

✓ Bootcamp Análisis de datos Nivel Innovador - Talento Tech

PROYECTO FINAL

Estudiante: Julian Ramirez Burbano

Haz doble clic (o pulsa Intro) para editar

- Link repositorio GitHub = <https://github.com/julirami/proyectoTTCH>
- Link Video YouTube = xxxxxxxxxxxxxxxxx

✓ Nombre proyecto: Análisis de Desempeño en la Prueba Saber 11

Introducción:

En este proyecto se realiza un completo análisis del desempeño de los estudiantes en la Prueba Saber 11 en el periodo 2019-2 específicamente en el departamento de Nariño (Colombia). Esta prueba es un examen obligatorio que se realiza en Colombia con el fin de medir las habilidades y conocimientos que han adquirido los estudiantes al terminar la educación media más conocida como bachillerato. Se usaron técnicas de visualización, análisis estadístico y modelado predictivo, para estudiar todos los datos y así lograr analizar distintos aspectos del desempeño estudiantil en las asignaturas evaluadas y su relación con el género y el municipio donde residen.

Datos Utilizados:

Los datos que utilicé para analizar fueron obtenidos del repositorio público DATOS ABIERTOS de Colombia en la Web WWW.GOV.CO específicamente en el link = <https://www.datos.gov.co/Educaci-n/PUNTAJE-ICFES-POR-DEPARTAMENTOS/x9vi-iv8c>

El conjunto de datos tiene información detallada sobre el rendimiento de los estudiantes en la Prueba Saber 11 en diferentes áreas como matemáticas, lectura crítica, ciencia y ciencias naturales, sociales y ciudadanas, y puntajes globales. También, hay información de la ubicación de los estudiantes, su género y el municipio de residencia.

✓ Objetivos

- Examinar la distribución de los puntajes obtenidos por los estudiantes en las diferentes áreas evaluadas, que incluyen matemáticas, lectura crítica, ciencias naturales, sociales y ciudadanas, e inglés. Esto permitirá identificar posibles tendencias o patrones en los puntajes globales, proporcionando una visión general del rendimiento académico en el departamento de Nariño durante el periodo de estudio.
- Evaluar las diferencias en los puntajes obtenidos por los estudiantes según su género. Además, se busca comparar los puntajes globales entre los diferentes municipios del departamento de Nariño. Esto ayudará a entender mejor cómo el género y el lugar de residencia pueden influir en el desempeño académico de los estudiantes en la Prueba Saber 11.
- Entender cómo tanto el nivel educativo de los padres como la disponibilidad de recursos tecnológicos y otros en el hogar influyen en el rendimiento académico de los estudiantes. Utilizando gráficos de calor y otras herramientas de visualización, se pretende identificar correlaciones significativas entre estos factores y los puntajes globales obtenidos en las pruebas. Este análisis proporcionará insights importantes para comprender la influencia de estas variables en el desempeño estudiantil, lo que podría guiar futuras intervenciones educativas.

✓ Variables

Variables Demográficas y Contextuales:

- ESTU_TIPODOCUMENTO: Tipo de documento del estudiante.
- ESTU_NACIONALIDAD: Nacionalidad del estudiante.

- ESTU_GENERO: Género del estudiante.
- ESTU_FECHANACIMIENTO: Fecha de nacimiento del estudiante.
- ESTU_DEPTO_RESIDE: Departamento de residencia del estudiante.
- ESTU_MCPIO_RESIDE: Municipio de residencia del estudiante.
- ESTU_DEPTO_PRESENTACION: Departamento de presentación de la prueba.
- ESTU_MCPIO_PRESENTACION: Municipio de presentación de la prueba.

Variables Familiares y Socioeconómicas:

- FAMI_ESTRATOVIVIENDA: Estrato de vivienda.
- FAMI_PERSONASHOGAR: Número de personas en el hogar.
- FAMI_CUARTOSHOGAR: Número de cuartos en el hogar.
- FAMI_EDUCACIONPADRE: Nivel educativo del padre.
- FAMI_EDUCACIONMADRE: Nivel educativo de la madre.
- FAMI_TIENEINTERNET: Si tienen internet en el hogar.
- FAMI_TIENEComputador: Si tienen computadora en el hogar.
- FAMI_TIENECONSOLAVIDEOJUEGOS: Si tienen consola de videojuegos en el hogar.
- FAMI_NUMLIBROS: Número de libros en el hogar.

Variables Académicas:

- COLE_NOMBRE_ESTABLECIMIENTO: Nombre del colegio.
- COLE_GENERO: Género del colegio (mixto, masculino, femenino).
- COLE_NATURALEZA: Naturaleza del colegio (oficial, privado).
- COLECALENDARIO: Calendario del colegio (A, B).
- COLE_CARACTER: Carácter del colegio (académico, técnico).
- COLE_AREA_UBICACION: Área de ubicación del colegio (urbano, rural).
- COLE_JORNADA: Jornada del colegio (mañana, tarde).

Variables de Desempeño:

- PUNT_LECTURA_CRITICA: Puntaje en lectura crítica.
- PUNT_MATEMATICAS: Puntaje en matemáticas.
- PUNT_C_NATURALES: Puntaje en ciencias naturales.
- PUNT_SOCIALES_CIUDADANAS: Puntaje en sociales y ciudadanas.
- PUNT_INGLES: Puntaje en inglés.
- PUNT_GLOBAL: Puntaje global.

```
# Se importan las librerías necesarias para el análisis
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import geopandas as gpd
import requests
from io import BytesIO
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score, accuracy_score, classification_report, confusion_matrix
from tabulate import tabulate

# Configuraciones iniciales
pd.set_option('display.max_columns', None)
sns.set(style="whitegrid")

# Cargar el dataset que se encuentra almacenado en un GitHub público
url = "https://raw.githubusercontent.com/julirami/proyectoTTCH/main/Saber_11_2019-2_20240516.csv"
data = pd.read_csv(url, sep=";")
```

▼ Proceso de Análisis:

El siguiente paso es analizar los datos, esto implica cargar y limpiar los datos, lo que involucra eliminar las filas que contienen valores nulos en las columnas de datos objetivo. También se realiza una exploración descriptiva de los datos.

Los resúmenes de las estadísticas, las visualizaciones, los histogramas, los gráficos de caja y los mapas geoespaciales, específicamente se usan para analizar cómo se distribuyen los puntajes en diferentes áreas, así como cualquier desigualdad que pueda existir basada en el género o el municipio.

```
# Se muestran las primeras filas del dataset para empezar el análisis exploratorio de datos
print(data.head())
```

| | | | |
|---|-------------------------------|-----------------------------|--------------------------------|
| 3 | N | 52835 | SAN ANDRÉS DE TUMACO |
| 4 | N | 52001 | PASTO |
| | | | |
| 0 | ESTU_DEPTO_PRESENTACION | ESTU_COD_DEPTO_PRESENTACION | PUNT_LECTURA_CRITICA \ |
| 1 | NARIÑO | 52 | 56 |
| 2 | NARIÑO | 52 | 41 |
| 3 | NARIÑO | 52 | 49 |
| 4 | NARIÑO | 52 | 39 |
| | | | |
| 0 | PERCENTIL_LECTURA_CRITICA | DESEMPL_LECTURA_CRITICA | PUNT_MATEMATICAS \ |
| 1 | 62 | 3 | 60 |
| 2 | 16 | 2 | 44 |
| 3 | 38 | 2 | 60 |
| 4 | 12 | 2 | 35 |
| | | | |
| 0 | 80 | 3 | 54 |
| | | | |
| 0 | PERCENTIL_MATEMATICAS | DESEMPL_MATEMATICAS | PUNT_C_NATURALES \ |
| 1 | 77 | 3 | 49 |
| 2 | 31 | 2 | 41 |
| 3 | 77 | 3 | 52 |
| 4 | 10 | 1 | 40 |
| | | | |
| 0 | 59 | 3 | 53 |
| | | | |
| 0 | PERCENTIL_C_NATURALES | DESEMPL_C_NATURALES | PUNT_SOCIALES_CIUDADANAS \ |
| 1 | 55 | 2 | 61 |
| 2 | 26 | 2 | 39 |
| 3 | 66 | 2 | 52 |
| 4 | 23 | 1 | 43 |
| | | | |
| 0 | 68 | 2 | 62 |
| | | | |
| 0 | PERCENTIL_SOCIALES_CIUDADANAS | DESEMPL_SOCIALES_CIUDADANAS | PUNT_INGLES \ |
| 1 | 88 | 3 | 57 |
| 2 | 33 | 1 | 45 |
| 3 | 68 | 2 | 50 |
| 4 | 43 | 2 | 35 |
| | | | |
| 0 | 88 | 3 | 49 |
| | | | |
| 0 | PERCENTIL_INGLES | DESEMPL_INGLES | PUNT_GLOBAL PERCENTIL_GLOBAL \ |
| 1 | 78 | A1 | 283 75 |
| 2 | 42 | A- | 208 25 |
| 3 | 57 | A1 | 265 65 |
| 4 | 15 | A- | 195 17 |
| | | | |
| 0 | 54 | A1 | 284 76 |
| | | | |
| 0 | ESTU_INSE_INDIVIDUAL | ESTU_NSE_INDIVIDUAL | ESTU_NSE_ESTABLECIMIENTO \ |
| 1 | 4,76492E+14 | 2.0 | 3 |
| 2 | 4,42588E+14 | 2.0 | 2 |
| 3 | 5,44029E+14 | 3.0 | 2 |
| 4 | 4,33939E+14 | 2.0 | 2 |
| | | | |
| 0 | 5,27653E+14 | 3.0 | 2 |
| | | | |
| 0 | ESTU_ESTADOINVESTIGACION | ESTU_GENERACION-E | |
| 1 | PUBLICAR | GENERACION E - GRATUIDAD | |
| 2 | PUBLICAR | GENERACION E - GRATUIDAD | |
| 3 | PUBLICAR | GENERACION E - GRATUIDAD | |
| 4 | PUBLICAR | GENERACION E - GRATUIDAD | |

```
# Se imprime información general del dataset
print(data.info())
```

```

32 FAMI_CUMECEREALFRUOSLEGUMBRE 15982 non-null object
33 FAMI_SITUACIONECONOMICA 16536 non-null object
34 ESTUDEDICACIONLECTURADIARIA 16872 non-null object
35 ESTUDEDICACIONINTERNET 15614 non-null object
36 ESTU_HORASSEMANATRABAJA 16884 non-null object
37 ESTUTIPOREMUNERACION 16270 non-null object
38 COLE_CODIGO_ICFES 16903 non-null int64
39 COLE_COD_DANE_ESTABLECIMIENTO 16903 non-null object
40 COLENOMBRE_ESTABLECIMIENTO 16903 non-null object
41 COLE_GENERO 16903 non-null object
42 COLE_NATURALEZA 16903 non-null object
43 COLE_CALENDARIO 16903 non-null object
44 COLE_BILINGUE 16903 non-null object
45 COLE_CARACTER 16434 non-null object
46 COLE_COD_DANE_SEDE 16903 non-null object
47 COLE_NOMBRE_SEDE 16903 non-null object
48 COLE_SEDE_PRINCIPAL 16903 non-null object
49 COLE_AREA_UBICACION 16903 non-null object
50 COLE_JORNADA 16903 non-null object
51 COLE_COD_MCPPIO_UBICACION 16903 non-null int64
52 COLE_MCPPIO_UBICACION 16903 non-null object
53 COLE_COD_DEPTO_UBICACION 16903 non-null int64
54 COLE_DEPTO_UBICACION 16903 non-null object
55 ESTU_PRIVADO_LIBERTAD 16903 non-null object
56 ESTU_COD_MCPPIO_PRESENTACION 16903 non-null int64
57 ESTU_MCPPIO_PRESENTACION 16903 non-null object
58 ESTU_DEPTO_PRESENTACION 16903 non-null object
59 ESTU_COD_DEPTO_PRESENTACION 16903 non-null int64
60 PUNT_LECTURA_CRITICA 16903 non-null int64
61 PERCENTIL_LECTURA_CRITICA 16903 non-null int64
62 DESEMP_LECTURA_CRITICA 16903 non-null int64
63 PUNT_MATEMATICAS 16903 non-null int64
64 PERCENTIL_MATEMATICAS 16903 non-null int64
65 DESEMP_MATEMATICAS 16903 non-null int64
66 PUNT_C_NATURALES 16903 non-null int64
67 PERCENTIL_C_NATURALES 16903 non-null int64
68 DESEMP_C_NATURALES 16903 non-null int64
69 PUNT_SOCIALES_CIUDADANAS 16903 non-null int64
70 PERCENTIL_SOCIALES_CIUDADANAS 16903 non-null int64
71 DESEMP_SOCIALES_CIUDADANAS 16903 non-null int64
72 PUNT_INGLES 16903 non-null int64
73 PERCENTIL_INGLES 16903 non-null int64
74 DESEMP_INGLES 16903 non-null object
75 PUNT_GLOBAL 16903 non-null int64
76 PERCENTIL_GLOBAL 16903 non-null int64
77 ESTU_INSE_INDIVIDUAL 16903 non-null object
78 ESTU_NSE_INDIVIDUAL 16191 non-null float64
79 ESTU_NSE_ESTABLECIMIENTO 16903 non-null int64
80 ESTU_ESTADOUNVESTIGACION 16903 non-null object
81 ESTU_GENERACION_E 16903 non-null object
dtypes: float64(1), int64(25), object(56)
memory usage: 10.6+ MB
None

```

```
# Explorando los datos
print(data.describe().T)
```

| | count | mean | std | min | \ |
|-------------------------------|---------|---------------|---------------|---------|---|
| PERIODO | 16903.0 | 20194.000000 | 0.000000 | 20194.0 | |
| ESTUCOD_RESIDE_DEPTO | 16903.0 | 52.000000 | 0.000000 | 52.0 | |
| ESTUCOD_RESIDE_MCPPIO | 16903.0 | 52366.028811 | 307.586957 | 52001.0 | |
| COLE_CODIGO_ICFES | 16903.0 | 119261.781400 | 164534.639443 | 9761.0 | |
| COLE_COD_MCPPIO_UBICACION | 16903.0 | 52381.887831 | 1445.789856 | 11001.0 | |
| COLE_COD_DEPTO_UBICACION | 16903.0 | 52.016151 | 1.406044 | 11.0 | |
| ESTUCOD_MCPPIO_PRESENTACION | 16903.0 | 52387.503283 | 1755.444520 | 5001.0 | |
| ESTUCOD_DEPTO_PRESENTACION | 16903.0 | 52.044312 | 1.725682 | 5.0 | |
| PUNT_LECTURA_CRITICA | 16903.0 | 51.222623 | 10.550399 | 0.0 | |
| PERCENTIL_LECTURA_CRITICA | 16903.0 | 47.643318 | 28.905991 | 1.0 | |
| DESEMP_LECTURA_CRITICA | 16903.0 | 2.550731 | 0.761231 | 1.0 | |
| PUNT_MATEMATICAS | 16903.0 | 50.683606 | 12.291657 | 15.0 | |
| PERCENTIL_MATEMATICAS | 16903.0 | 50.458262 | 29.450054 | 1.0 | |
| DESEMP_MATEMATICAS | 16903.0 | 2.437082 | 0.764202 | 1.0 | |
| PUNT_C_NATURALES | 16903.0 | 48.507543 | 11.165361 | 0.0 | |
| PERCENTIL_C_NATURALES | 16903.0 | 50.940839 | 29.778247 | 1.0 | |
| DESEMP_C_NATURALES | 16903.0 | 2.039401 | 0.783503 | 1.0 | |
| PUNT_SOCIALES_CIUDADANAS | 16903.0 | 45.617701 | 11.922236 | 17.0 | |
| PERCENTIL_SOCIALES_CIUDADANAS | 16903.0 | 48.785068 | 28.633698 | 1.0 | |
| DESEMP_SOCIALES_CIUDADANAS | 16903.0 | 1.859847 | 0.804929 | 1.0 | |
| PUNT_INGLES | 16903.0 | 47.115837 | 11.969232 | 0.0 | |
| PERCENTIL_INGLES | 16903.0 | 47.608413 | 28.713529 | 1.0 | |
| PUNT_GLOBAL | 16903.0 | 244.311306 | 52.061527 | 95.0 | |
| PERCENTIL_GLOBAL | 16903.0 | 49.085369 | 29.387936 | 1.0 | |
| ESTU_NSE_INDIVIDUAL | 16191.0 | 1.880921 | 0.869013 | 1.0 | |
| ESTU_NSE_ESTABLECIMIENTO | 16903.0 | 1.836301 | 0.673028 | 1.0 | |

| | 25% | 50% | 75% | max |
|-------------------------------|---------|---------|----------|----------|
| PERIODO | 20194.0 | 20194.0 | 20194.0 | 20194.0 |
| ESTU_COD_RESIDE_DEPTO | 52.0 | 52.0 | 52.0 | 52.0 |
| ESTU_COD_RESIDE_MCPIO | 52001.0 | 52356.0 | 52678.0 | 52885.0 |
| COLE_CODIGO_ICFES | 11932.0 | 70342.0 | 130427.0 | 725218.0 |
| COLE_COD_MCPIO_UBICACION | 52001.0 | 52356.0 | 52678.0 | 86320.0 |
| COLE_COD_DEPTO_UBICACION | 52.0 | 52.0 | 52.0 | 86.0 |
| ESTU_COD_MCPIO_PRESENTACION | 52001.0 | 52356.0 | 52678.0 | 86749.0 |
| ESTU_COD_DEPTO_PRESENTACION | 52.0 | 52.0 | 52.0 | 86.0 |
| PUNT_LECTURA_CRITICA | 44.0 | 51.0 | 59.0 | 100.0 |
| PERCENTIL_LECTURA_CRITICA | 22.0 | 47.0 | 72.0 | 100.0 |
| DESEMPL_LECTURA_CRITICA | 2.0 | 3.0 | 3.0 | 4.0 |
| PUNT_MATEMATICAS | 42.0 | 51.0 | 60.0 | 100.0 |
| PERCENTIL_MATEMATICAS | 25.0 | 50.0 | 76.0 | 100.0 |
| DESEMPL_MATEMATICAS | 2.0 | 3.0 | 3.0 | 4.0 |
| PUNT_C_NATURALES | 40.0 | 48.0 | 57.0 | 100.0 |
| PERCENTIL_C_NATURALES | 24.0 | 52.0 | 77.0 | 100.0 |
| DESEMPL_C_NATURALES | 1.0 | 2.0 | 3.0 | 4.0 |
| PUNT_SOCIALES_CIUDADANAS | 36.0 | 45.0 | 54.0 | 100.0 |
| PERCENTIL_SOCIALES_CIUDADANAS | 24.0 | 48.0 | 73.0 | 100.0 |
| DESEMPL_SOCIALES_CIUDADANAS | 1.0 | 2.0 | 2.0 | 4.0 |
| PUNT_INGLES | 38.0 | 47.0 | 55.0 | 100.0 |
| PERCENTIL_INGLES | 22.0 | 48.0 | 72.0 | 100.0 |
| PUNT_GLOBAL | 205.0 | 242.0 | 282.0 | 452.0 |
| PERCENTIL_GLOBAL | 23.0 | 49.0 | 75.0 | 100.0 |
| ESTU_NSE_INDIVIDUAL | 1.0 | 2.0 | 2.0 | 4.0 |
| ESTU_NSE_ESTABLECIMIENTO | 1.0 | 2.0 | 2.0 | 4.0 |

```
# Se verifica si hay valores faltantes en el conjunto de datos y devuelve un mensaje indicando si existen ("True") o no ("False").
print(f"Hay valores desconocidos: {data.isnull().values.any()}")
```

→ Hay valores desconocidos: True

```
# Se limpia el dataset con el fin de eliminar filas con valores faltantes en variables de objetivo
columnas_interes = [
    'ESTU_GENERO', 'PUNT_GLOBAL', 'PUNT_MATEMATICAS', 'PUNT_LECTURA_CRITICA', 'PUNT_SOCIALES_CIUDADANAS', 'PUNT_C_NATURALES', 'PUNT_INGLES'

data = data.dropna(subset=columnas_interes)
```

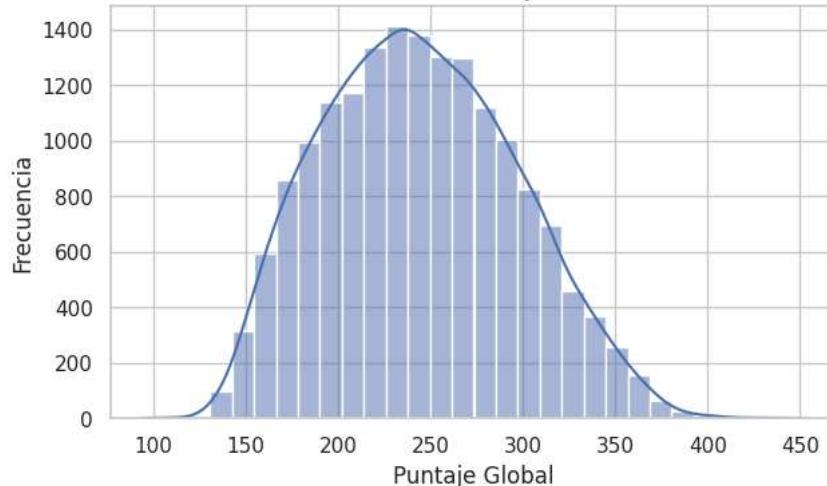
```
# Se describen de forma estadística de las variables objetivo
print(data[columnas_interes].describe().T)
```

| | count | mean | std | min | 25% | 50% | \ |
|--------------------------|---------|------------|-----------|------|-------|-------|---|
| PUNT_GLOBAL | 16903.0 | 244.311306 | 52.061527 | 95.0 | 205.0 | 242.0 | |
| PUNT_MATEMATICAS | 16903.0 | 50.683606 | 12.291657 | 15.0 | 42.0 | 51.0 | |
| PUNT_LECTURA_CRITICA | 16903.0 | 51.222623 | 10.550399 | 0.0 | 44.0 | 51.0 | |
| PUNT_SOCIALES_CIUDADANAS | 16903.0 | 45.617701 | 11.922236 | 17.0 | 36.0 | 45.0 | |
| PUNT_C_NATURALES | 16903.0 | 48.507543 | 11.165361 | 0.0 | 40.0 | 48.0 | |
| PUNT_INGLES | 16903.0 | 47.115837 | 11.969232 | 0.0 | 38.0 | 47.0 | |
| | | 75% | max | | | | |
| PUNT_GLOBAL | | 282.0 | 452.0 | | | | |
| PUNT_MATEMATICAS | | 60.0 | 100.0 | | | | |
| PUNT_LECTURA_CRITICA | | 59.0 | 100.0 | | | | |
| PUNT_SOCIALES_CIUDADANAS | | 54.0 | 100.0 | | | | |
| PUNT_C_NATURALES | | 57.0 | 100.0 | | | | |
| PUNT_INGLES | | 55.0 | 100.0 | | | | |

```
# Histograma de los puntajes globales
plt.figure(figsize=(7, 4))
sns.histplot(data['PUNT_GLOBAL'], bins=30, kde=True)
plt.title('Distribución de Puntajes Globales')
plt.xlabel('Puntaje Global')
plt.ylabel('Frecuencia')
plt.show()
```



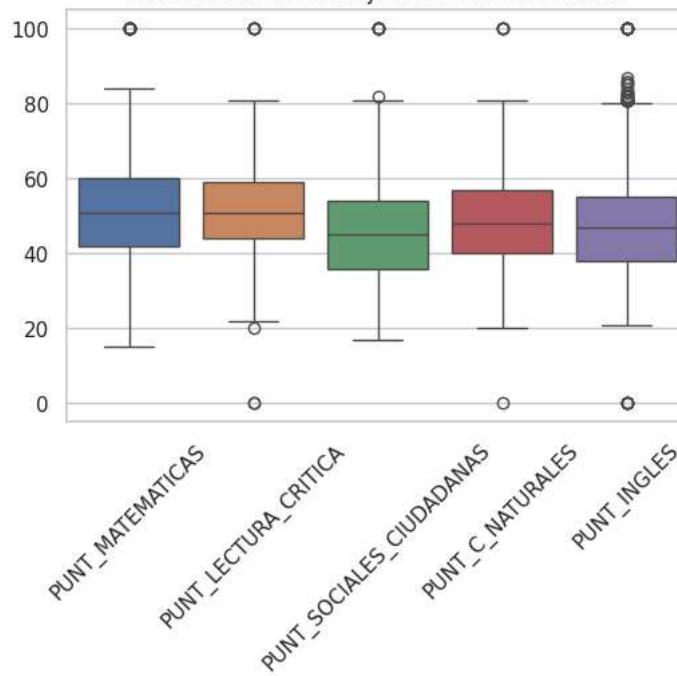
Distribución de Puntajes Globales



```
# Boxplots para los puntajes en las diferentes asignaturas
plt.figure(figsize=(6, 4))
sns.boxplot(data=data[columnas_interes[2:]])
plt.title('Distribución de Puntajes en Diferentes Áreas')
plt.xticks(rotation=45)
plt.show()
```



Distribución de Puntajes en Diferentes Áreas



```
# Se definen las variables objetivo pero excluyendo la variable "PUNT_GLOBAL", para trabajar algunas graficas
columnas_grafica = ['PUNT_MATEMATICAS', 'PUNT_LECTURA_CRITICA', 'PUNT_SOCIALES_CIUDADANAS', 'PUNT_C_NATURALES', 'PUNT_INGLES']

# Se crear una grafica con subplots enfocado en observar la distribucion por municipios
fig, axs = plt.subplots(len(columnas_grafica), 1, figsize=(13, 15), sharex=True)

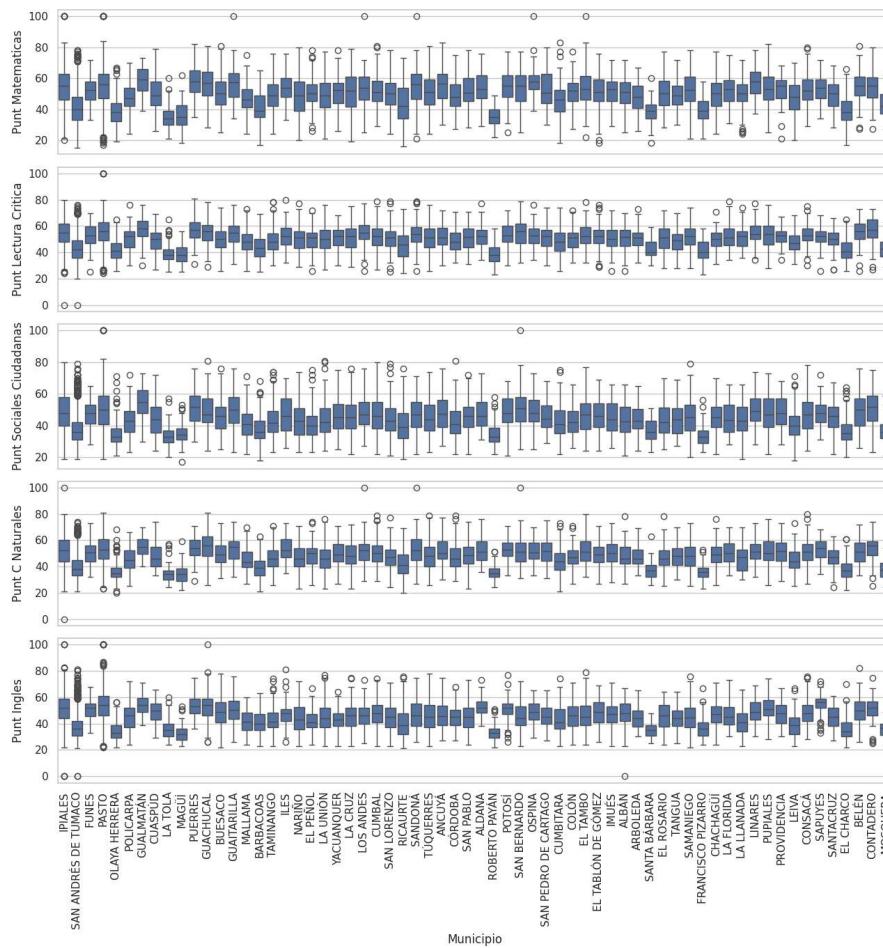
for i, col in enumerate(columnas_grafica):
    sns.boxplot(x='ESTU_MCPIO_RESIDE', y=col, data=data, ax=axs[i])
    axs[i].set_ylabel(col.replace("_", " ").title())
    if i < len(columnas_grafica) - 1:
        axs[i].set_xlabel('')
    axs[i].tick_params(axis='x', rotation=90)

axs[-1].set_xlabel('Municipio')
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
fig.suptitle('Distribución de Puntaje por Municipio', fontsize=16)

plt.show()
```



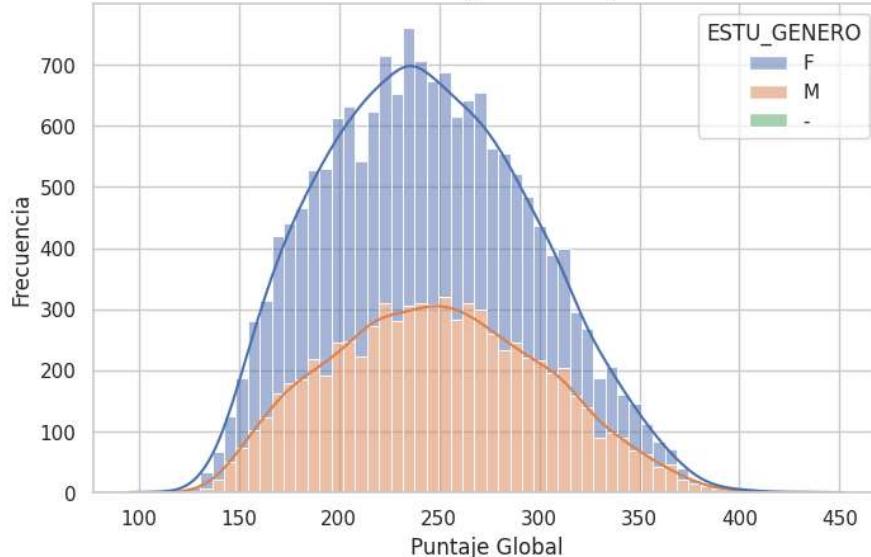
Distribución de Puntaje por Municipio



```
# Gráfico de barras apiladas por género
plt.figure(figsize=(8, 5))
sns.histplot(data=data, x='PUNT_GLOBAL', hue='ESTU_GENERO', multiple='stack', kde=True)
plt.title('Distribución de Puntajes Globales por Género')
plt.xlabel('Puntaje Global')
plt.ylabel('Frecuencia')
plt.show()
```



Distribución de Puntajes Globales por Género

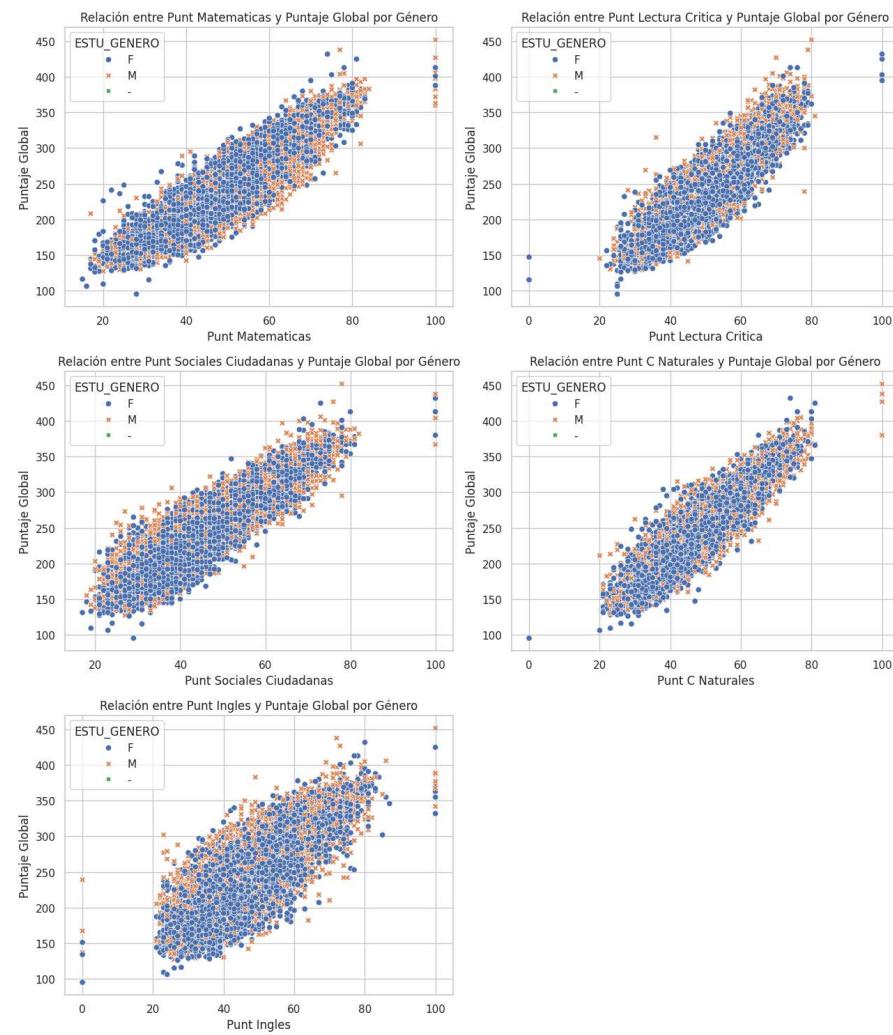


```
# Se crea un grafico de dispepcion para ver las relaciones entre el genero y los puntajes
fig, axs = plt.subplots(3, 2, figsize=(13, 15))
```

```
for i, col in enumerate(columnas_grafica):
    row = i // 2
    col_pos = i % 2
    sns.scatterplot(x=col, y='PUNT_GLOBAL', hue='ESTU_GENERO', style='ESTU_GENERO', data=data, ax=axs[row, col_pos])
    axs[row, col_pos].set_title(f'Relación entre {col.replace("_", " ").title()} y Puntaje Global por Género')
    axs[row, col_pos].set_xlabel(col.replace("_", " ").title())
    axs[row, col_pos].set_ylabel('Puntaje Global')

# importante, aqui elimina graficos vacios
if len(columnas_grafica) % 2 != 0:
    fig.delaxes(axs[-1, -1])

plt.tight_layout()
plt.show()
```



```
# URL del archivo GeoJSON de los límites de los municipios de Nariño en Google Drive
geojson_url = 'https://drive.google.com/uc?id=1rge0WctTRXHAKkdoniWlZWBoH8v0IYn'

# Descargar el archivo GeoJSON
response = requests.get(geojson_url)
geojson_data = BytesIO(response.content)

# Cargar el archivo GeoJSON
gdf = gpd.read_file(geojson_data)

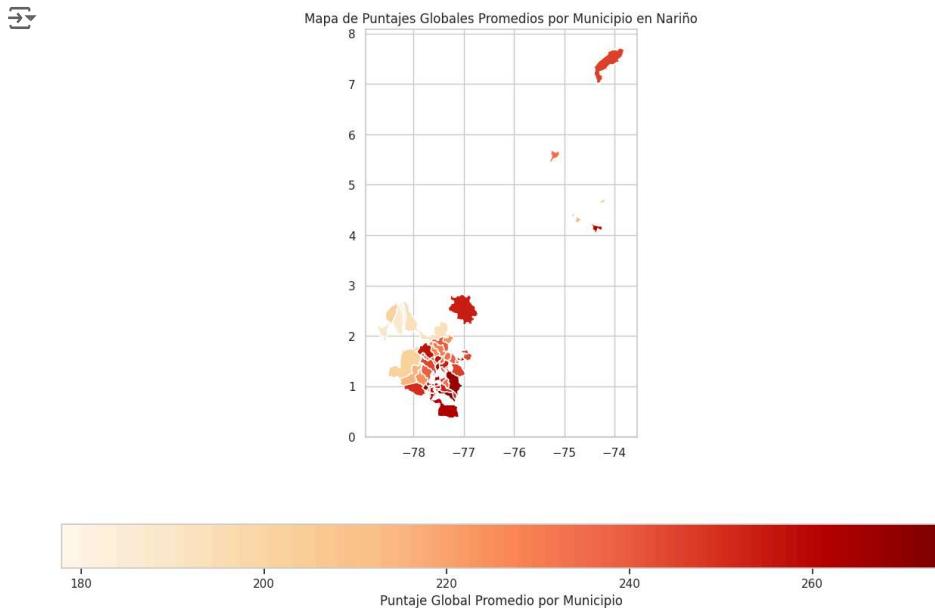
# Asumir que la columna correcta se llama 'MPIO_CNMBR', ajusta esto según corresponda
municipality_column = 'MPIO_CNMBR'

# Agrupar los datos por municipio y calcular el puntaje promedio
municipios = data.groupby('ESTU_MCPIO_RESIDE')[['PUNT_GLOBAL']].mean().reset_index()

# Renombrar las columnas para que coincidan con el GeoDataFrame
municipios.columns = [municipality_column, 'PUNT_GLOBAL']

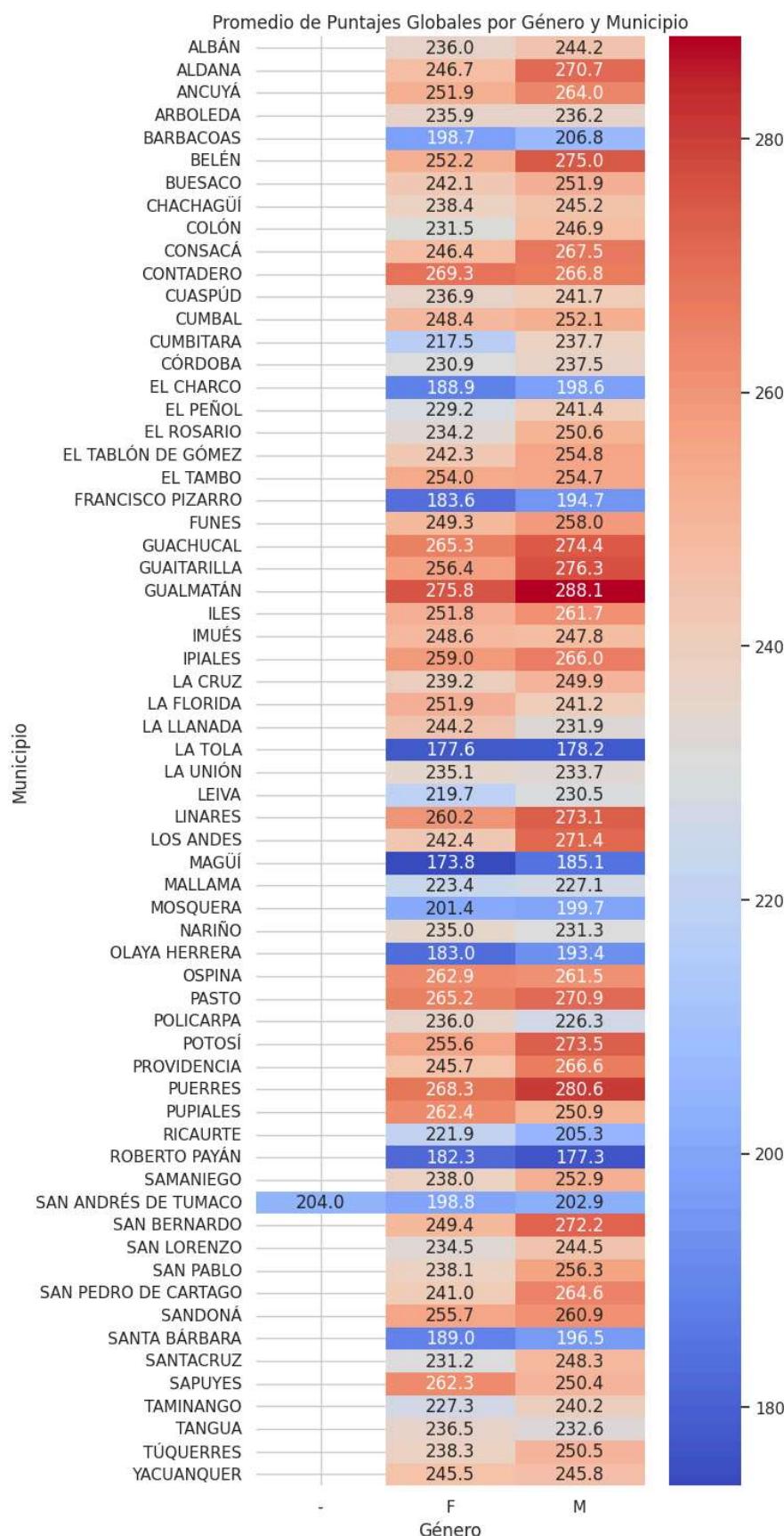
# Unir los datos geográficos con los puntajes promedio
gdf = gdf.merge(municipios, on=municipality_column, how='left')

# Crear el mapa
fig, ax = plt.subplots(1, 1, figsize=(15, 10))
gdf.plot(column='PUNT_GLOBAL', ax=ax, legend=True, cmap='OrRd', legend_kwds={'label': "Puntaje Global Promedio por Municipio", 'orientation': 'vertical'})
plt.title('Mapa de Puntajes Globales Promedios por Municipio en Nariño')
plt.show()
```



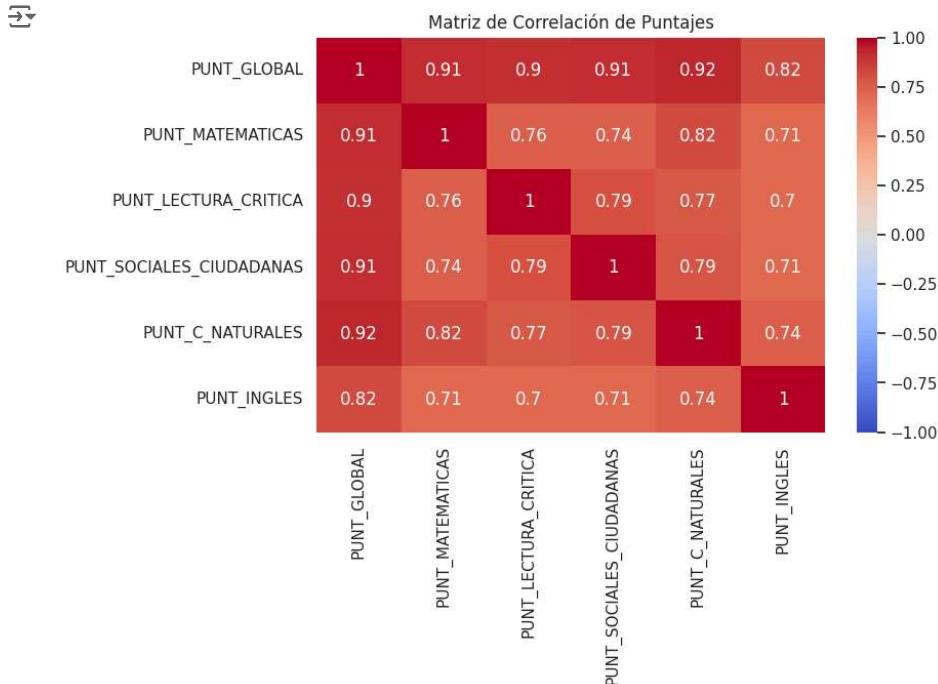
El anterior análisis geoespacial, ayuda a entender las diferencias en el desempeño académico a nivel municipal. Se logra observar que esta información es muy importante para identificar los municipios que necesitan mejorar y se tendrá que diseñar planes de mejora continua con el fin de mejorar los resultados de estas pruebas.

```
# Se crea la tabla de calor para observar los puntajes globales por género y municipio
pivot_table = data.pivot_table(values='PUNT_GLOBAL', index='ESTU_MCPIO_RESIDE', columns='ESTU_GENERO', aggfunc='mean')
plt.figure(figsize=(6, 18))
sns.heatmap(pivot_table, annot=True, fmt=".1f", cmap="coolwarm")
plt.title('Promedio de Puntajes Globales por Género y Municipio')
plt.xlabel('Género')
plt.ylabel('Municipio')
plt.show()
```



```
# Aquí se seleccionan sólo las variables objetivo que sean numéricas para la matriz de correlación
columnas_numericas = ['PUNT_GLOBAL', 'PUNT_MATEMATICAS', 'PUNT_LECTURA_CRITICA', 'PUNT_SOCIALES_CIUDADANAS', 'PUNT_C_NATURALES', 'PUNT_INGLES']

# Matriz de correlación
plt.figure(figsize=(8, 5))
correlation_matrix = data[columnas_numericas].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Matriz de Correlación de Puntajes')
plt.show()
```



Las altas correlaciones entre los puntajes de cada área y el puntaje global muestran que el desempeño en cada área contribuye significativamente al puntaje total, esta información es vital y con ella se puede identificar aspectos clave de mejora y para diseñar estrategias educativas más integrales y dentro de este análisis, nos ayuda a identificar que existe una correlación positiva fuerte ya que si los puntajes de cada área aumentan, los puntajes globales también tienden a aumentar significativamente.

▼ Entrenamiento de modelos de regresión

En este proyecto voy a utilizar dos métodos diferentes en el análisis con el fin de mejorar la precisión y validez de las predicciones que se van a realizar al final, también usar 2 modelos de estudio genera un análisis más completo y detallado de los datos. En general esto permite hacer recomendaciones más válidas de acuerdo a los datos, que son la evidencia de este proyecto.

Al final la conclusión será que la evaluación de los dos métodos ayudará a la toma de decisiones después del análisis y que estas sirvan para mejorar el desempeño estudiantil de las pruebas en un futuro.

```
# Se seleccionan las variables independientes y dependientes
X = data[['PUNT_MATEMATICAS', 'PUNT_LECTURA_CRITICA', 'PUNT_SOCIALES_CIUDADANAS', 'PUNT_C_NATURALES', 'PUNT_INGLES']]
y = data['PUNT_GLOBAL']
```

```
# Se dividen los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
```

```
# Se entrena el modelo de REGRESION LINEAL
model = LinearRegression()
model.fit(X_train, y_train)
```

↳ ▾ `LinearRegression`
`LinearRegression()`

```
# Se realizan las predicciones con el conjunto de prueba
y_pred_rl = model.predict(X_test)
```

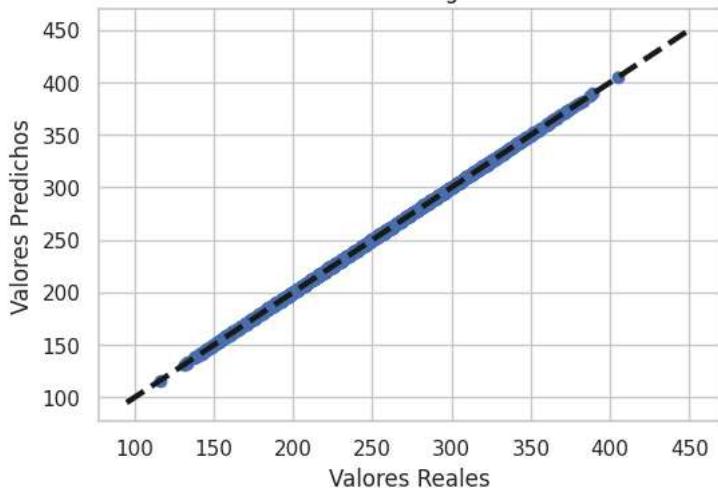
```
# Se evaluar el modelo
mse_rl = mean_squared_error(y_test, y_pred_rl)
r2_rl = r2_score(y_test, y_pred_rl)
```

```
# Se imprimen los resultados evaluados
print(f"Mean Squared Error: {mse_rl}")
print(f"R-squared: {r2_rl}")
```

↳ Mean Squared Error: 0.08660050616047052
R-squared: 0.9999676459464607

```
# Se visualizan los resultados
plt.figure(figsize=(6, 4))
plt.scatter(y_test, y_pred_rl)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=3)
plt.xlabel('Valores Reales')
plt.ylabel('Valores Predichos')
plt.title('Real vs Predicho: Regresión Lineal')
plt.show()
```

↳ **Real vs Predicho: Regresión Lineal**



```
# Se entrena un modelo de RANDOM FOREST
model_rf = RandomForestRegressor(random_state=42)
model_rf.fit(X_train, y_train)
```

↳ ▾ `RandomForestRegressor`
`RandomForestRegressor(random_state=42)`

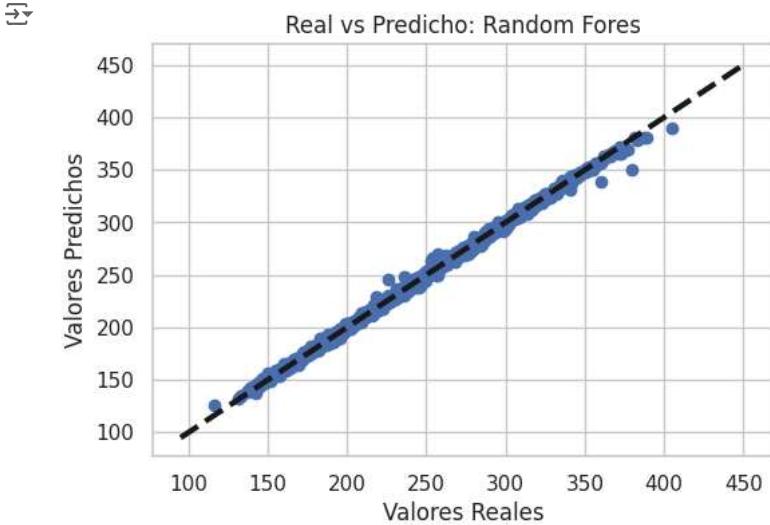
```
# Se realizan las predicciones con el conjunto de prueba
y_pred_rf = model_rf.predict(X_test)

# Se evalua el modelo
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)

# Se imprimen los resultados evaluados
print("\nModelo de Bosques Aleatorios:")
print(f"Mean Squared Error: {mse_rf}")
print(f"R-squared: {r2_rf}")
```

→
Modelo de Bosques Aleatorios:
Mean Squared Error: 4.986242578356004
R-squared: 0.9981371337594596

```
# Se vizualizan los resultados
plt.figure(figsize=(6, 4))
plt.scatter(y_test, y_pred_rf)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=3)
plt.xlabel('Valores Reales')
plt.ylabel('Valores Predichos')
plt.title('Real vs Predicho: Random Fores')
plt.show()
```



```
# Se crea un DataFrame con los resultados de ambos modelos para compararlos
resultados = pd.DataFrame({
    'Modelo': ['Regresión Lineal', 'Random Forest'],
    'Mean Squared Error': [mse_rl, mse_rf],
    'R-squared': [r2_rl, r2_rf]
})

# Se muestran en una tabla
print("Comparación de Resultados de los Modelos:")
print(tabulate(resultados, headers='keys', tablefmt='pretty', showindex=False))
```

→ Comparación de Resultados de los Modelos:

| Modelo | Mean Squared Error | R-squared |
|------------------|---------------------|--------------------|
| Regresión Lineal | 0.08660050616047052 | 0.9999676459464607 |
| Random Forest | 4.986242578356004 | 0.9981371337594596 |

Hallazgos y Conclusiones:

- Existe una gran variedad en los datos de puntajes obtenidos por los estudiantes en cada área. En algunos casos hay una diferencia significativa por géneros entre el puntaje promedio total y sus áreas de estudio, en otras el puntaje de una área supera al otro.
- En algunos municipios se presentan diferencias muy grandes en sus resultados y muchos de estos municipios tienen características socioeconómicas diferentes, lo que muestra que los factores socioeconómicos influyen directamente en los estudiantes.

- Al predecir el puntaje global de los estudiantes enfocado en sus resultados, se ajustan los modelos de regresión: Regresión lineal y Random Forest. Los dos modelos se ajustan muy bien a los datos, pero cuando se trata de predicción, Random Forest están ligeramente adelantados.

Recomendaciones:

- Estos resultados podrían utilizarse para identificar áreas de mejora en el sistema educativo de Nariño y con ellos diseñar planes específicos dirigidos a mejorar el desempeño estudiantil.
- Estos hallazgos pueden mostrar la necesidad de un análisis adicional con el fin de explorar la relación entre el desempeño estudiantil y variables como el nivel socioeconómico, el tipo de institución educativa y el acceso a recursos educativos.
- Después de este proyecto sería muy importante hacer un seguimiento a largo plazo de los estudiantes para evaluar el efecto de cualquier plan de mejora.

▼ Espacio para realizar predicciones

```
# Aquí se realiza una predicción con nuevos datos

# Se deberá agregar datos como puntaje de cada asignatura para que el modelo pueda mostrar el promedio total
nuevo_estudiante = pd.DataFrame({
    'PUNT_MATEMATICAS': [20],
    'PUNT_LECTURA_CRITICA': [60],
    'PUNT_SOCIALES_CIUDADANAS': [90],
    'PUNT_C_NATURALES': [100],
    'PUNT_INGLES': [20]
})

# Predicción con el modelo de Regresión Lineal
prediccion_rl = model.predict(nuevo_estudiante)
print(f"Predicción del Puntaje Global para el nuevo estudiante (Regresión Lineal): {prediccion_rl[0]}")

# Predicción con el modelo de Random Forest
prediccion_rf = model_rf.predict(nuevo_estudiante)
print(f"Predicción del Puntaje Global para el nuevo estudiante (Random Forest) : {prediccion_rf[0]}")

⤵ Predicción del Puntaje Global para el nuevo estudiante (Regresión Lineal): 319.2126794138088
⤵ Predicción del Puntaje Global para el nuevo estudiante (Random Forest) : 344.31
```

▼ Sección de SQL

En un proyecto de análisis de datos, el lenguaje SQL es muy importante como herramienta de análisis para extraer información valiosa de conjuntos de datos grandes.

Con este tipo de consultas SQL se pueden aplicar análisis desde la exploración y limpieza de datos hasta la generación de consultas particulares que generan conocimientos del dataset, como patrones dentro de los datos o tendencias de los mismos.

```
from sqlalchemy import create_engine

# Se crea una base de datos SQLite en memoria
db_memoria = create_engine('sqlite:///PruebaSaber11.db')

# Se guarda el df en la base de datos antes creada
data.to_sql('PruebaSaber11', con=db_memoria, index=False, if_exists='replace')

⤵ 16903
```

```
# SQL para consultar municipios con mayor promedio general de mayor a menor

municipios = """
SELECT ESTU_MCPIO_RESIDE, AVG(PUNT_GLOBAL) as AVG_PUNT_GLOBAL
FROM PruebaSaber11
GROUP BY ESTU_MCPIO_RESIDE
ORDER BY AVG_PUNT_GLOBAL DESC
"""

municipios_promedio = pd.read_sql(municipios, con=db_memoria)
print("Municipios con mayor promedio general:")
print(municipios_promedio.head(10))

# Definir una paleta de colores personalizada
custom_palette = sns.color_palette("pastel")

# Generar la gráfica de barras con el parámetro y asignando la paleta de colores personalizada
plt.figure(figsize=(4, 3))
sns.barplot(x='AVG_PUNT_GLOBAL', y='ESTU_MCPIO_RESIDE', data=municipios_promedio.head(10), hue='ESTU_MCPIO_RESIDE', palette=custom_palette,
            plt.title('Municipios con Mayor Promedio General de Puntaje')
            plt.xlabel('Promedio Puntaje Global')
            plt.ylabel('Municipio')
            plt.show()
```

↳ Municipios con mayor promedio general:

| | ESTU_MCPIO_RESIDE | AVG_PUNT_GLOBAL |
|---|-------------------|-----------------|
| 0 | GUALMATÁN | 281.661972 |
| 1 | PUERRES | 274.038835 |
| 2 | GUACHUCAL | 269.813278 |
| 3 | CONTADERO | 268.157143 |
| 4 | PASTO | 267.817403 |
| 5 | GUAITARILLA | 266.198276 |
| 6 | LINARES | 265.306931 |
| 7 | POTOSÍ | 262.311927 |
| 8 | IPIALES | 262.293318 |
| 9 | OSPIÑA | 262.263889 |



```
# SQL para calcular el puntaje global promedio por género
```

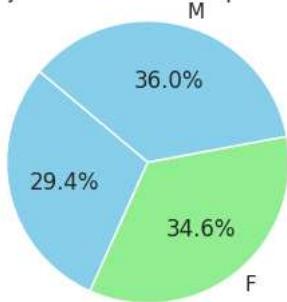
```
puntaje_genero = """
SELECT ESTU_GENERO, AVG(PUNT_GLOBAL) as AVG_PUNT_GLOBAL
FROM PruebaSaber11
GROUP BY ESTU_GENERO
"""

puntaje_genero = pd.read_sql(puntaje_genero, con=db_memoria)

# Datos para el gráfico de pastel
labels = puntaje_genero['ESTU_GENERO']
sizes = puntaje_genero['AVG_PUNT_GLOBAL']
colors = ['skyblue', 'lightgreen']

# Crear el gráfico de pastel
plt.figure(figsize=(3, 3))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.title('Puntaje Global Promedio por Género')
plt.axis('equal')
plt.show()
```

→ Puntaje Global Promedio por Género



```
# SQL para consultar los mejores 10 colegios de mayor a menor
```

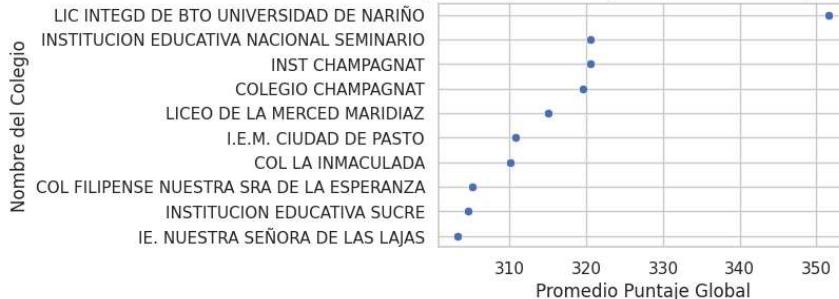
```
mejores_colegios = """
SELECT COLE_NOMBRE_ESTABLECIMIENTO, ESTU_MCPPIO_RESIDE AS MUNICIPIO, AVG(PUNT_GLOBAL) as AVG_PUNT_GLOBAL
FROM PruebaSaber11
GROUP BY COLE_NOMBRE_ESTABLECIMIENTO
ORDER BY AVG_PUNT_GLOBAL DESC
LIMIT 10
"""

mejores_colegios = pd.read_sql(mejores_colegios, con=db_memoria)

# Crear la gráfica de puntos para los 10 mejores colegios
plt.figure(figsize=(5, 3))
sns.scatterplot(x='AVG_PUNT_GLOBAL', y='COLE_NOMBRE_ESTABLECIMIENTO', data=mejores_colegios)
plt.title('Top 10 Colegios con Mayor Promedio de Puntaje Global')
plt.xlabel('Promedio Puntaje Global')
plt.ylabel('Nombre del Colegio')
plt.show()
```

→

Top 10 Colegios con Mayor Promedio de Puntaje Global



```

colegios_pasto = """
SELECT COLE_NOMBRE_ESTABLECIMIENTO , AVG(PUNT_GLOBAL) as AVG_PUNT_GLOBAL
FROM PruebaSaber11
WHERE ESTU_MPIO_RESIDE = 'PASTO'
GROUP BY COLE_NOMBRE_ESTABLECIMIENTO
ORDER BY AVG_PUNT_GLOBAL DESC
LIMIT 10
"""

colegios_promedio = pd.read_sql(colegios_pasto, con=db_memoria)
print("Colegios en Pasto con mayor promedio general:")
print(colegios_promedio)

# Datos para el diagrama de pastel
labels = colegios_promedio['COLE_NOMBRE_ESTABLECIMIENTO']
sizes = colegios_promedio['AVG_PUNT_GLOBAL']

# Colores personalizados
# Definir una paleta de colores personalizada
colors = sns.color_palette("pastel")

# Crear el diagrama de pastel
plt.figure(figsize=(4, 5))
plt.pie(sizes, labels=labels, colors=colors, autopct='%.1f%%', startangle=140)
plt.title('Colegios en Pasto con Mayor Promedio General')
plt.axis('equal')
plt.show()

```

☞ Colegios en Pasto con mayor promedio general:

| | COLE_NOMBRE_ESTABLECIMIENTO | AVG_PUNT_GLOBAL |
|---|---|-----------------|
| 0 | LIC INTEGD DE BTO UNIVERSIDAD DE NARIÑO | 351.727273 |
| 1 | INST CHAMPAGNAT | 320.541985 |
| 2 | LICEO DE LA MERCED MARIDIAZ | 315.145161 |
| 3 | I.E.M. CIUDAD DE PASTO | 310.756000 |
| 4 | COL LA INMACULADA | 310.100000 |
| 5 | COL FILIPENSE NUESTRA SRA DE LA ESPERANZA | 305.115385 |
| 6 | CENTRO EDUCATIVO CONOCIMIENTO Y SABIDURIA | 304.000000 |
| 7 | COL SAN FELIPE NERI | 297.180851 |
| 8 | I.E.M. NORMAL SUPERIOR DE PASTO | 295.757009 |
| 9 | INSTITUCION EDUCATIVA LAS MESAS INSEM | 291.000000 |

Colegios en Pasto con Mayor Promedio General



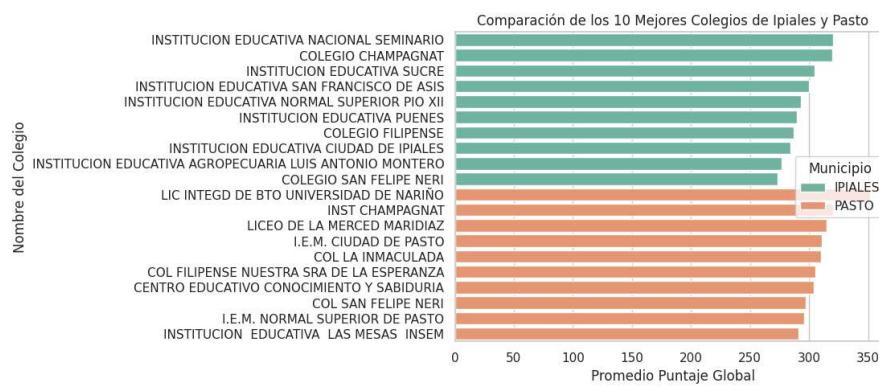
```
# Consulta SQL para comparar los 10 mejores colegios de Ipiales y Pasto
mejores_colegios_ipiales_pasto = """
SELECT COLE_NOMBRE_ESTABLECIMIENTO, ESTU_MCPIO_RESIDE AS MUNICIPIO, AVG(PUNT_GLOBAL) as AVG_PUNT_GLOBAL
FROM PruebaSaber11
WHERE ESTU_MCPIO_RESIDE IN ('IPIALES', 'PASTO')
GROUP BY ESTU_MCPIO_RESIDE, COLE_NOMBRE_ESTABLECIMIENTO
ORDER BY ESTU_MCPIO_RESIDE, AVG_PUNT_GLOBAL DESC
"""

mejores_colegios = pd.read_sql(mejores_colegios_ipiales_pasto, con=db_memoria)

# Mostrar los 10 mejores colegios de Ipiales y los 10 mejores de Pasto
top_10_ipiales = mejores_colegios[mejores_colegios['MUNICIPIO'] == 'IPIALES'].head(10)
top_10_pasto = mejores_colegios[mejores_colegios['MUNICIPIO'] == 'PASTO'].head(10)

# Concatenar los datos de los 10 mejores colegios de Ipiales y Pasto
top_10 = pd.concat([top_10_ipiales, top_10_pasto])

# Crear la gráfica de barras agrupadas
plt.figure(figsize=(7, 5))
sns.barplot(x='AVG_PUNT_GLOBAL', y='COLE_NOMBRE_ESTABLECIMIENTO', hue='MUNICIPIO', data=top_10, palette='Set2')
plt.title('Comparación de los 10 Mejores Colegios de Ipiales y Pasto')
plt.xlabel('Promedio Puntaje Global')
plt.ylabel('Nombre del Colegio')
plt.legend(title='Municipio')
plt.show()
```



```
# SQL para calcular el Impacto del Estrato Socioeconómico

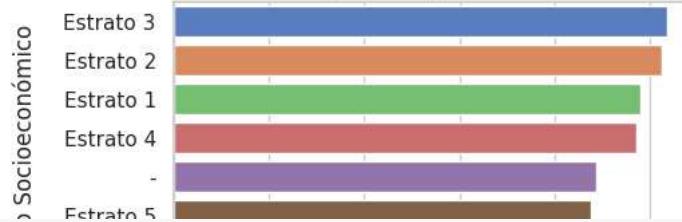
estrato_sql = """
SELECT FAMI_ESTRATOVIVIENDA, AVG(PUNT_GLOBAL) as AVG_PUNT_GLOBAL
FROM PruebaSaber11
GROUP BY FAMI_ESTRATOVIVIENDA
ORDER BY AVG_PUNT_GLOBAL DESC
"""

estrato_promedio = pd.read_sql(estrato_sql, con=db_memoria)

plt.figure(figsize=(5, 3))
sns.barplot(x='AVG_PUNT_GLOBAL', y='FAMI_ESTRATOVIVIENDA', hue='FAMI_ESTRATOVIVIENDA', data=estrato_promedio, palette='muted', dodge=False,
plt.title('Promedio de Puntaje Global por Estrato Socioeconómico')
plt.xlabel('Promedio Puntaje Global')
plt.ylabel('Estrato Socioeconómico')
plt.show()
```



Promedio de Puntaje Global por Estrato Socioeconómico



SQL para calcular Impacto de la Dedicación Diaria a la Lectura

```
lectura_sql = """
SELECT ESTUDEDICACIONLECTURADIARIA, AVG(PUNT_GLOBAL) as AVG_PUNT_GLOBAL
FROM PruebaSaber11
GROUP BY ESTUDEDICACIONLECTURADIARIA
ORDER BY AVG_PUNT_GLOBAL DESC
```