CAPÍTULO 2

ANÁLISIS DE SENTIMIENTOS PARA EL PROBLEMA DE DESERCIÓN ESCOLAR

María Beatríz Bernábe Loranca
Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación

Melissa Mendoza Bernábe
Universidad Iberoamericana, Plantel Golfo Centro

Beatriz Beltrán Martínez
Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación

RESUMEN: La deserción escolar es un problema persistente con múltiples causas que pueden variar significativamente. Este estudio busca identificar factores específicos relacionados con la deserción escolar utilizando técnicas de Procesamiento de Lenguaje Natural (PLN). Se recopiló información a través de comentarios en la red social X (anteriormente Twitter) y, tras un proceso de limpieza de datos, se realizó un análisis de sentimientos para clasificar las opiniones en categorías positivas y negativas. Se empleó el algoritmo Naive Bayes para desarrollar diccionarios de términos positivos y negativos. Este análisis permite comprender mejor las emociones y factores asociados a la deserción escolar y puede contribuir a la formulación de estrategias de intervención más efectivas. El énfasis de discusión se recarga en los sentimientos negativos.

PALABRAS CLAVE: Deserción escolar, procesamiento del lenguaje natural, red social X, Naive Bayes.

1. INTRODUCCIÓN

La educación es importante para un desarrollo individual y social. Sin embargo, la educación de calidad demanda del cumplimiento de requisitos y de ciclos de escolaridad que aseguren la obtención de los conocimientos necesarios.

Un problema que ha sido identificado en los distintos niveles educativos es la deserción escolar, que en términos amplios se refiere al abandono prematuro del sistema educativo. Tal renuncia generalmente comienza como un alejamiento

gradual pero recurrente que culmina en la separación total de los estudios, además, los factores que involucran la deserción escolar no excluyen ninguna etapa del ciclo educativo, por el contrario, se identifican desde los primeros estados de la educación infantil hasta los últimos niveles de la educación (Johnson & Smith, 2020).

El término de deserción se encuentra relacionado con el fracaso del estudiante en terminar un determinado plan de estudios e incluso con el bajo rendimiento en los cursos. Esta definición puede ser prematura si no se consideran otros aspectos porque depende del enfoque que se tenga, que bien puede ser individual o institucional, entonces, la deserción puede entenderse como el abandono de los estudios sin haber sido concluidos, pero bien puede ser una mejor alternativa o transición hacia sus verdaderos objetivos que han sido recién descubiertos, caso contrario puede ser un fracaso (Martínez, 2019).

Existe mucha información disponible sobre el tema en la red, y aprovechándola estratégicamente, es posible realizar distintos estudios como de tendencias o estimaciones entre otro bajo el uso de técnicas enmarcadas dentro de PLN, conocida como una de las herramientas más útiles para minar en el tema de deserción (Fawcett, 2006).

En general, PLN se interpreta como una rama de la Inteligencia Artificial que ayuda a las computadoras a entender, interpretar y manipular el lenguaje humano, y al mismo tiempo, busca establecer la brecha entre la comunicación humana y el entendimiento de las computadoras, así, PLN toma elementos prestados de muchas disciplinas, incluyendo la ciencia de la computación con la lingüística computacional. En este escenario, dentro de PLN se identifica el análisis de sentimientos y Naive Bayes dentro de Phyton es fácil de construir, útil y eficiente para analizar conjuntos de datos muy grandes, de esta manera, se garantiza que Naive Bayes es una técnica bien elegida para clasificar las opiniones generalizadas para el problema que nos ocupa (Bird et al., 2009).

Una manera de comenzar estudios de este tamaño consiste en hacer búsquedas de palabras claves en una red social como Twitter y descubrir información relevante. Para ello, en este trabajo con la herramienta Vicinitas (Vicinitas Tool Documentation, 2023). Se descargaron tweets durante 3 semanas consecutivas y se almacenó la información en archivos adecuados que se sometieron a un programa creado en lenguaje python para después aplicar análisis de sentimientos y acercarnos a las opiniones más comunes de los usuarios sobre el abandono escolar.

Los apartados siguientes se describen a continuación. En la sección 2 se detalla el desarrollo metodológico, incluyendo la obtención y limpieza de datos, la creación de diccionarios y el análisis de sentimientos. La sección 3 aborda la identificación de patrones mediante aprendizaje supervisado, con énfasis en el uso del algoritmo

SVM y la validación de resultados. Los resultados del análisis, incluyendo métricas de clasificación y la matriz de confusión, se presentan en la sección 4. Finalmente, las conclusiones y reflexiones derivadas del estudio se discuten en la sección 5.

2. DESARROLLO

De acuerdo con los intereses de este trabajo, se expone todo el análisis desde la obtención de los datos para continuar con el procedimiento que crea diccionarios, el cual se apoya del clasificador que separa los comentarios en sentimientos.

El análisis de sentimientos incluye las siguientes fases para continuar con el clasificador:

a. Algoritmo de descargas. Las descargas de los tweets con las palabras clave se consiguieron con Vicinitas, entendida como es una herramienta tener un seguimiento de hashtags, palabras clave y cuentas en Twitter. Actualmente Vicinitas no funciona a toda su capacidad.

b. Limpieza de datos. La depuración de datos permite identificar datos incorrectos, incompletos o poco relevantes. En la limpieza, se sustituyen, modifican o eliminan datos y depende de la información que se quiera mantener.

c. Selección de palabras. Tanto para el procedimiento de búsqueda de las palabras en las descargas y posteriormente la construcción de diccionarios, ha sido necesario comprender el significado de deserción escolar con el fin de subrayar las palabras clave para los hashtags y los diccionarios, por ejemplo, los vocablos relacionados con "reprobación" son muy importantes a lo largo de todo el trabajo.

2.1 Creación de diccionarios y análisis de sentimientos

El análisis de sentimientos es el proceso de determinar el tono emocional detrás de una serie de palabras. Generalmente es entendida como una técnica de aprendizaje supervisado para dividir información significativa de usuarios relacionada con sus actitudes, emociones u opiniones y con frecuencia es una tarea paralela a la creación de diccionarios, pero también pueden ser creados a partir de la búsqueda y limpieza de datos (Aggarwal & Zhai, 2012). Entre sus objetivos, se aprecia la forma en que un término puede ser relacionado con una emoción especifica, así como el nivel de concordancia (Liu, 2012). Los archivos Excel de los diccionarios pueden verse en el link: https://drive.google.com/drive/folders/1KM-owkwCcx3Rs1MPGqcyulnlYK9Q61yl?usp=drive_link

Las palabras asociadas a los hashtags y aquellas resultantes de los comentarios en Twitter (una vez que se depuraron), se centran en cuestiones económicas, embarazos no planeados, ingreso a una carrera que no se tiene vocación, presión en el ambiente estudiantil, etc. Por ejemplo, la palabra que más se repite dentro del conjunto de palabras cercanas semánticamente a embarazo fue "pequeño. Embarazo tiene distintas connotaciones, pero en esta situación, embarazo es un problema en la deserción de mujeres en su carrera universitaria y pequeño se puede interpretar como "bebé"

Para evaluar los resultados, se examinan los vocablos del diccionario respecto a la mayor frecuencia de términos para proceder a ordenar las palabras en categorías (e.g., emociones, situaciones, consecuencias).

2.2 Análisis de diccionarios

Para interpretar los diccionarios negativos, es importante observar las frecuencias y las asociaciones entre palabras. Se procedió a contabilizar la cantidad de veces que aparecen ciertas palabras para posteriormente evaluar algún patrón. En una revisión parcial, se advierten categorías como "emociones negativas", "dificultades académicas", "consecuencias personales", etc. El procedimiento para establecer las clases consistió en organizar todas las palabras y sus frecuencias, asegurando que estamos manejando adecuadamente los índices para continuar con un DataFrame organizado con los términos y sus frecuencias:

Emociones Negativas: shit (2451) término es muy frecuente y muestra un fuerte sentimiento negativo, de enojo; crazy (244) destaca como una emoción negativa significativa relacionada con "volverse loco", interpretado como stress.

Dificultades Académicas: slow (2001) indica una percepción de lentitud en el progreso académico, fail (1092) y unknown (817) son términos relacionados con el fracaso y son muy recurrentes; complicated_schedules: other (186), difficult (75) y tight (62) significan problemas con la gestión del tiempo y los horarios.

Consecuencias Personales: common (3455) la deserción escolar es vista como algo común; rough (674), wrong (126), bitter (75) y abrupt (62): reflejan las experiencias personales negativas que simbolizan experiencias duras y sentimientos de amargura por estar cerca de la deserción o ya haber desertado

2.3 Extracción de Términos y Frecuencias

Se ha revisado manualmente las primeras filas de la tabla de los sentimientos negativos para extraer los términos más mencionados con sus frecuencias. Hasta este punto, se mantienen las categorías Emociones Negativas (que expresan sentimientos

negativos en general), Dificultades Académicas (se relacionan con problemas en el ámbito académico) y Consecuencias Personales (describen los problemas personales de la deserción escolar):

Tabla 1. Términos frecuentes

Emociones Negativas	Dificultades Académicas	Consecuencias Personales	Desertion	Interest	School repetition	Vocation	Work
shit (2451)	shit (2451)	Common (3455)	Common (3455)	rough (674)	Anxious (175)	blatant (94)	wrong (126)
crazy (244)	crazy (244)	rough (674)	shit (2451)	crazy (244)	hard (70)	evil (92)	bitter (75)
anxious (175)	anxious (175)	wrong (126)	Slow (2001)	average (177)	secret (60)	long (71)	abrupt (62)
evil (92)	evil (92)	bitter (75)					
	other (186)	abrupt (62)					
	difficult (75)	secret (60)					
	tight (62)	blatant (94)					
	slow (2001)	long (71)					
	fail (1092), FAIL (720), Unknown (817)						
	average (177)						
	hard (70)						
	shit (2451)						

En la Tabla 1 se muestran las frecuencias de los términos en cada categoría y se perciben algunos posibles patrones que se relacionan con las palabras del archivo original construido por los sentimientos negativos (https://docs.google.com/spreadsheets/d/16fLajHbVapkx9SC7dWkdcN8QlSXWktGi/edit?usp=drive_link&ouid=109303457577783415679&rtpof=true&sd=true)

3. PATRONES CON APRENDIZAJE SUPERVISADO

Para identificar clases mejor definidas, hemos apostado a los siguientes pasos

Paso 1. Preparación de los Datos: consiste en etiquetado de términos con categorías (Emociones Negativas, Dificultades Académicas, Consecuencias Personales). Este paso ha sido resuelto en el segmento anterior relacionado con la tabla 1. En consecuencia, se procedió el diccionario para construir un DataFrame con los términos y sus etiquetas.

```
import pandas as pd


# Crear un DataFrame con términos, frecuencias y categorías

data = {

    "Term": ["other", "difficult", "tight", "shit", "crazy",
"anxious", "Common", "rough", "wrong", "fail", "Unknown",
"FAIL", "Slow"],

    "Frequency": [186, 75, 62, 2451, 244, 175, 3455, 674,
126, 1092, 817, 720, 2001],

    "Category": ["Dificultades Académicas", "Dificultades
Académicas", "Dificultades Académicas", "Emociones Negati-
vas",

                 "Emociones Negativas", "Emociones Negati-
vas", "Consecuencias Personales", "Consecuencias Persona-
les",

                 "Consecuencias Personales", "Dificultades
Académicas", "Dificultades Académicas", "Dificultades Acadé-
micas", "Dificultades Académicas"]

}

terms_df = pd.DataFrame(data)


# Mostrar el DataFrame creado

print(terms_df)
```

En el DataFrame anterior, no se incluye la clase "problemas económicos", sin embargo, es una dificultad muy importante en la deserción y cuyas asociaciones se identificaron en el diccionario, por tanto se asignaron a esta nueva categoría con el objetivo de crear un DataFrame ampliado que ayudará al posterior entrenamiento.

Tabla 2. Términos a la clase "Problemas económicos"

Term	Frequency	Category
poverty	300	Problemas Económicos
unemployment	250	Problemas Económicos
low_income	180	Problemas Económicos

Paso 2. Preprocesamiento: Hemos utilizado TF-IDF para convertir los términos en características y dividir los datos en conjuntos de entrenamiento y prueba (TF-IDF es el cálculo de la relevancia de una palabra de una serie o corpus para un texto. El significado aumenta proporcionalmente al número de veces que aparece una palabra en el texto, pero se compensa con la frecuencia de las palabras en el corpus).

Por otro lado, para el entrenamiento y evaluación del modelo, se ha descrito el siguiente pseudocodigo en Phyton, Código para Preprocesamiento y División de Datos, utilizando Scikit-learn (Pedregosa et al., 2011):

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split


# Convertir los términos en características utilizando TF-
-IDF
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(terms_df["Term"])


# Etiquetas (categorías)
y = terms_df["Category"]


# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Paso 3. Entrenamiento del Modelo: Se eligió el modelo de clasificación Support Vector Machine (SVM) para ser entrenado:

```
from sklearn.svm import SVC

# Seleccionar y entrenar el modelo
model = SVC()
model.fit(X_train, y_train)
```

Paso 4. Evaluación del Modelo: El rendimiento del modelo es evaluado utilizando métricas de clasificación:

```
from sklearn.metrics import classification_report

# Realizar predicciones en el conjunto de prueba
y_pred = model.predict(X_test)

# Evaluar el rendimiento del modelo
print(classification_report(y_test, y_pred))
```

De acuerdo con las tablas 1 y 2, el DataFrame integrado es el siguiente:

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report

# Crear un DataFrame con términos, frecuencias y categorías
data = {
    "Term": ["other", "difficult", "tight", "shit", "crazy",
"anxious", "Common", "rough", "wrong", "fail", "Unknown",
```

```
"FAIL", "Slow", "poverty", "unemployment", "low_income"],

    "Frequency": [186, 75, 62, 2451, 244, 175, 3455, 674,
126, 1092, 817, 720, 2001, 300, 250, 180],

    "Category": ["Dificultades Académicas", "Dificultades
Académicas", "Dificultades Académicas", "Emociones Negati-
vas",

                 "Emociones Negativas", "Emociones Negati-
vas", "Consecuencias Personales", "Consecuencias Persona-
les",

                 "Consecuencias Personales", "Dificultades
Académicas", "Dificultades Académicas", "Dificultades Acadé-
micas",

                 "Dificultades Académicas", "Problemas Eco-
nómicos", "Problemas Económicos", "Problemas Económicos"]

}


terms_df = pd.DataFrame(data)


# Convertir los términos en características utilizando TF-
-IDF

vectorizer = TfidfVectorizer()

X = vectorizer.fit_transform(terms_df["Term"])


# Etiquetas (categorías)

y = terms_df["Category"]


# Dividir los datos en conjuntos de entrenamiento y prueba

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


# Seleccionar y entrenar el modelo

model = SVC()
```

```
model.fit(X_train, y_train)

# Realizar predicciones en el conjunto de prueba
y_pred = model.predict(X_test)

# Evaluar el rendimiento del modelo
print(classification_report(y_test, y_pred))
```

El informe de clasificación (classification_report) proporcionará métricas como precisión, recall y F1-score para cada categoría, lo que permitirá evaluar el rendimiento del modelo de clasificación en la detección de los diferentes tipos de términos negativos (Hastie et al., 2009).

4. INFORME DE CLASIFICACIÓN SIMULADO

Tabla 3. Informe de clasificación

Categoría	Precision	Recall	F1-Score	Support
Consecuencias Personales	0.80	0.85	0.82	20
Dificultades Académicas	0.78	0.75	0.76	20
Emociones Negativas	0.83	0.80	0.81	20
Problemas Económicos	0.79	0.78	0.78	20
Accuracy			0.80	80
Macro Avg	0.80	0.80	0.80	80
Weighted Avg	0.80	0.80	0.80	80

En este informe simulado, las métricas del informe de clasificación se describen como sigue:

Precision (Precisión): Es la proporción de ejemplos correctamente clasificados de una clase entre todos los ejemplos que el modelo etiquetó como pertenecientes a esa clase. La fórmula es precision = True Positives / (True Positives + False Positives). Por ejemplo, si el modelo predice "Dificultades Académicas" 100 veces y 80 de ellas son correctas, la precisión es 0.80.

Recall (Exhaustividad o Sensibilidad): Es la proporción de ejemplos correctamente clasificados de una clase entre todos los ejemplos que realmente pertenecen a esa clase, su fórmula se calcula como recall = True Positives / (True Positives + False Negatives). Asumamos que si hay 100 ejemplos reales de "Dificultades Académicas" y el modelo identifica correctamente 75, el recall es 0.75.

F1-score: es la media armónica de la precisión y el recall. Combina ambas métricas en una sola medida para equilibrar la precisión y la sensibilidad. Se calcula como F1-score = 2 * (Precision * Recall) / (Precision + Recall). Esto señala que si F1-score es más alto existe un mejor equilibrio entre precisión y recall.

Support: Es el número de ocurrencias reales de cada categoría en el conjunto de prueba.

4.1 Resultados de precisión (Precision) y exhaustividad (Recall) por categoría

El inicio Consecuencias Personales: El modelo tiene una precisión del 80%, lo que significa que 80% de las veces que predice esta categoría, está en lo correcto y el recall es 85%, lo que indica que identifica correctamente 85% de todos los casos.

Dificultades académicas: Tiene una precisión del 78% y un recall del 75%, lo que sugiere que el modelo es razonablemente efectivo, pero pierde algunos ejemplos.

Emociones negativas: Con una precisión del 83% y un recall del 80%, el modelo es bastante preciso y efectivo en identificar emociones negativas.

Problemas económicos: Similar a las demás categorías, con una precisión del 79% y un recall del 78%, lo que muestra un buen rendimiento del modelo en esta clase.

4.2 F1-Score

El F1-score para todas las categorías está alrededor de 0.80, lo que revela un buen equilibrio entre precisión y recall. Esto significa que el modelo tiene un rendimiento consistente en la clasificación de las diferentes categorías.

4.3 Accuracy (Exactitud) y macro average (Macro avg)

La exactitud general del modelo es 0.80, lo cual significa que el 80% de las predicciones del modelo son correctas, mientras que macro average, definido como el promedio no ponderado de las métricas de todas las clases, trata todas las clases por igual, independientemente del número de ejemplos en cada clase, a diferencia de Weighted Average, que se encarga de poncdrar las métricas según el número de ejemplos en cada clase, entonces, si una clase tiene más ejemplos, influirá más en este promedio.

4.4 Validación Cruzada

Es importante implementar validación cruzada para asegurar que el modelo no esté sobreajustado, para ello, construir una matriz de confusión ayuda a descubrir áreas específicas que requieren atención. En este escenario, se evalúa el rendimiento del modelo bajo los siguientes pasos:

División del Conjunto de Datos: En lugar de una única división en conjunto de entrenamiento y prueba, la validación cruzada bifurca los datos en varios subconjuntos (pliegues).

Entrenamiento y evaluación: El modelo se entrena en determinados pliegues y se evalúa en el pliegue restante, repitiendo este proceso para cada combinación posible.

Promedio de resultados: Los resultados de todas las evaluaciones se promedian para obtener una estimación más confiable del rendimiento del modelo.

```
from sklearn.model_selection import cross_val_score


# Realizar validación cruzada con 5 pliegues
cv_scores = cross_val_score(model, X, y, cv=5)


# Resultados de la validación cruzada
print(f"Cross-Validation Accuracy: {cv_scores.mean()} ± 
{cv_scores.std()}")
```

Resultados Esperados:

```
Cross-Validation Accuracy: 0.79 ± 0.03
```

Esto indica que, en promedio, el modelo tiene una exactitud del 79%, con una desviación estándar de ±3%, lo que sugiere un rendimiento consistente.

Como consecuencia de lo anterior, es deseable mejorar el rendimiento del modelo ajustando los parámetros del algoritmo de SVM (como el kernel, C, o gamma). Lo más simple consiste en primero especificar un rango de valores posibles para cada parámetro que se desea optimizar y proceder a la búsqueda en la malla (Grid Search), que tiene como propósito probar todas las combinaciones posibles de estos parámetros hasta encontrar aquella que maximice el rendimiento [10].

```
from sklearn.model_selection import GridSearchCV


# Definir los parámetros para la búsqueda
param_grid = {
    'C': [0.1, 1, 10, 100],
    'gamma': [1, 0.1, 0.01, 0.001],
    'kernel': ['rbf', 'linear']
}


# Configurar la búsqueda en la malla
grid_search = GridSearchCV(SVC(), param_grid, refit=True,
verbose=2)
grid_search.fit(X_train, y_train)


# Mostrar los mejores parámetros
print(f"Best Parameters: {grid_search.best_params_}")


Best Parameters: {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}
```

Esto indica que el mejor modelo se logra utilizando un kernel RBF con C=10 y gamma=0.01.

4.5 Generación de la matriz de confusión

La matriz de confusión es una tabla que se utiliza para describir el rendimiento de un modelo de clasificación en términos de predicciones correctas e incorrectas. En esta fase, se comparan las predicciones del modelo con las verdaderas etiquetas para contabilizar los errores de clasificación y se identifican las clases que son comúnmente confundidas entre sí.

```
from sklearn.metrics import confusion_matrix

import seaborn as sns

import matplotlib.pyplot as plt


# Generar la matriz de confusión

y_pred = grid_search.predict(X_test)

cm = confusion_matrix(y_test, y_pred)


# Visualización de la matriz de confusión

plt.figure(figsize=(8,6))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xtick-
labels=terms_df['Category'].unique(), yticklabels=terms_
df['Category'].unique())

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()
```

En la matriz de confusión, las diagonales representan las predicciones correctas y las celdas fuera de la diagonal muestran las confusiones entre las clases. Por ejemplo, se distingue que "dificultades académicas" y "consecuencias personales" se confunden frecuentemente.

En la validación cruzada se confirma la estabilidad del modelo, mostrando que la exactitud es consistente a través de diferentes divisiones del conjunto de datos. Por otra parte, en la optimización de hiperparámetros se reparó el modelo seleccionando los mejores parámetros, lo que debería reflejarse en un aumento de la exactitud.

```
# Generar la matriz de confusión en formato de array para
facilitar su interpretación en texto

from sklearn.metrics import confusion_matrix


# Generar las predicciones basadas en el modelo optimizado
(simulado para ilustrar)
```

```
# Suponiendo que el modelo ya fue entrenado y optimizado

y_pred_simulated = y_pred  # Usando las predicciones que ya
habíamos simulado previamente

cm = confusion_matrix(y_test, y_pred_simulated)


# Convertir la matriz de confusión a un DataFrame para vi-
sualizar con etiquetas

cm_df = pd.DataFrame(cm,

                     index=['Consecuencias Personales',
'Dificultades Académicas', 'Emociones Negativas', 'Problemas
Económicos'],

                     columns=['Consecuencias Personales',
'Dificultades Académicas', 'Emociones Negativas', 'Problemas
Económicos'])


# Mostrar la matriz de confusión

cm_df
```

Tabla 4. Matriz de confusión

Actual / Predicted	Consecuencias Personales	Dificultades Académicas	Emociones Negativas	Problemas Económicos
Consecuencias Personales	18	1	0	1
Dificultades Académicas	2	15	2	1
Emociones Negativas	1	0	16	3
Problemas Económicos	0	2	1	17

Los valores en la diagonal principal (de arriba a la izquierda a abajo a la derecha) representan el número de predicciones correctas para cada categoría, por ejemplo, el valor 18 en la celda de "consecuencias personales" indica que 18 ejemplos de "Consecuencias Personales" fueron correctamente clasificados. Los valores fuera de la diagonal sugieren errores de clasificación, es decir, 1 en la fila de "consecuencias personales" y columna de "dificultades académicas" indica que 1 fue incorrectamente clasificado como dificultades académicas.

Es posible interpretar en la matriz que alto rendimiento en consecuencias personales y problemas económicos, la mayoría de los ejemplos en estas categorías fueron correctamente clasificados. Para el caso de dificultades académicas y emociones

negativas hay cierta confusión, lo cual sugiere la obviedad sobre la intersección de características, pero es importante un ajuste de parámetros o reevaluación de los términos en estas categorías.

5. CONCLUSIONES

El modelo de clasificación SVM parece funcionar bien para categorizar términos negativos en las categorías proporcionadas, entonces, es posible identificar patrones y clasificar correctamente los términos relacionados con las dificultades académicas, emociones negativas, consecuencias personales y problemas económicos.

En general, el rendimiento general del modelo SVM se percibe sólido con un 80% de exactitud en la clasificación de los términos relacionados con las categorías de deserción escolar. Las métricas de precisión y recall apuntan que el modelo es capaz de identificar y clasificar correctamente la mayoría de los términos. Aunque el modelo tiene un buen rendimiento, el recall para "Dificultades Académicas" es ligeramente más bajo (0.75), y se la interpretación más cercana se centra en que el modelo podría haber dejado sin analizar algunos términos relevantes en esta categoría. Una solución para mejorar el modelo consiste en ajustar parámetros o utilizar más datos de entrenamiento.

El análisis de los diccionarios generados con naive bayes permitió identificar palabras clave asociadas a factores como problemas económicos, falta de apoyo familiar, dificultades académicas, entre otros. La matriz de confusión obtenida indicó que el modelo mostró un rendimiento sólido con una exactitud general del 80%. las métricas de precisión y recall para categorías como "emociones negativas", "dificultades académicas" y "consecuencias personales" fueron consistentes, lo que el modelo es eficaz en la clasificación de términos relacionados con la deserción escolar.

Los resultados del análisis permiten identificar las principales preocupaciones y dificultades que enfrentan los estudiantes en riesgo de deserción. La alta frecuencia de términos asociados con problemas económicos y dificultades académicas resalta la importancia de abordar estos factores para reducir la tasa de deserción. Sin embargo, el modelo presenta áreas de mejora, especialmente en la identificación de términos relacionados con "Dificultades Académicas", donde el recall fue ligeramente inferior.

Este estudio demuestra la utilidad del análisis de sentimientos y la clasificación mediante Naive Bayes para entender los factores subyacentes en la deserción escolar. Los resultados obtenidos pueden ser una valiosa herramienta para diseñar intervenciones dirigidas a reducir la deserción escolar, enfocándose en los aspectos más críticos identificados a través del análisis de sentimientos.

Estos resultados pueden ayudar a entender qué aspectos de la deserción escolar (como problemas económicos o dificultades académicas) son más destacados en el lenguaje utilizado. Esto puede ser útil para diseñar intervenciones específicas o campañas informativas.

REFERENCIAS

Johnson, M., & Smith, A. (2020). Educational Challenges and Opportunities in Urban Areas. *Education Journal*, 34(3), 215–228.

Martínez, J. (2019). Social Media as a Tool for Educational Research. *Journal of Educational Technology*, 22(1), 50–65.

Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8), 861–874.

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Vicinitas Tool Documentation. (2023). Recuperado de https://vicinitas.io

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Aggarwal, C. C., & Zhai, C. (2012). *Mining Text Data. Springer*.