**ELECCIONES EEUU 2020 – DOCUMENTACIÓN PROCESO ANALÍTICO**

**DATOS ENTRADA**

Partimos de un TXT con la transcripción del debate entre Biden y Trump de 2020.

Fuente de datos Kaggle: <https://www.kaggle.com/datasets/headsortails/us-election-2020-presidential-debates/data>

**PRIMER PROCESO ANALITICO**

1. Leo con Spark el fichero TXT y lo cargo en un RDD.
2. Creo una función group\_speaker\_blocks que va a recibir un bloque de líneas y las va a agrupar por bloques (‘SPEAKER’, ‘TIME’, ‘TEXT’)
3. # Paso 2: Agrupar bloques (speaker + texto)
4. def group\_speaker\_blocks(lines):
5. result = []
6. current\_speaker = None
7. current\_time = None
8. current\_text = []
9. for line in lines: #Bucle de lectura de lineas
10. match = re.match(r"^(.\*?):\s\*\((\d{2}:\d{2})\)", line) #\*\*
11. if match: #Devuelve TRUE si reconoce el patron Speaker : (mm:ss)
12. # Si ya había un bloque anterior, lo guardamos
13. if current\_speaker:
14. result.append((current\_speaker, current\_time, " ".join(current\_text)))
15. current\_speaker = match.group(1) #Guarda Speaker
16. current\_time = match.group(2) #Tiempo
17. current\_text = [line[match.end():].strip()]  # y Texto
18. else:
19. if current\_speaker:
20. current\_text.append(line.strip()) #Sigue añadiendo lineas de texto mientras no encuentre nuevo patron.
21. # Añadir el último bloque
22. if current\_speaker: #Acaban las lineas asique añado el ultimo patron y devuelvo res que es una LISTA.
23. result.append((current\_speaker, current\_time, "".join(current\_text)))
24. return result

**\*\* Explicación patrón r"^(.\*?):\s\*\((\d{2}:\d{2})\)\s\*(.\*)"**

* **^(.\*?)**: desde el principio, captura todo hasta el primer “**:”** (el nombre del hablante).
* **\s\***: posibles espacios en blanco.
* **\((\d{2}:\d{2})\)**: entre paréntesis, captura el timestamp tipo 01:20.
* **\s\***: más espacios.
* **(.\*)**: el resto de la línea, es el texto hablado.

3. Llamo a la función con mi RDD. grouped\_data = group\_speaker\_blocks(rdd.collect())

4. Creo RDD con los bloques. parsed\_rdd = sc.parallelize(grouped\_data)

5. Creo una lista de palabras que no quiero que se analicen porque no aportan mucha información analítica interesante. Esta lista la he ido variando en función de las analíticas que he ido haciendo.

stopwords = set(["this", "that", "what", "which", "with", "have", "from", "would", "could", "should", "their", "there", "about", "because", "before", "after", "over","under", "again", "some", "them", "been", "being", "then", "than","themselves", "does", "did", "doing", "into", "just", "a", "an", "the","more", "most", "other", "out", "such", "too", "very", "where", "when", "how", "who", "whom", "whose", "why", "also", "only", "every", "much","many", "each", "same", "own", "any", "both", "few", "while", "though","through", "against", "among", "those", "etc", "like", "even", "still","in", "on", "at", "by", "for", "with", "about", "against", "between", "into", "through", "during", "before", "after", "above", "below", "to", "from", "up", "down", "out", "off", "over", "under","are", "be","is","of", "but", "it", "if","was","were", "and","I", "i","you", "he", "she", "it", "we", "they", "me", "him", "her", "us", "them", "my", "your", "his", "her", "its", "our", "their", "mine", "yours", "hers", "ours", "theirs", "myself", "yourself", "himself", "herself", "itself", "ourselves", "yourselves", "themselves", "who", "whom", "whose", "which", "that", "anyone", "anybody", "someone", "somebody", "everyone", "everybody", "no one", "nobody", "each", "either", "neither", "none", "one", "another", "others", "everything", "something", "nothing", "anything", "not", "no", "as", "so","all"])

6. Ya tengo el RDD parseado ahora necesito analizarlo. Primero filtro las palabras que no me interesen.

def filter\_short\_words(record):

    speaker, time, text = record

    # Filtramos palabras de lista stopwords

    words = [word for word in text.split() if word.isalpha() and word.lower() not in stopwords]

    # Reconstruimos el texto limpio

    clean\_text = " ".join(words)

    return (speaker, time, clean\_text)

# Aplicamos el filtrado al RDD

filtered\_rdd = parsed\_rdd.map(filter\_short\_words)

7. Ya podemos comenzar con el conteo de palabras , creando pares ((speaker,word), 1)

# Paso 1: crear pares ((speaker, word), 1)

word\_pairs\_rdd = filtered\_rdd.flatMap(

    lambda record: [((record[0], word.lower()), 1) for word in record[2].split()]

)

8. Y Reduciendo por clave y hacemos suma acumulada.

# Paso 2: Reducir por clave (speaker, word)

word\_counts\_rdd = word\_pairs\_rdd.reduceByKey(lambda a, b: a + b)

9. Hacemos un GROUP BY speaker para agrupar todas las palabras que ha dicho cada speaker y su conteo.

# Paso 3: reorganizar como (speaker, (word, count))

grouped\_by\_speaker\_rdd = word\_counts\_rdd.map(

    lambda pair: (pair[0][0], (pair[0][1], pair[1])))

# Agrupar todas las palabras por speaker

speaker\_word\_list\_rdd = grouped\_by\_speaker\_rdd.groupByKey().mapValues(list)

10. Sacar el TOP 20 de cada speaker.

top\_20\_words\_per\_speaker = speaker\_word\_list\_rdd.mapValues(

    lambda word\_counts: sorted(word\_counts, key=lambda x: x[1], reverse=True)[:20]

)

11.Creo un dataframe de columnas speaker, word, count para poder visualizar/exportar los datos analíticos.

from pyspark.sql import Row

# Aplanamos los datos

flattened = top\_10\_words\_per\_speaker.flatMap(lambda x: [Row(speaker=x[0], word=w[0], count=w[1]) for w in x[1]])

df = spark.createDataFrame(flattened)

**VISUALIZACIÓN POWER BI**

No he realizado la analítica de la mano de linguistas especializados que me orientasen en que palabras filtrar. Ha sido un mero experimento práctico para comprender mejor el uso de los RDD en Spark y practicar con un caso de uso realista donde es interesante su uso.

1. **Conteo de palabras de los 2 protagonistas incluyendo pronombres**

He incluido pronombres en esta analítica para obtener unas primeras hipótesis.

**Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.**

Es interesante ver como **Trump** emplea mucho el pronombre “**you**” lo que sugiere que puede estar criticando o poniendo en dudas las ideas de Biden, increpándole a el directamente. Sin embargo, los datos sugieren que **Biden** tomó otra estrategia completamente distinta. El empleó el pronombre “**he**” lo que sugiere que no entró en tanta confrontación directa con Trump, si no que optó por una táctica de apelación al espectador, hablando sobre su contrincante y sus ideas, queriéndole restar protagonismo y queriendo tomar un papel de “yo estoy con la gente” y a “El” es a quien debemos derrotar/derrocar (porque Trump era el presidente en ese momento)

Por seguir en orden de palabras con mayores ocurrencias, vemos que **Trump** emplea mucho mas el pronombre personal **“I”** refiriéndose a el mismo lo que sugiere que su discurso contiene propuestas y promesas. Y en concreto promesas personificadas en su propia persona, lo que quiere transmitir una posición de Liderazgo y de enfoque de la población a una sola persona, él mismo, apelando a la pasión del público y sabiendo que tiene un público fanático de su persona. Aunque no descarta el uso del **“we”** para hablar de su partido. **Biden** sin embargo, lo emplea bastante menos, aunque también lo hace y se decanta mas por el uso del **“we”** refiriéndose a su propio partido como grupo y poniendo menos peso en su propia figura como gran lider.

1. **Conteo de palabras eliminando pronombres personales**

**Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.**

En este caso elimino los pronombres para centrarme en expresiones verbales destacables. La primera que destacaría sería el conteo de las palabras **“people” y “want”**. **Trump** hace un uso de ambas muy elevado lo que puede suponer hipotéticamente que esté intentando dejar claro a Biden lo que la gente quiere “This is what people want …” o confirmar los propios pensamientos de los espectadores. Esto es una suposición hecha sin visualizar el debate, también puede venir relacionado con la técnica de rebatir a **Biden** con frases como “What you want is not possible …” o con la técnica de exposición de ideas con el pronombre “I” como “I want to do …” que suena contundente y directa. Aunque quien mas emplea la palabra “**people**” es Biden. Esto puede deberse a una estrategia alineada con el uso de pronombres, mas enfocada a dirigirse a los espectadores, crear sensación de comunidad y de hacer frente a un lider cuya técnica es muy individualista.

Por otro lado esto se puede complementar con la negación “**not**” que realmente puede emplearse con muchos verbos diferentes y en circunstancias diversas pero puede emplearse en frases parecidas a las vistas previamente “people want …”, **“people do not want …” o “I am not going to …”, “We are not going to …”.** Ambos emplean esta palabra, pero **Biden** en mucha mayor medida, lo que puede sugerir desacuerdo, aclaración de comentarios de Trump con los que no está de acuerdo. En definitiva defensa.

1. **Elimino "not", "no", "as", "so","all"**

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Eliminando estas palabras para ver otras con mas claridad podemos observar términos como **fact**, **number o tax** que podrían sugerir un discurso mas analítico o de uso de datos y apelación a términos económicos en determinadas partes por parte de **Biden. Trump** no emplea exactamente esas, pero no se descarta que hable de datos para nada, ya que emplea mucho la palabra “**million**” lo cual se refiere a datos económicos o de número de personas con alta probabilidad. Hablar de cantidades altas de dinero puede resultar impactante para la gente y puede ser empleado como recurso.