ArgxEdu

Hackatón 2020

Desafío 1: predecir desempeño en las pruebas Aprender

Equipo: El 22

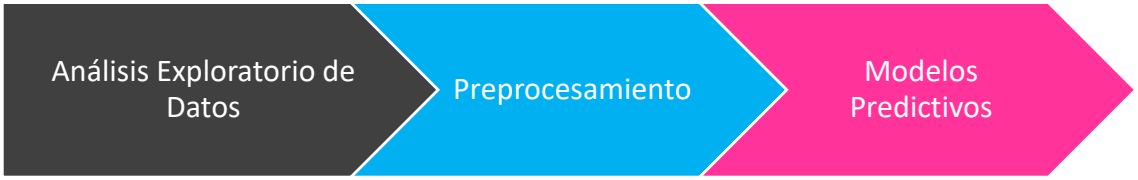
Integrante: Diego Lopez Yse (único integrante)

Junio 2020

DESAFÍO SELECCIONADO Y ENFOQUE ELEGIDO

He seleccionado el desafío 1: "predecir desempeño en las pruebas Aprender". La idea ha sido responder algunas preguntas como: ¿qué factores inciden sobre los resultados de las pruebas Aprender? ¿Son factores endógenos o exógenos al sistema educativo? En otras palabras, ¿son factores sobre los cuales el sistema educativo puede incidir?

Bajo la hipótesis que es posible predecir el desempeño en las pruebas Aprender basándome en las variable detalladas en el dataset "app_alumno.csv", he definido los siguientes pasos para trabajar los datos:



Para ello he elegido el lenguaje de programación Python, ya que el tamaño y extensión del dataset justifica la selección de tecnologías más potentes que las usualmente utilizables (ejemplo Microsoft Excel). He incluido el código utilizado a lo largo del desarrollo de este trabajo para facilitar el entendimiento.

DATOS UTILIZADOS. DESCRIPCIÓN DE VARIABLES

Para plantear responder la pregunta del desafío, he utilizado el dataset "app_alumno.csv", que se compone de las siguientes 25 variables:

- id
- sexo
- indice_socioeconomico
- nivel_desemp_matematica
- nivel_desemp_lengua
- nivel_desemp_ciencias_sociales
- nivel_desemp_ciencias_naturales
- ponderador_lengua
- ponderador_matematica
- ponderador_ciencias_naturales
- ponderador_ciencias_sociales
- tiene_notebook
- tiene_pc
- tiene_tablet
- tiene_celular
- tiene_smartphone
- tiene_consola
- tiene_smarttv
- tiene_cable
- tiene_internet

- repeticion_primaria
- repeticion_secundaria
- escuela_id
- nivel_id
- year_id

El dataset se compone de 1.835.710 registros, con una cantidad significativa de valores faltantes. La mayoría de variables son categóricas, lo que implica desafíos de procesamiento diferentes a dataset netamente numéricos. Los pasos que seguí en esta instancia fueron:

- **Análisis Univariado**
- **Análisis Bivariado**
- **Missing Values**

Por la naturaleza de los datos no me detuve en el análisis de outliers. A continuación, detallo cada uno de los pasos que seguí:

ANÁLISIS UNIVARIADO

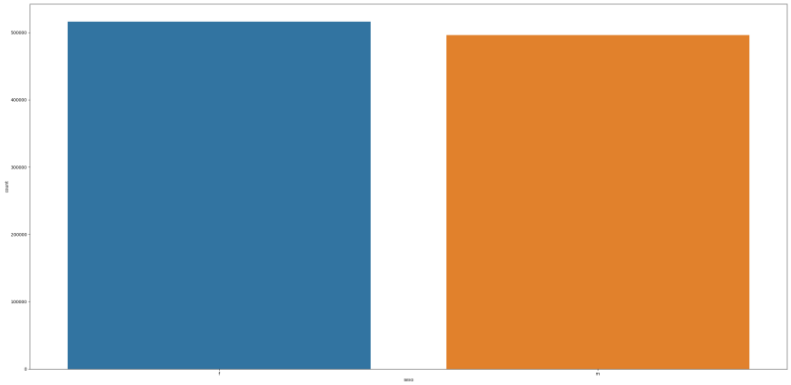
Como paso inicial, he realizado un análisis individual sobre las variables de interés. En primer lugar, he importado las librerías necesarias para el análisis:

```
import pandas as pd

import numpy as np

import seaborn as sns
```

Luego he comenzado el trabajo de exploración sobre el total de registros y las siguientes variables:

- **Sexo**

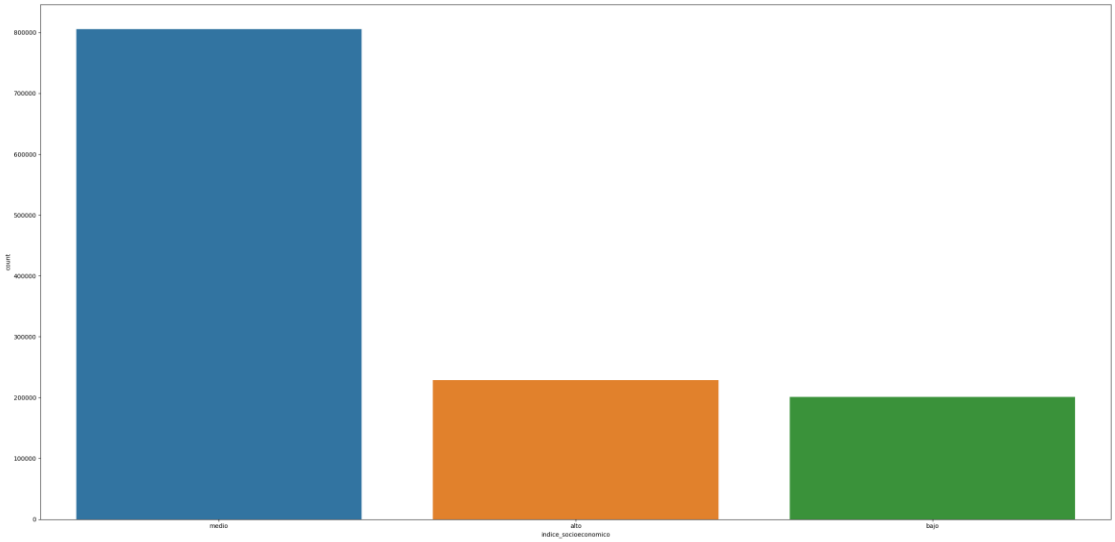
```
sns.countplot(x='sexo',data=df)
```



Las clases sexo femenino y masculino se encuentran equilibradas (femenino levemente superior).

- **Índice socioeconomico**

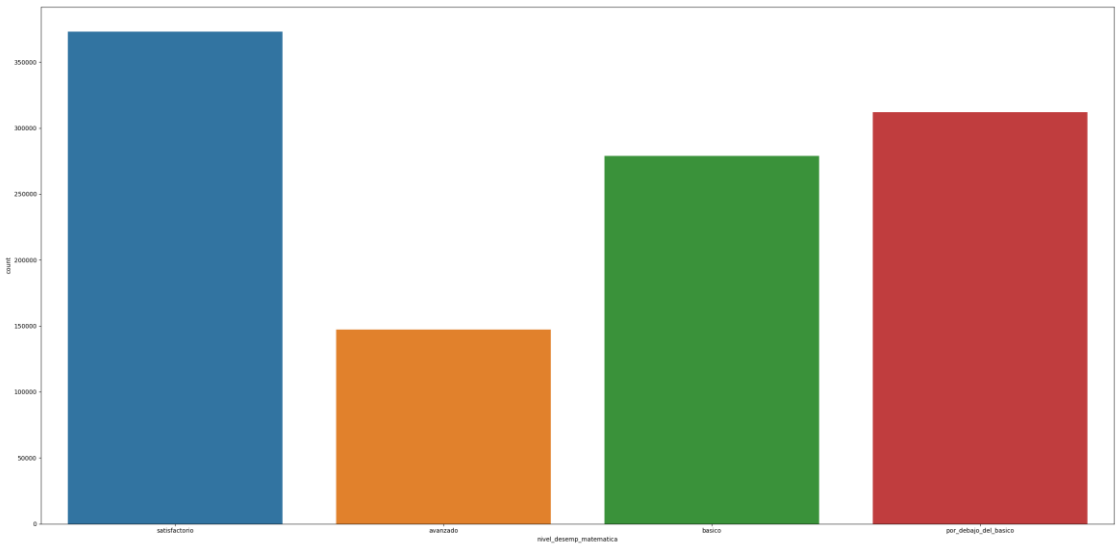
```
sns.countplot(x='indice_socioeconomico',data=df)
```



La amplia **mayoría** de observaciones se corresponden con el **sector medio**, siendo los sectores altos y bajos muy similares.

- **Nivel de desempeño en matemática**

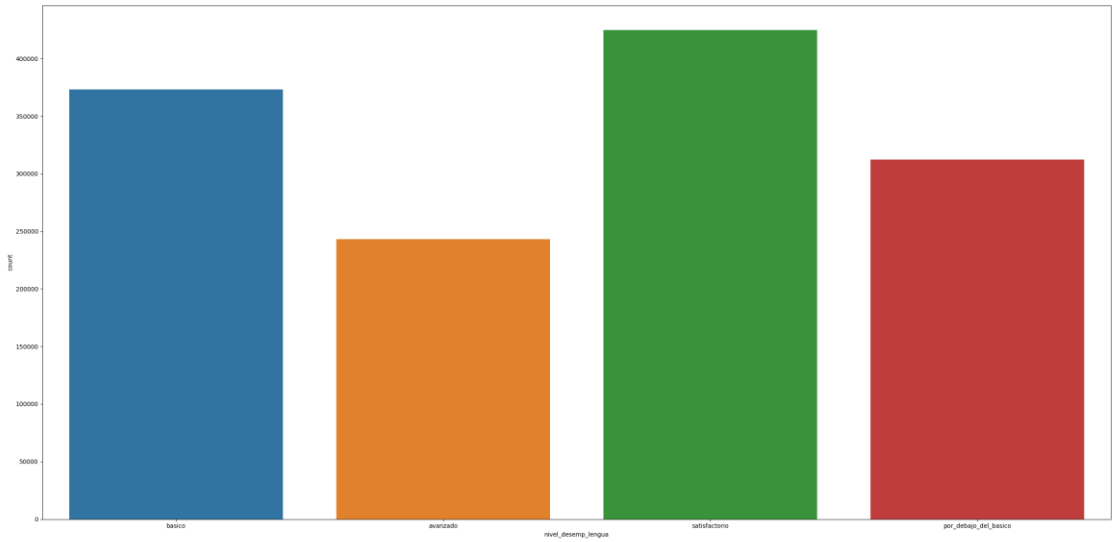
```
sns.countplot(x='nivel_desemp_matematica',data=df)
```



La **mayoría** de casos son de **nivel satisfactorio**, y existen niveles considerables de resultados básico y por debajo del básico.

- **Nivel de desempeño en lengua**

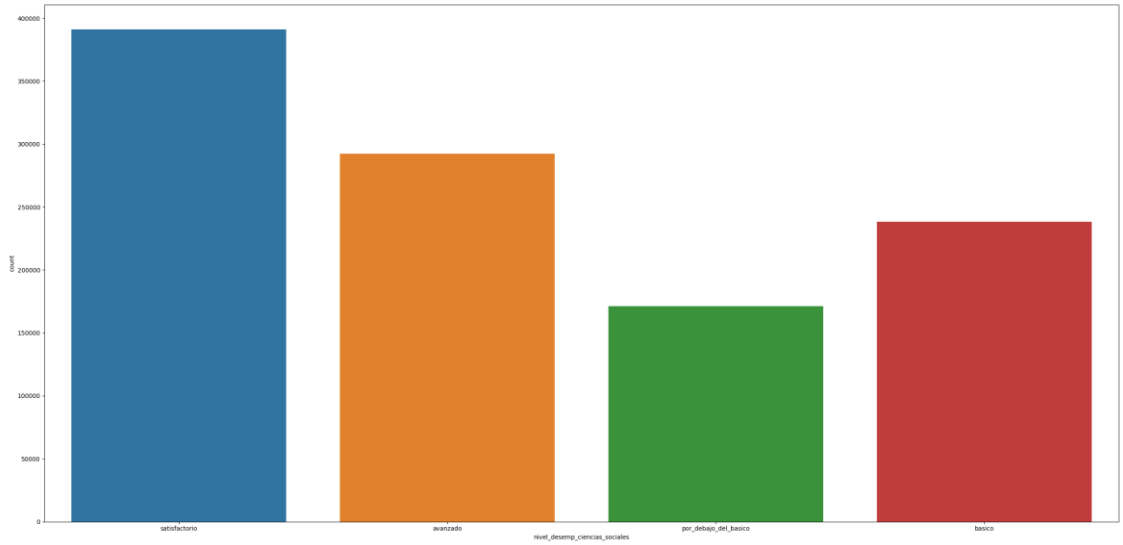
```
sns.countplot(x='nivel_desemp_lengua',data=df)
```



La **mayoría** de casos son de **nivel satisfactorio**, y existen niveles considerables de resultados básico y por debajo del básico.

- **Nivel de desempeño en ciencias sociales**

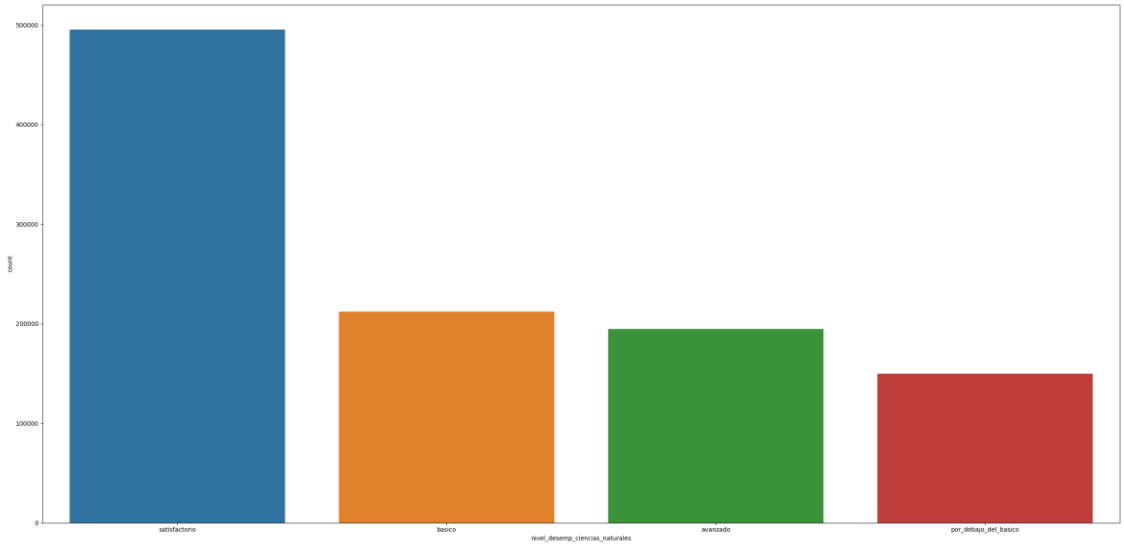
```
sns.countplot(x='nivel_desemp_ciencias_sociales',data=df)
```



La **mayoría** de casos son de **nivel satisfactorio**, y existen niveles considerables de resultados avanzados y básico. En este caso, el conjunto de resultados es más positivo que para matemáticas y lengua.

- **Nivel de desempeño en ciencias naturales**

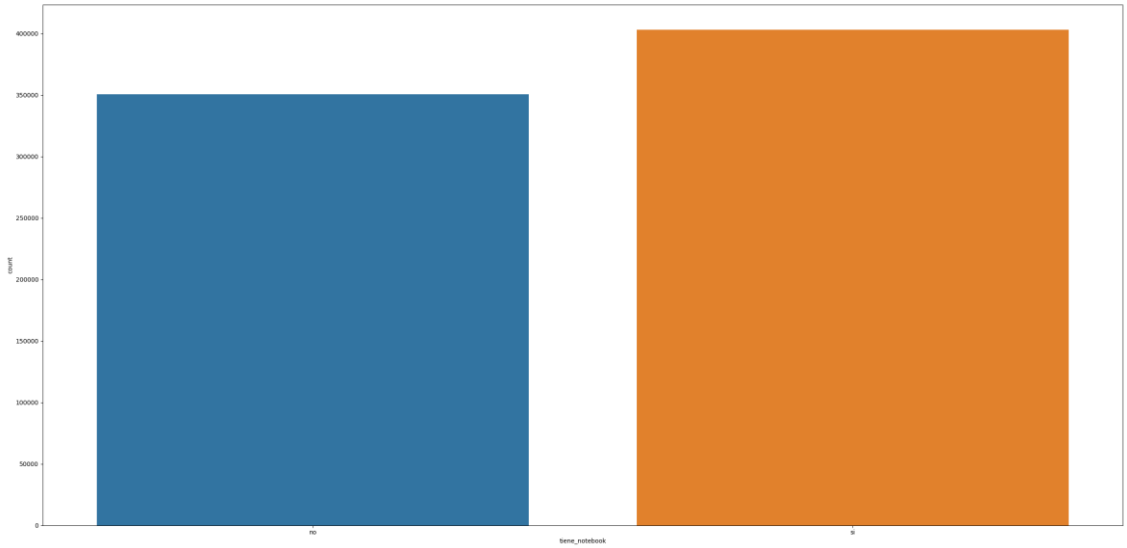
```
sns.countplot(x='nivel_desemp_ciencias_naturales',data=df)
```



La **mayoría** de casos son de **nivel satisfactorio**, siendo los niveles restantes muy similares entre sí.

- **Tiene notebook**

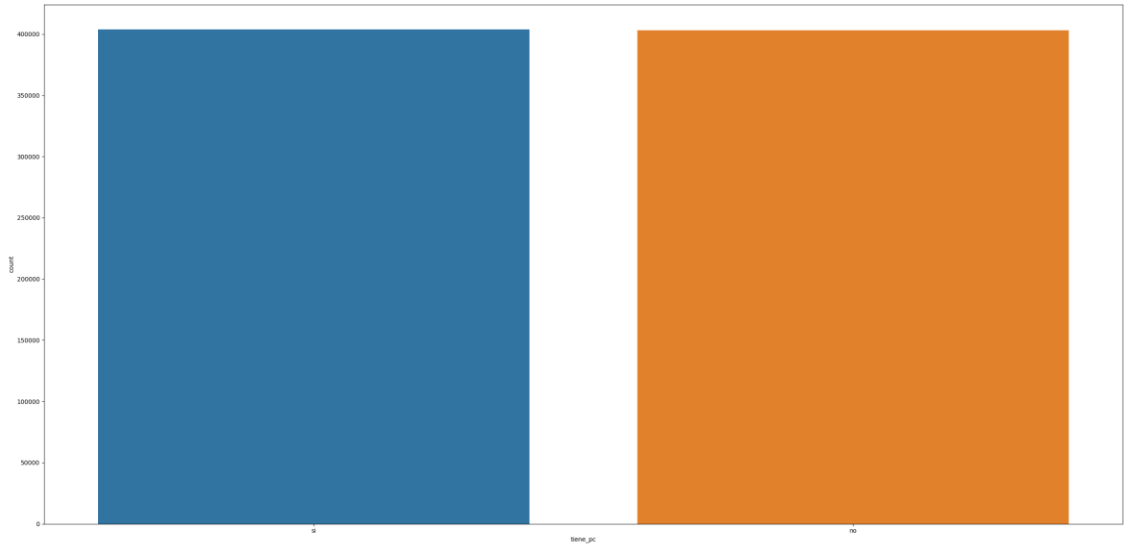
```
sns.countplot(x='tiene_notebook',data=df)
```



La mayoría tiene notebook.

- **Tiene pc**

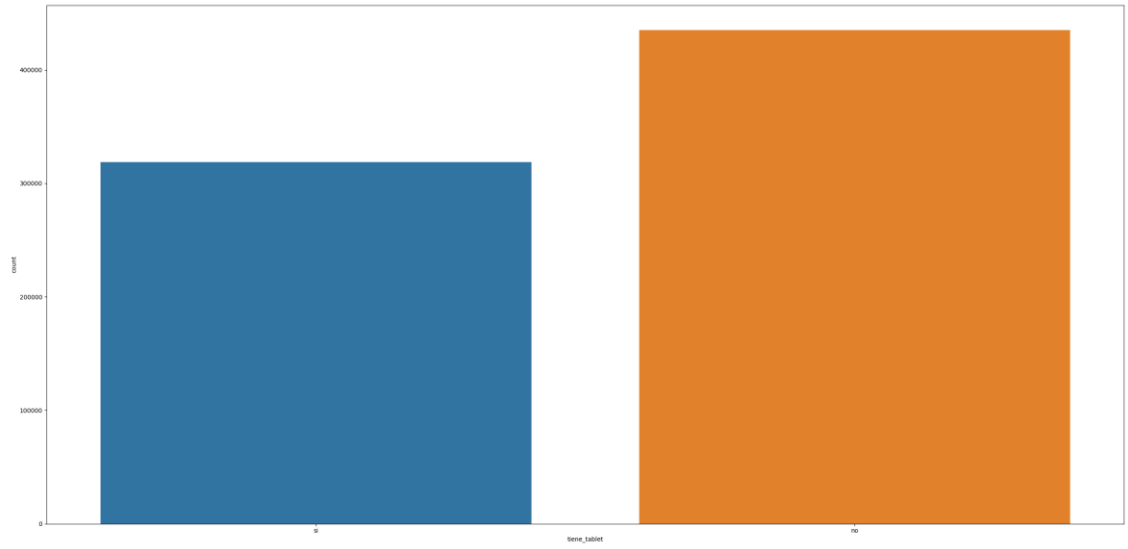
```
sns.countplot(x='tiene_pc',data=df)
```



Casi la misma cantidad de observaciones tienen y no tienen pc

- **Tiene tablet**

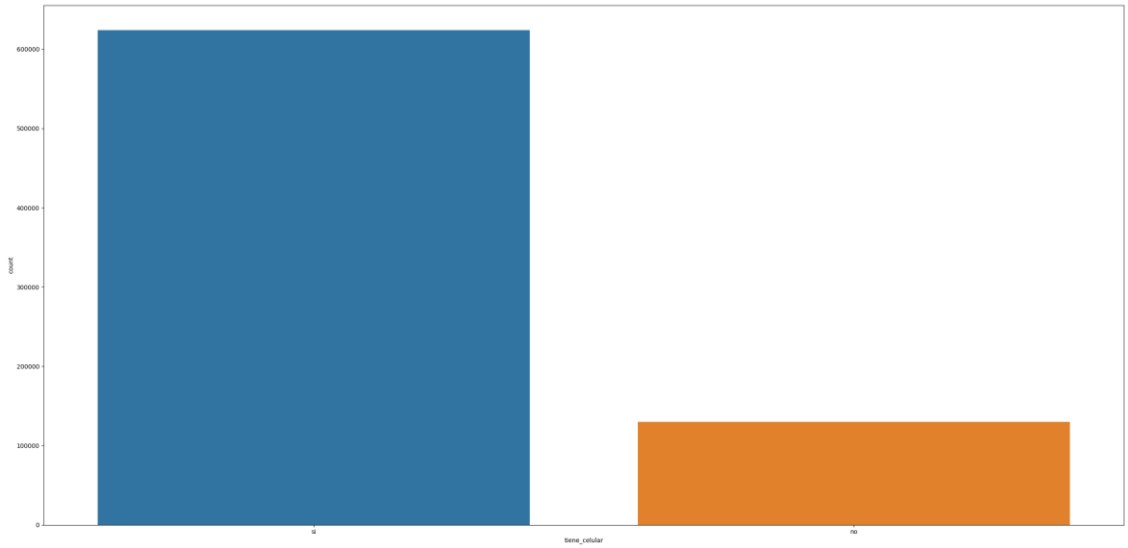
```
sns.countplot(x='tiene_tablet',data=df)
```



La mayoría no tiene tablet.

- **Tiene celular**

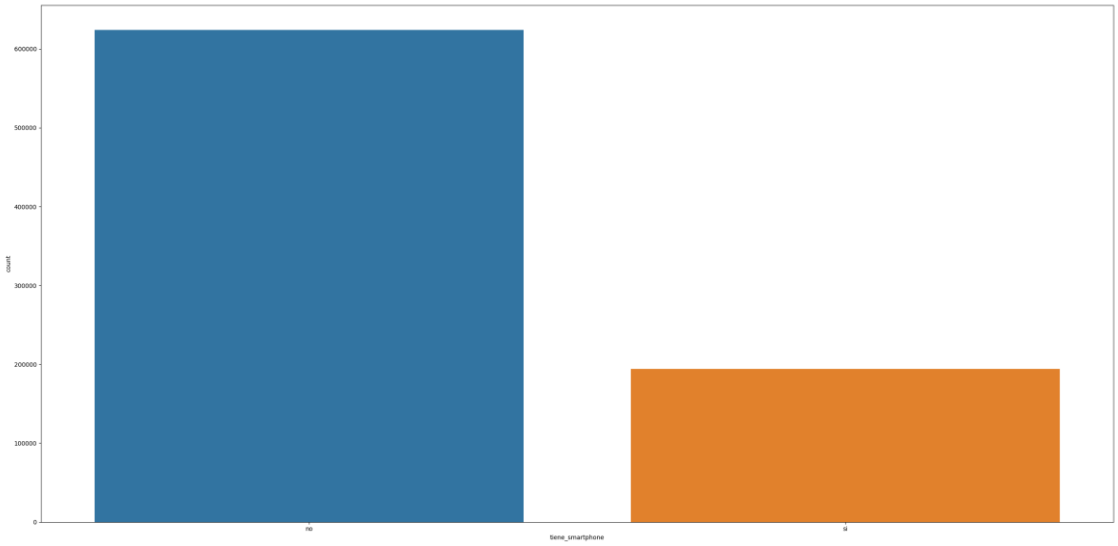
```
sns.countplot(x='tiene_celular',data=df)
```



La **gran mayoría tiene celular**. Se observa un marcado desequilibrio entre las clases.

- **Tiene smartphone**

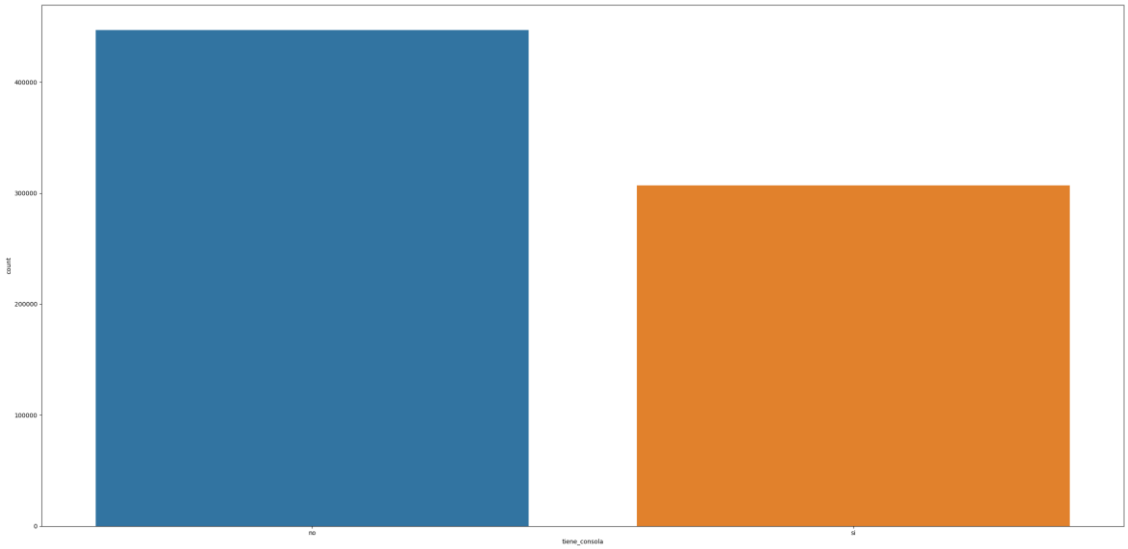
```
sns.countplot(x='tiene_smartphone',data=df)
```



La **enorme mayoría no cuenta con smartphone**.

- **Tiene consola**

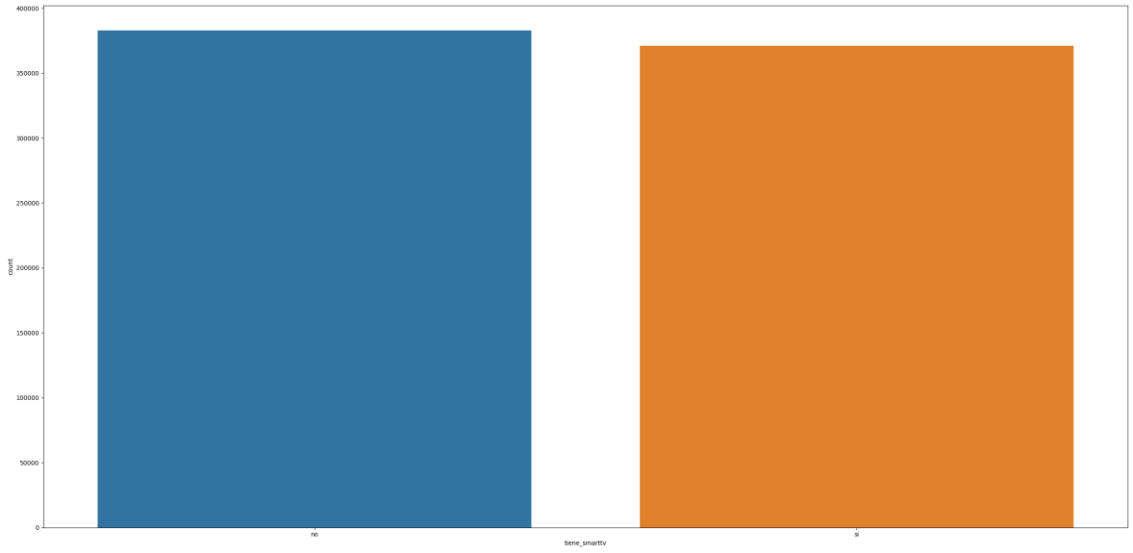
```
sns.countplot(x='tiene_consola',data=df)
```



La mayoría no tiene consola.

- **Tiene Smart tv**

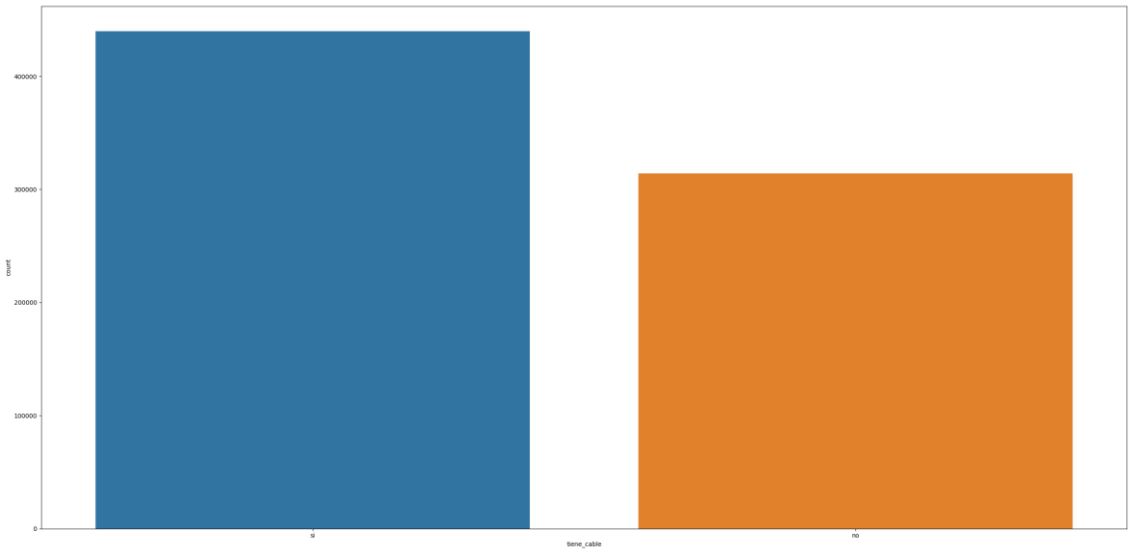
```
sns.countplot(x='tiene_smarttv',data=df)
```



Las cantidades de observaciones entre quienes tienen Smart tv y quienes no tienen son casi equilibradas.

- **Tiene cable**

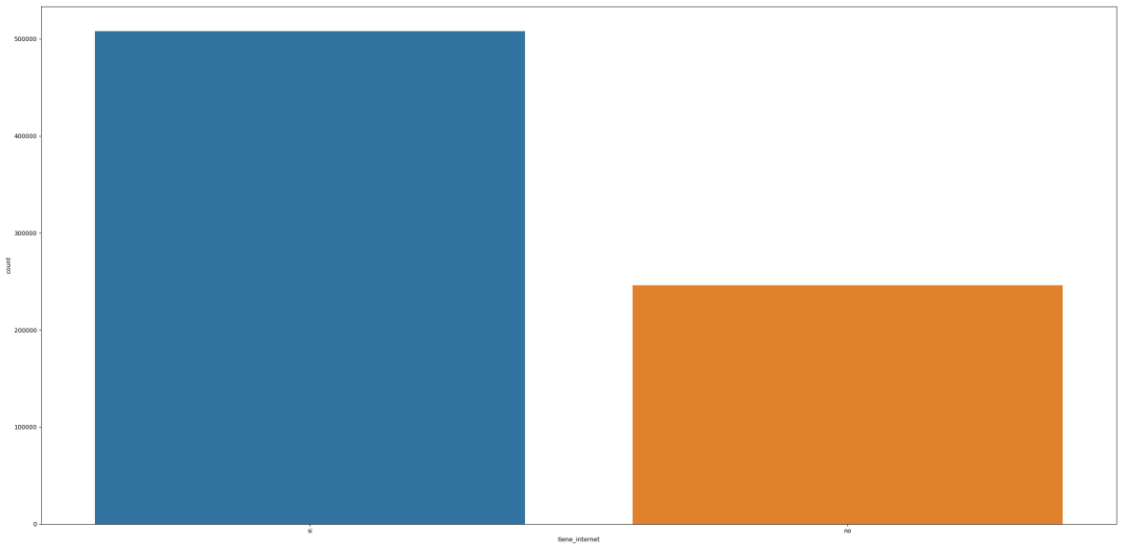
```
sns.countplot(x='tiene_cable',data=df)
```



La mayoría de observaciones tiene cable.

- **Tiene internet**

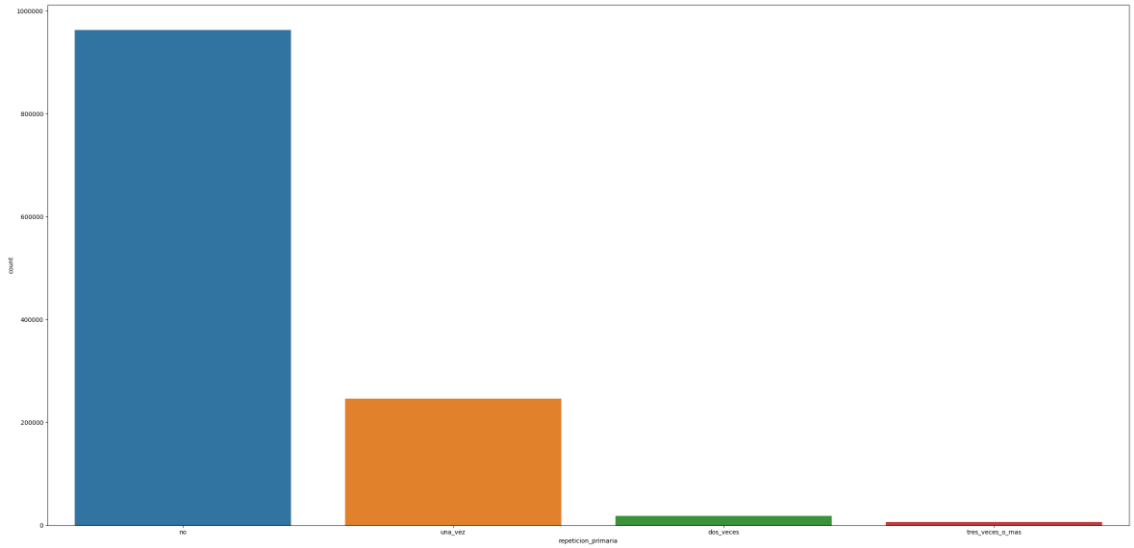
```
sns.countplot(x='tiene_internet',data=df)
```



La **mayoría** de observaciones **tiene internet**.

- **Repetición primaria**

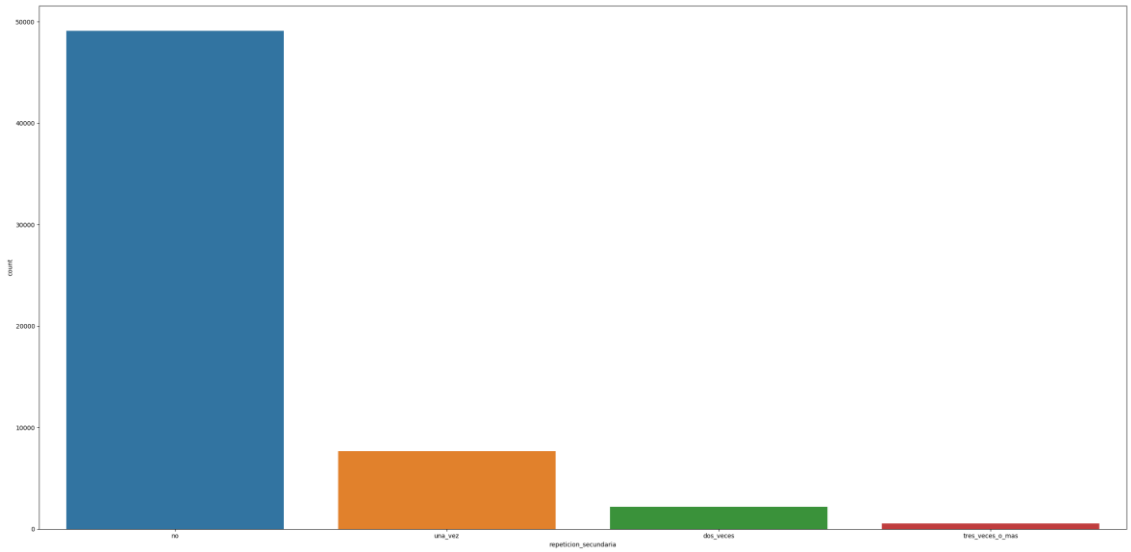
```
sns.countplot(x='repeticion_primaria',data=df)
```



La amplia mayoría de casos no ha repetido el primario.

- **Repetición secundaria**

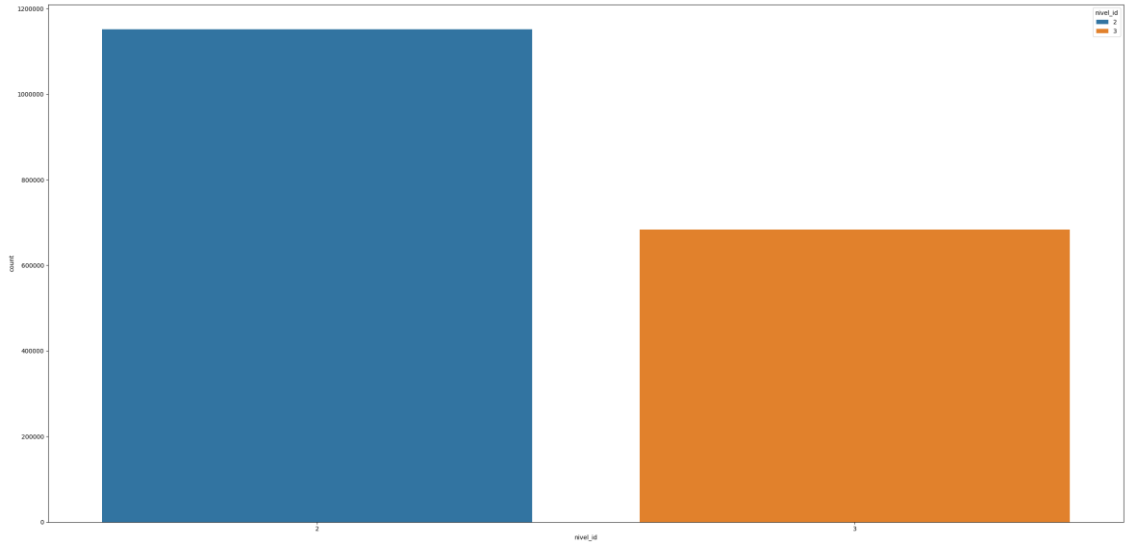
```
sns.countplot(x='repeticion_secundaria',data=df)
```



La amplia mayoría de casos no ha repetido el secundario, y **aumenta la importancia relativa de quienes repitieron 2 veces con respecto al nivel pri**mario.

- **Nivel id**

```
sns.countplot(x='nivel_id',data=df)
```



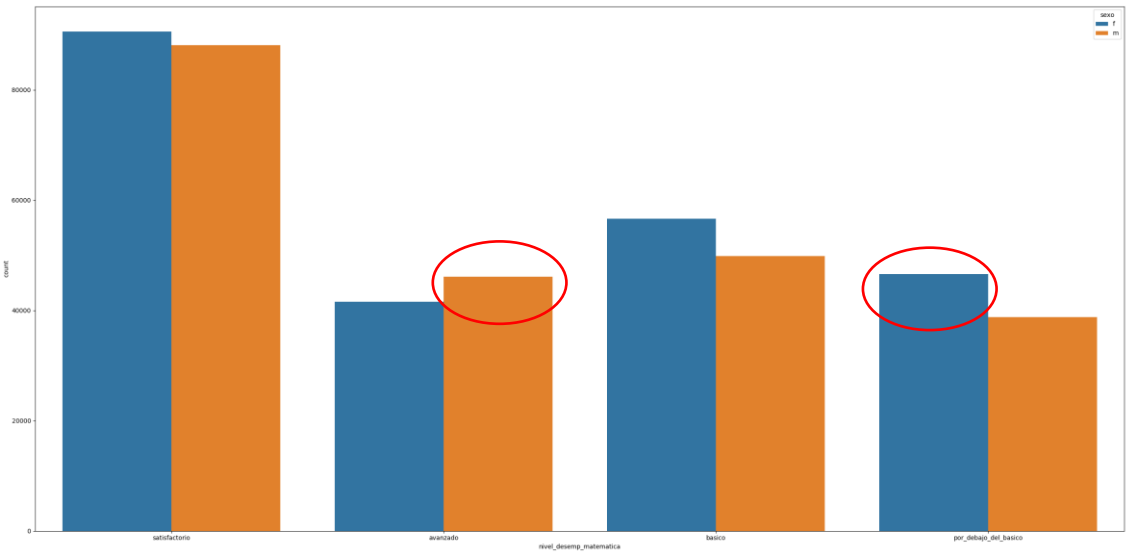
Se observa una mayor cantidad de observaciones de primario (nivel_id = 2) que secundario.

ANÁLISIS BIVARIADO

Luego de analizar las variables de manera individual, realicé algunas combinaciones para intentar descubrir tendencias no observables a simple vista:

1) **Por sexo**
    a. **Nivel de desempeño en matemática**

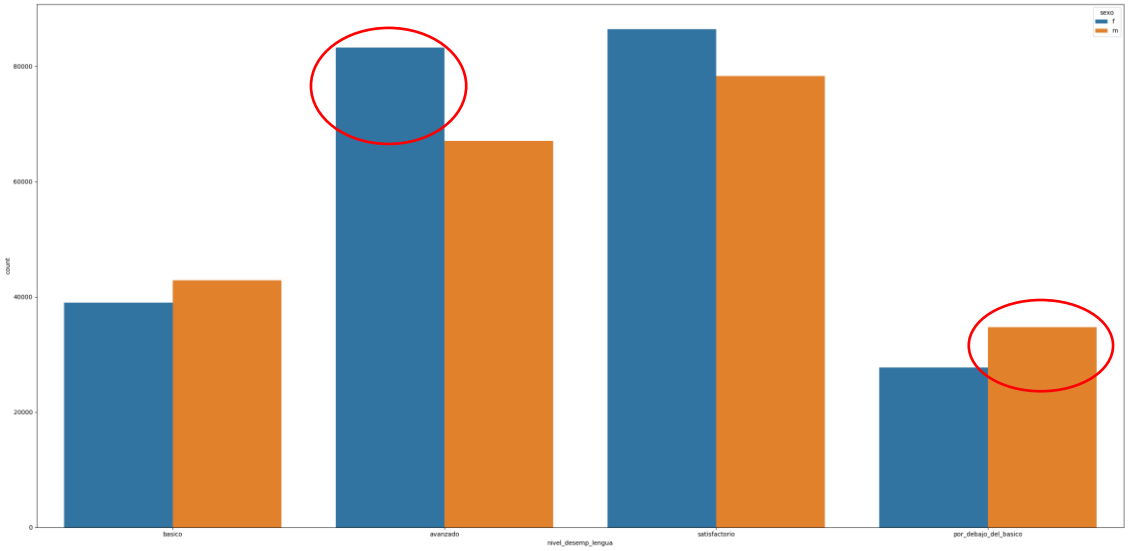
```
sns.countplot(x='nivel_desemp_matematica', hue='sexo', data=df)
```



No se observan diferencias significativas en el desempeño en matemáticas clasificado por sexo, pero el sexo femenino tuvo una performance superior en el nivel avanzado y participación inferior en el nivel por debajo del básico.

b. Nivel de desempeño en lengua

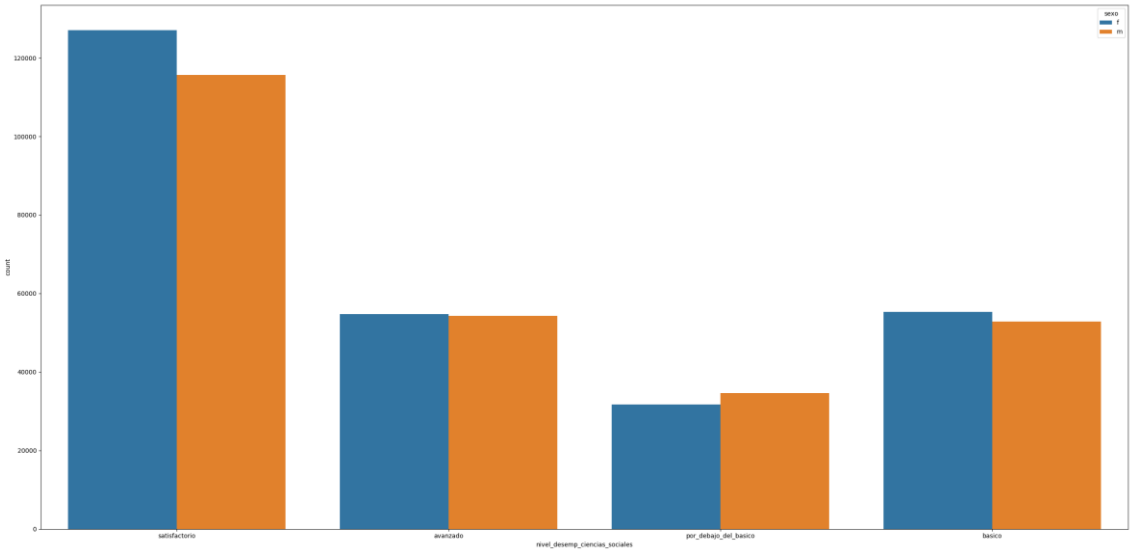
```
sns.countplot(x='nivel_desemp_lengua', hue='sexo', data=df)
```



No se observan diferencias significativas en el desempeño en matemáticas clasificado por sexo, pero el sexo masculino tuvo una participación superior en el nivel avanzado y participación inferior en el nivel por debajo del básico.

c. Nivel de desempeño en ciencias sociales

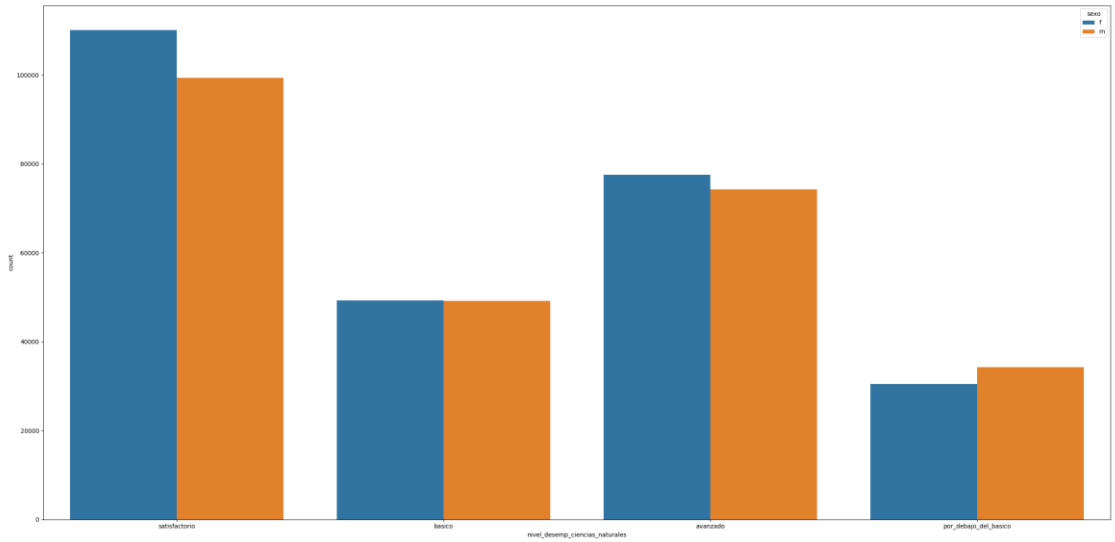
```
sns.countplot(x='nivel_desemp_ciencias_sociales', hue='sexo', data=df)
```



No se observan diferencias significativas en el desempeño en ciencias sociales clasificado por sexo.

d. Nivel de desempeño en ciencias naturales

```
sns.countplot(x='nivel_desemp_ciencias_naturales', hue='sexo', data=df)
```

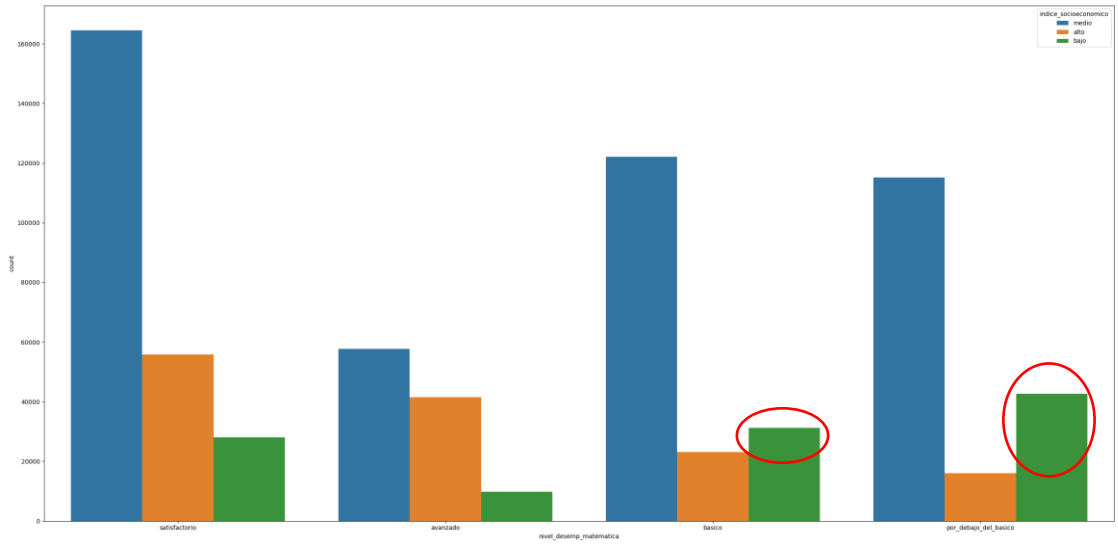


No se observan diferencias significativas en el desempeño en ciencias naturales clasificado por sexo.

2) Por nivel socioeconómico

a. Nivel de desempeño en matemática

```
sns.countplot(x='nivel_desemp_matematica', hue='indice_socioeconomico', data=df)
```



En línea con el análisis univariado, el sector socioeconómico medio es la clase mayoritaria en todos los niveles de desempeño en matemática. Es interesante observar el **aumento en participación del sector socioeconómico bajo a medida que se reduce el nivel de desempeño**, superando la participación del nivel socioeconómico alto en los niveles básico y por debajo del básico. Existe una hipótesis de correlación entre estas variables. Para validarla, realizo un test

de chi-cuadrado entre las variables índice socioeconómico y nivel de desempeño en matemáticas:

```
from scipy.stats import chi2_contingency

table = pd.crosstab(df.indice_socioeconomico, df.nivel_desemp_matematica)

chi2, p, dof, expected = chi2_contingency(table.values)

print ('Chi-square:', chi2)

print ('p-value:', p)
```

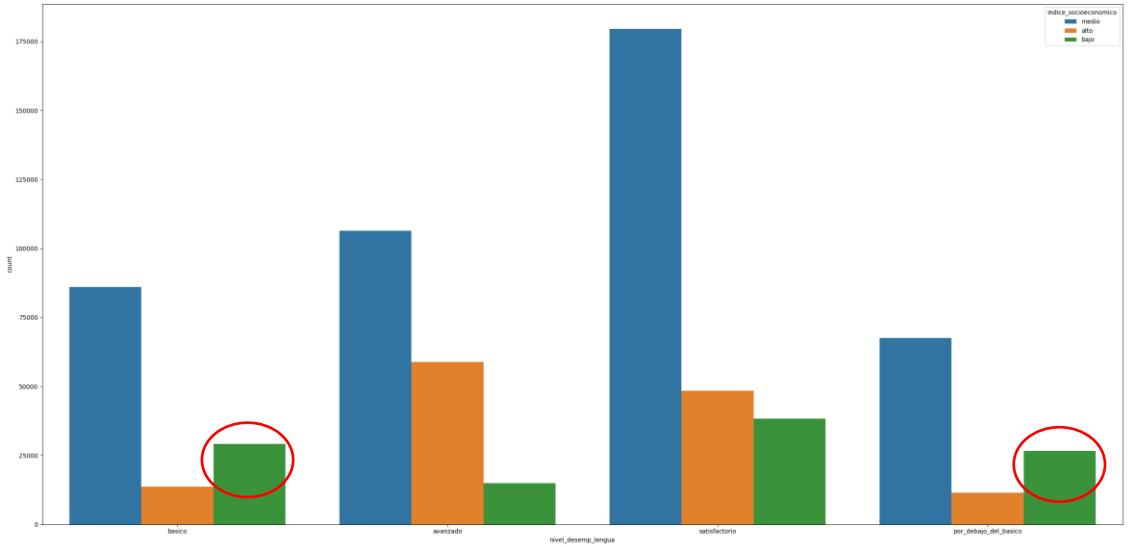
Chi-square: 52002.15534227134

p-value: 0.0

Con un p-value mayor a 0.05 (utilizando un valor de significancia del 95%), sería posible concluir que las variables bajo análisis son independientes. Como el resultado es menor a 0.05, se rechaza la hipótesis nula y podemos concluir que existe relación entre ambas variables.

b. Nivel de desempeño en lengua

```
sns.countplot(x='nivel_desemp_lengua', hue='indice_socioeconomico', data=df)
```



Como analizamos con matemáticas, se observa un **aumento en participación del sector socioeconómico bajo a medida que se reduce el nivel de desempeño en lengua**, superando la participación del nivel socioeconómico alto en los niveles básico y por debajo del básico.

Realizo nuevamente un test chi-cuadrado para analizar la relación entre las variables:

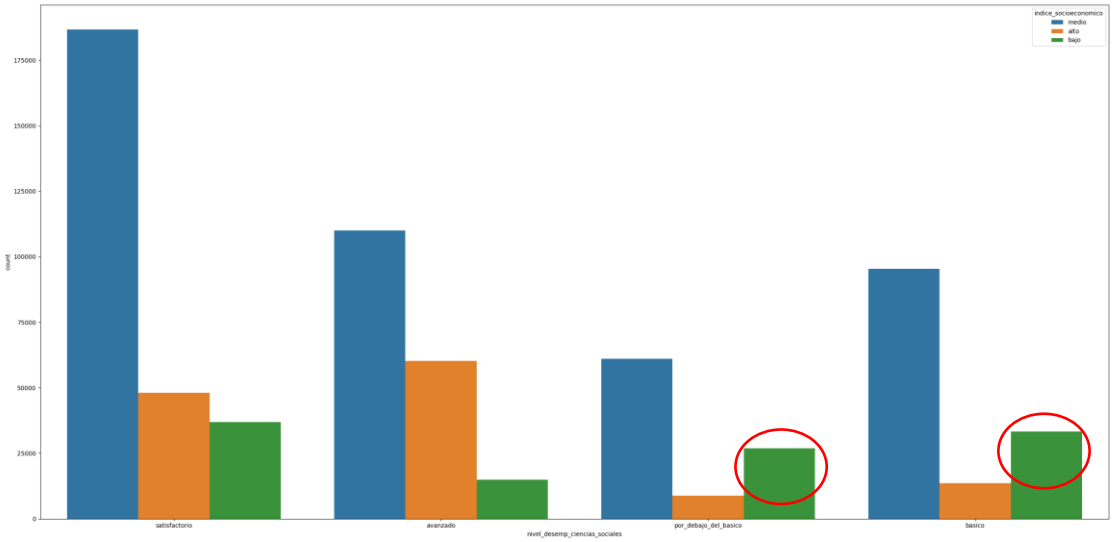
```
table = pd.crosstab(df.indice_socioeconomico, df.nivel_desemp_lengua)

chi2, p, dof, expected = chi2_contingency(table.values)

print ('Chi-square:', chi2)

print ('p-value:', p)
```

Chi-square: 43109.11075218844

p-value: 0.0

c. Nivel de desempeño en ciencias sociales

```
sns.countplot(x='nivel_desemp_ciencias_sociales', hue='indice_socioeconomico', data=df)
```



Nuevamente, se observa un **aumento en participación del sector socioeconómico bajo a medida que se reduce el nivel de desempeño en ciencias sociales**, superando la participación del nivel socioeconómico alto en los niveles básico y por debajo del básico.

Realizo nuevamente un test chi-cuadrado para analizar la relación entre las variables:

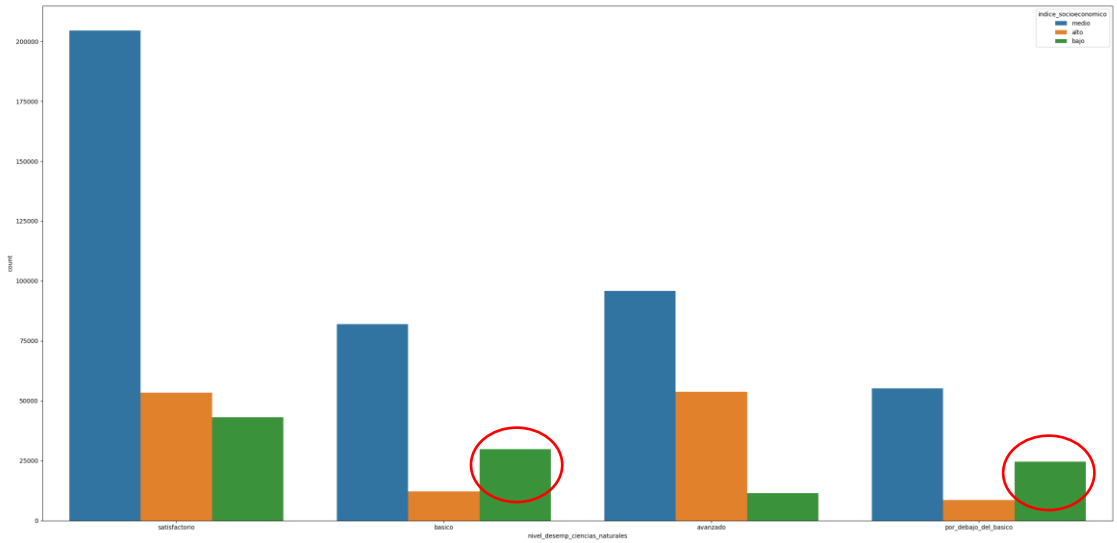
```
table = pd.crosstab(df.indice_socioeconomico, df.nivel_desemp_ciencias_sociales)

chi2, p, dof, expected = chi2_contingency(table.values)

print ('Chi-square:', chi2)

print ('p-value:', p)
```

Chi-square: 53436.63736354612

p-value: 0.0

d. Nivel de desempeño en ciencias naturales

```
sns.countplot(x='nivel_desemp_ciencias_naturales', hue='indice_socioeconomico', data=df)
```



En este caso también se observa un **aumento en participación del sector socioeconómico bajo a medida que se reduce el nivel de desempeño en ciencias naturales**, superando la participación del nivel socioeconómico alto en los niveles básico y por debajo del básico.

Realizo nuevamente un test chi-cuadrado para analizar la relación entre las variables:

```
table = pd.crosstab(df.indice_socioeconomico, df.nivel_desemp_ciencias_naturales)

chi2, p, dof, expected = chi2_contingency(table.values)

print ('Chi-square:', chi2)

print ('p-value:', p)
```

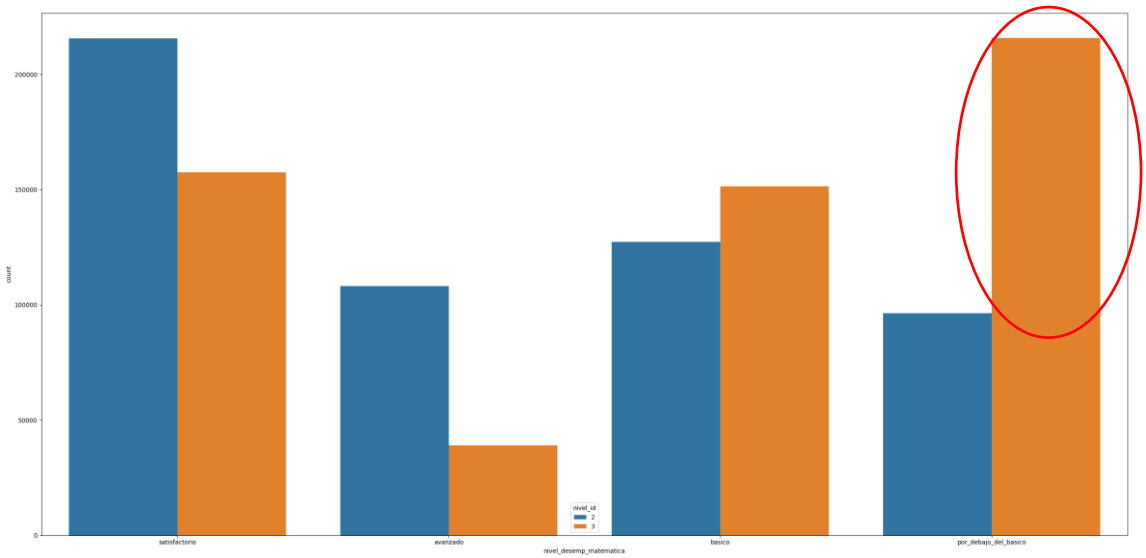
Chi-square 49546.07437475268

p-value: 0.0

3) Por nivel educativo (primario=2 o secundario=3)

a. Nivel de desempeño en matemática

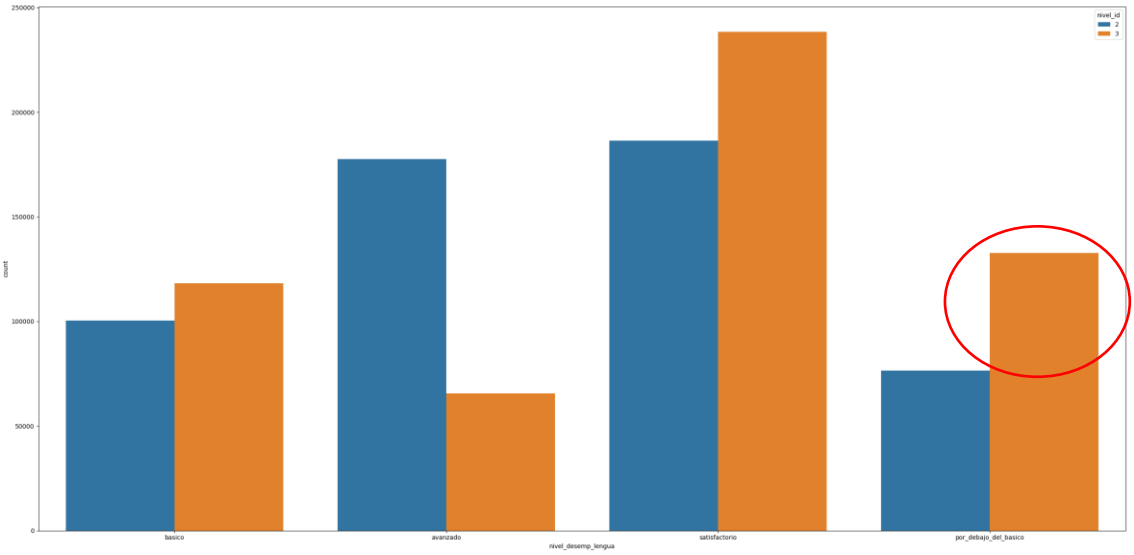
```
sns.countplot(x='nivel_desemp_matematica', hue='nivel_id', data=df)
```



Es interesante observar que a pesar que la población total es superior en primario (detallado en el análisis univariado), los registros de nivel de desempeño **por debajo del básico en matemáticas son muy superiores en el secundario**, indicando un fuerte desfasaje en el aprendizaje o el esquema de evaluación entre primario y secundario.

b. Nivel de desempeño en lengua

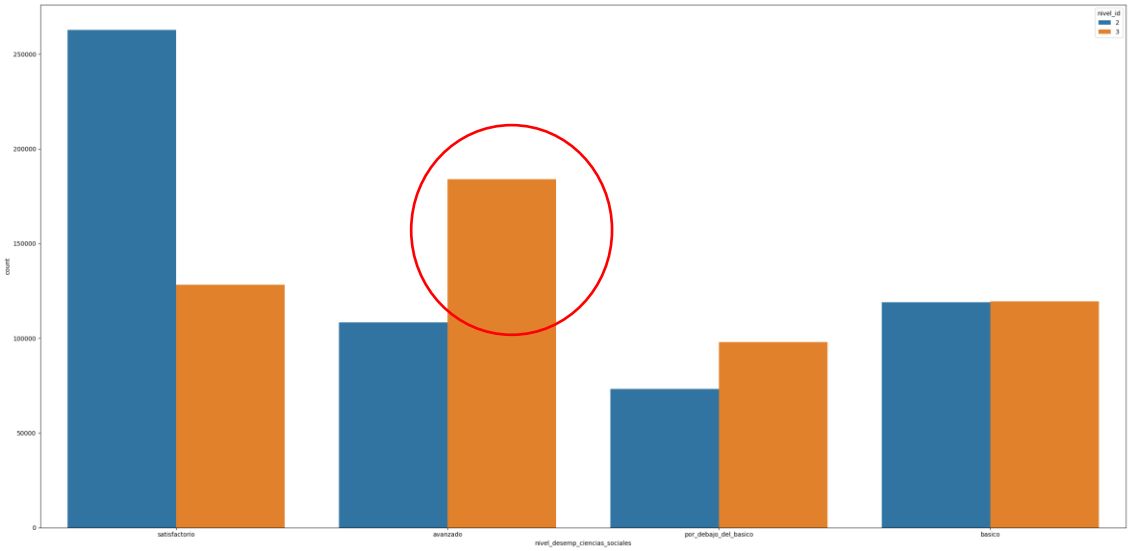
```
sns.countplot(x='nivel_desemp_lengua', hue='nivel_id', data=df)
```



En el caso de **lengua**, los registros de nivel de desempeño **por debajo del básico son también superiores en el secundario**.

c. Nivel de desempeño en ciencias sociales

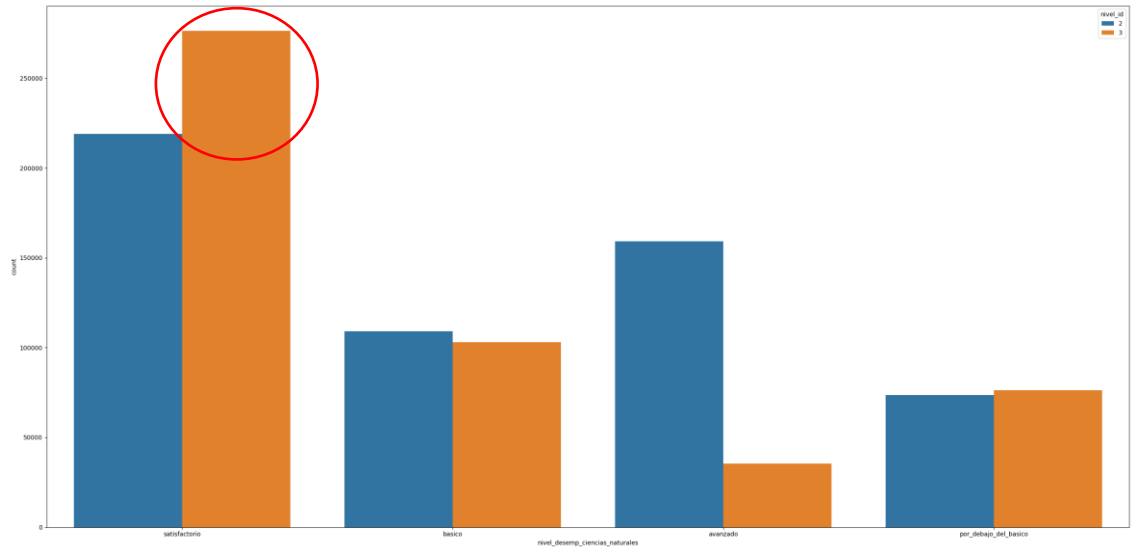
```
sns.countplot(x='nivel_desemp_ciencias_sociales', hue='nivel_id', data=df)
```



En el caso de ciencias sociales, **la tendencia con respecto a matemáticas y lengua se modifica en el nivel avanzado** y la participación de la proporción en secundario es mayor que en el primario.

d. Nivel de desempeño en ciencias naturales

```
sns.countplot(x='nivel_desemp_ciencias_naturales', hue='nivel_id', data=df)
```

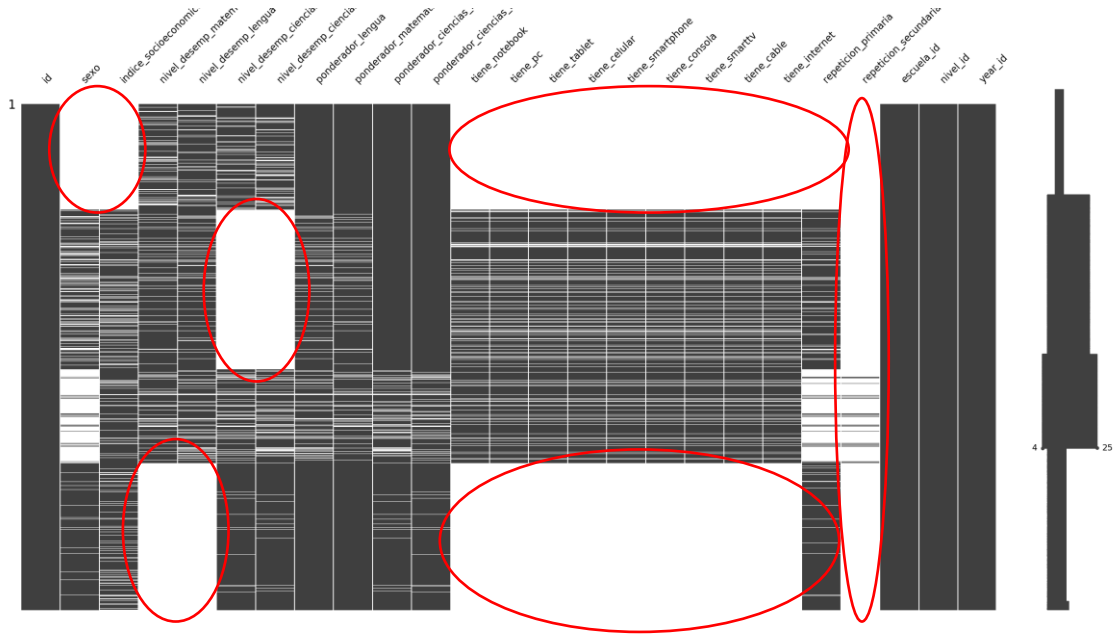


En el caso de ciencias naturales, se observa una **mayor participación de la proporción de nivel satisfactorio en secundario que en el primario**, invirtiendo la relación de la población total detallada en el análisis univariado.

MISSING VALUES

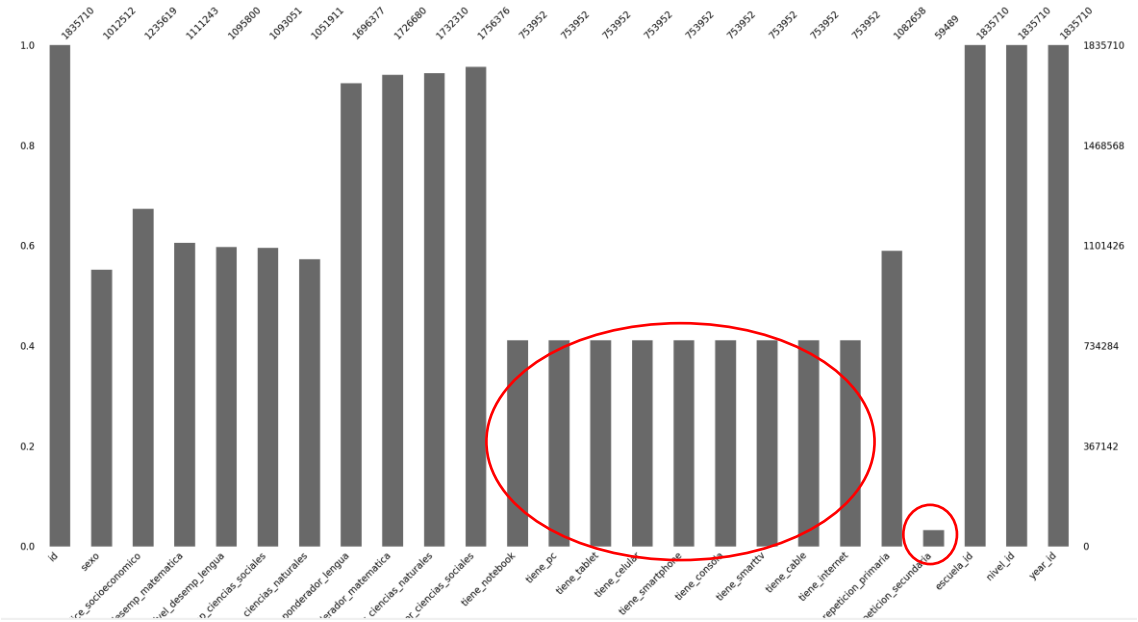
Más allá de la descripción general de datos obtenida a través de líneas como .info(), realicé un análisis detallado de valores faltantes.

```
import missingno as msno

msno.matrix(df)
```



Existen **secciones muy significativas de valores faltantes** en muchas variables (particularmente en las preguntas de acceso a tecnología). Se observa preliminarmente que los bloques faltantes en las variables nivel_desempeño van a presentar un desafío central para el modelo predictivo.

```
msno.bar(df)
```

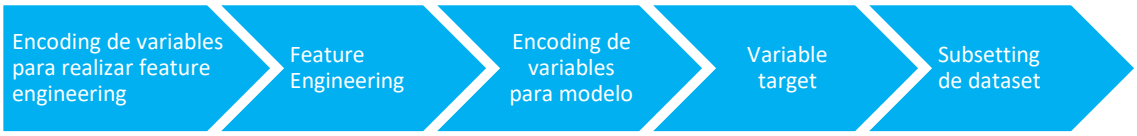


De manera acumulada, se observa que las **variables asociadas a preguntas de tecnología tienen un promedio de 60% de valores faltantes**, y las variables **repetición_secundaria** cuenta con la menor cantidad de registros completos **(+90% de registros faltantes).**

PROCESAMIENTO DE INFORMACIÓN. HALLAZGOS Y DESAFÍOS

Sin dudas esta fue la actividad que me representó mayor esfuerzo y cantidad de horas de desarrollo. Las definiciones de encoding me llevaron mucho tiempo (probé distintos métodos como one-hot enconding y ordinal enconding para distintas variables, experimenté con distintas escalas y distintas binarizaciones), así como el desarrollo de features que reflejaran lo que pretendía. Luego de realizar muchas pruebas y errores, he tratado de priorizar que los datos se mantuvieran transparentes e interpretables para el modelo predictivo. En mi caso, convertí al problema en un desafío de **regresión**, intentando predecir el valor de una variable target que integra el desempeño de las distintas materias. Los pasos que seguí fueron:

- ***Preprocesamiento***



- ***Modelos predictivos***

A continuación, detallaré cada uno de estos pasos para preprocesamiento y modelado.

PREPROCESAMIENTO

1) Encoding de variables para realizar feature engineering

Este paso de encoding lo enfoqué en preparar ciertas variables para optimizar el próximo paso de feature engineering. Como se verá, luego de realizar feature engineering volví a realizar un proceso de encoding para preparar el dataset para el siguiente paso de modelado.

Luego de testear distintas estrategias para realizar encoding de variables categóricas, decidí dividir el universo de variables en **2 grupos**:

- Definí un grupo de variables sobre el cual realizar un **encoding ordinal**, contemplando:
  - indice_socioeconomico

```
indice_socioeconomico_dict = {          'alto':1,
```

```
                                        'medio':2,
                                        'bajo':3}
df['indice_socioeconomico_ord'] = df.indice_socioeconomico.map(indice_socioeconomico_dict)
df['indice_socioeconomico_ord'] = df['indice_socioeconomico_ord'].fillna(0)
```

- nivel_desemp_matematica
- nivel_desemp_lengua
- nivel_desemp_ciencias_sociales
- nivel_desemp_ciencias_naturales

```
nivel_desemp_dict = {   'avanzado':4,
                        'satisfactorio':3,
                        'basico':2,
                        'por_debajo_del_basico':1}
df['nivel_desemp_matematica_ord'] = df.nivel_desemp_matematica.map(nivel_desemp_dict)
df['nivel_desemp_lengua_ord'] = df.nivel_desemp_lengua.map(nivel_desemp_dict)
df['nivel_desemp_ciencias_sociales_ord'] = df.nivel_desemp_ciencias_sociales.map(nivel_desemp_dict)
df['nivel_desemp_ciencias_naturales_ord'] = df.nivel_desemp_ciencias_naturales.map(nivel_desemp_dict)
df[['nivel_desemp_matematica_ord', 'nivel_desemp_lengua_ord', 'nivel_desemp_ciencias_sociales_ord',
'nivel_desemp_ciencias_naturales_ord']] = df[['nivel_desemp_matematica_ord', 'nivel_desemp_lengua_ord',
'nivel_desemp_ciencias_sociales_ord', 'nivel_desemp_ciencias_naturales_ord']].fillna(0)
```

- repeticion_primaria
- repeticion_secundaria

```
repeticion_dict = {  'no':4,
                     'una_vez':3,
                     'dos_veces':2,
                     'tres_veces_o_mas':1}
df['repeticion_primaria_ord'] = df.repeticion_primaria.map(repeticion_dict)
df['repeticion_secundaria_ord'] = df.repeticion_secundaria.map(repeticion_dict)
df[['repeticion_primaria_ord', 'repeticion_secundaria_ord']] = df[['repeticion_primaria_ord',
'repeticion_secundaria_ord']].fillna(0)
```

- Definí otro grupo de variables sobre el cual realizar un **encoding binario**, incluyendo:
  - sexo

```
df['sexo_bin'] = df.sexo.replace({'f':1, 'm':-1})
df['sexo_bin'] = df['sexo_bin'].fillna(0)
```

- tiene_notebook

```
df['tiene_notebook_bin'] = df.tiene_notebook.replace({'si':1, 'no':-1})
```

```
df['tiene_notebook_bin'] = df['tiene_notebook_bin'].fillna(0)
```

- tiene_pc

```
df['tiene_pc_bin'] = df.tiene_pc.replace({'si':1, 'no':-1})
df['tiene_pc_bin'] = df['tiene_pc_bin'].fillna(0)
```

- tiene_tablet

```
df['tiene_tablet_bin'] = df.tiene_tablet.replace({'si':1, 'no':-1})
df['tiene_tablet_bin'] = df['tiene_tablet_bin'].fillna(0)
```

- tiene_celular

```
df['tiene_celular_bin'] = df.tiene_celular.replace({'si':1, 'no':-1})
df['tiene_celular_bin'] = df['tiene_celular_bin'].fillna(0)
```

- tiene_smartphone

```
df['tiene_smartphone_bin'] = df.tiene_smartphone.replace({'si':1, 'no':-1})
df['tiene_smartphone_bin'] = df['tiene_smartphone_bin'].fillna(0)
```

- tiene_consola

```
df['tiene_consola_bin'] = df.tiene_consola.replace({'si':1, 'no':-1})
df['tiene_consola_bin'] = df['tiene_consola_bin'].fillna(0)
```

- tiene_smarttv

```
df['tiene_smarttv_bin'] = df.tiene_smarttv.replace({'si':1, 'no':-1})
df['tiene_smarttv_bin'] = df['tiene_smarttv_bin'].fillna(0)
```

- tiene_cable

```
df['tiene_cable_bin'] = df.tiene_cable.replace({'si':1, 'no':-1})
df['tiene_cable_bin'] = df['tiene_cable_bin'].fillna(0)
```

- tiene_internet

```
df['tiene_internet_bin'] = df.tiene_internet.replace({'si':1, 'no':-1})
df['tiene_internet_bin'] = df['tiene_internet_bin'].fillna(0)
```

2) Feature Engineering (incluyendo variable target)

Basándome en el análisis exploratorio realizado previamente, decidí **desarrollar variables que contemplen aspectos de acceso a tecnologías, desempeño en materias y nivel socioeconómico**.

Las variables generadas son:

- **Tecnologías**

- variable "acceso a herramientas"

```
df['acceso_herramientas'] = df['tiene_notebook_bin'] + df['tiene_pc_bin'] + df['tiene_tablet_bin'] +
df['tiene_celular_bin'] + df['tiene_smartphone_bin'] + df['tiene_consola_bin'] + df['tiene_smarttv_bin'] +
df['tiene_cable_bin'] + df['tiene_internet_bin']

df['acceso_herramientas'] = df['acceso_herramientas'].fillna(0)
```

- variable "acceso a herramientas de sociabilización"

```
df['acceso_herram_soc'] = df['tiene_notebook_bin'] + df['tiene_pc_bin'] + df['tiene_celular_bin'] +
df['tiene_smartphone_bin'] + df['tiene_internet_bin']

df['acceso_herram_soc'] = df['acceso_herram_soc'].fillna(0)
```

- variable "acceso a herramientas de educación"

```
df['acceso_herram_educ'] = df['tiene_tablet_bin'] + + df['tiene_internet_bin']

df['acceso_herram_educ'] = df['acceso_herram_educ'].fillna(0)
```

- variable "acceso a herramientas de esparcimiento"

```
df['acceso_herram_esparc'] = df['tiene_consola_bin'] + df['tiene_smarttv_bin'] + df['tiene_cable_bin']

df['acceso_herram_esparc'] = df['acceso_herram_esparc'].fillna(0)
```

- **Desempeño**

- variables "desempeño por materia (ponderando por materia) vinculado a repetición primario"

```
df['desemp_mat_repeticion_prim'] = (df['nivel_desemp_matematica_ord'] * df.ponderador_matematica) *
df['repeticion_primaria_ord']

df['desemp_len_repeticion_prim'] = (df['nivel_desemp_lengua_ord'] * df.ponderador_lengua) *
df['repeticion_primaria_ord']

df['desemp_cs_s_repeticion_prim'] = (df['nivel_desemp_ciencias_sociales_ord'] * df.ponderador_ciencias_sociales)
* df['repeticion_primaria_ord']

df['desemp_cs_n_repeticion_prim'] = (df['nivel_desemp_ciencias_naturales_ord'] *
df.ponderador_ciencias_naturales) * df['repeticion_primaria_ord']

df[['desemp_mat_repeticion_prim', 'desemp_len_repeticion_prim', 'desemp_cs_s_repeticion_prim',
'desemp_cs_n_repeticion_prim']] = df[['desemp_mat_repeticion_prim', 'desemp_len_repeticion_prim',
'desemp_cs_s_repeticion_prim', 'desemp_cs_n_repeticion_prim']].fillna(0)
```

- variables "desempeño por materia (ponderando por materia) vinculado a repetición secundario"

```
df['desemp_mat_repeticion_sec'] = (df['nivel_desemp_matematica_ord'] * df.ponderador_matematica) *
df['repeticion_secundaria_ord']

df['desemp_len_repeticion_sec'] = (df['nivel_desemp_lengua_ord'] * df.ponderador_lengua) *
df['repeticion_secundaria_ord']
```

```
df['desemp_cs_s_repeticion_sec'] = (df['nivel_desemp_ciencias_sociales_ord'] * df.ponderador_ciencias_sociales) * df['repeticion_secundaria_ord']

df['desemp_cs_n_repeticion_sec'] = (df['nivel_desemp_ciencias_naturales_ord'] * df.ponderador_ciencias_naturales) * df['repeticion_secundaria_ord']

df[['desemp_mat_repeticion_sec', 'desemp_len_repeticion_sec', 'desemp_cs_s_repeticion_sec', 'desemp_cs_n_repeticion_sec']] = df[['desemp_mat_repeticion_sec', 'desemp_len_repeticion_sec', 'desemp_cs_s_repeticion_sec', 'desemp_cs_n_repeticion_sec']].fillna(0)
```

- variables "desempeño por materia (ponderando por materia) vinculado a nivel socioeconómico"

```
df['desemp_mat_indice_socio'] = (df['nivel_desemp_matematica_ord'] * df.ponderador_matematica) * df['indice_socioeconomico_ord']

df['desemp_len_indice_socio'] = (df['nivel_desemp_lengua_ord'] * df.ponderador_lengua) * df['indice_socioeconomico_ord']

df['desemp_cs_s_indice_socio'] = (df['nivel_desemp_ciencias_sociales_ord'] * df.ponderador_ciencias_sociales) * df['indice_socioeconomico_ord']

df['desemp_cs_n_indice_socio'] = (df['nivel_desemp_ciencias_naturales_ord'] * df.ponderador_ciencias_naturales) * df['indice_socioeconomico_ord']

df[['desemp_mat_indice_socio', 'desemp_len_indice_socio', 'desemp_cs_s_indice_socio', 'desemp_cs_n_indice_socio']] = df[['desemp_mat_indice_socio', 'desemp_len_indice_socio', 'desemp_cs_s_indice_socio', 'desemp_cs_n_indice_socio']].fillna(0)
```

- **Desempeño y Tecnología**

- variables "desempeño por materia (ponderando por materia) vinculado a variable "acceso a herramientas"

```
df['desemp_mat_acceso_herramientas'] = (df['nivel_desemp_matematica_ord'] * df.ponderador_matematica) * df['acceso_herramientas']

df['desemp_len_acceso_herramientas'] = (df['nivel_desemp_lengua_ord'] * df.ponderador_lengua) * df['acceso_herramientas']

df['desemp_cs_c_acceso_herramientas'] = (df['nivel_desemp_ciencias_sociales_ord'] * df.ponderador_ciencias_sociales) * df['acceso_herramientas']

df['desemp_cs_n_acceso_herramientas'] = (df['nivel_desemp_ciencias_naturales_ord'] * df.ponderador_ciencias_naturales) * df['acceso_herramientas']

df[['desemp_mat_acceso_herramientas', 'desemp_len_acceso_herramientas', 'desemp_cs_c_acceso_herramientas', 'desemp_cs_n_acceso_herramientas']] = df[['desemp_mat_acceso_herramientas', 'desemp_len_acceso_herramientas', 'desemp_cs_c_acceso_herramientas', 'desemp_cs_n_acceso_herramientas']].fillna(0)
```

- variables "desempeño por materia (ponderando por materia) vinculado a variable "acceso a herramientas de sociabilización"

```
df['desemp_mat_acceso_herram_soc'] = (df['nivel_desemp_matematica_ord'] * df.ponderador_matematica) * df['acceso_herram_soc']
```

```
df['desemp_len_acceso_herram_soc'] = (df['nivel_desemp_lengua_ord'] * df.ponderador_lengua) *
df['acceso_herram_soc']

df['desemp_cs_c_acceso_herram_soc'] = (df['nivel_desemp_ciencias_sociales_ord'] *
df.ponderador_ciencias_sociales) * df['acceso_herram_soc']

df['desemp_cs_n_acceso_herram_soc'] = (df['nivel_desemp_ciencias_naturales_ord'] *
df.ponderador_ciencias_naturales) * df['acceso_herram_soc']

df[['desemp_mat_acceso_herram_soc', 'desemp_len_acceso_herram_soc', 'desemp_cs_c_acceso_herram_soc',
'desemp_cs_n_acceso_herram_soc']] = df[['desemp_mat_acceso_herram_soc', 'desemp_len_acceso_herram_soc',
'desemp_cs_c_acceso_herram_soc', 'desemp_cs_n_acceso_herram_soc']].fillna(0)
```

- variables "desempeño por materia (ponderando por materia) vinculado a variable "acceso a herramientas de educación"

```
df['desemp_mat_acceso_herram_educ'] = (df['nivel_desemp_matematica_ord'] * df.ponderador_matematica) *
df['acceso_herram_educ']

df['desemp_len_acceso_herram_educ'] = (df['nivel_desemp_lengua_ord'] * df.ponderador_lengua) *
df['acceso_herram_educ']

df['desemp_cs_c_acceso_herram_educ'] = (df['nivel_desemp_ciencias_sociales_ord'] *
df.ponderador_ciencias_sociales) * df['acceso_herram_educ']

df['desemp_cs_n_acceso_herram_educ'] = (df['nivel_desemp_ciencias_naturales_ord'] *
df.ponderador_ciencias_naturales) * df['acceso_herram_educ']

df[['desemp_mat_acceso_herram_educ', 'desemp_len_acceso_herram_educ',
'desemp_cs_c_acceso_herram_educ', 'desemp_cs_n_acceso_herram_educ']] =
df[['desemp_mat_acceso_herram_educ', 'desemp_len_acceso_herram_educ',
'desemp_cs_c_acceso_herram_educ', 'desemp_cs_n_acceso_herram_educ']].fillna(0)
```

- variables "desempeño por materia (ponderando por materia) vinculado a variable "acceso a herramientas de esparcimiento"

```
df['desemp_mat_acceso_herram_esparc'] = (df['nivel_desemp_matematica_ord'] * df.ponderador_matematica) *
df['acceso_herram_esparc']

df['desemp_len_acceso_herram_esparc'] = (df['nivel_desemp_lengua_ord'] * df.ponderador_lengua) *
df['acceso_herram_esparc']

df['desemp_cs_c_acceso_herram_esparc'] = (df['nivel_desemp_ciencias_sociales_ord'] *
df.ponderador_ciencias_sociales) * df['acceso_herram_esparc']

df['desemp_cs_n_acceso_herram_esparc'] = (df['nivel_desemp_ciencias_naturales_ord'] *
df.ponderador_ciencias_naturales) * df['acceso_herram_esparc']

df[['desemp_mat_acceso_herram_esparc', 'desemp_len_acceso_herram_esparc',
'desemp_cs_c_acceso_herram_esparc', 'desemp_cs_n_acceso_herram_esparc']] =
df[['desemp_mat_acceso_herram_esparc', 'desemp_len_acceso_herram_esparc',
'desemp_cs_c_acceso_herram_esparc', 'desemp_cs_n_acceso_herram_esparc']].fillna(0)
```

3) Encoding de variables enfocado en el modelo predictivo

Decidí hacer **one hot encoding** de ciertas variables para, buscando **mayor robustez** al momento de incluirlas en el dataset del modelo predictivo:

```
categorical_cols = ['indice_socioeconomico',
```

```
                    'sexo',
                    'tiene_notebook',
                    'tiene_pc',
                    'tiene_tablet',
                    'tiene_celular',
                    'tiene_smartphone',
                    'tiene_consola',
                    'tiene_smarttv',
                    'tiene_cable',
                    'tiene_internet']
df = pd.get_dummies(df, columns = categorical_cols)
```

4) Variable target

Como mencionado, decidí generar un indicador numérico que integre el desempeño por registro en las distintas materias. Sobre este indicador que llamo "nivel de desempeño total", realizaré el modelo predictivo:

$$\begin{aligned}&\text{Nivel de desempeño total}\\&= Nivel\ desempeño\ matemática\ \times Ponderador\ matemática\\&+ Nivel\ desempeño\ lengua\ \times Ponderador\ lengua\\&+ Nivel\ desempeño\ ciencias\ sociales\ \times Ponderador\ ciencias\ sociales\\&+ Nivel\ desempeño\ ciencias\ naturales\ \times Ponderador\ ciencias\ naturales\end{aligned}$$

```
df['nivel_desemp_total'] = (df.nivel_desemp_matematica_ord * df.ponderador_matematica) +
(df.nivel_desemp_lengua_ord * df.ponderador_lengua) + (df.nivel_desemp_ciencias_sociales_ord *
df.ponderador_ciencias_sociales) + (df.nivel_desemp_ciencias_naturales_ord * df.ponderador_ciencias_naturales)

df['nivel_desemp_total'] = df['nivel_desemp_total'].fillna(0)
```

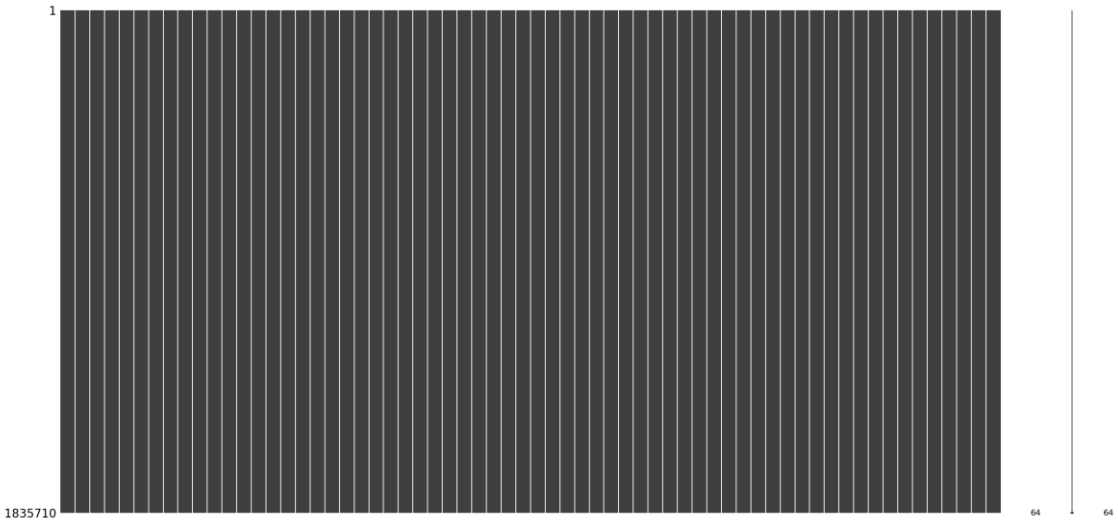
5) Subsetting de dataset

Como paso previo al desarrollo del modelo predictivo, realicé un subsetting del dataframe para **quitar variables redundantes** o empleadas anteriormente para otros fines:

```
df1 = df.copy()

df1 = df1.iloc[:,12:]

vbles_elim = [      'indice_socioeconomico_ord',
                    'sexo_bin',
                    'tiene_notebook_bin',
                    'tiene_pc_bin',
```

```
                    'tiene_tablet_bin',
                    'tiene_celular_bin',
                    'tiene_smartphone_bin',
                    'tiene_consola_bin',
                    'tiene_smarttv_bin',
                    'tiene_cable_bin',
                    'tiene_internet_bin']
df1.drop(vbles_elim, inplace= True, axis=1)
```

El nuevo dataframe depurado se compone de 1.835.710 registros y 64 variables, siendo la variable "nivel_desemp_total" la variable target. Realizo nuevamente un análisis de missing values para validar que el preprocesamiento previo ocurrió correctamente:



Perfecto, todas las variables se encuentran completar y puedo avanzar hacia la etapa de modelado.

MODELOS PREDICTIVOS

Modelo predictivo 1 – benchmark

Como mencionado anteriormente, definí al problema como un desafío de **regresión**. Para eso elegí utilizar **Random Forests** como algoritmo para el modelo de predicción, con la intención de poder estimar la variable target.

1) Features y target

Luego de haber definido el dataframe final para el modelo predictivo, separo el mismo en features y target. Primero importo las librerías que voy a necesitar para el modelado y luego hago la separación de variables:

```
from sklearn.ensemble import RandomForestRegressor

from sklearn.model_selection import train_test_split

from sklearn import metrics

from sklearn.metrics import r2_score

X = df1.iloc[:,:63].copy()

y = df1.iloc[:,63].copy()
```

2) Separación en train y test

Realizo una separación en set de train y de test (método hold out), utilizando el 70% de los datos para entrenar el modelo y el 30% para testearlo:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

3) Random Forest e hiperparámetros

Realicé un setting standard de Random Forest, eligiendo MSE como criterio de partición, 10 estimadores y topeando la profunidad máxima de los árboles en 10:

```
rf = RandomForestRegressor(n_estimators = 10, criterion = 'mse', max_depth = 10)
```

4) Entrenamiento y Testing

```
model_r = rf.fit(X_train, y_train)

y_pred = model_r.predict(X_test)
```

5) Medición de performance

Para medir la performance de este modelo inicial y los siguientes, utilizo las métricas MAE, MSE, RMSE y R^2:

```
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))

print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

print('R Squared Score is:', r2_score(y_test, y_pred))
```

Los resultados son:

```
Mean Absolute Error: 4.04281418905499
Mean Squared Error: 1057.6526326835485
Root Mean Squared Error: 32.52157180524257
R Squared Score is: 0.36919583455279525
```

Ya tenemos un benchmark para intentar superarlo. Realicé un listado de feature importance y los primeros resultados fueron:

```
for importance, name in sorted(zip(rf.feature_importances_, X_train.columns),reverse=True):
    print (name, importance)
```

```
nivel_id 0.6214652396596263
year_id 0.11626870887467691
nivel_desemp_ciencias_naturales_ord 0.07594687829317909
nivel_desemp_matematica_ord 0.06484915015252254
nivel_desemp_lengua_ord 0.05449011394020551
nivel_desemp_ciencias_sociales_ord 0.032913095348389654
desemp_cs_n_acceso_herram_soc 0.007436768104920069
desemp_cs_s_indice_socio 0.006123947189454774
desemp_len_repeticion_prim 0.004104877009118648
desemp_len_acceso_herram_soc 0.0034104515718223283
desemp_mat_repeticion_prim 0.003006733791496315
desemp_cs_c_acceso_herram_soc 0.002106185414468056
desemp_cs_s_repeticion_prim 0.0019885543951420033
desemp_len_indice_socio 0.0016231200356084594
desemp_cs_n_repeticion_prim 0.0007683145515132398
desemp_mat_indice_socio 0.0007001698607428156
desemp_cs_n_indice_socio 0.0006807174018044678
```

Como se observa, nuestro algoritmo particiona por variables como nivel de educación y año en primera medida, luego utilizando las variables desarrolladas dentro de feature importance.

A continuación, realicé 2 nuevos modelos también con Random Forest, uno con datos agrupados por nivel de educación primario y el otro nivel de educación secundario, ya que mi hipótesis es que de esa manera podré obtener un modelo más apropiado a la problemático que intento resolver.

Modelo predictivo 2 – nivel de educación primario

Utilicé el mismo dataset que para modelo 1, solamente filtrado por registros con nivel de educación **primario**, dando un resultado de 1.151.819 observaciones y 64 columnas. Siguiendo la misma estrategia de hold out y utilizando el mismo algoritmo de Random Forest con exactamente los mismos hiperparámtros que en el modelo 1, el resultado del modelo es:

```
Mean Absolute Error: 5.3852083629678305
Mean Squared Error: 1673.4078589768433
Root Mean Squared Error: 40.90730813652792
R Squared Score is: 0.3647514125773089
```

Aunque hemos especificado un modelo para el primario, la performance es casi la misma (levemente inferior) a la del modelo general. Veamos qué sucedió con el feature importance:

```
year_id 0.7598195730914381
nivel_desemp_matematica_ord 0.07549474407681164
nivel_desemp_lengua_ord 0.06896464882990944
nivel_desemp_ciencias_naturales_ord 0.048477175274775755
nivel_desemp_ciencias_sociales_ord 0.030680489349936828
desemp_len_repeticion_prim 0.005934839116144548
desemp_cs_n_repeticion_prim 0.005120707993735601
desemp_mat_repeticion_prim 0.0027890664700541893
desemp_cs_s_repeticion_prim 0.0019320348869924086
repeticion_primaria_ord 0.00043652391196402705
desemp_len_indice_socio 0.00026630438463092095
desemp_mat_indice_socio 3.644283566184077e-05
desemp_cs_n_indice_socio 2.497935055135475e-05
indice_socioeconomico_bajo 6.894983704916209e-06
desemp_cs_s_indice_socio 6.344713568864159e-06
desemp_len_acceso_herram_educ 2.2724419782288523e-06
desemp_len_acceso_herram_esparc 1.7798004780844152e-06
desemp_len_acceso_herramientas 1.135433050146485e-06
desemp_len_acceso_herram_soc 8.568421813370233e-07
```

El año pasó a ser la variable con mayor poder predictivo, y el peso relativo de las otras variables desarrolladas en el ejercicio de feature engineering se mantuvieron relativamente iguales. Veamos qué sucede con el siguiente modelo enfocado en registros de secundario.

Modelo predictivo 3 – nivel de educación secundario

Utilicé el mismo dataset que para modelo 1, solamente filtrado por registros con nivel de educación **secundario**, dando un resultado de 683.891 observaciones y 64 columnas. Siguiendo la misma estrategia de hold out y utilizando el mismo algoritmo de Random Forest con exactamente los mismos hiperparámtros que en el modelo 1, el resultado del modelo es:

```
Mean Absolute Error: 1.5173093742030939
Mean Squared Error: 7.169764850466768
Root Mean Squared Error: 2.6776416583379428
R Squared Score is: 0.9042825493879428
```

El modelo ahora dio un resultado excelente, muy distintos a los modelos 1 y 2 detallados anteriormente. Es más, si miramos el R^2 podría considerarse que la performance fue tan buena que el modelo posiblemente esté overfitteando.

Veamos qué sucedió con el feature importance:

```
nivel_desemp_ciencias_naturales_ord 0.2747829886548231
desemp_cs_s_indice_socio 0.15592459202464543
nivel_desemp_ciencias_sociales_ord 0.12528942577648552
nivel_desemp_lengua_ord 0.09436230121647435
desemp_len_indice_socio 0.08459700620738317
nivel_desemp_matematica_ord 0.0717751272884817
desemp_cs_n_indice_socio 0.038633804164805365
year_id 0.03845204813756955
desemp_mat_indice_socio 0.033434903939542
desemp_len_acceso_herram_soc 0.014402728086763673
indice_socioeconomico_bajo 0.013529474252099095
desemp_cs_n_acceso_herram_soc 0.012335100005991547
indice_socioeconomico_alto 0.00822817347733999
desemp_len_repeticion_prim 0.0057680245858085346
desemp_cs_c_acceso_herram_soc 0.00474520278487338
indice_socioeconomico_medio 0.0036668993225058683
acceso_herram_soc 0.0030674146085563577
desemp_cs_c_acceso_herram_educ 0.0028282690397604766
desemp_mat_acceso_herram_soc 0.002541128817170865
tiene_celular_si 0.0021310159588552656
desemp_mat_repeticion_prim 0.0019341687979535249
```

El modelo está particionando de buena manera utilizando las variables creadas en la etapa de feature engineering, lo que hace pensar que funciona adecuadamente para este dataset (a diferencia del modelo 2). Rankean en mejor posición no solamente las variables asociadas exclusivamente al desempeño, sino también las que incorporan aspectos del **nivel socioeconómico**, afirmando la hipótesis surgida del análisis exploratorio bivariado donde se detectó una relación entre desempeño y nivel socioeconómico.

A continuación, detallo algunas conclusiones de todo este ejercicio.

CONCLUSIONES

Una de las primeras observaciones producto del análisis exploratorio de datos han sido los niveles de desempeño en matemáticas, lengua, ciencias sociales y naturales, y el hecho que en todos los casos la clase mayoritaria es la satisfactoria: ¿es esto producto de un **sesgo en la evaluación hacia la media?** Sería interesante estresar las clases para observar la verdadera distribución de casos.

Ya que la gran mayoría de registros que respondieron el cuestionario **tienen celular**, debería **acentuarse la comunicación** por esta vía con los alumnos para dar seguimiento a las asistencias, inscripciones en exámenes, etc.

Otra observación es que la mayoría de registros que respondieron el cuestionario **tienen cable y tienen internet**, lo que indica una oportunidad para **explotar** aún más **sistemas de educación remotos y digitales**, mientras se trabaja en la inclusión de quienes no tienen acceso a estos canales de comunicación.

La tasa de respuesta a las preguntas vinculadas a aspectos tecnológicos/digitales es menor al 50%, por lo que hay una oportunidad para **mejorar la captura de datos** que puedan usarse como predictores de comportamiento.

Del análisis exploratorio bivariado se observa una **fuerte relación entre el nivel socioeconómico y los niveles de desempeño** en todas las materias, con una marcada tendencia hacia menores niveles de desempeño en niveles socioeconómicos bajos, resaltando el flagelo de la desigualdad económica en nuestro país. Es precisamente esto lo que quise reflejar en la etapa de modelado, viéndose plasmado claramente el modelo 3 (utilizado con registros del nivel de educación secundario).

De los modelos predictivos surgen varias conclusiones interesantes. Por un lado, se observa que al intentar **predecir de manera general** el resultado para primario y secundario **la performance del modelo es baja**, sin embargo, cuando intentamos **predecir el resultado del nivel secundario** de manera específica, **la performance se incrementa** abruptamente. Otra sorpresa es que el modelo para primario es casi idéntico al modelo general (no solo en performance sino también en partición), por lo que el modelo no parece adaptarse bien a la estructura de datos, pero al aplicar el mismo modelo sobre la población de secundario el mismo es tan preciso que roza el overfitting.

Me hubiese gustado incorporar otros datasets que se encontraban disponibles para aumentar la dimensionalidad del universo de datos, pero sinceramente no llegué con los tiempos ya que preferí comprender en profunidad el postulado inicial.

Como oportunidades de mejora sería bueno mejorar la optimización de hiperparámetro (ejemplo aplicando random search), y explorar otras variantes de modelos, ya que aunque Random Forest cuenta excelentes características de flexibilidad y adaptación, tal vez no pueda capturar las características intrínsecas del dataset.