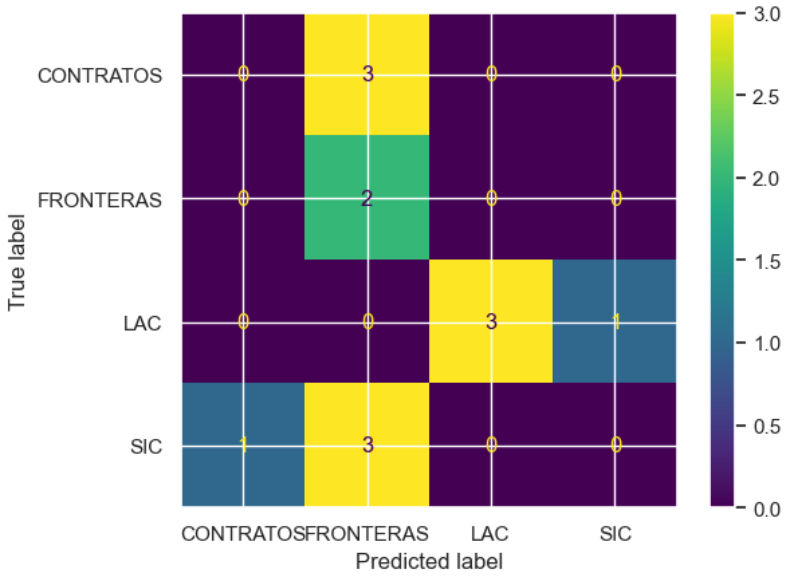

Figura 7. Matriz de confusión Naive Bayes

B. Caso 2

Para el caso 2 se tiene que existe un comportamiento similar al Caso 1 (Ver Figura 8), donde el algoritmo de *Random Forest*, sin embargo, similarmente, el rendimiento del modelo no es lo suficientemente alto para implementar.

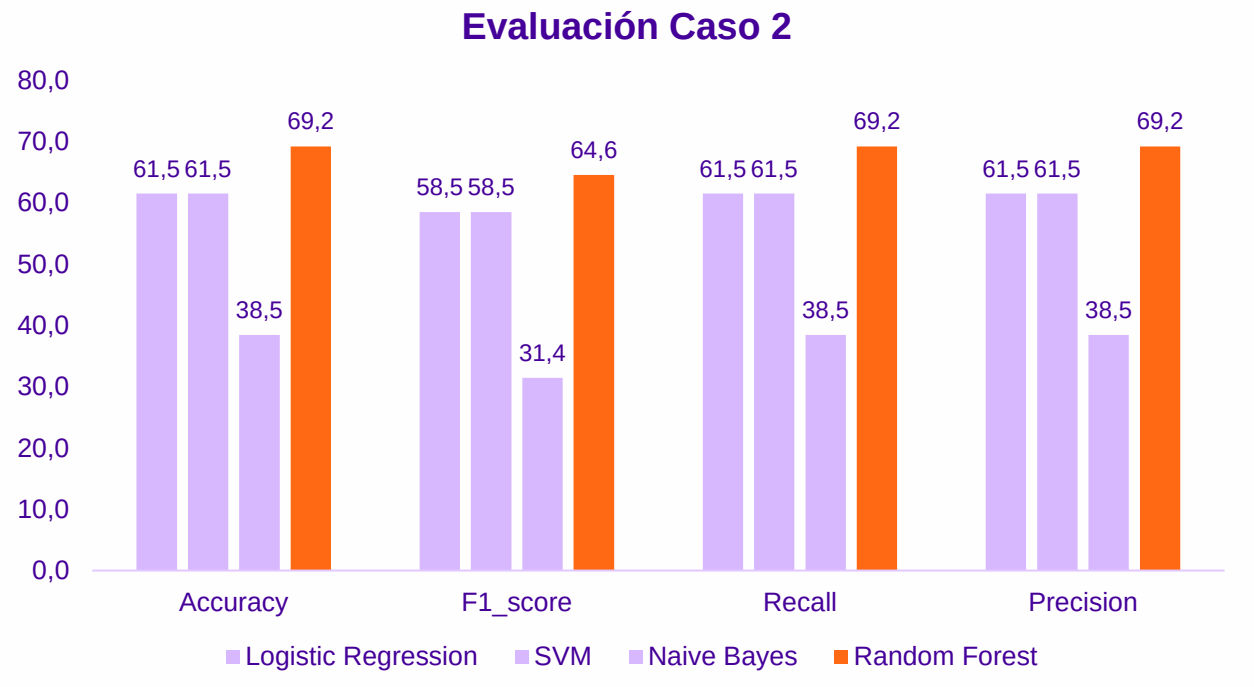

Figura 8. Evaluación modelos Caso 2

Adicionalmente, analizando las matrices de confusión, se tiene que:

1. *Logistic Regression*

Está matriz, comparte características con la matriz de confusión del caso 1 para el modelo de Logistic Regression. Donde, tanto para Contratos cómo para LAC, se tienen aproximaciones ajustadas, sin embargo, para Fronteras y SIC, las clasificaciones fallan.

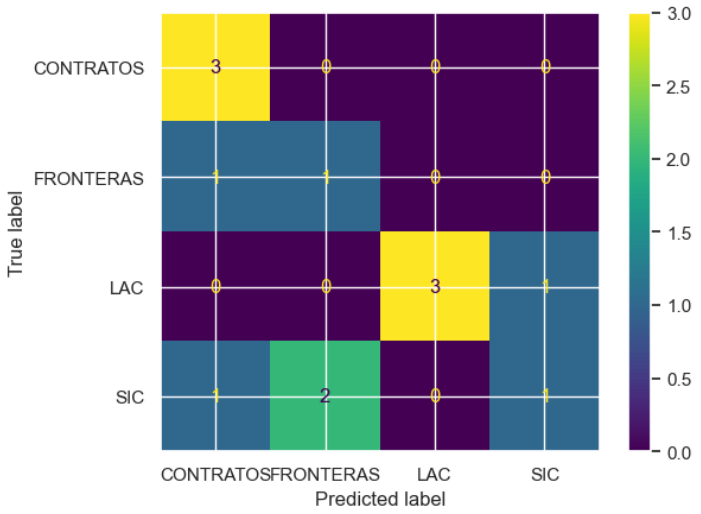

Figura 9. Matriz de confusión Logistic Regression

2. *Support Vector Machine*

Por tanto, de manera similar tanto para Fronteras y SIC se requiere mayor cantidad de datos de prueba y modelado para obtener resultados más confiables.

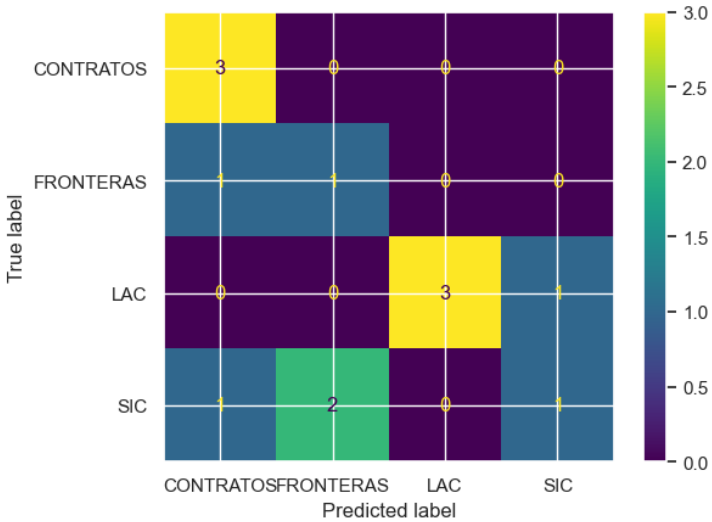

Figura 10. Matriz de confusión SVM

3. Naive Bayes

Para el modelo de Naive bayes, los resultados concluyen que sólo para LAC, el modelo realiza predicciones viables (Figura 11).

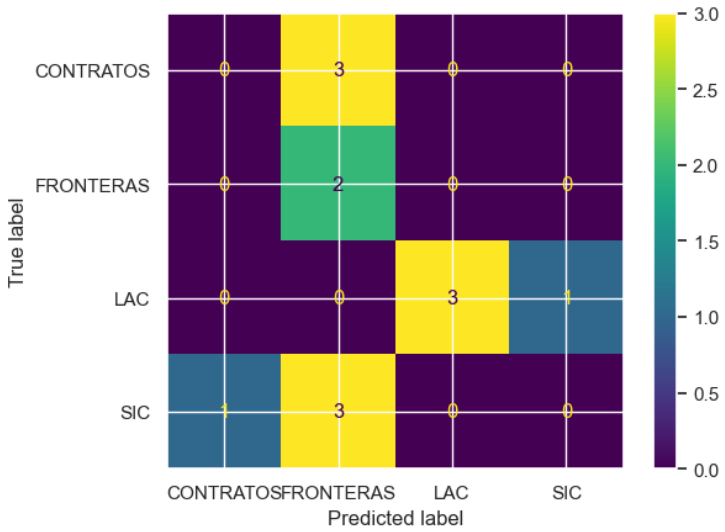

Figura 11. Matriz de confusión Naive Bayes.

C. Caso 3

Para el Caso 3 debido a la metodología plantea el banco de resoluciones se reduce a 30 resoluciones, ya que a que, al realizar la reducción de palabras, es posibles que resoluciones no tengan palabras disponibles y por tanto no se analizan. El banco de resoluciones, por tanto, quedó consolidado cómo se evidencia en la Figura 12.

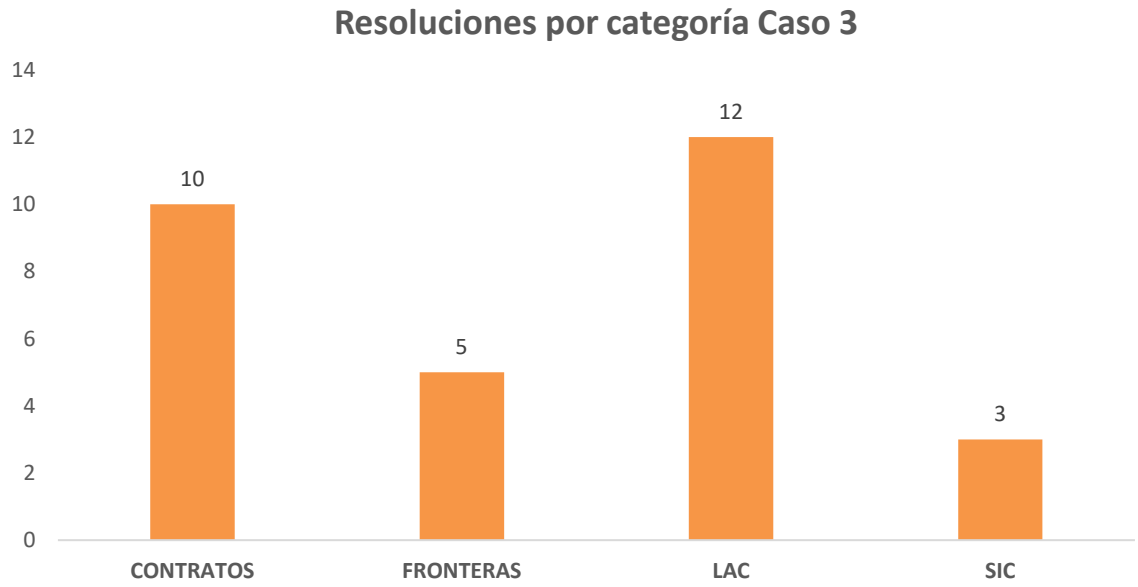

Figura 12. Banco de resoluciones Caso 3

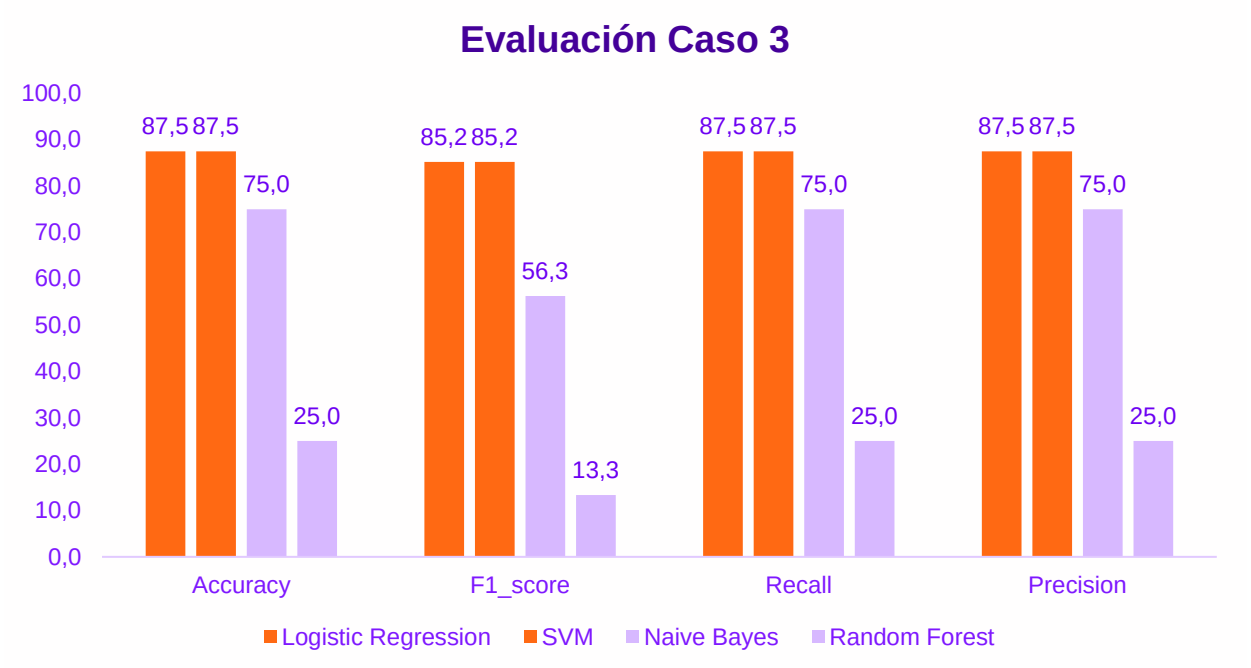

Figura 13. Evaluación modelos Caso 3

Debido a la sub-entrenación del modelo, el modelo basado en *Random Forest*, cuenta con una evaluación baja, debido a que no se pueden generar suficientes subconjuntos para la clasificación satisfactoria de los datos. Además, aunque los otros modelos tienen medidas congruentes y a priori aceptables, al analizar las matrices de confusión, se concluye que no hay suficientes casos de pruebas para un correcto análisis.

1. *Logistic Regression*

Con sólo 8 casos de pruebas el modelo efectivamente logra predecir de forma confiable las resoluciones pertenecientes al LAC (Ver Figura 14), sin embargo, no existe evaluación para SIC y para fronteres existe una falta de precisión a la hora de la clasificación.

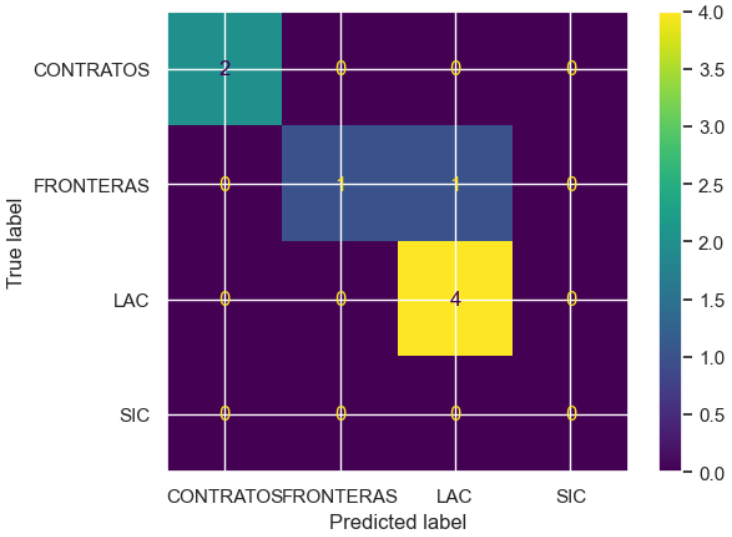

Figura 14. Matriz de confusión Logistic Regression

2. *Support Vector Machine*

Para el caso de SVM se tiene que similar al modelo de *Logistic Regression* no se consideran casos de prueba para el SIC, por tanto, los resultados podrían ser no fiel representación de los datos (Ver Figura 15)

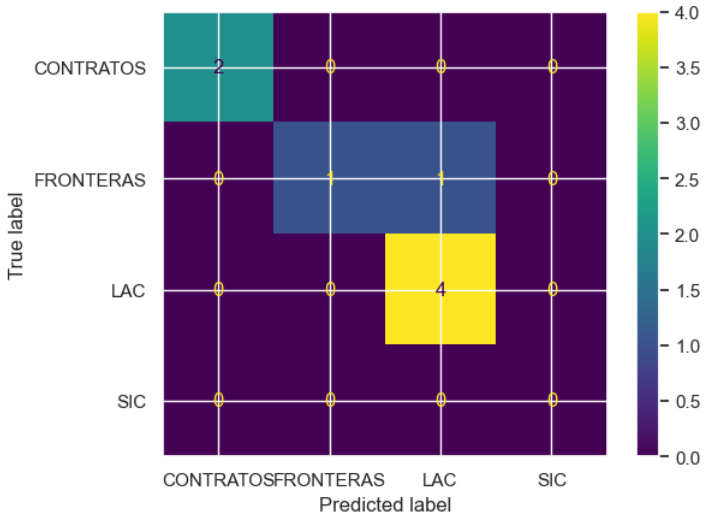

Figura 15. Matriz de confusión SVM

3. *Naive Bayes*

Para el caso de Naive Bayes se tiene que el modelo al igual que *SVM* y *Logistic Regression*, el modelo requiere mayor cantidad de datos de prueba, especialmente para SIC. (Ver Figura 16)

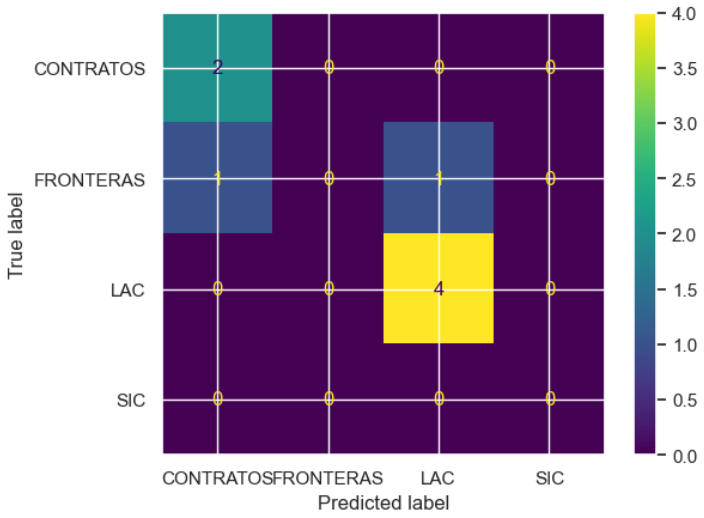

Figura 16. Matriz de confusión Naive Bayes.

Por tanto, al analizar la métrica F1 score la cual permite una comparación directa entre modelos y estrategias, para todos los Casos (Ver Figura 17), podemos evidenciar que a pesar de que el Caso 3 tiene mejores resultados que el Caso 1 y 2 y si se analizan por sí sólo se podría concluir que usando modelos de *Logistic Regression* o *SVM* el proceso podría ser aplicable, sin embargo, si realizamos el análisis conjunto de las matrices de confusión se puede evidenciar que los resultados podrían no ser una confiable representación de la realidad debido a la poca cantidad de datos de entrenamiento y prueba.

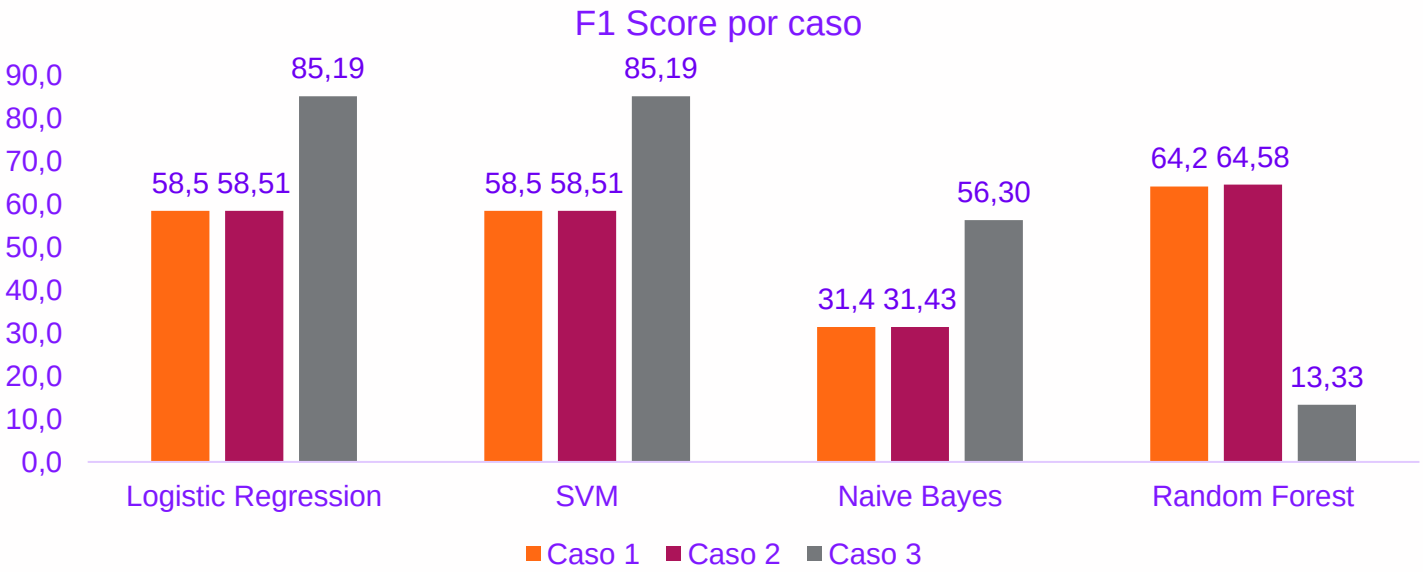

Figura 17. Evaluación F1 Score por Caso y por Modelo

VI. RECOMENDACIONES Y TAREAS ADICIONALES

Según los resultados encontrados en el proyecto, se recomienda crear una nueva estrategia de manejo de datos, la cual se base en una fragmentación por artículos clasificando artículo por artículo, donde se deberá:

1. Dividir las resoluciones artículo por artículo
2. Establecer estrategias de clasificación y obtención de datos.
3. Cambiar el flujo de código para poder considerar la información por artículo
4. Evaluar los resultados obtenidos por los diferentes casos.

Ahora, adicionalmente a estas tareas dentro del marco del proyecto de prácticas, también dentro de la dirección participé en:

1. **Implementación Resolución CREG 101 018 de 2023:**

Esta resolución habla sobre la vigilancia del poder de mercado de los agentes generadores dentro del SIN, en ese proceso participé dentro:

1. **Análisis de impactos regulatorios:** Analizando a que equipos impacta esta resolución, además que procesos se deben de cambiar y/o modificar para el cumplimiento de los hitos regulatorios.
2. **Gestionar equipo de implementación:** Una vez identificado los equipos y el personal de implementación, ayudé a gestionar que procedimientos se deben de cumplir y que hitos regulatorios se deben de cumplir
3. **Implementar:** Estos hitos se contemplan bajo los procedimientos publicados en la Circular CREG 068 de 2023[1] donde participé en la concepción y verificación de procedimientos para la aplicación de la resolución
4. **Crear planes B:** Desarrollé el Plan B para la aplicación del informe de poder de mercado, el cual ejecuta el equipo de Liquidación SIC de manera diaria, bajo los procedimientos de la Circular CREG 068 de 2023
5. **Liderar el desarrollo del aplicativo para el Reporte de Situación de Control:** Liderar la implementación del aplicativo de RSC dentro de RPM (Reporte de Parámetros del Mercado) para que los agentes generadores puedan realizar el reporte de situación de

[1] Circular CREG 068 de 2023: https://gestornormativo.creg.gov.co/gestor/entorno/docs/originales/Circular_CREG_068_2023/

control que se hablar en el Artículo 4 de la Resolución CREG 101 018 de 2023 y Artículo 5 de la Resolución 079 de 2018.

2. **Apoyo en el entendimiento regulatorio**

Apoyar en el entendimiento y gestión de comentarios de proyectos de resoluciones CREG y decretos del MME, dentro de las cuales se encuentran resoluciones que afectan resoluciones de Contratos, Garantías, Subastas, Liquidación SIC y Compensación del STR y STN. En las resoluciones en las cuales participé:

- Crg101-018-23
- Crg101-020-23
- Crg101-021-23
- Crg101-022-23
- Crg101-023-23
- Crg101-024-23
- Crg101-025-23
- Crg101-027-23
- Crg101-028-23
- Crg101-029-23
- Crg701-016-23
- Crg701-017-23
- Crg701-018-23
- Crg701-019-23
- Crg701-020-23
- Crg701-021-23
- Crg701-022-23
- Crg701-023-23
- Crg701-023A-23
- Crg701-024-23
- Crg701-025-23

3. **Liderar el Proyecto de Simplex Operativo – Ideal y Migración del Modelo**

Donde fui el encargado de implementar el módulo de despacho ideal, que reemplazará al DRP, aplicativo en el que se ejecuta el despacho diario y que requiere una migración a sistemas más modernos. Además, también de participar en la migración del modelo matemático de despacho de OPL a GAMS usando Gurobi cómo optimizador. Estos proyectos son claves para el desarrollo de las actividades de XM y en las cuales tuve un rol activo en el buen desarrollo de estos.

VII. CONCLUSIONES

1. Las estrategias de procesamiento de datos, a pesar de que no son suficientes para la correcta clasificación de las resoluciones, presentan una base para futuros trabajos los cuales, podrían implementar estrategias similares, tal vez no a nivel de resoluciones completas, si no, a nivel de artículos dentro de resoluciones.
2. La evaluación conjunta del F1 score y las matrices de confusión dan suficiente claridad del rendimiento promedio de cada modelo, además de presentar las herramientas necesarias para la comparación de los diferentes modelos.
3. Las matrices de confusión presentan de manera clara y eficiente el funcionamiento del clasificador, además de presentar la necesidad de aumentar el tamaño de la base de datos.
4. A pesar de que, con el *Caso 1 y 2* usando modelos de SVM y Regresiones logísticas se lograban clasificar de manera correcta categorías cómo LAC y Contratos, la poca cantidad de datos puede llevar a conclusiones apresuradas, por tanto, se es necesario aumentar el conjunto de datos de entrenamiento y prueba

REFERENCIAS

[1] N. Indurkhya and F. Damerau, *NATURAL LANGUAGE PROCESSING SECOND EDITION*, vol. 2. 2010. doi: https://doi.org/10.1201/9781420085938.

[2] H. Thimm, "Data modeling and NLP-based scoring method to assess the relevance of environmental regulatory announcements," *Environ Syst Decis*, vol. 43, no. 3, pp. 416–432, Sep. 2023, doi: 10.1007/s10669-023-09900-7.

[3] D. Jurafsky and J. H. Martin, "Speech and Language Processing an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition Draft Summary of Contents."

[4] Q. Li, J. Dong, J. Zhong, Q. Li, and C. Wang, "A neural model for type classification of entities for text," *Knowl Based Syst*, vol. 176, pp. 122–132, Jul. 2019, doi: 10.1016/j.knosys.2019.03.025.

[5] H. Chen, X. Fang, and H. Fang, "Multi-task prediction method of business process based on BERT and Transfer Learning," *Knowl Based Syst*, vol. 254, p. 109603, Oct. 2022, doi: 10.1016/j.knosys.2022.109603.

[6] S. Lai, J. Wu, Z. Ma, and C. Ye, "BTextCAN: Consumer fraud detection via group perception," *Inf Process Manag*, vol. 60, no. 3, p. 103307, May 2023, doi: 10.1016/j.ipm.2023.103307.

[7] "NLP Unlocked: Lemmatization #003. NLP unleashed: unleashing the power of… | by Arts2Survive | Medium." Accessed: Dec. 16, 2023. [Online]. Available: https://medium.com/@pankajchandravanshi/nlp-unlocked-lemmatization-003-c1bc406581b0

[8] *Reviewing the Stock of Regulation*. OECD, 2020. doi: 10.1787/1a8f33bc-en.

[9] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," Jan. 2012.

[10] J. Tolles and W. J. Meurer, "Logistic Regression," *JAMA*, vol. 316, no. 5, p. 533, Aug. 2016, doi: 10.1001/jama.2016.7653.

[11] Freedman and David A, *Statistical Models Theory and Practice*, vol. 1. 2009.

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Mach Learn*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[13] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Stat Comput*, vol. 14, no. 3, pp. 199–222, Aug. 2004, doi: 10.1023/B:STCO.0000035301.49549.88.

[14] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, “Support Vector Clustering,” *J. Mach. Learn. Res.*, vol. 2, pp. 125–137, Mar. 2002.

[15] “Funcionamiento de SVM - Documentación de IBM.” Accessed: Dec. 16, 2023. [Online]. Available: https://www.ibm.com/docs/es/spss-modeler/saas?topic=models-how-svm-works

[16] E. Anguiano-Hernández, “Naive Bayes Multinomial para Clasificación de Texto Usando un Esquema de Pesado por Clases,” 2009. Accessed: Dec. 17, 2023. [Online]. Available: https://ccc.inaoep.mx/~esucar/Clases-mgp/Proyectos/MGP_RepProy_Abr_29

[17] J. M. Cabrera Jiménez and F. O. Pérez Pérez, “Clasificación de Documentos usando Naive Bayes Multinomial y Representaciones Distribucionales,” México, 2011. Accessed: Dec. 17, 2023. [Online]. Available: https://ccc.inaoep.mx/~esucar/Clases-mgp/Proyectos/2011/Reporte_Proyecto_Clasificacion_de_Documentos.pdf

[18] Sriram, “Multinomial Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2023 | upGrad blog,” Artificial Intelligence. Accessed: Dec. 17, 2023. [Online]. Available: https://www.upgrad.com/blog/multinomial-naive-bayes-explained/

[19] R. Yehoshua, “Random Forests. Random forests is a powerful machine… | by Dr. Roi Yehoshua | Medium,” medium. Accessed: Dec. 17, 2023. [Online]. Available: https://medium.com/@roiyeho/random-forests-98892261dc49

[20] J. G. Gómez Ramírez, “Métricas De Evaluación De Modelos En El Aprendizaje Automático,” DataSource.ai. Accessed: Dec. 17, 2023. [Online]. Available: https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico

[21] Snigdho8869, “Multiclass Text Classification with Machine Learning and Deep Learning,” *GitHub*. GitHub, Apr. 09, 2023. Accessed: Dec. 17, 2023. [Online]. Available: https://github.com/Snigdho8869/Multiclass-Text-Classification/tree/main