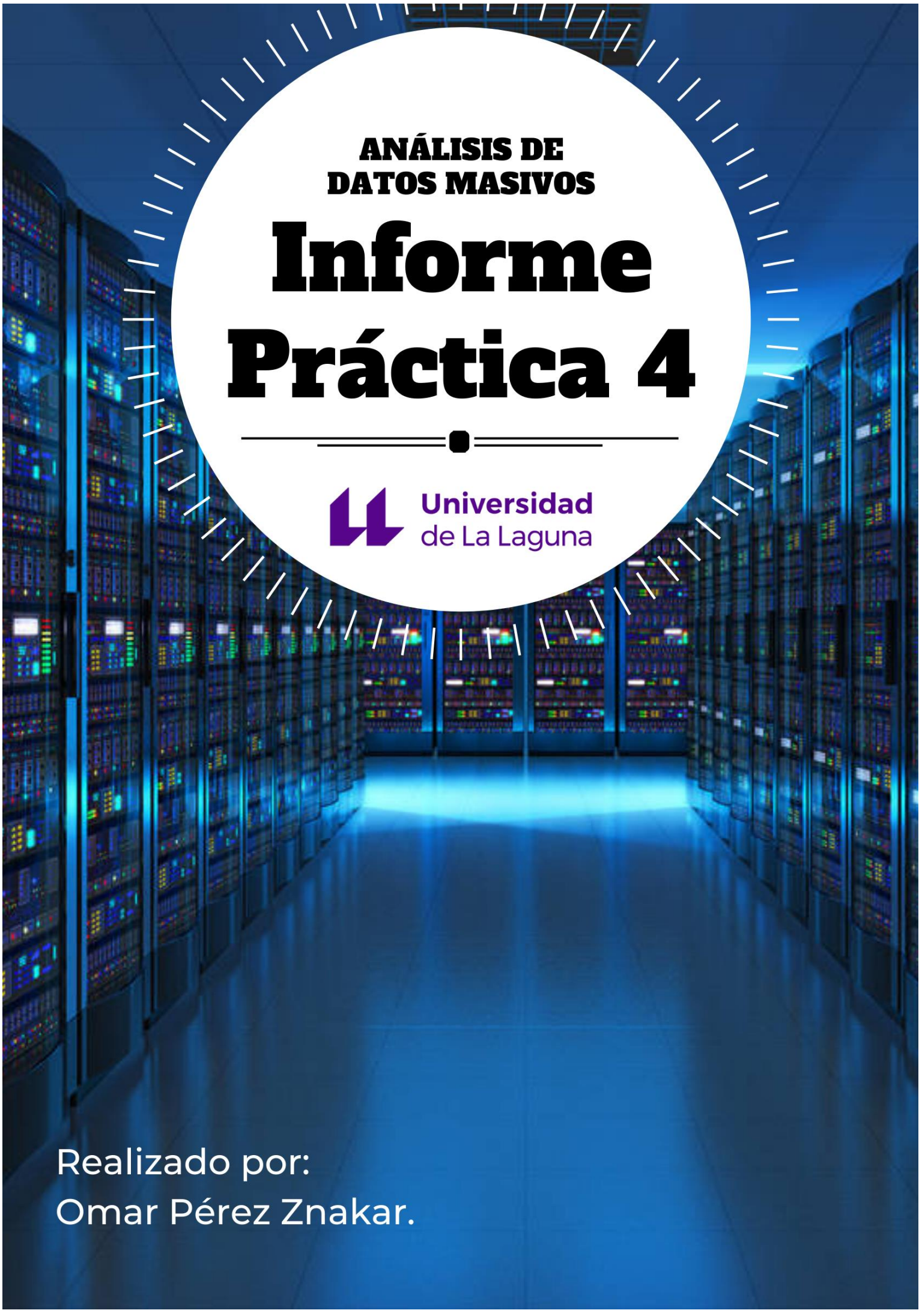


ANÁLISIS DE DATOS MASIVOS

Informe Práctica 4

Universidad de La Laguna

Realizado por:
Omar Pérez Znakar.

ÍNDICE

1.- Representación de datos utilizando los mapas geográficos. 3
1.1 Explicación. 3
1.2 Ejemplo desarrollado. 5
2.- Algoritmo de aprendizaje. 6
2.1 Descripción estructura. **6**
2.2 Aprendizajes supervisados. 7
2.2.1 Clasificación. 7
2.2.2 Regresión. 8
2.3 Aprendizaje no supervisado basado en Clustering. 10
2.4 Subida proyecto a CodeCloud. 11

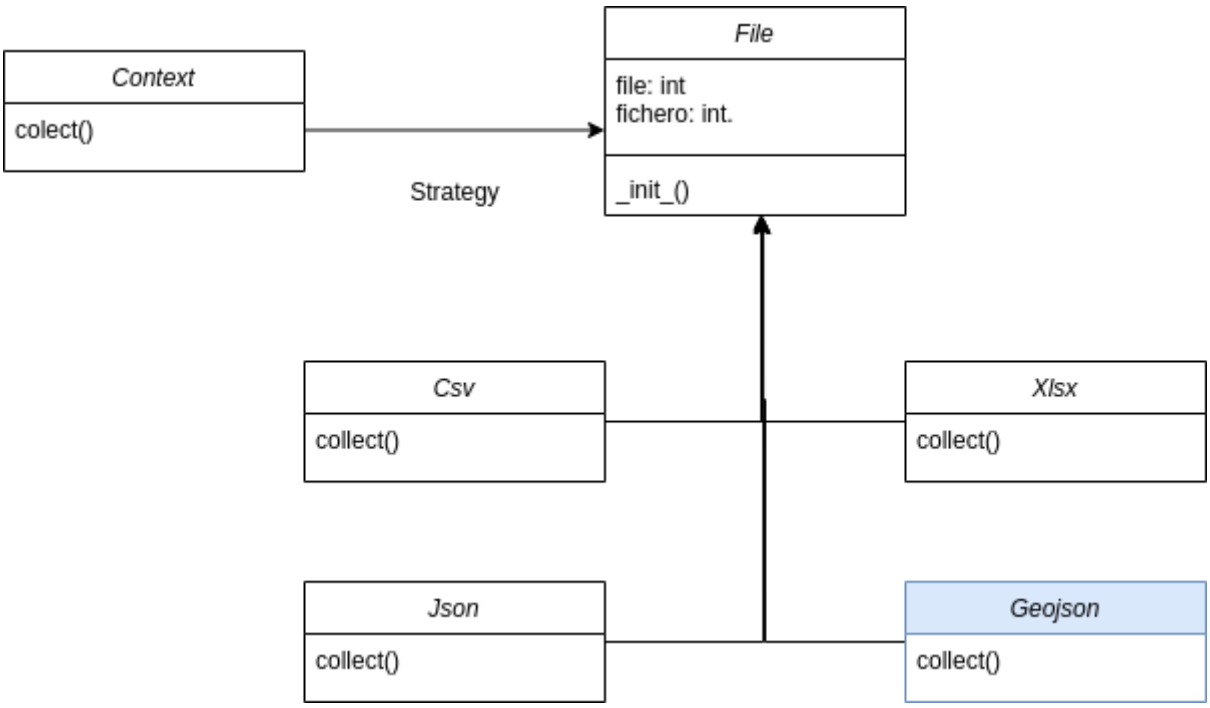
1.- Representación de datos utilizando los mapas geográficos.

1.1 Explicación.

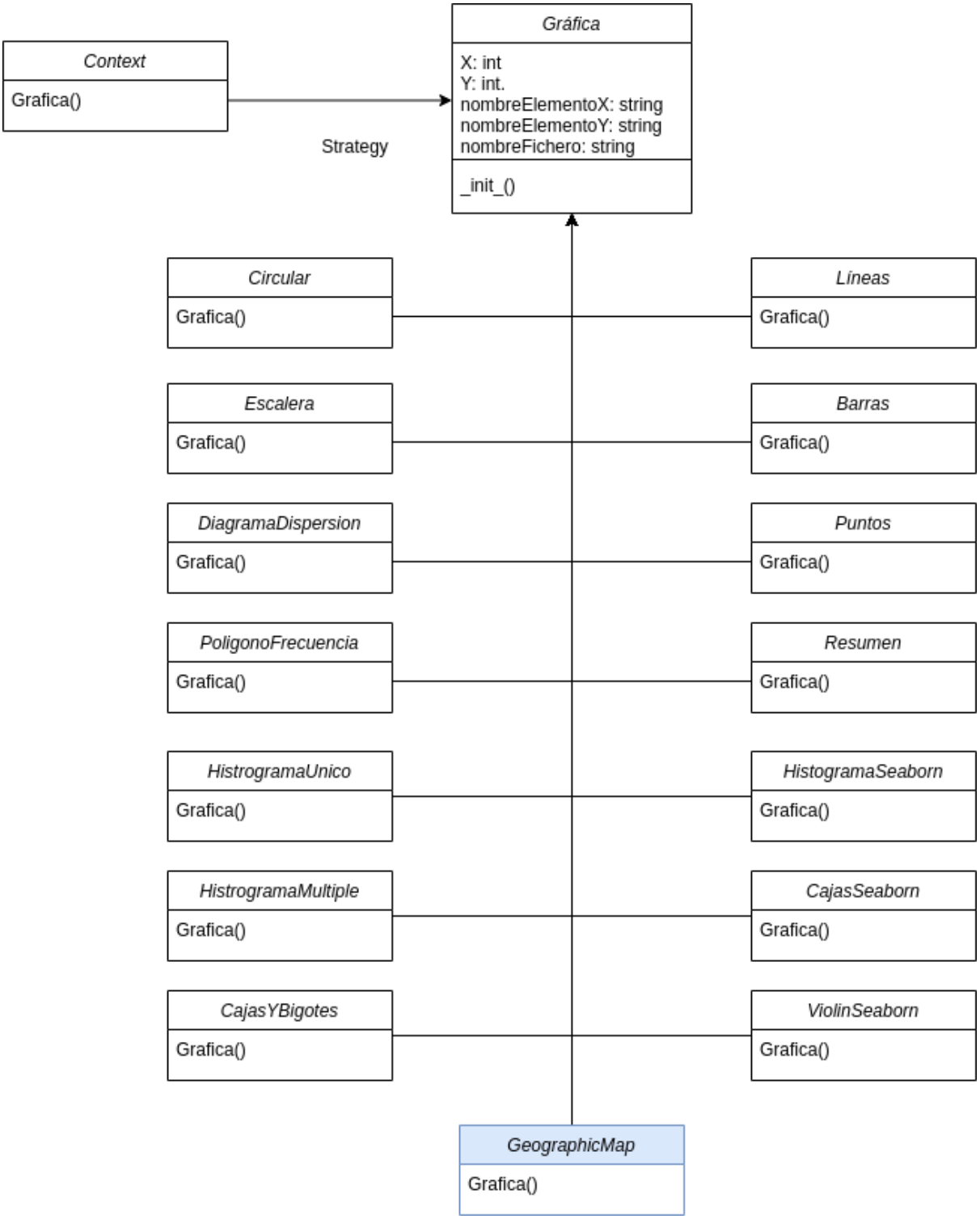
Para este apartado, se ha usado el tutorial que podremos ver en el siguiente enlace "http://www.geomapik.com/desarrollo-programacion-gis/mapas-con-python-geopandas-matplotlib/". En el nos explica como generar un mapa geográfico usando Python junto con las librerías denominadas "geopandas" y "matplotlib".

En relación a las capas de abstracción, se han añadido dos elementos (uno a cada capa). Estos son el propio diagrama de mapas geográficos y, poder leer un nuevo tipo de dato (".geoson"). Estas representaciones y sus modificaciones (marcadas en azul) las podremos ver a continuación:

Abstracción Tipos de Ficheros

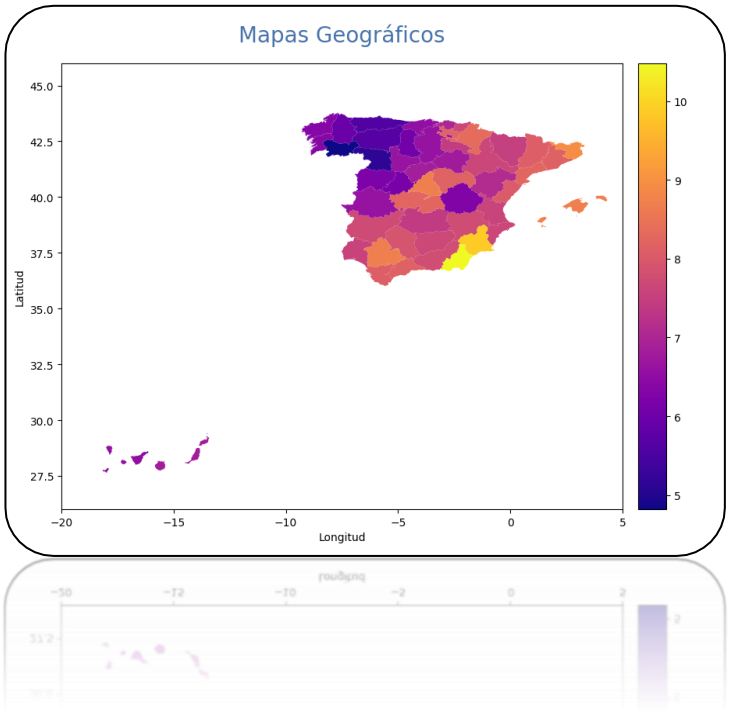


Abstracción Gráficas

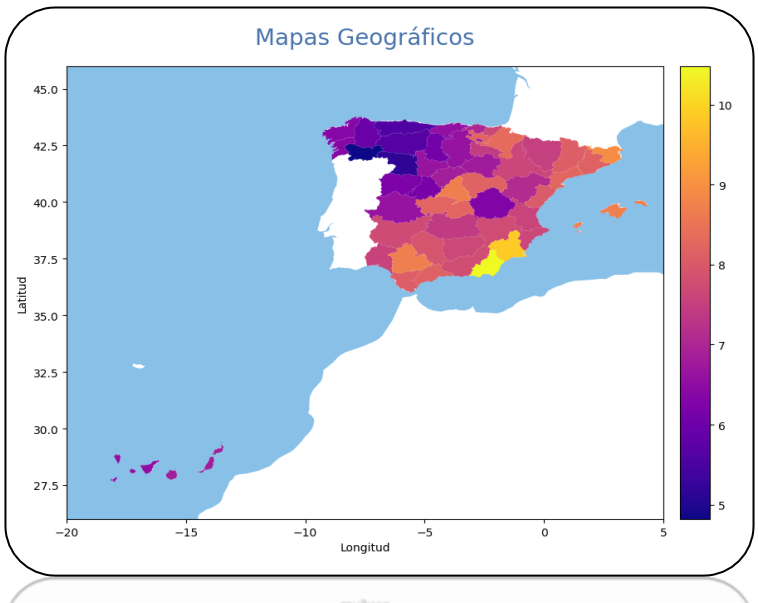


1.2 Ejemplo desarrollado.

En cuanto a los datos, se han usado los datos referentes a la natalidad española en el año 2018, categorizada por provincias. Posteriormente, se ha probado el código desarrollado obteniendo la siguiente salida:



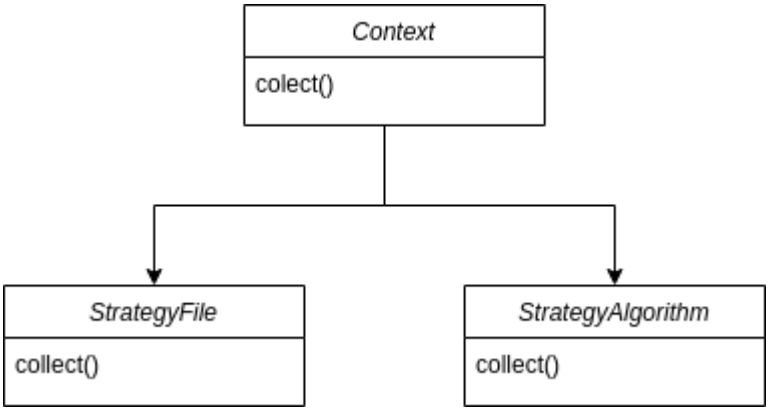
Por último, se ha modificado el código añadiendo una base para cargar los contornos de los países y el mar. La salida de esta modificación la podremos ver a continuación:



2.- Algoritmo de aprendizaje.

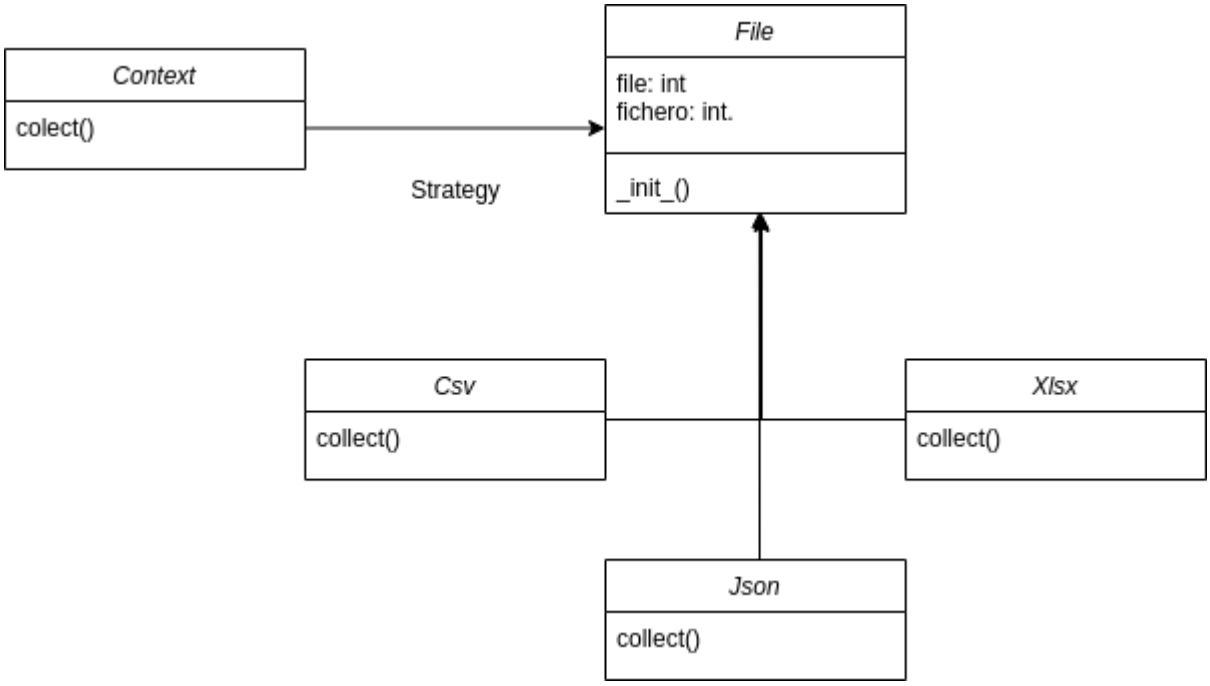
2.1 Descripción estructura.

Como el profesor nos ha ido introduciendo poco a poco, para este ejercicio se ha intentado realizar diferentes capas de abstracción para simplificar el entendimiento de este. Por tanto, la estructura del código realizado sería:

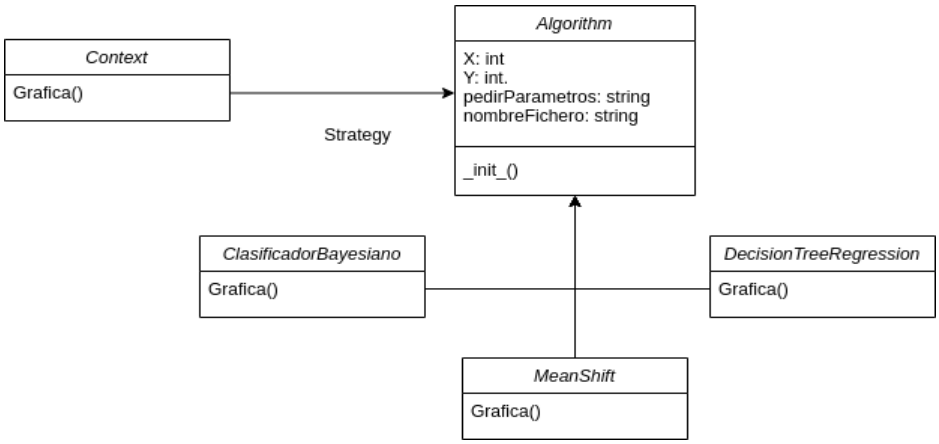


En donde:

- **Context:** es el programa principal.
- **StrategyFile:** es la clase que nos permitirá poder entender diferentes tipos de documentos. Esta sigue la siguiente estructura:



- **StrategyAlgorithm:** es la clase que nos permitirá realizar el algoritmo seleccionado con anterioridad. Esta sigue la siguiente estructura:

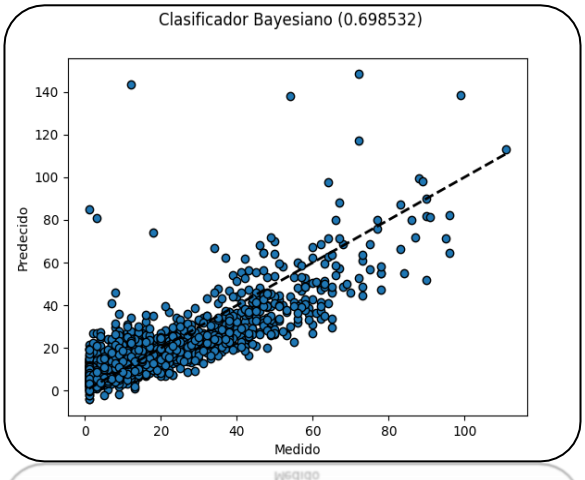


2.2 Aprendizajes supervisados.

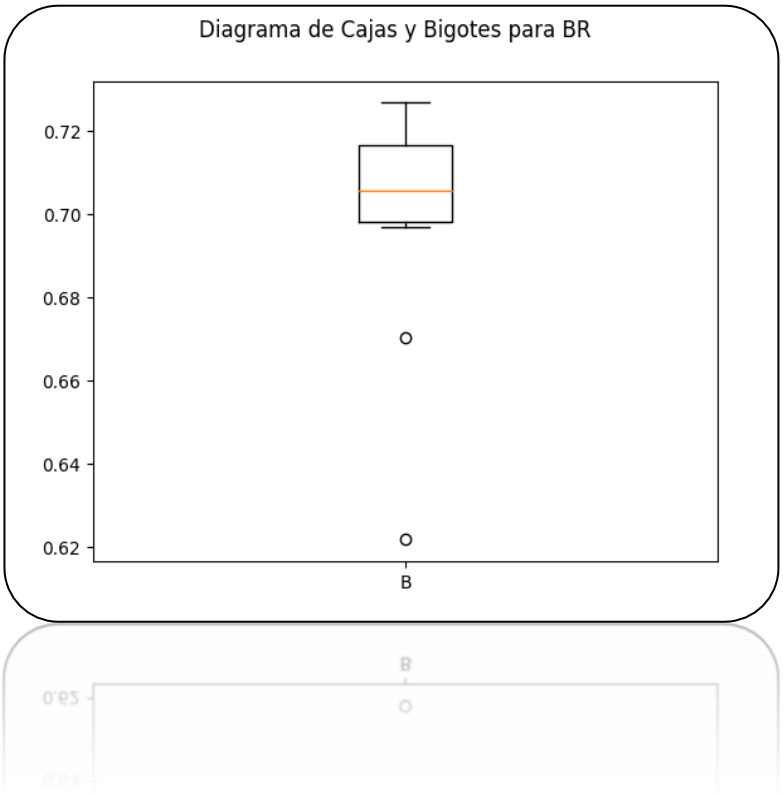
2.2.1 Clasificación.

Para el algoritmo de aprendizaje supervisado usando clasificación se ha optado por el algoritmo denominado “Bayesiano”. De igual forma, podremos seleccionar la columna para la que se quiere hacer la clasificación. Esta consta de diferentes partes que podremos ver a continuación:

- Se divide el vector en dos: uno para entrenar y el otro para para saber el porcentaje de acierto.
- Se realizar un estudio usando las librerías denominadas “sklearn” y “pandas”.
- Mostramos por pantalla el estudio calculado. Un ejemplo podría ser usar el csv de Tome Cano e intentar clasificar el valor de NO2. La salida obtenida para este caso sería:



- Realizamos un diagrama de cajas y bigotes para que nos dé información sobre el porcentaje de acierto que nos brinda este algoritmo. Para el ejemplo visto en el apartado anterior la salida sería:

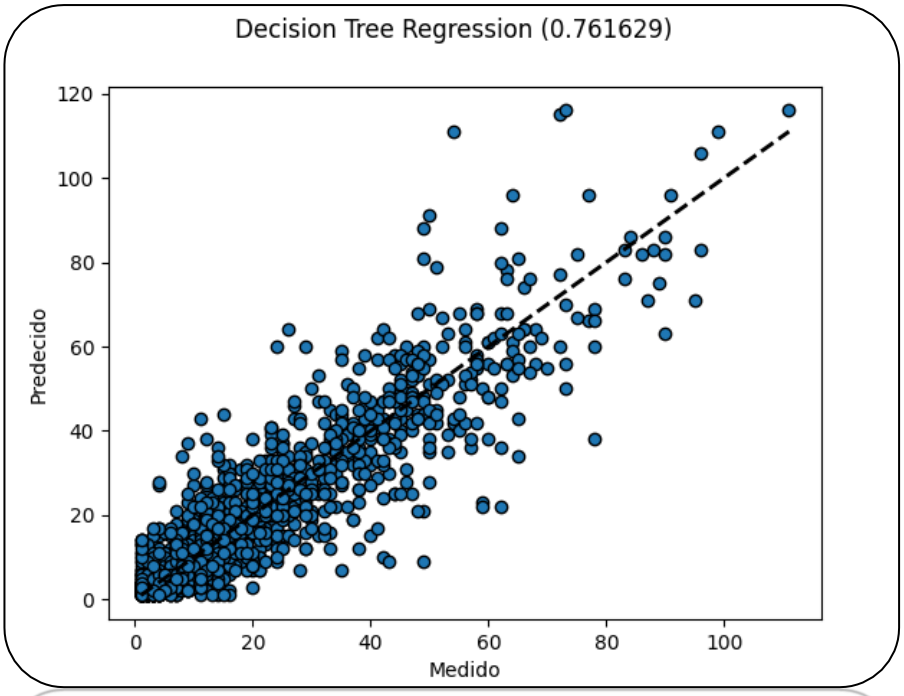


Como podremos ver, este algoritmo nos brinda un porcentaje (para este caso particular) de un 69,85% de aciertos. Por tanto, lo ideal para este tipo de casos es realizar un estudio y ver cual es el algoritmo que mejor se asemeja a nuestra problemática.

2.2.2 Regresión.

Para este apartado, se ha optado por utilizar el algoritmo denominado “Decision Tree Regression” (o en español “Regresión del árbol de decisiones”). Con este podremos seleccionar el conjunto de columnas a usar. De igual forma, se ha seguido el siguiente planteamiento:

- Se divide el vector en dos: uno para entrenar y el otro para para saber el porcentaje de acierto.
- Se realizar un estudio (usando este algoritmo) usando las librerías denominadas “sklearn” y “pandas”.
- Mostramos por pantalla el estudio calculado. Un ejemplo podría ser usar el csv de Tome Cano. La salida obtenida para este caso sería:


Decision Tree Regression (0.761629)
Predecido
Medido
0
20
40
60
80
100
120

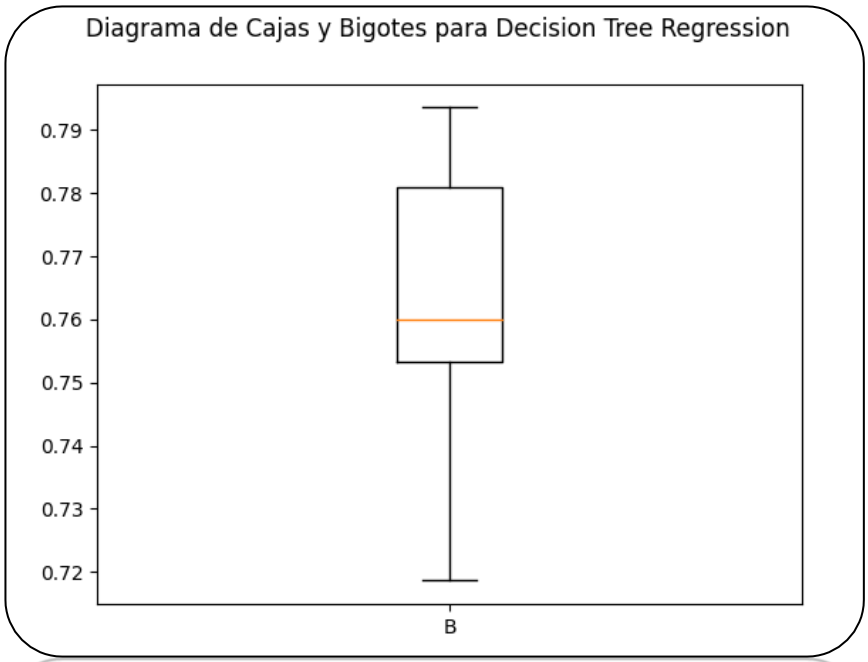
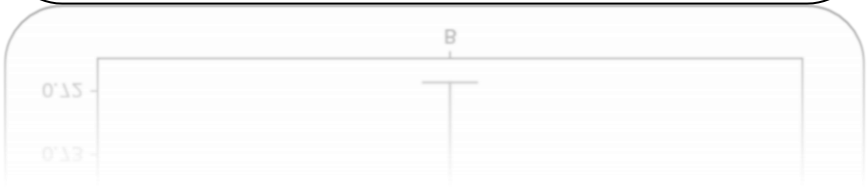


Diagrama de Cajas y Bigotes para Decision Tree Regression
0.79
0.78
0.77
0.76
0.75
0.74
0.73
0.72
B

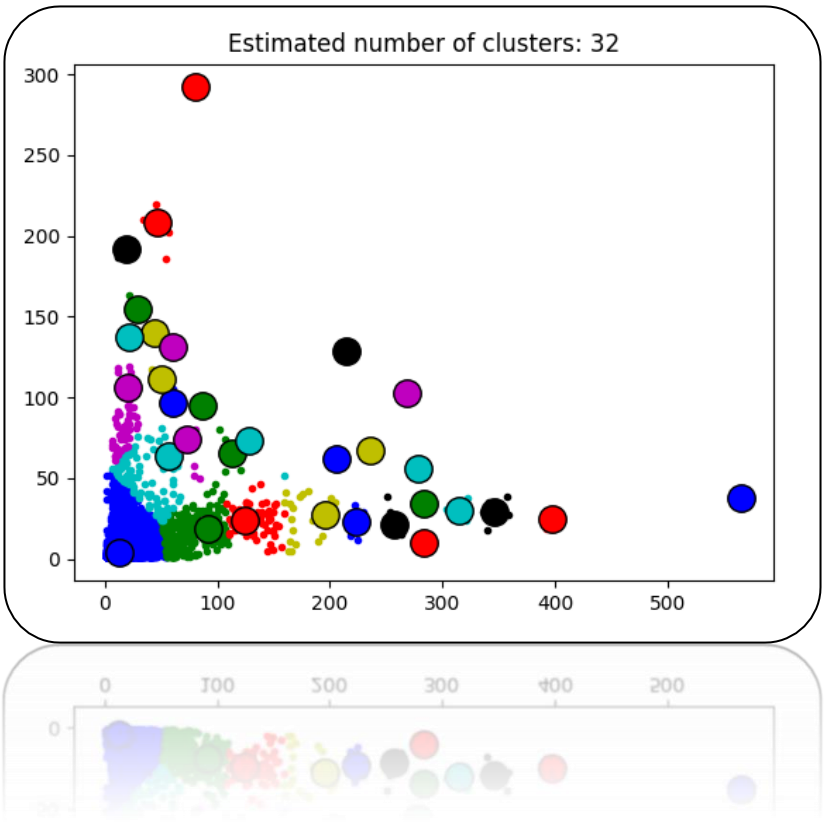


Como podemos ver en este apartado, este algoritmo nos aporta un 76% de aciertos. Por tanto, lo lógico sería usar diversos algoritmos y, elegir el que mejor se adapte a nuestra problemática.

2.3 Aprendizaje no supervisado basado en Clustering.

Para este apartado, se ha usado el algoritmo denominado “Mean Shift”. Este algoritmo me ha parecido interesante debido a que nos permite estimar el número de clusters. De igual forma, aparte de esto se ha representado el resultado obtenido usando las librerías: matplotlib y pandas.

Para comprobar el funcionamiento de este algoritmo, se ha usado el csv de Tomé Cano. El resultado obtenido ha sido:



Como podemos ver en la imagen anterior, se estima un total de 32 clúster para estos datos. De igual forma, cabe resaltar que los colores repetidos son clústeres diferentes.

2.4 Subida proyecto a CodeCloud.

Para este apartado, se ha creado el archivo ".json" de configuración para el código desarrollado. Este lo podremos ver a continuación:

```
"Name":"MachineLearning",
"file": "TomeCano.csv",
"Description": "Método que permite aplicar funciones de Machine Learning determinadas.",
"Elements": [
  {
      "Name":"tipoGrafica",
      "value":"2",
      "DescriptionShort":"Campo que nos permite elegir la gráfica deseada.",
      "DescriptionLong":"Valores entre 1-3 ()."
  },
  {
      "Name":"columnaSeleccionadaInicial",
      "value":"4",
      "DescriptionShort":"Campo que nos permite elegir la columna inicial deseada.",
      "DescriptionLong":"Valor número empezando por 0 de la columna desea para nuestra representación gráfica."
  },
  {
      "Name":"columnaSeleccionada",
      "value":"6",
      "DescriptionShort":"Campo que nos permite elegir la columna final deseada.",
      "DescriptionLong":"Valor número empezando por 0 de la columna desea para nuestra representación gráfica."
  }
]
```

Posteriormente, se ha subido dicho archivo y un comprimido con el código realizado. Por último, se ha comprobado el funcionamiento. Una prueba de ello la podremos ver a continuación: