

# MÁSTER EN DATA SCIENCE

CURSO 2023-2024

MÓDULO: Tecnología y herramientas Big Data

PROFESOR: Abel González Durán

ALUMNO: Yonatan Eleuterio Rubio



## Análisis de Datos de Participación Electoral en España

### Introducción

Este informe documenta el proceso de análisis de datos sobre la participación electoral en las elecciones españolas, a través del estudio de los votos emitidos en diferentes municipios y comunidades autónomas. Utilizando un conjunto de datos que contiene información de participación electoral, partidos políticos y resultados por mesa de votación, se realizaron diversas consultas y comparaciones para evaluar cómo se comportan los diferentes municipios según su tamaño (grandes y pequeños) y la influencia de factores como la provincia, comunidad autónoma y la participación.

El análisis fue realizado en un entorno de procesamiento de datos con el uso de DataFrames en PySpark, lo que permitió gestionar grandes volúmenes de información y realizar cálculos eficientes, incluyendo agregaciones, comparaciones y la visualización de tendencias.

### Planteamiento

El objetivo principal del análisis fue responder a varias preguntas relacionadas con la participación electoral y los partidos políticos ganadores en las elecciones. Para ello, los datos fueron cargados y procesados en DataFrames utilizando PySpark, lo que permitió aprovechar el paralelismo para manejar grandes volúmenes de información.

El trabajo se estructuró en las siguientes fases:

1. **Carga y Preprocesamiento de Datos:** Los conjuntos de datos fueron cargados en DataFrames que se juntaron a partir de los datasets brindados PECelecciones.csv y PECMunicipios.csv. Se limpiaron para asegurar que los datos estuvieran listos para el análisis. Esto incluyó la verificación de columnas, la conversión de tipos de datos y la creación de nuevas columnas necesarias para el análisis (por ejemplo, el cálculo de la participación por municipio o comunidad).
2. **Cálculo de Participación Electoral:** La participación electoral fue calculada como el porcentaje de votos emitidos respecto al total de votantes registrados, tanto para

municipios grandes como pequeños. Para los municipios grandes, se extrajeron los 20 municipios con mayor población y, para los municipios pequeños, los 20 con menos de 10,000 habitantes.

3. **Análisis Comparativo:** Se realizaron comparaciones entre los municipios grandes y pequeños para evaluar cómo la participación afecta los resultados de los partidos. Se observaron las variaciones en los votos por partido, en la participación y cómo se distribuyen los votos en función del tamaño de los municipios.
4. **Resultados y Justificación de las Respuestas a las Preguntas:** Las preguntas clave se respondieron utilizando los siguientes enfoques:

¿Qué partido tiene más participación en los municipios grandes vs. pequeños?

¿Cómo se distribuyen los votos en función de la participación?

¿Cuáles son los municipios con mayor diferencia de participación por comunidad?

¿La provincia o comunidad autónoma tiene un impacto significativo en el resultado?

## Ejecución y Resultados

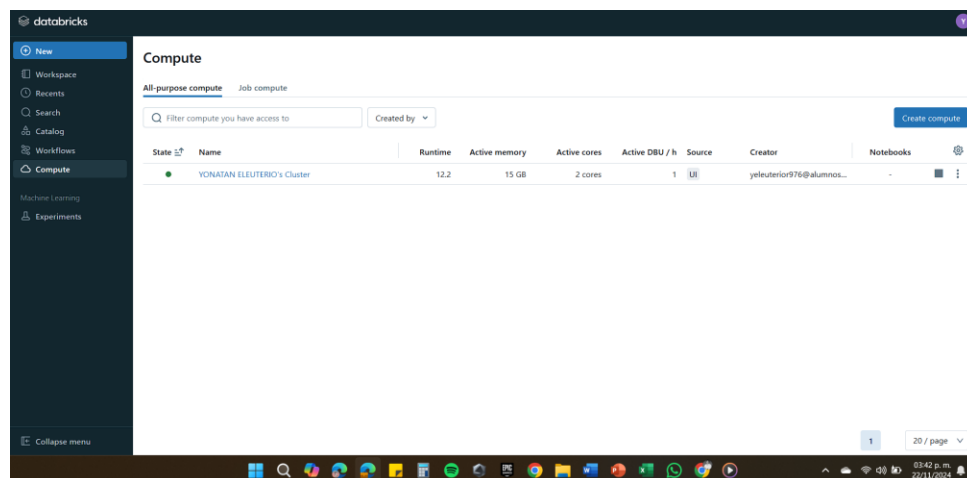
### 1. Razonar y justificar qué herramienta utilizaremos en cada paso y para qué

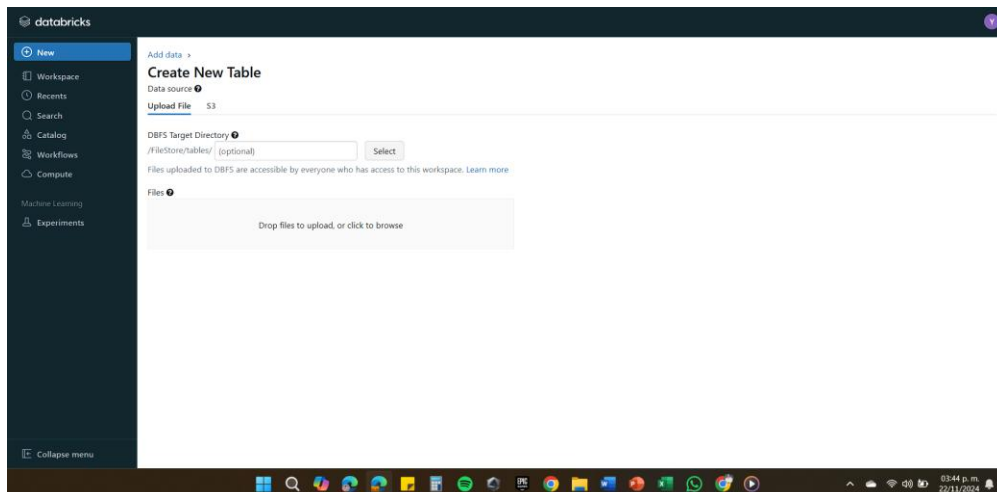
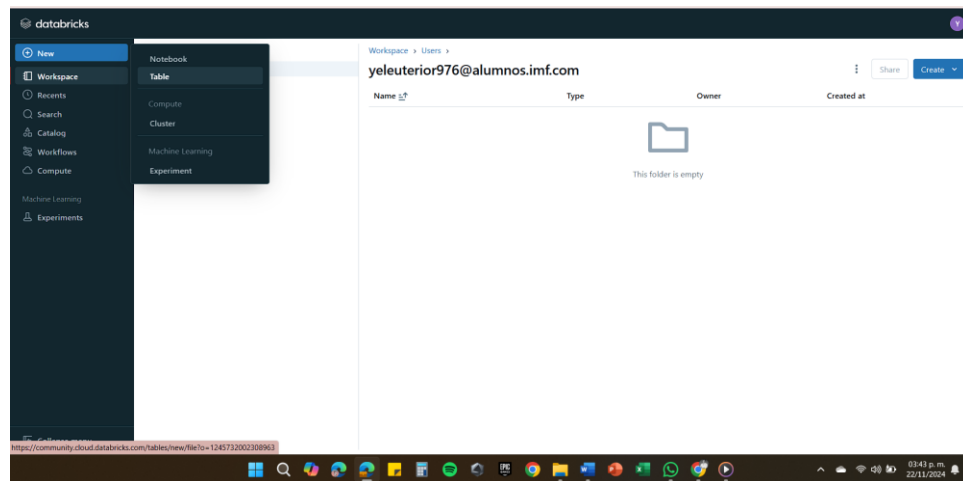
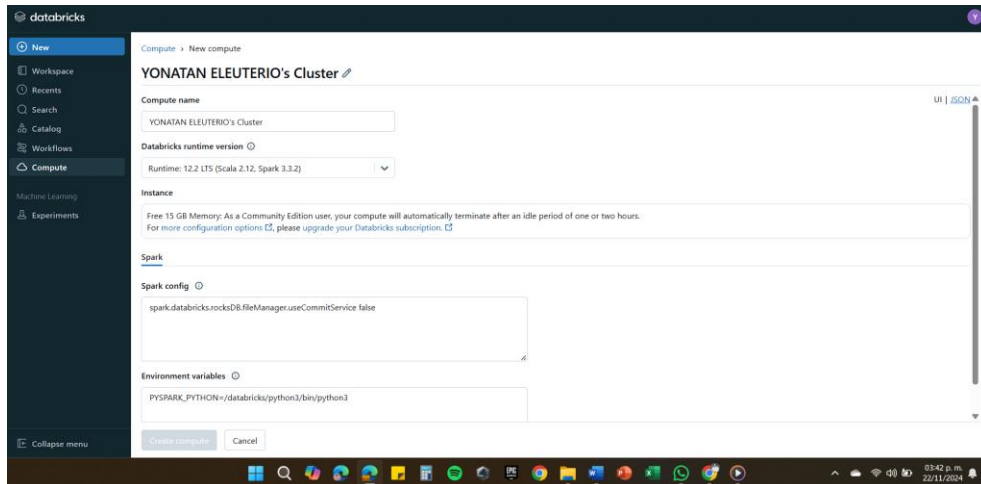
Usaremos Databricks porque nos olvidamos de usar una máquina virtual y tenemos todas las funciones SQL, lenguaje python, spark, etc. que nos ayudan a manipular la información.

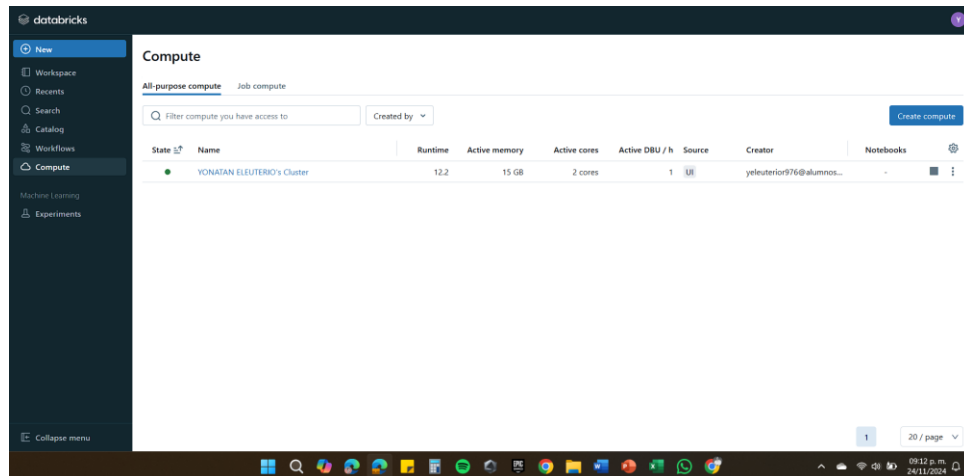
### 2. Cargar los diferentes ficheros en las herramientas seleccionadas y sacar un listado de sus contenidos por pantalla o a fichero.

Para esto en Databricks tenemos que generar un clúster donde podamos procesar la información.

## Creación de Clúster



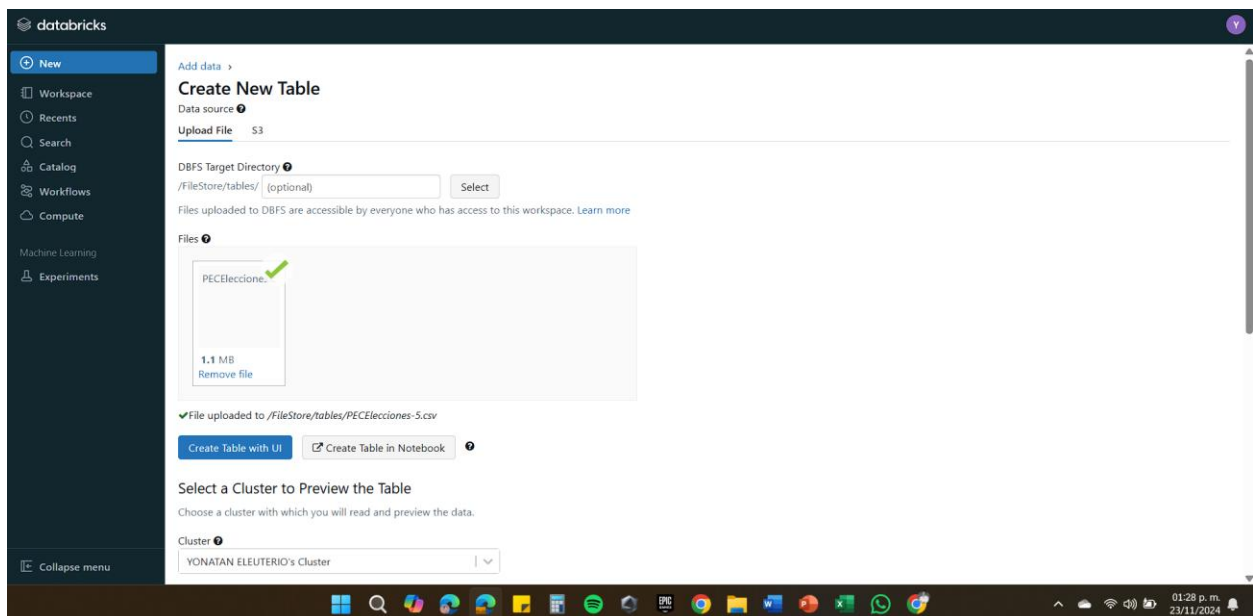




Una vez creado y activo empezaremos a cargar las tablas

## Carga de CSVs

Cargamos una tabla donde laa podamos definir para que nos cree un código para poder cargarla con código en el Notebook y poder manipular la información fácilmente como se muestra abajo



New

Workspace

Recents

Search

Catalog

Workflows

Compute

Machine Learning

Experiments

2024-11-22 - PEC Big Data Yonatan Eleuterio

Python

Run all

YONATAN ELEUTERIO's...

Share

Publish

File Edit View Run Help

Last edit was 7 minutes ago

Overview

This notebook will show you how to create and query a table or DataFrame that you uploaded to DBFS. [DBFS](#) is a Databricks File System that allows you to store data for querying inside of Databricks. This notebook assumes that you have a file already inside of DBFS that you would like to read from.

This notebook is written in **Python** so the default cell type is Python. However, you can use different languages by using the `%LANGUAGE` syntax. Python, Scala, SQL, and R are all supported.

4 minutes ago (22s)

2

Python

# File location and type  
file\_location = "/FileStore/tables/PECElecciones.csv"  
file\_type = "csv"  
  
# CSV options  
infer\_schema = "true"  
first\_row\_is\_header = "true"  
delimiter = ";"  
  
# The applied options are for CSV files. For other file types, these will be ignored.  
df = spark.read.format(file\_type) \\\n .option("inferSchema", infer\_schema) \\\n .option("header", first\_row\_is\_header) \\\n .option("sep", delimiter) \\\n

Choose a cluster with which you will read and preview the data.

Cluster  
YONATAN ELEUTERIO's Cluster

Preview Table

Specify Table Attributes

Specify the Table Name, Database and Schema to add this to the data UI for other users to access

Table Name  
pecelecciones\_csv

Create in Database  
default

File Type  
CSV

Column Delimiter  
;

☒ First row is header

☒ Infer schema

☐ Multi-line

Create Table

Create Table in Notebook

Table Preview

Codigo	Mesas	Censo	Votantes	Validos	Blanco
INT	INT	INT	INT	INT	INT
4001	2	1062	823	814	5
4002	2	1039	748	740	2
4003	26	17357	11157	11075	64
4004	1	425	329	329	0
4005	1	556	437	431	3
4006	13	7022	4792	4749	30

New

Workspace

Recents

Search

Catalog

Workflows

Compute

Machine Learning

Experiments

2024-11-22 - PEC Big Data Yonatan Eleuterio

Python

Run all

YONATAN ELEUTERIO's...

Share

Publish

File Edit View Run Help

Last edit was 7 minutes ago

Overview

This notebook will show you how to create and query a table or DataFrame that you uploaded to DBFS. [DBFS](#) is a Databricks File System that allows you to store data for querying inside of Databricks. This notebook assumes that you have a file already inside of DBFS that you would like to read from.

This notebook is written in **Python** so the default cell type is Python. However, you can use different languages by using the `%LANGUAGE` syntax. Python, Scala, SQL, and R are all supported.

4 minutes ago (22s)

2

Python

# File location and type  
file\_location = "/FileStore/tables/PECElecciones.csv"  
file\_type = "csv"  
  
# CSV options  
infer\_schema = "true"  
first\_row\_is\_header = "true"  
delimiter = ";"  
  
# The applied options are for CSV files. For other file types, these will be ignored.  
df = spark.read.format(file\_type) \\\n .option("inferSchema", infer\_schema) \\\n .option("header", first\_row\_is\_header) \\\n .option("sep", delimiter) \\\n

2024-11-22 - PEC Big Data Yonatan Eleuterio Python

File Edit View Run Help Last edit was 8 minutes ago

df: pyspark.sql.dataframe.DataFrame = [Codigo: integer, Mesas: integer ... 56 more fields]

	Codigo	Mesas	Censo	Votantes	Validos	Blanco	Nulos	PP	PSOE	POC
25	4026	1	66	54	51	0	3	36	7	
26	4027	1	250	159	159	0	0	82	37	
27	4028	1	221	191	190	1	1	44	114	
28	4029	14	8859	5834	5790	35	44	2195	2239	
29	4030	2	1119	819	812	5	7	319	354	
30	4031	4	2338	1772	1762	2	10	589	880	
31	4032	11	5586	3642	3623	23	19	1272	1302	
32	4033	1	135	97	93	1	4	25	52	
33	4034	1	160	99	99	2	0	56	20	
34	4035	16	8364	5810	5766	28	44	2328	2248	
35	4036	1	226	175	175	0	0	86	70	
36	4037	3	1314	969	955	6	14	365	383	
37	4038	5	3084	2140	2126	16	14	750	754	
38	4041	1	322	240	238	6	2	96	64	

8,125 rows | 21.55 seconds runtime Refreshed 4 minutes ago

2024-11-22 - PEC Big Data Yonatan Eleuterio Python

File Edit View Run Help Last edit was 10 minutes ago

```
# Create a view on table

temp_table_name = "PECElecciones_csv"

df.createOrReplaceTempView(temp_table_name)
```

```
/* Query the created temp table in a SQL cell */

select * from "PECElecciones_csv"
```

(1) Spark Jobs

\_sqlid: pyspark.sql.dataframe.DataFrame = [Codigo: integer, Mesas: integer ... 56 more fields]

	Codigo	Mesas	Censo	Votantes	Validos	Blanco	Nulos	PP	PSOE	POC
1	4001	2	1062	823	814	5	9	267	356	
2	4002	2	1039	748	740	2	8	212	342	

## Unión de las tablas

Una vez cargadas las tablas las unimos y quitamos el código para tener la base de datos limpia y fácil de utilizar son caracteres especiales y que se pueda visualizar de manera correcta

PEC Big Data Yonatan Eleuterio (Python)

Import Notebook

15

```

%sql

/* Query the created temp table in a SQL cell */

select * from elecciones_join

```

Table

	i3 codigo	A3 comunidad	A3 provincia	A3 municipio	i3 poblacion	i3 mesas	i3 censo	i3 votantes	i3 v...
1	4001	Andalucia	Almeria	Abia	1342	2	1062	823	
2	4002	Andalucia	Almeria	Abrucena	1279	2	1039	748	
3	4003	Andalucia	Almeria	Adria	24670	26	17357	11157	
4	4004	Andalucia	Almeria	Albanchez	805	1	425	329	
5	4005	Andalucia	Almeria	Alboloduy	653	1	556	437	
6	4006	Andalucia	Almeria	Albox	11429	13	7022	4792	
7	4007	Andalucia	Almeria	Alcolea	812	1	734	524	
8	4008	Andalucia	Almeria	Alcontar	570	2	505	423	
9	4009	Andalucia	Almeria	Alcudia de Monteagud	168	1	162	123	
10	4010	Andalucia	Almeria	Alhabia	700	1	576	444	
11	4011	Andalucia	Almeria	Alhama de Almeria	3763	4	2765	2119	
12	4012	Andalucia	Almeria	Alicun	220	1	191	161	
13	4013	Andalucia	Almeria	Almeria	194203	212	141380	94872	
14	4014	Andalucia	Almeria	Almocita	173	1	144	117	

8,125 rows

### 3. Generar un fichero con el top 10 de población de municipios de España y otro con el bottom 10 de población. Necesitamos el nombre de los municipios, autonomía, provincia y población.

Nos interesan Municipio, Comunidad, Provincia y Poblacion. Convertimos Poblacion a un tipo numérico para poder ordenar. Ordenar y filtrar:

orderBy ordena por población en orden descendente (Top 10) o ascendente (Bottom 10). limit(10) extrae solo los primeros 10 registros después del ordenamiento. Guardar los resultados:

Usamos el método write.csv para exportar los resultados en formato CSV. La opción header=True asegura que las cabeceras estén incluidas en el archivo.

PEC Big Data Yonatan Eleuterio (Python)

Import Notebook

21

display(top\_10)

Table

	Municipio	Comunidad	Provincia	Poblacion
1	Madrid	Comunidad de Madr...	Madrid	3141991
2	Barcelona	Catalunya	Barcelona	1604555
3	Valencia	Comunitat Valenciana	Valencia / Valencia	786189
4	Sevilla	Andalucia	Sevilla	693878
5	Zaragoza	Aragon	Zaragoza	664953
6	Malaga	Andalucia	Malaga	569130
7	Murcia	Region de Murcia	Murcia	439889
8	Palma de Mallorca	Illes Balears	Illes Balears	400578
9	Las Palmas de Gran Canaria	Canarias	Las Palmas	379766
10	Bilbao	Pais Vasco	Bizkaia	345141

10 rows

22

Windows taskbar: 09:23 p.m. 24/11/2024

PEC Big Data Yonatan Eleuterio (Python)

Import Notebook

22

display(bottom\_10)

Table

	Municipio	Comunidad	Provincia	Poblacion
1	Jaramillo Quemado	Castilla y Leon	Burgos	5
2	Illan de Vacas	Castilla - La Mancha	Toledo	6
3	Estepa de San Juan	Castilla y Leon	Soria	7
4	Castilnuevo	Castilla - La Mancha	Guadalajara	8
5	Villanueva de Gormaz	Castilla y Leon	Soria	8
6	Valdemadera	La Rioja	La Rioja	8
7	Villarroya	La Rioja	La Rioja	8
8	Quinoneria	Castilla y Leon	Soria	9
9	Torremochuela	Castilla - La Mancha	Guadalajara	11
10	Reinoso	Castilla y Leon	Burgos	11

10 rows

Windows taskbar: 09:23 p.m. 24/11/2024

- Queremos saber los 10 municipios donde ha habido más participación (el porcentaje de votos respecto el censo es más alto) y donde ha habido más abstención. Generaremos un fichero del top y uno del bottom con los datos del municipio y el % de participación.

**Cálculo de participación y abstención:**

CAST asegura que las divisiones se realicen correctamente como números flotantes.

La fórmula para participación y abstención está incluida en el selectExpr.


**Ordenamiento:**



`orderBy("Participacion", ascending=False):` Ordena en orden descendente para el Top 10.

`orderBy("Abstencion", ascending=False):` Ordena en orden descendente para el Bottom 10. Guardar resultados:

Los archivos se guardan en la ruta `dbfs:/mnt/tmp/` para accesibilidad. Incluyen encabezados gracias a `header=True`.

 **PEC Big Data Yonatan Eleuterio** (Python) Import Notebook


Table

	Municipio	Comunidad	Provincia	1.2 Participacion	1.2 Abstencion
1	Castilnuevo	Castilla - La Mancha	Guadalajara	100	0
2	Boada de Campos	Castilla y Leon	Palencia	100	0
3	Cincovillas	Castilla - La Mancha	Guadalajara	100	0
4	Banuelos	Castilla - La Mancha	Guadalajara	100	0
5	Ocentejo	Castilla - La Mancha	Guadalajara	100	0
6	La Zoma	Aragon	Teruel	100	0
7	Balconchan	Aragon	Zaragoza	100	0
8	Salcedillo	Aragon	Teruel	100	0
9	Alcolea de las Penas	Castilla - La Mancha	Guadalajara	100	0
10	Torremochuela	Castilla - La Mancha	Guadalajara	100	0

10 rows

27

Windows taskbar: 09:25 p.m. 24/11/2024

 **PEC Big Data Yonatan Eleuterio** (Python) Import Notebook

```
# Leer y mostrar el Bottom 10 de abstención
bottom_abs_df = spark.read.csv("dbfs:/mnt/tmp/bottom_abstencion.csv", header=True, inferSchema=True)
display(bottom_abs_df)
```

Table

	Municipio	Comunidad	Provincia	1.2 Participacion	1.2 Abstencion
1	Arano	Comunidad Foral de Navarra	Navarra	42.42424242424242	57.57575757575758
2	Les Valls de Valira	Catalunya	Lleida	43.728813559322035	56.271186440677965
3	Baliarrain	Pais Vasco	Gipuzkoa	46.31578947368421	53.68421052631579
4	Belena	Castilla y Leon	Salamanca	46.666666666666664	53.333333333333336
5	Ulobera	Catalunya	Lleida	47.42857142857143	52.57142857142857
6	Balboa	Castilla y Leon	Leon	47.51552795031056	52.48447204968944
7	Albiztur	Pais Vasco	Gipuzkoa	48.01587301587302	51.98412698412698
8	Hernialde	Pais Vasco	Gipuzkoa	48.23529411764706	51.76470588235294
9	Ezkurra	Comunidad Foral de Navarra	Navarra	48.25174825174825	51.74825174825175
10	Purujosa	Aragon	Zaragoza	48.57142857142857	51.42857142857143

10 rows

5. ¿Existe algún municipio que haya votado al 100% a un partido? Si es así, ¿cuál es y a qué partido? Si no es así, sacar una lista de los 10 municipios donde su concentración de voto porcentual haya sido mayor y a qué partido.

Windows taskbar: 09:26 p.m. 24/11/2024

```
# Guardar resultados
top_10_municipios.write.csv("dbfs:/mnt/tmp/top_10_concentracion.csv", header=True, mode="overwrite")
```

Municipios con 100% de votos a un partido:

Municipio	Partido_Maximo	Max_Porcentaje
Castilnuevo	psoe	100.0
Congostrina	psoe	100.0
Rebollosa de Jadr...	psoe	100.0
La Vid de Bureba	psoe	100.0
Portillo de Soria	psoe	100.0
Valdemadera	psoe	100.0

Como podemos ver Castilnuevo, Congostrina, Rebollosa, La vid de Bureba, Portillo de Soria y Valdemadera fueron los municipios con 100% de votos para el partido PSOE

32

```
# Leer municipios con 100% de votos
```

32

```
# Leer municipios con 100% de votos
municipios_100_df = spark.read.csv("dbfs:/mnt/tmp/municipios_100.csv", header=True, inferSchema=True)
display(municipios_100)
```

Table

	Municipio	Comunidad	Provincia	1.2 VotosValidos	1.2 pp	1.2 psoe	1.2 podemosuequo	1
1	Castilnuevo	Castilla - La Mancha	Guadalajara	7	7	0	0	0
2	Congostrina	Castilla - La Mancha	Guadalajara	10	10	0	0	0
3	Rebollosa de Jadrake	Castilla - La Mancha	Guadalajara	9	9	0	0	0
4	La Vid de Bureba	Castilla y Leon	Burgos	11	11	0	0	0
5	Portillo de Soria	Castilla y Leon	Soria	12	12	0	0	0

6 rows

6. Necesitamos comparar los datos de participación de la 'España vacía' con los de la 'España llena'. Saquemos el índice de participación (votos/censo) por provincia, ordenado de mayor a menor.

Como podemos ver Castilnuevo, Congostrina, Rebollosa, La vid de Bureba, Portillo de Soria y Valdemadera fueron los municipios con 100% de votos para el partido PSOE

6. Necesitamos comparar los datos de participación de la 'España vacía' con los de la 'España llena'. Saquemos el índice de participación (votos/censo) por provincia, ordenado de mayor a menor.

```
df_participacion_unweighted.write.csv('dbfs:/mnt/emp/indice_participacion_provincias.csv', mode='true', mode='overwrite')
```

Índice de participación por provincia:

Provincia	indice_participacion
Segovia	75.83481877599525
Avila	75.59051531049981
Valladolid	75.56431381386316
Valencia / Valencia	75.43376381209732
Cuenca	75.4200108389894
La Rioja	74.70781702175164
Madrid	74.26218486313643
Guadalajara	74.13460193088008
Palencia	74.11032808936196
Castellon / Castello	73.8008608827231
Cantabria	73.2672360974288
Burgos	73.28762779817225
Toledo	72.9733611127505
Albacete	72.88202091603256
Salamanca	72.57564543982153
Zaragoza	72.200368673421
Teruel	72.1377796808368

35

```
#filtro para 'España vacía' (provincias de baja densidad)
provincias_vacia = ["Soria", "Teruel", "Cuenca", "Zamora", "Palencia", "Ávila", "Segovia"]
```

Participación en España vacía:

Provincia	total_votantes	total_censo	indice_participacion
Segovia	89341.0	117810.0	75.83481877599525
Cuenca	116898.0	154996.0	75.4200108389894
Palencia	101377.0	136792.0	74.11032808936196
Teruel	75604.0	104805.0	72.1377796808368
Soria	49994.0	70435.0	70.97891673173848
Zamora	109330.0	154088.0	70.9529619438243

Participación en España llena:

Provincia	total_votantes	total_censo	indice_participacion
Madrid	3462738.0	4662855.0	74.26218486313643
Sevilla	1041748.0	1501017.0	69.4028115604287
Barcelona	2647828.0	3971665.0	66.66795915566897

7. Queremos saber si los municipios grandes son representativos en los resultados de las comunidades. Debemos sacar para cada comunidad el municipio que tiene más población y comparar el partido con más participación del municipio con la comunidad. ¿Coinciden? ¿No coinciden? ¿Tiene que ver con que represente más de un determinado porcentaje de población de la comunidad?

```
display(comparacion)
```

Table						
	comunidad	municipio	partido	votos_municipio	partido_comunidad	votos_comunidad
1	Andalucía	Sevilla	pp	129961	pp	
2	Aragón	Zaragoza	pp	121660	pp	
3	Canarias	Las Palmas de Gran Canaria	pp	64080	pp	
4	Cantabria	Santander	pp	43755	pp	
5	Castilla - La Mancha	Albacete	pp	38470	pp	
6	Castilla y León	Valladolid	pp	75414	pp	
7	Cataluña	Barcelona	erc-cats	132722	ercats	
8	Ciudad de Ceuta	Ceuta	pp	15956	pp	
9	Ciudad de Melilla	Melilla	pp	13478	pp	
10	Comunidad Foral de Navarra	Pamplona / Iruña	pp	36344	pp	
11	Comunidad de Madrid	Madrid	pp	696804	pp	
12	Comunitat Valenciana	Valencia	pp	159079	pp	
13	Extremadura	Badajoz	pp	35531	pp	
14	Galicia	Vigo	pp	52297	pp	

19 rows

Los municipios más grandes, independientemente de su porcentaje de población en la comunidad, tienden a ser representativos del partido dominante en la comunidad. Este efecto se observa tanto en comunidades donde los municipios más grandes tienen una influencia significativa como en aquellas donde representan una fracción menor. Sin embargo, en regiones con porcentajes muy pequeños, la coincidencia podría ser fortuita y necesitaría análisis adicionales.

8. **Vamos a analizar los datos de los municipios grandes y de los municipios pequeños.**
9. **Sacaremos la participación y el top 5 de partidos votados en los 20 municipios con más población de España.**
  - **Sacaremos la participación y el top 5 de partidos votados en los 20 primeros municipios con menos de 10000 habitantes de España.**
  - **Comparemos resultados. ¿Cómo se comportan los diferentes municipios y partidos? ¿Tiene que ver la participación con que gane un partido u otro? ¿Y la provincia o autonomía?**

databricks

PEC Big Data Yonatan Eleuterio (Python)

Import Notebook

```
display(top_partidos_grandes)
```

	Año municipio	Año provincia	Año comunidad	1.2 participacion	% pp	% psoe	% podemosuequo
1	Madrid	Madrid	Comunidad de Madr...	73.77538730593457	696804	329947	36752
2	Barcelona	Barcelona	Catalunya	67.59953324404191	116255	107621	
3	Valencia	Valencia / Valencia	Comunitat Valenciana	76.54331440657171	159079	76793	
4	Sevilla	Sevilla	Andalucia	71.08096289777099	129961	107150	7991
5	Zaragoza	Zaragoza	Aragon	72.55865129086952	121660	82192	7852
6	Malaga	Malaga	Andalucia	67.5082094558808	96359	69196	5928
7	Murcia	Murcia	Region de Murcia	74.00989150021087	109773	38700	3512
8	Palma de Mallorca	Illes Balears	Illes Balears	62.0171207113545	58394	33911	4435
9	Las Palmas de Gran Canaria	Las Palmas	Canarias	64.5707453222139	64080	43871	4562
10	Bilbao	Bizkaia	Pais Vasco	67.50430330340208	31165	26257	5008
11	Alicante / Alacant	Alicante / Alacant	Comunitat Valenciana	70.55868582193023	62244	33915	
12	Cordoba	Cordoba	Andalucia	69.86459322235845	71843	38361	3825
13	Valladolid	Valladolid	Castilla y Leon	75.64742209401535	75414	42241	3189
14	Vigo	Pontevedra	Galicia	71.50963565763733	52297	40205	

20 rows

48

09:38 p.m.  
24/11/2024

*Participación Municipios grandes:* Promedio de participación: 70.6% La participación es alta en general, con municipios como Valencia y Madrid superando el 73%.

*Municipios con menor participación:* Palma de Mallorca (62%) y L'Hospitalet de Llobregat (65%). Municipios pequeños:

*Promedio de participación:* 70.5% Similar a los municipios grandes, pero con una mayor dispersión.

*Municipios con participación más alta:* Buñol (78.8%) y Castalla (78.2%). Municipios con menor participación: Zumaia (65%) y Ordizia (66.7%).

La participación promedio es comparable entre municipios grandes y pequeños. Sin embargo, los municipios pequeños tienden a mostrar una mayor variabilidad en participación.

*Municipios grandes:*

PP: Dominante en la mayoría de los municipios grandes. Por ejemplo, obtiene mayoría absoluta en Madrid (696,804 votos) y Valencia (159,079 votos). PSOE: En segundo lugar en muchos municipios grandes como Sevilla, Málaga y Bilbao. Podemos-IU-Equo: Fuerte en algunas localidades, como Bilbao (50,083 votos) y Zaragoza (78,527 votos). VOX: Baja representación general; su mayor desempeño está en Madrid, aunque sigue siendo marginal.

databricks

PEC Big Data Yonatan Eleuterio (Python)

Import Notebook

```
display(top_partidos_pequenos)
```

	A <sub>1</sub> municipio	A <sub>2</sub> provincia	A <sub>3</sub> comunidad	1.2 participacion	1 <sup>o</sup> pp	1 <sup>o</sup> psOE	1 <sup>o</sup> podemosIU-Equo	1 <sup>o</sup>
1	Castalla	Alicante / Alacant	Comunitat Valenciana	78.20370882922779	2274	879	0	
2	Miajadas	Caceres	Extremadura	69.15266984206568	2221	1977	689	
3	Mengíbar	Jaen	Andalucia	70.14550264550265	1915	2256	445	
4	Ribadeo	Lugo	Galicía	67.8712382831771	2378	1106	0	
5	Zumarraga	Gipuzkoa	País Vasco	66.92793931731985	410	1246	1626	
6	Daganzo de Arriba	Madrid	Comunidad de Madr...	76.86980609418282	1794	810	1046	
7	Grinon	Madrid	Comunidad de Madr...	75.49429128376497	2734	654	793	
8	Campos	Illes Balears	Illes Balears	68.20229617788112	2362	604	778	
9	Foz	Lugo	Galicía	71.17494760202194	2349	1352	0	
10	Caspe	Zaragoza	Aragon	66.44674835061262	1247	1320	751	
11	Ordizia	Gipuzkoa	País Vasco	66.75250357653792	340	544	1129	
12	Gelves	Sevilla	Andalucia	71.02640086206897	1692	1536	992	
13	Caldas de Reis	Pontevedra	Galicía	67.47937446127324	2290	1434	0	
14	Fortuna	Murcia	Region de Murcia	72.07678883071553	2441	952	672	

20 rows

### Municipios pequeños:

PP: Sigue dominando en la mayoría de los municipios pequeños, pero con márgenes menores. PSOE: Más competitivo en municipios pequeños, llegando a ser el partido más votado en localidades como Tocina (2,843 votos) y Mengíbar (2,256 votos). Podemos-IU-Equo: Consigue un desempeño relevante en algunos municipios como Zumaia (1,317 votos) y Ordizia (1,129 votos). Otros partidos: En municipios pequeños hay más fragmentación del voto hacia partidos locales o regionales.

El PP domina en ambos contextos, pero el PSOE es más competitivo en municipios pequeños. Además, la fragmentación del voto en municipios pequeños indica un mayor peso de partidos locales y minoritarios.

Relación con provincias y autonomías En Catalunya, partidos regionales como ERC (en municipios grandes como Barcelona) y otros partidos de izquierda como Podemos son más fuertes. En el País Vasco, Podemos-IU-Equo y otros partidos minoritarios tienen más presencia que en otras regiones. En Andalucía, el PP y PSOE se disputan las mayorías en municipios grandes y pequeños. En Comunitat Valenciana, el PP domina tanto en grandes como pequeños municipios, aunque con fragmentación significativa en municipios pequeños.

**10. En el caso actual la mayoría de respuestas se pueden plantear en entornos relacionales o pseudorelacionales. Si tuviésemos una versión turbo de las urnas de votación en la que cada vez que se produce una votación se generase un mensaje al respecto (obviamente no de quien vota ni a quién, sólo de un voto) para tener la participación en tiempo real, ¿podríamos considerar una base de datos NoSQL para almacenar la información? ¿Qué tipo de base de datos utilizaríamos y qué información se te ocurre que almacenaríamos?**

**Debemos considerar que hay una mesa de votación por cada 500 votantes, por lo que asumiendo 30 millones de votantes y una participación del 100% tendríamos un total de 60.000 urnas emitiendo información de votación. Haciéndolo lineal en las 11 horas que suele durar la jornada de votaciones, hablamos de un máximo de 45.000 votos por minuto.**

Sí, una base de datos NoSQL sería una opción ideal para manejar este caso debido al volumen, velocidad y la estructura simple de los datos que generaría este sistema. Para esto el sistema tener lo siguiente:

- Alta velocidad de escritura: Manejar un flujo constante de 45,000 votos/minuto, con picos potenciales superiores.
- Distribución geográfica: La información proviene de 60,000 urnas distribuidas en todo el país, lo que implica la necesidad de una infraestructura escalable y de baja latencia.
- Escalabilidad horizontal: La base de datos debe crecer fácilmente agregando nodos a medida que aumente la participación o el número de votantes.
- Consulta en tiempo real: Generar estadísticas como participación total, por comunidad, provincia o municipio sin afectar las operaciones de escritura.
- Simplitud del modelo de datos: Cada mensaje contiene poca información (un voto registrado), lo que permite una estructura de datos sencilla.

#### *Tipo de Base de Datos NoSQL Ideal*

Base de datos de series temporales (Time Series Database):

Ejemplos: InfluxDB, Amazon Timestream, TimescaleDB. Diseñadas para manejar flujos de datos con marca de tiempo. Optimizadas para consultas de agregación y métricas en tiempo real. Compresión eficiente de datos históricos.

Base de datos orientada a documentos:

Ejemplos: MongoDB, Couchbase. Flexibles en cuanto al esquema, permiten agregar datos adicionales fácilmente. Buen soporte para agregaciones y búsquedas por índices.

Base de datos de clave-valor con pub-sub:

Ejemplos: Redis Streams, Apache Kafka (para almacenamiento y procesamiento). Ideales para manejar flujos de datos en tiempo real. Redis puede actuar como almacenamiento temporal para analítica rápida, mientras que Kafka puede registrar los mensajes para análisis más extensos.

Modelo de Datos Cada mensaje puede contener la siguiente información mínima:

{

```
"timestamp": "2024-11-24T09:15:34Z",    // Marca de tiempo
"mesa_id": "123456",                    // Identificador único de la mesa
"municipio_id": "ES-28079",             // Código del municipio
"provincia": "Madrid",                  // Provincia de la mesa
"comunidad": "Comunidad de Madrid",     // Comunidad Autónoma
"mesa_participacion": 250,              // Participación acumulada en la mesa
"total_municipio": 8000                 // Participación acumulada en el municipio
}
```

## Conclusión

Este análisis ha proporcionado una visión detallada de cómo la participación electoral varía según el tamaño de los municipios y las comunidades autónomas. El uso de PySpark para procesar y analizar grandes volúmenes de datos permitió obtener insights valiosos sobre la distribución de los votos y la participación electoral. Las diferencias en la participación y los resultados por comunidad autónoma destacan la importancia de entender el contexto geográfico y político al analizar los resultados electorales.