

```
<!--Estudio Mining-->
```

```
MODELO PREDICTIVO DE  
LA ASISTENCIA A  
CITAS MÉDICAS {
```

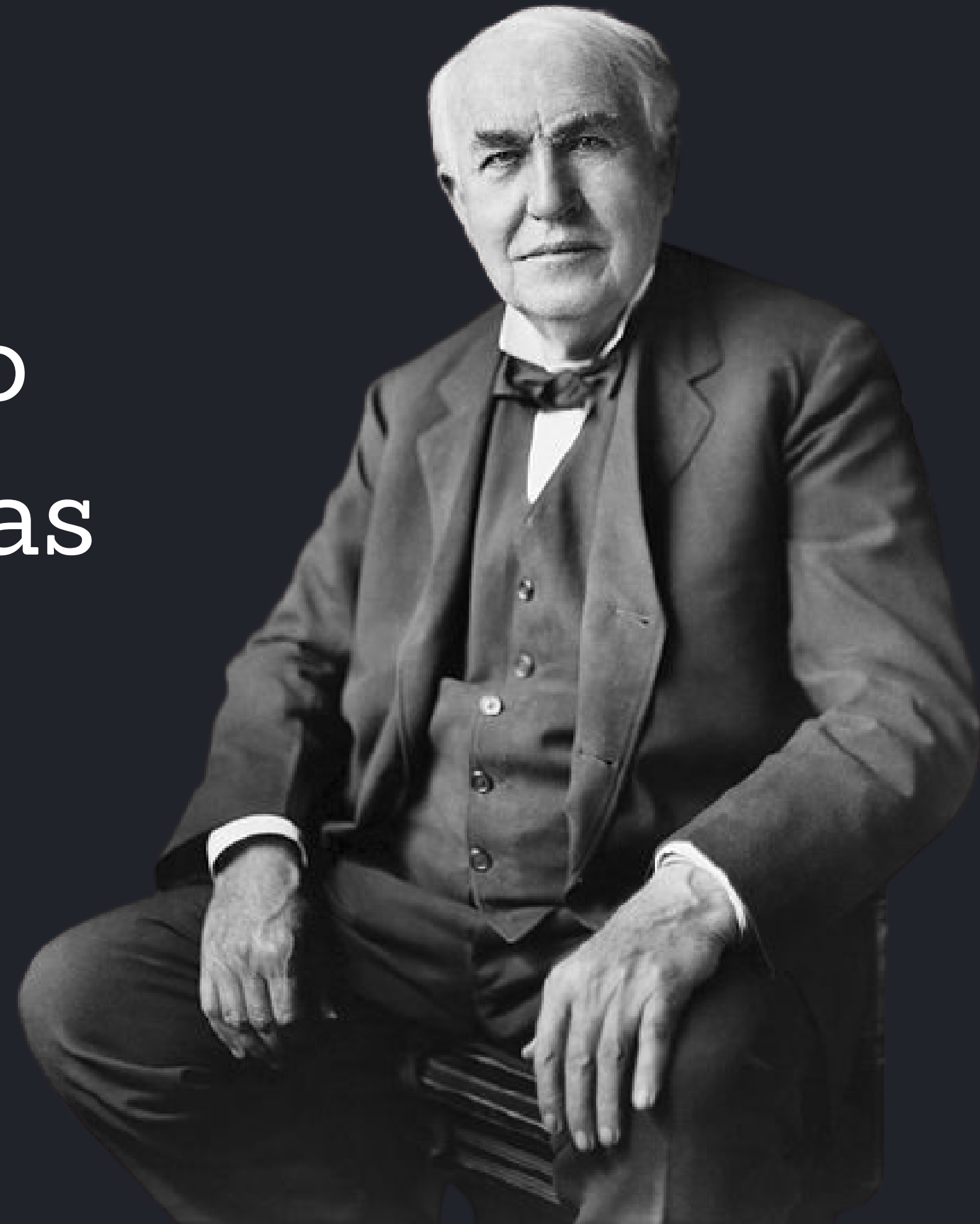
```
<Por="Mirian Cayo Molloni"/>
```

```
}
```



“ No he fracasado.
Sólo me he topado
con 10000 maneras
que no funcionan

Thomas Edison



En memoria de

HONORIO APAZA



Introducción {

Este trabajo explora el desarrollo y la implementación de un modelo predictivo de asistencia a citas médicas en una clínica dental, con el objetivo de optimizar la eficiencia operativa y mejorar la calidad de la atención proporcionada. A través de la aplicación de algoritmos de clasificación, evaluaremos el rendimiento del modelo utilizando métricas específicas y estableceremos un marco para la mejora continua, asegurando así una adaptación efectiva a las dinámicas cambiantes del entorno clínico y de los pacientes.



}

Objetivos{

void Objetivo_General(){

Desarrollar un modelo predictivo preciso y confiable que utilice técnicas de entrenamiento de machine learning, como Random Forest, VSM y otros, para predecir la asistencia a citas médicas con el fin de optimizar la gestión de recursos y mejorar la eficiencia en los servicios de atención al cliente en la clínica dental.

}

void Objetivos_Específicos(){

01

Implementar un sistema de recolección de datos.

02

Explorar y preprocesar los datos.

03

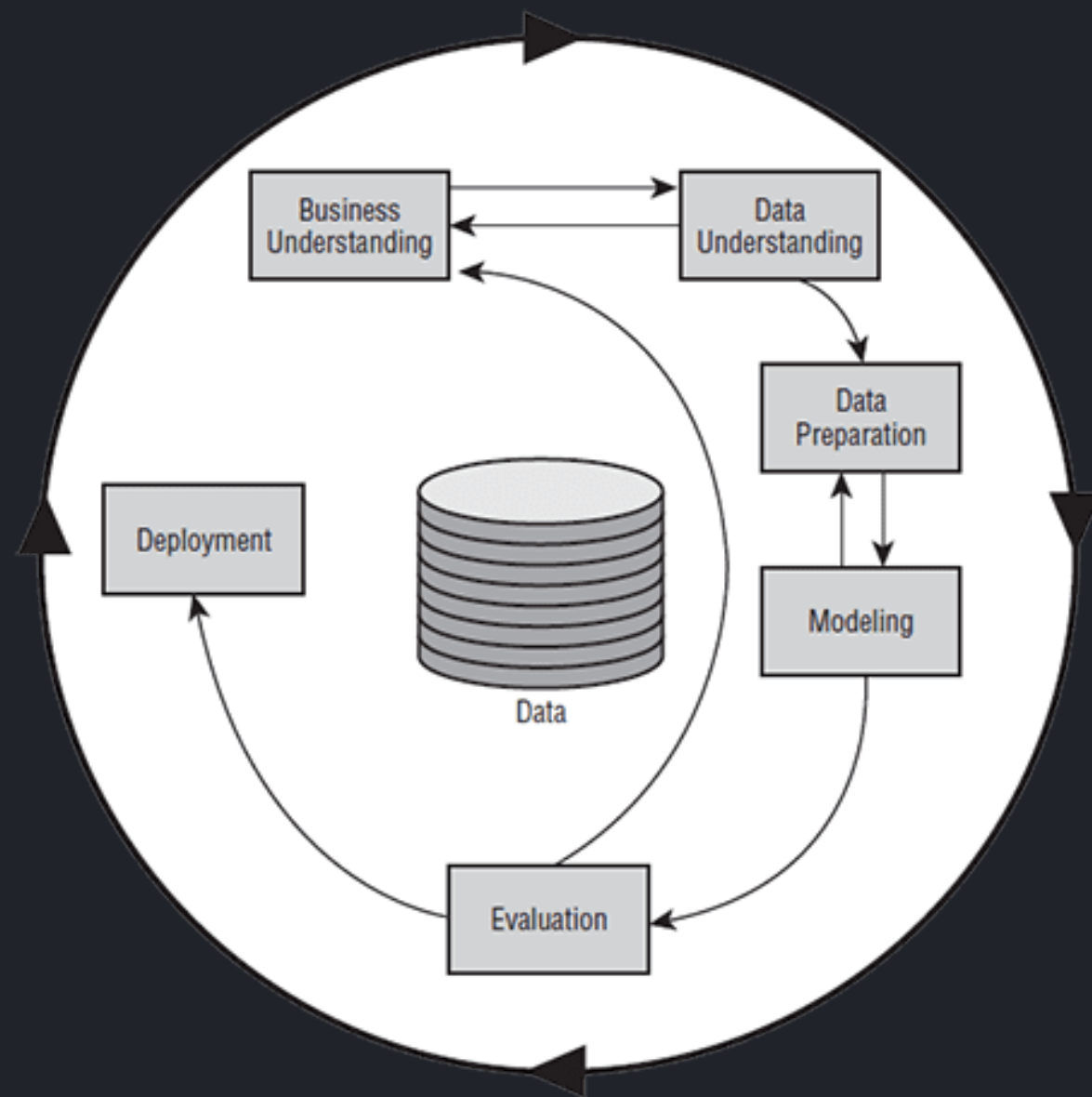
Desarrollar y entrenar modelos predictivos.

04

Evaluar y optimizar la precisión del modelo. }

}

Metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) {



- Comprensión empresarial: ¿qué necesita la empresa?
- Comprensión de datos: ¿qué datos tenemos o necesitamos? ¿Está limpio?
- Preparación de datos: ¿cómo organizamos los datos para el modelado?
- Modelado: ¿Qué técnicas de modelado debemos aplicar?
- Evaluación: ¿Qué modelo se adapta mejor a los objetivos comerciales?
- Implementación: ¿cómo acceden las partes interesadas a los resultados?

}

Imputacion de Datos {

Data imputation (Imputación de Datos)

```
[ ] import pandas as pd

# Reemplazar los valores NaN en las columnas específicas con ceros
columnas_a_imputar = ['pres_monto_total', 'descuento_total', 'descuento_estado']
data[columnas_a_imputar] = data[columnas_a_imputar].fillna(0)
print(data.isnull().sum())
```

```
pac_sex          0
cme_fech_inicial  0
cme_fech_final    0
cme_titulo        0
cme_estado        0
pres_monto_total  0
descuento_total   0
descuento_estado  0
dtype: int64
```

Verificar valores nulos

```
[ ] print(data.isnull().sum())
```

```
pac_sex          0
cme_fech_inicial  0
cme_fech_final    0
cme_titulo        0
cme_estado        0
pres_monto_total  0
descuento_total   0
descuento_estado  0
dtype: int64
```

```
[ ] print(data.dtypes)
data.head()
```

```
pac_sex          object
cme_fech_inicial  object
cme_fech_final    object
cme_titulo        object
cme_estado        object
pres_monto_total  float64
descuento_total   float64
descuento_estado  object
dtype: object
```

}

Categorical Encoding{

```
[ ] print(pd.value_counts(data['cme_e
```

```
CONCLUIDO      15013
ANULADO        1477
NO_ASISTIO     1466
ABIERTO         63
EN_PROCESO       8
Name: cme_estado, dtype: int64
```

```
1: ANULADO
2: CONCLUIDO
3: PROCESO
4: NO ASISTIO
```

Categorical Encoding (Codificación categórica)

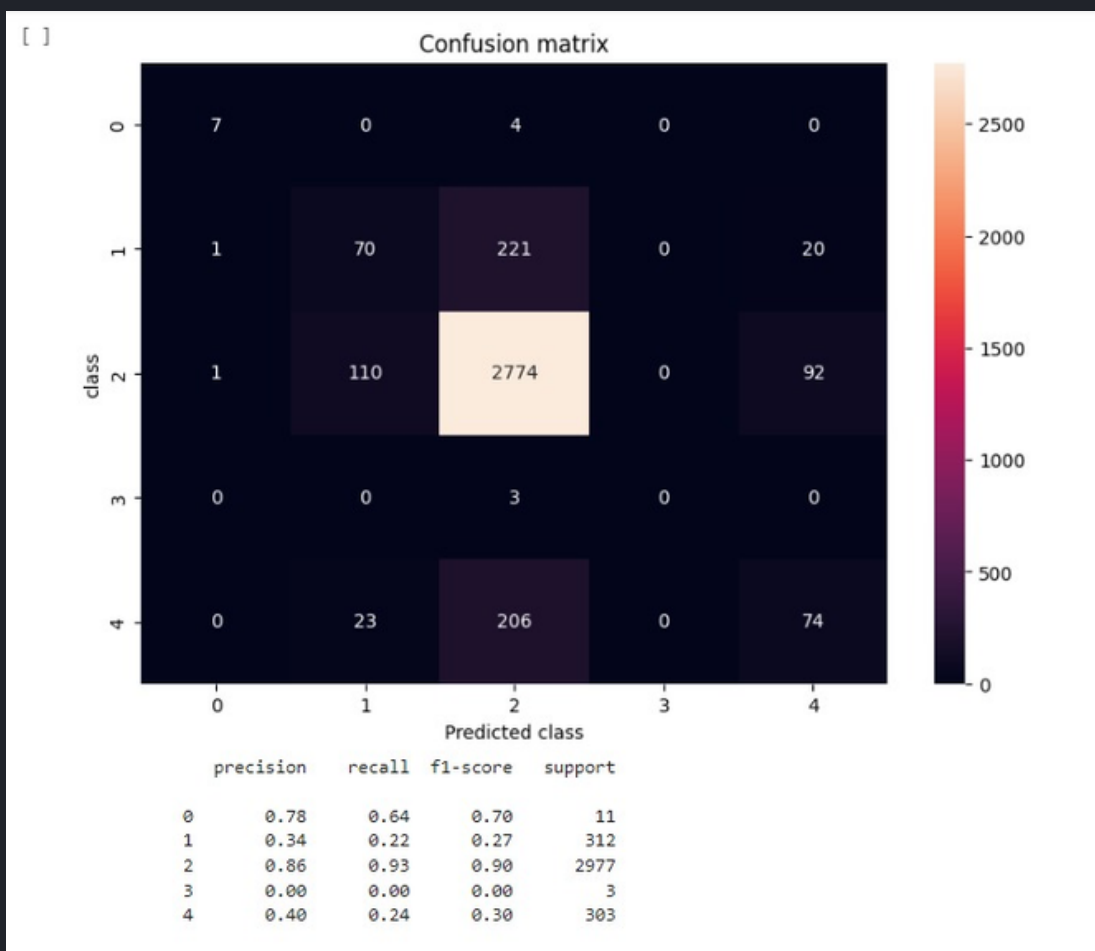
```
object_columns = data.select_dtypes(include=['object']).columns
data[object_columns] = data[object_columns].astype('category')
numeric_columns = data.select_dtypes(include=['int64', 'float64']).columns
categorical_data = data.drop(columns=numeric_columns)
category_mapping = dict(enumerate(categorical_data['cme_estado'].cat.categories))
for column in categorical_data.columns:
    categorical_data[column] = categorical_data[column].cat.codes
print(category_mapping)
#Se mostrará la data categórica recogida de la data principal
categorical_data.head()
```

```
{0: 'ABIERTO', 1: 'ANULADO', 2: 'CONCLUIDO', 3: 'EN_PROCESO', 4: 'NO_ASISTIO'}
```

	pac_sex	cme_fech_inicial	cme_fech_final	cme_titulo	cme_estado	descuento_estado
0	0	1576	1643	368	2	0
1	0	1621	1689	835	2	0
2	0	1645	1714	824	2	0
3	0	1721	1799	824	2	0
4	0	1740	1821	824	2	0

}

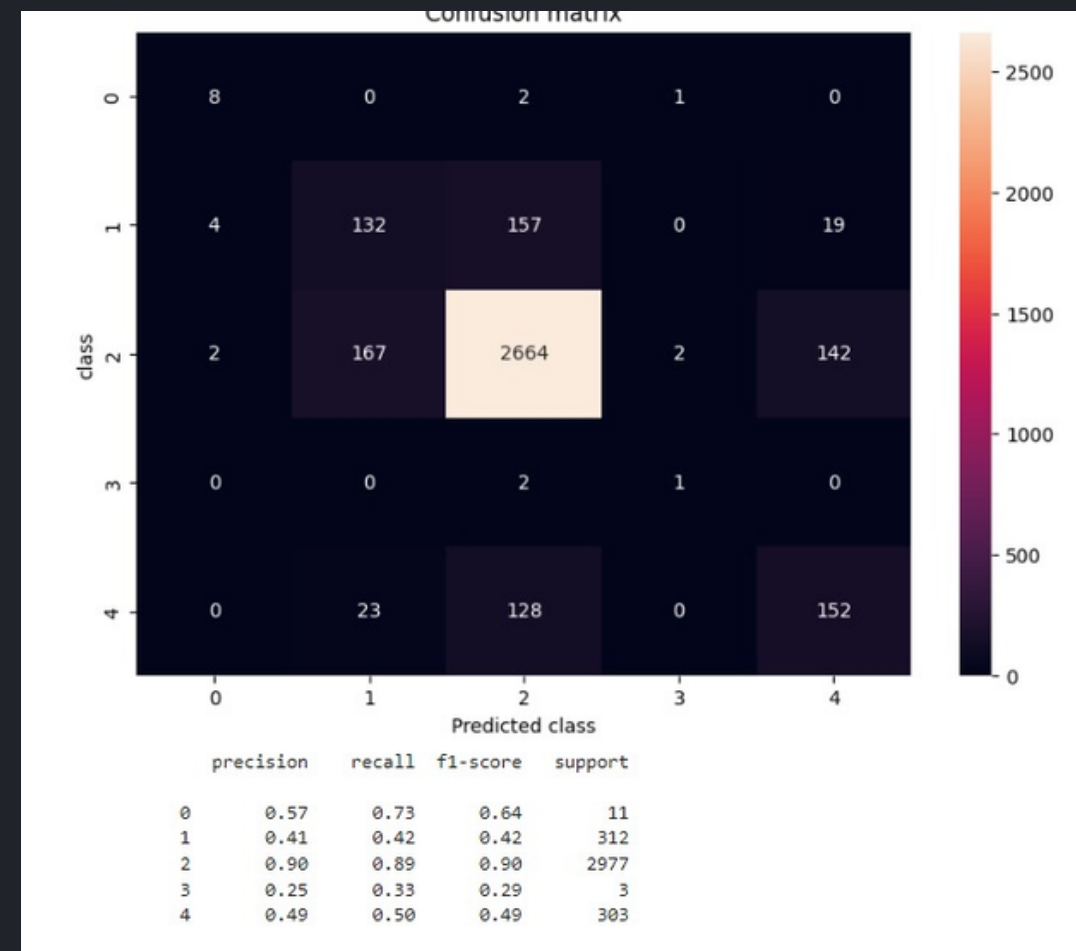
Random Forest {



}

Exactitud de 81%

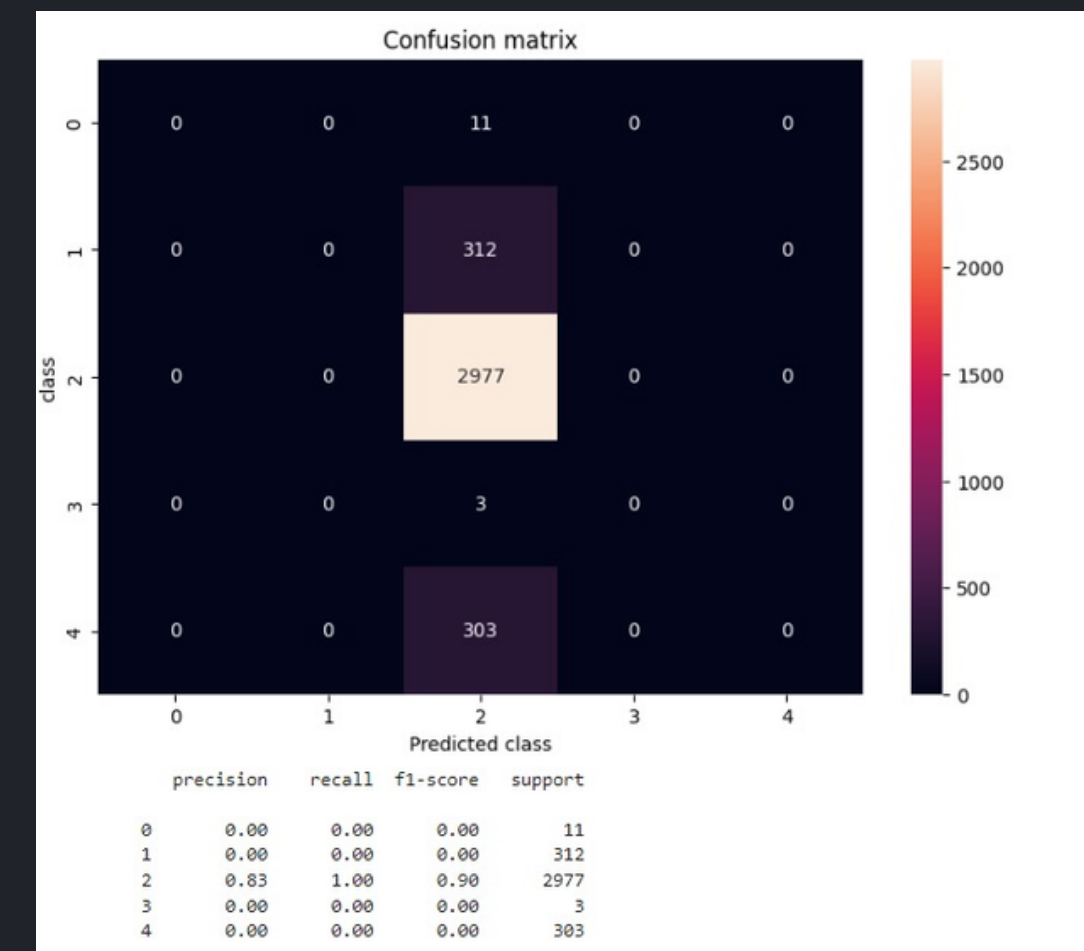
Desicion Tree {



}

Exactitud de 82%

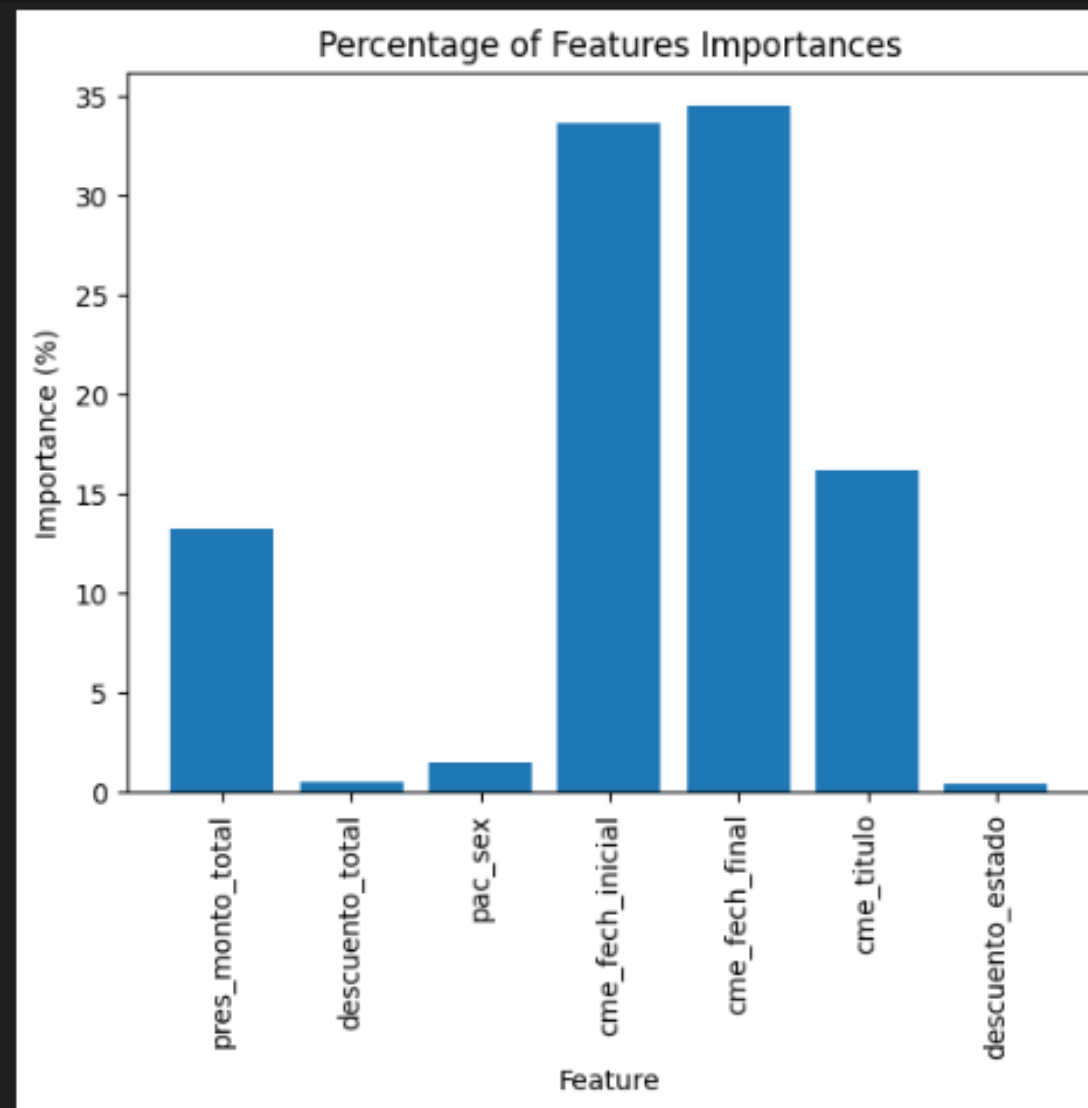
Regresion Logistica Multinomial {



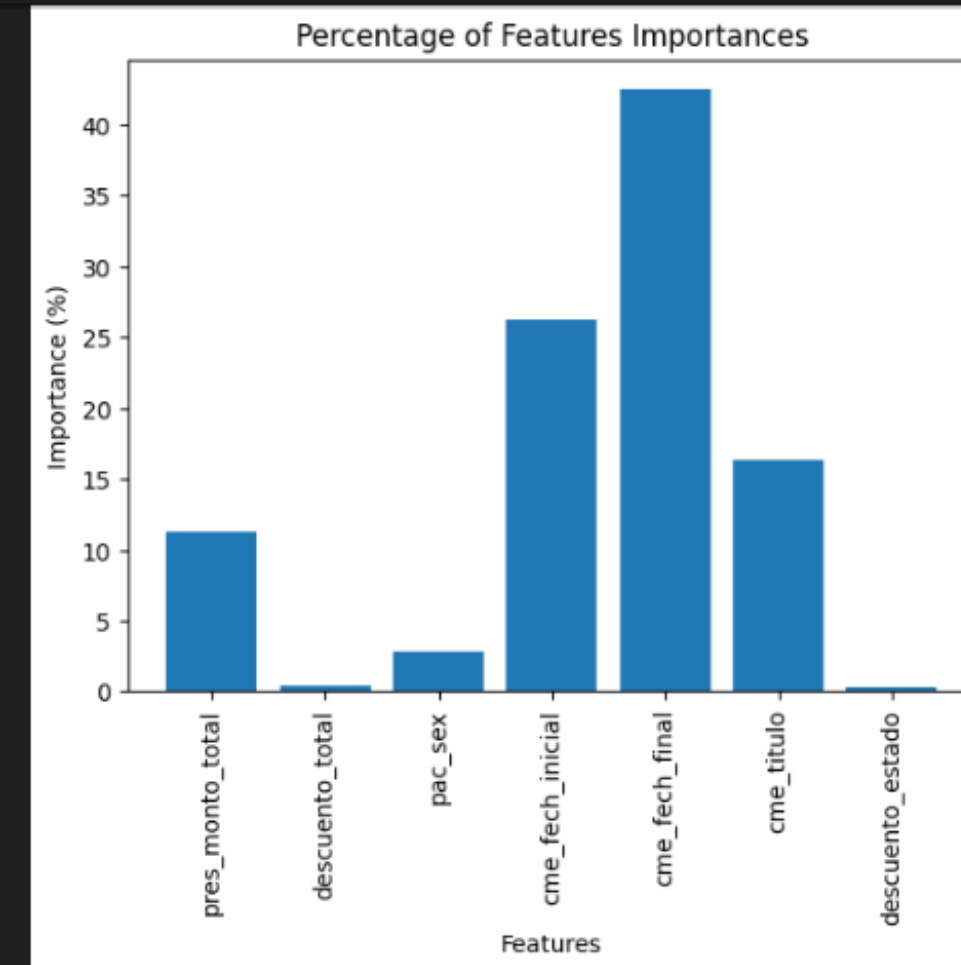
}

Exactitud de 83%

Features Importances { Verificación de columna relevante



```
pres_monto_total: 13.29%
descuento_total: 0.51%
pac_sex: 1.56%
cme_fech_inicial: 33.59%
cme_fech_final: 34.45%
cme_titulo: 16.21%
descuento_estado: 0.40%
```



```
... pres_monto_total: 11.32%
descuento_total: 0.42%
pac_sex: 2.89%
cme_fech_inicial: 26.29%
cme_fech_final: 42.48%
cme_titulo: 16.31%
descuento_estado: 0.30%
```

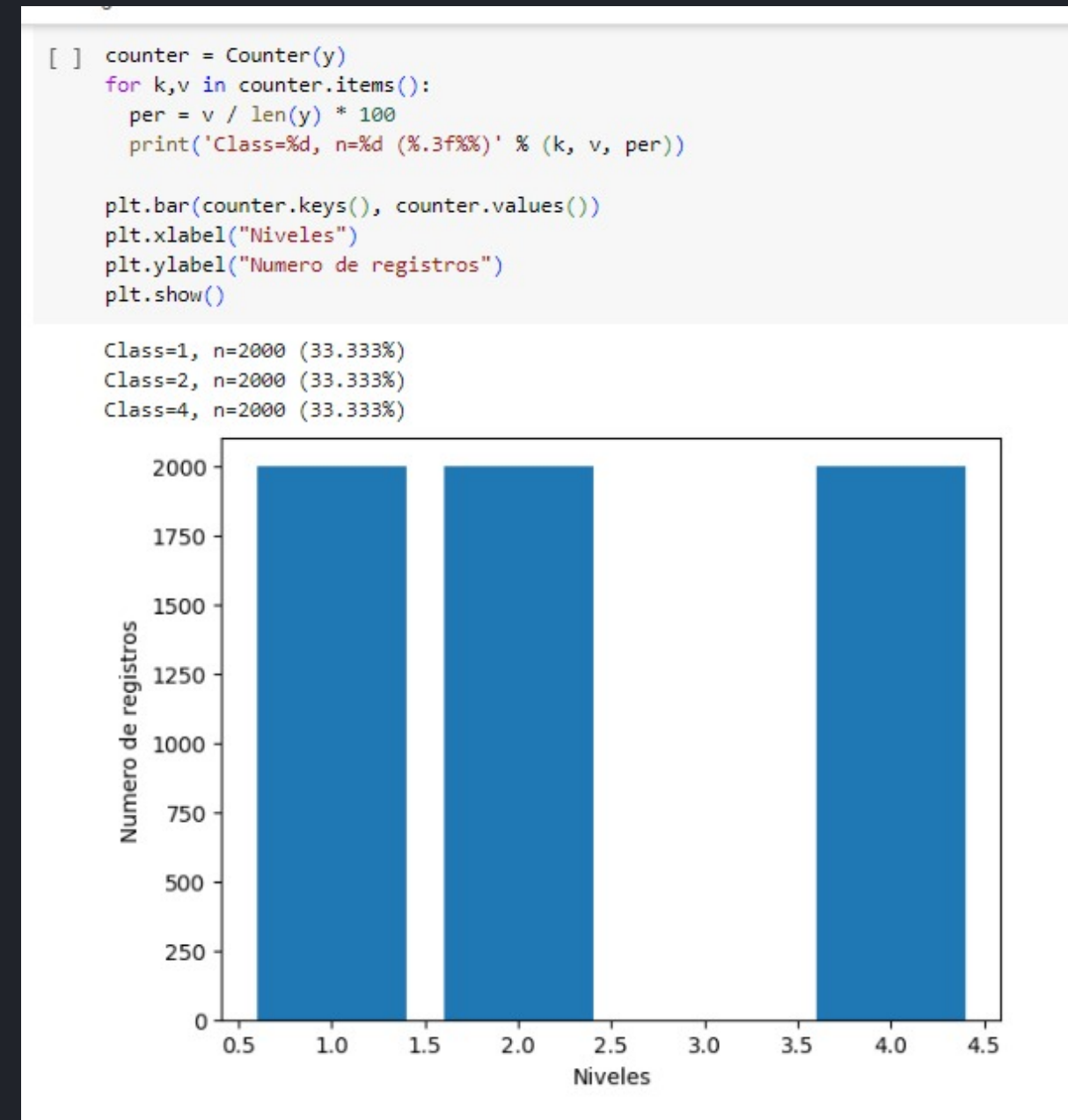
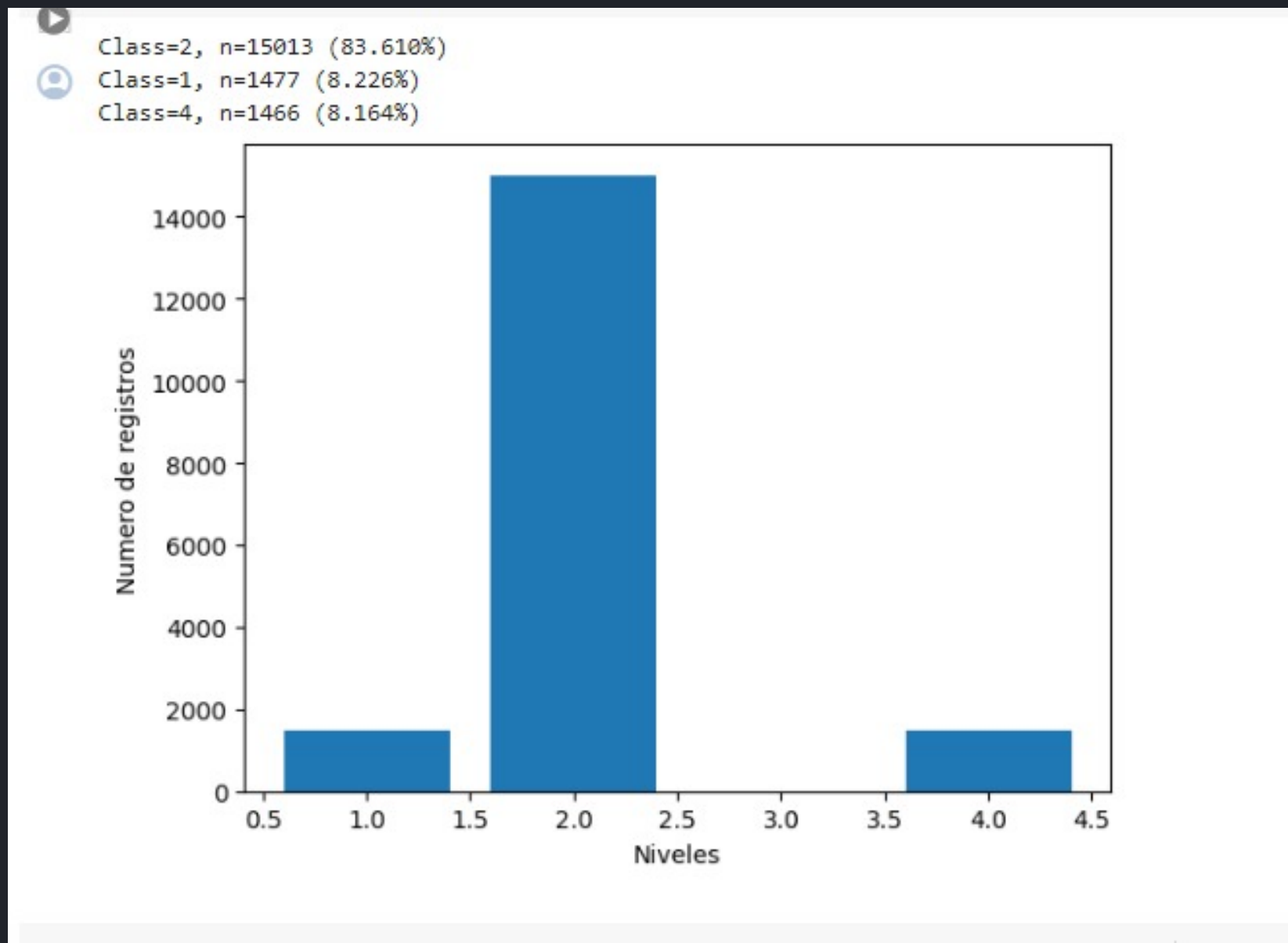
}

Balanceaminto {

2: ANULADO

4: NO ASISTIO

```
undersampling_strategy = {2:  
int(2000)} y oversampling_strategy =  
{1: int(2000), 4: int(2000)}
```



}

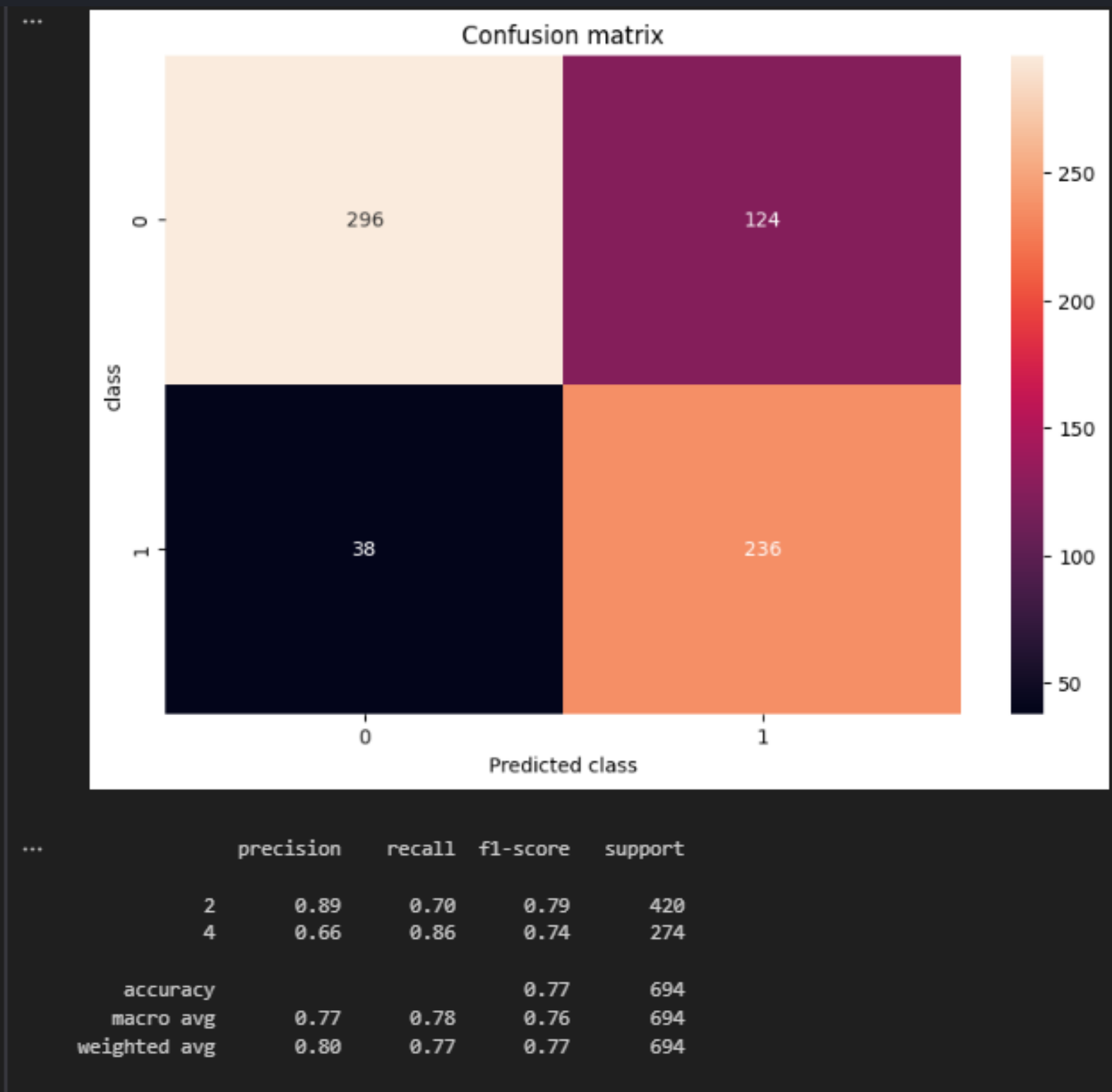
Random Forest {

2: ANULADO

4: NO ASISTIO

Random Forest es un conjunto de árboles de decisión que trabajan de manera conjunta. Este modelo ha demostrado una capacidad moderada para predecir la asistencia a citas médicas.

Exactitud de 77%



}

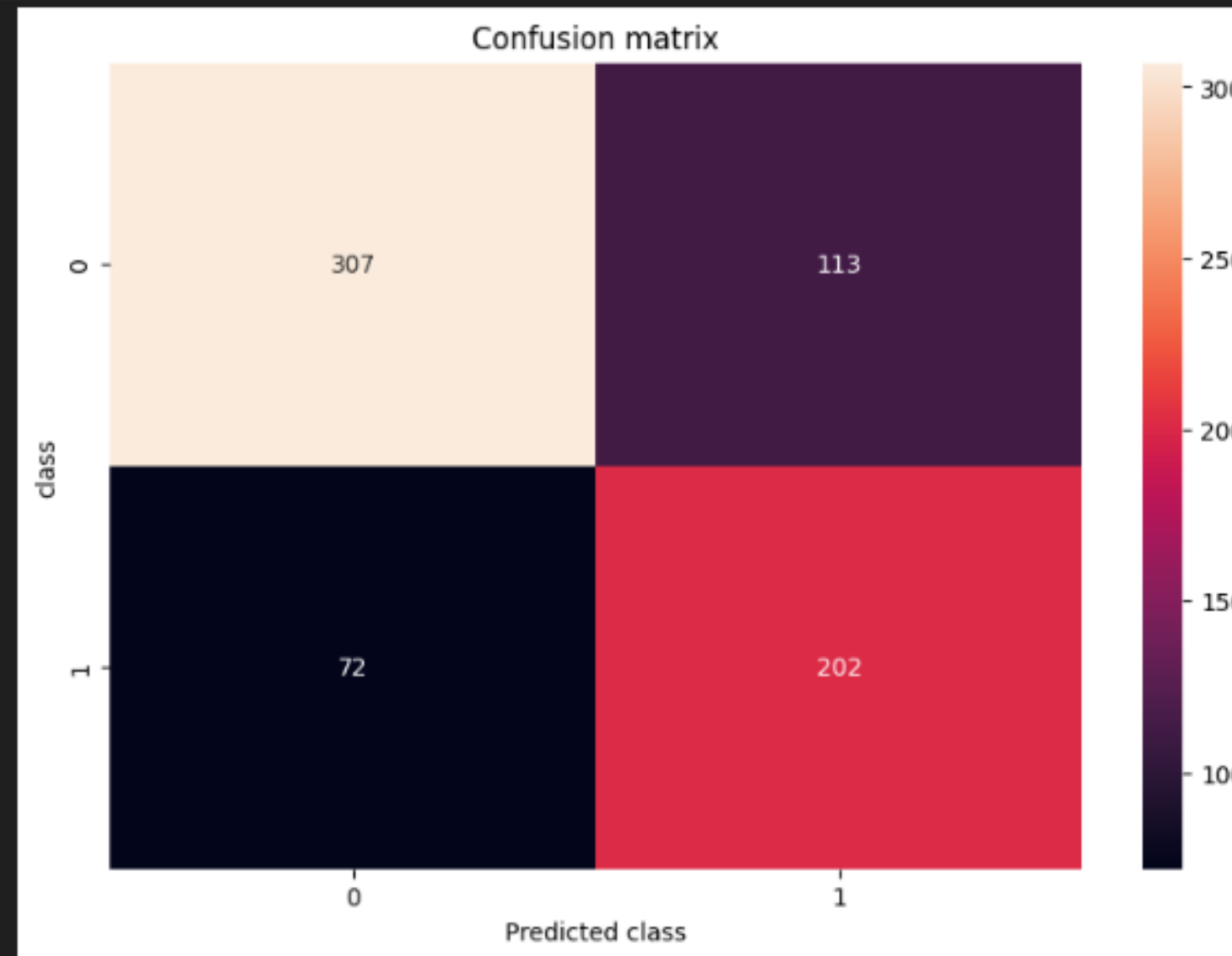
Desicion Tree {

2: ANULADO

4: NO ASISTIO

Un árbol de decisión es un modelo que divide el conjunto de datos en nodos basados en características específicas. Este modelo ha mostrado un rendimiento ligeramente menor en comparación con Random Forest, con una exactitud del 77%.

Exactitud de 73%



	precision	recall	f1-score	support
2	0.81	0.73	0.77	420
4	0.64	0.74	0.69	274
accuracy			0.73	694
macro avg	0.73	0.73	0.73	694
weighted avg	0.74	0.73	0.74	694

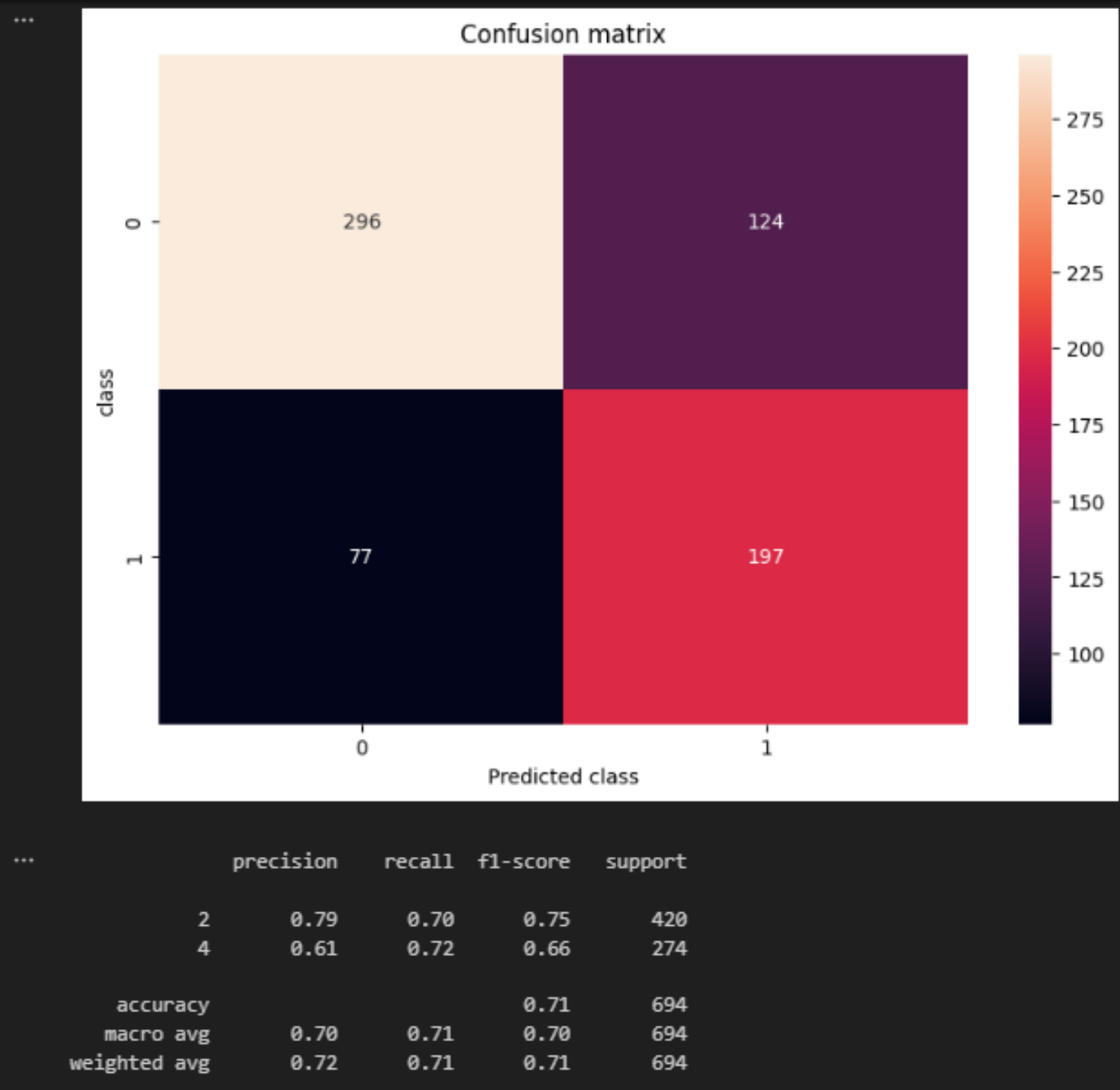
}

Regresión logística multinominal {

2: ANULADO
4: NO ASISTIO

La regresión logística multinominal es un modelo que se utiliza comúnmente para problemas de clasificación multiclase. La exactitud del 71% indica que este modelo puede enfrentar desafíos en la clasificación precisa de las distintas clases relacionadas con las citas médicas

Exactitud de 71%



}

Support Vector Machine

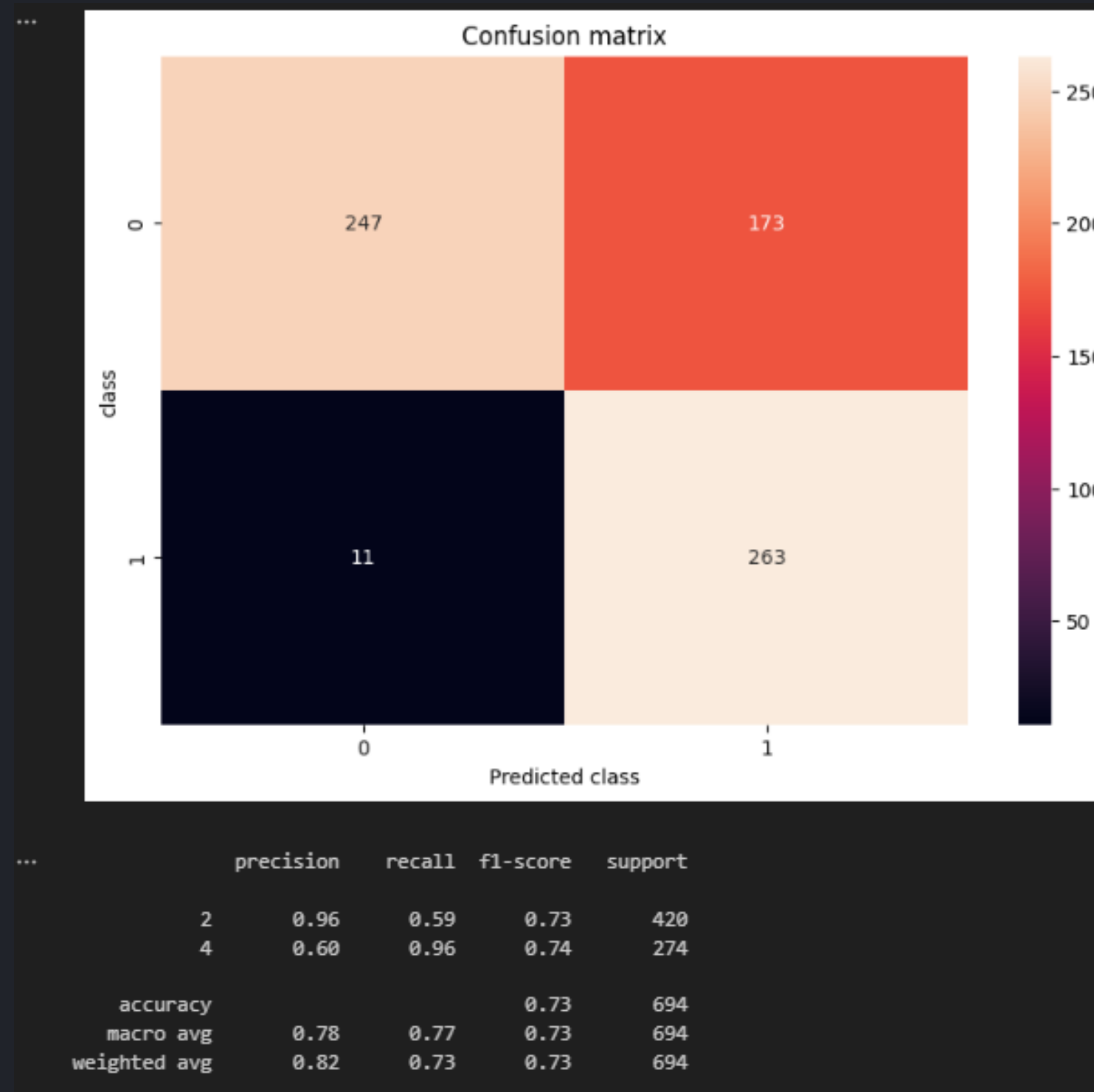
{

2: ANULADO

4: NO ASISTIO

Las máquinas de soporte vectorial (SVM) son eficaces en la clasificación, pero en este caso, la exactitud del 73% sugiere un rendimiento similar a una predicción aleatoria.

Exactitud de 73%



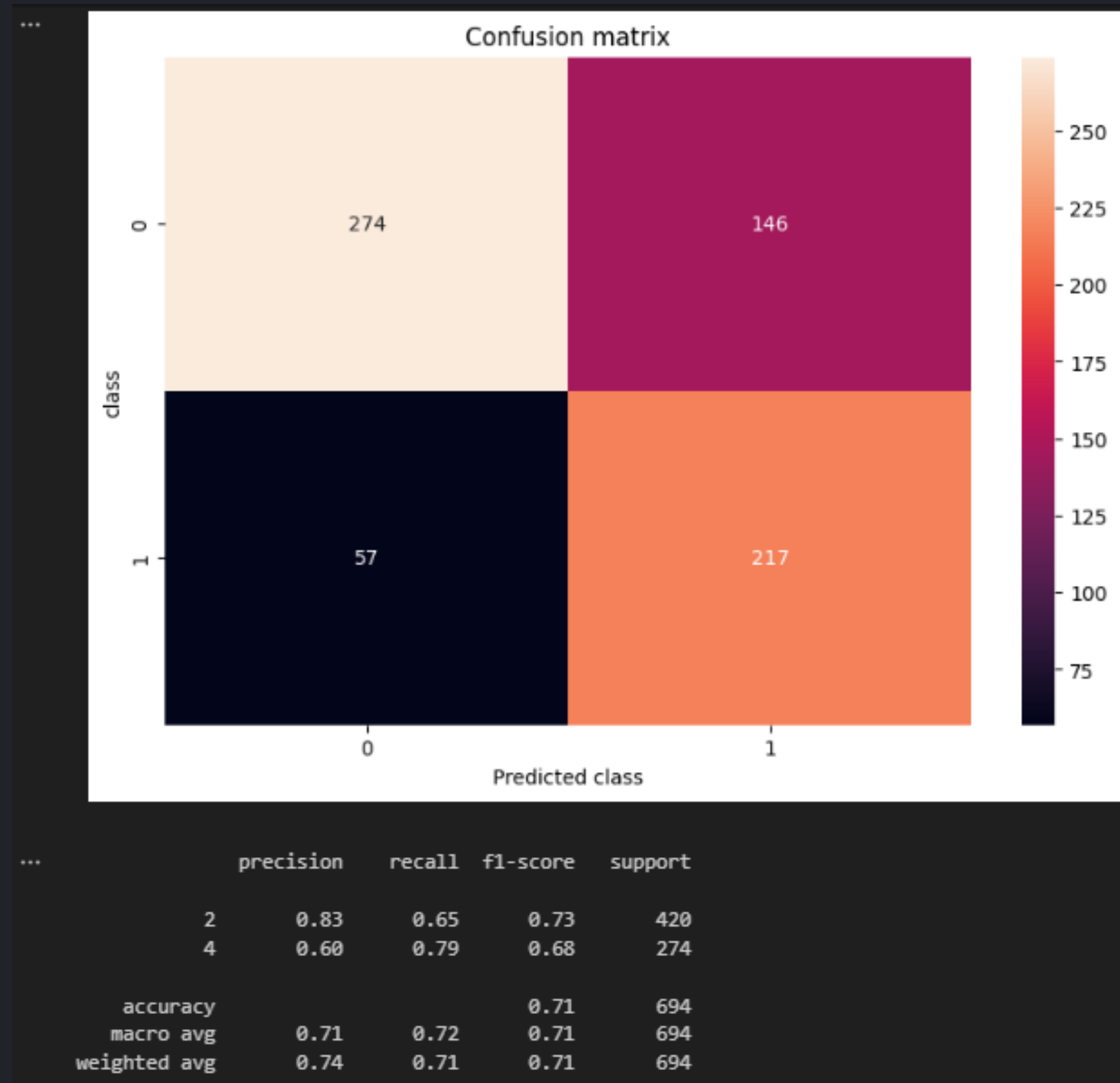
}

Naibe Bayes{

1: ANULADO
2: CONCLUIDO
4: NO ASISTIO

Naive Bayes es un modelo probabilístico que asume independencia condicional entre las características. Una exactitud del 50% sugiere un rendimiento similar al azar.

Exactitud de 71%



}

Conclusiones {

Primera Conclusión.

El modelo de Decision Tree se destacó como el más efectivo entre los modelos evaluados para predecir la asistencia a citas dentales basándose en los datos recopilados. Aunque alcanzó un porcentaje del 64% de precisión, es fundamental tener en cuenta que aún hay margen para mejorar la exactitud predictiva.

Segunda Conclusión.

Este resultado indica que el árbol de decisiones pudo capturar patrones significativos en los datos de los pacientes de la clínica dental. Sin embargo, sería beneficioso continuar refinando el modelo, explorando otras características, ajustando parámetros y considerando técnicas adicionales de ingeniería de características para potenciar aún más su desempeño.

}

Antecedentes {

Título: "Predicting Attendance at Medical Appointments Using Machine Learning"

- Revista: Journal of Medical Internet Research, 2019
- Descripción: Este estudio desarrolló un modelo de aprendizaje automático basado en árboles de decisión para predecir la asistencia a citas médicas. El modelo se entrenó con un conjunto de datos de 12.000 citas médicas y alcanzó una precisión del 72,2%.

}

```
<!--Data Mining-->
```

Gracias {

```
<Por="Mirian Cayo Molloni"/>
```

}