Pontificia Universidad
JAVERIANA
Cali

PROYECTO 1 (data cleaning + MLlib)

Presentado por:
Julian Paredes Conde
Juan Camilo Hernandez Saavedra
Augustin Valencia

PROCESAMIENTO DE GRANDES VOLÚMENES DE DATOS

CALI, SEPTIEMBRE DE 2021

- **Contexto:**

  Este conjunto de datos refleja los incidentes de delitos denunciados (con la excepción de los asesinatos) que ocurrieron en la ciudad de Chicago desde 2001 hasta el 2017. Los datos se extraen del sistema CLEAR (Análisis e informes de aplicación de la ley ciudadana) del Departamento de Policía de Chicago. Para proteger la privacidad de las víctimas de delitos, las direcciones se muestran solo a nivel de bloque y no se identifican ubicaciones específicas. El dataset se encuentra dividido en 4 archivos .csv donde cada archivo contiene información en un rango de 3 años, a excepción del último archivo que contiene información en un rango de 4 años (2012-2017). Para nuestro caso de estudio se trabajará con el último archivo pues consideramos que este cuenta con la cantidad de información que creemos justa para el estudio.

- **Estructura del Dataset:**

  El dataset cuenta con 23 atributos de los cuales inicialmente se utilizarán 22 para poder predecir la característica deseada. Las características que posee el dataset son:

  - ID: identificador único del registro.
  - Número de caso: el número RD del Departamento de Policía de Chicago (número de división de registros), que es exclusivo del incidente.
  - Fecha: fecha en que ocurrió el incidente.
  - Bloque: la dirección parcialmente redactada donde ocurrió el incidente, colocándola en el mismo bloque que la dirección real.
  - IUCR - El código uniforme de denuncia de delitos de Illinois. Esto está directamente relacionado con el Tipo principal y la Descripción.
  - Tipo principal: la descripción principal del código IUCR.
  - Descripción: la descripción secundaria del código IUCR, una subcategoría de la descripción primaria.
  - Descripción de la ubicación: descripción de la ubicación donde ocurrió el incidente.
  - Arresto: indica si se realizó un arresto.

- Doméstico: indica si el incidente estuvo relacionado con el hogar según lo define la Ley de Violencia Doméstica de Illinois.

- Beat: Un beat se define como el territorio y horario en el que una patrulla realiza su trabajo. Esta columna determina el beat en el que ocurrió el incidente. Un beat es el área geográfica policial más pequeña: cada beat tiene un automóvil policial dedicado. Tres a cinco beat conforman un sector policial y tres sectores conforman un distrito policial. El Departamento de Policía de Chicago tiene 22 distritos policiales.

- Distrito: indica el distrito policial donde ocurrió el incidente.

- Área de la comunidad: indica el área de la comunidad donde ocurrió el incidente. Chicago tiene 77 áreas comunitarias.

- Código del FBI: indica la clasificación de delitos según se describe en el Sistema nacional de informes basados en incidentes (NIBRS) del FBI.

- Coordenada X: la coordenada x de la ubicación donde ocurrió el incidente en la proyección de State Plane Illinois East NAD 1983. Esta ubicación se desplaza de la ubicación real para la redacción parcial, pero cae en el mismo bloque.

- Coordenada Y: la coordenada y de la ubicación donde ocurrió el incidente en la proyección de State Plane Illinois East NAD 1983. Esta ubicación se desplaza de la ubicación real para la redacción parcial, pero cae en el mismo bloque.

- Año: año en que ocurrió el incidente.

- Actualizado: fecha y hora en que se actualizó por última vez el registro.

- Latitud: la latitud de la ubicación donde ocurrió el incidente. Esta ubicación se desplaza de la ubicación real para la redacción parcial, pero cae en el mismo bloque.

- Longitud: la longitud de la ubicación donde ocurrió el incidente. Esta ubicación se desplaza de la ubicación real para la redacción parcial, pero cae en el mismo bloque.

- Ubicación: la ubicación donde ocurrió el incidente en un formato que permite la creación de mapas y otras operaciones geográficas en este portal de datos. Esta ubicación se desplaza de la ubicación real para la redacción parcial, pero cae en el mismo bloque.

- **¿Qué se va a predecir?:**

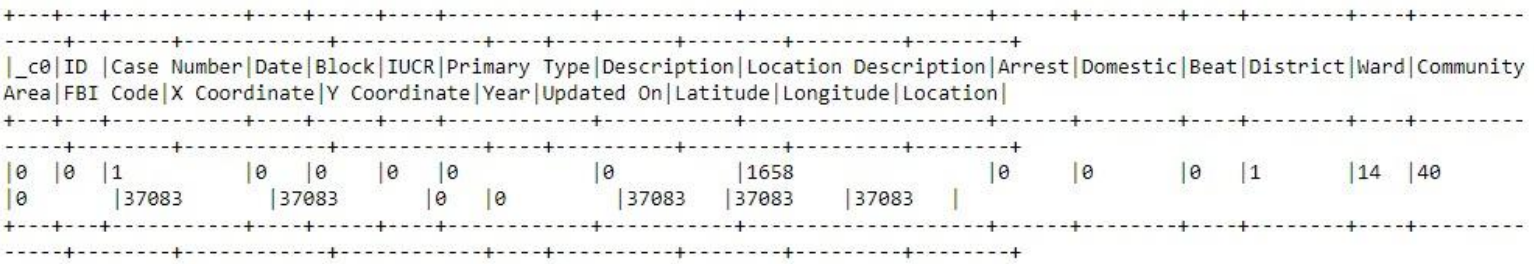
  Para nuestro proyecto se busca predecir si dado un delito se va a realizar un arresto o no. Todo esto con base del tipo de delito cometido, información de la ubicación donde se cometió el delito, descripción demográfica del delito, fecha y hora, y patrulla o patrullas que atendieron el delito (beat).

  Esto es un problema de clasificación binaria, pues las salidas de los modelos estarán representados de manera binaria, donde 1 significa que se realizará un arresto y 0 significa que no se realizará un arresto.

- **Descripción Dataset:**

  El dataset nos proporciona información sobre los delitos realizados en la ciudad de Chicago desde el año 2012 hasta el año 2017. Inicialmente el conjunto de datos contaba con:

  Número de filas o instancias: 1.456.714
  Número de columnas o características: 23
  Número de datos nulos Total: 187.129



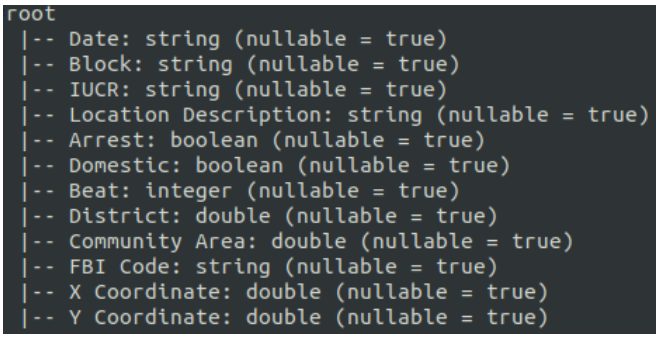
_c0	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	Beat	District	Ward	Community Area	FBI Code	X Coordinate	Y Coordinate	Year	Updated On	Latitude	Longitude	Location
0	0	1	0	0	0	0	0	1658	0	0	0	1	14	40	0	37083	37083	0	0	37083	37083	37083

Valores nulos de cada columna.

Como se puede observar los atributos donde mayor se presenta pérdida de datos son en las columnas que se relacionan con la ubicación donde ocurrieron los crímenes además, se puede observar que la cantidad de datos nulos son exactamente iguales en varias de estas columnas. Esto puede ocurrir debido a la confidencialidad que se tuvo con ciertos crímenes dentro del dataset.

- **Transformaciones Realizadas:**

    - **Selección de atributos:** A simple vista se pudo identificar ciertas columnas que eran redundantes o innecesarias para el desarrollo del modelo. Dentro de estos atributos podemos encontrar; los atributos relacionados con la ubicación de un delito, el año de un delito, el id de un delito, el número de caso, tipo y descripción del delito, y el número de distrito municipal, y fecha de actualización del delito.
      Por otro lado, mucha de la información relacionada con la ubicación del delito es la misma representada de una forma diferente, es por eso, que se llegó a la decisión de eliminar las columnas longitud, latitud y localización, y dejar únicamente las columnas que hacen referencia a las coordenadas x, y de la ubicación del delito.



    - **Manipulación columnas y filas nulas:** Como se pudo observar en la sesión anterior se obtuvo una cantidad relativamente pequeña de datos nulos, donde mayormente estos provienen de las columnas relacionadas con la ubicación del delito. Una vez eliminadas las columnas que no se utilizaran, se procede a eliminar las filas que contienen datos nulos.

Nuevo número de filas o instancias: 1.418.379

- **Serialización de atributos:** En nuestro dataset, se encontraron ciertos atributos que se podían serializar ya que estos en nuestro caso, hacen referencia a códigos policiales o federales (FBI) con los cuales estas entidades clasifican los delitos o situaciones. Los atributos serializados son IUCR, FBI Code, Location Description y bloque o dirección donde ocurrió el delito. En el dataset los atributos serializados los representamos con la abreviatura "_index" al final del nombre del atributo.

- **Manipulación de los atributos booleanos:**
  La función de serialización que provee pyspark no permite serializar atributos tipo booleanos. Para poder dejar estas columnas en el tipo de dato entero, se debe utilizar métodos de sql que provee pyspark.

```
#serializar columnas
columnasSerializar = ['IUCR', 'Location Description', 'FBI Code', 'Block']
indexers = [StringIndexer(inputCol=column, outputCol=column+"_index").fit(df) for column in columnasSerializar]
pipeline = Pipeline(stages=indexers)
df = pipeline.fit(df).transform(df)
df = df.select([column for column in df.columns if column not in columnasSerializar])

#conervirt booleanos a enteros
df = df.withColumn("Arrest", when(col("Arrest"), lit(1)).otherwise(0)).withColumn("Domestic", when(col("Domestic"),
                lit(1) ).otherwise(0))
```

- **Transformación de Datos Fecha:**
  Como se puede observar en nuestro dataset resultante, quedaron ciertos atributos que contienen fechas pero en formato cadena. Para poder manipular las fechas de manera más cómoda para futuros proyectos, primeramente transformamos esta columna a el formato timestamp. Estos atributos poseen la abreviatura "ts".

  Para mejorar la precisión general de los modelos, se decidió descomponer el atributo fecha del delito en 4 nuevos atributos, mes, día, hora y minuto del delito. Se decidió excluir el año del delito pues este atributo puede empeorar la precisión de los modelos ya que, al ser un dataset que se encuentra entre los años 2012 y 2017 intentar predecir un delito que ocurrió fuera de este rango de años, teniendo el atributo año dentro del modelo, puede generar problemas de precisión.

```
root
 |-- Arrest: integer (nullable = true)
 |-- Domestic: integer (nullable = true)
 |-- Beat: integer (nullable = true)
 |-- District: double (nullable = true)
 |-- Community Area: double (nullable = true)
 |-- X Coordinate: double (nullable = true)
 |-- Y Coordinate: double (nullable = true)
 |-- IUCR_index: double (nullable = true)
 |-- Location Description_index: double (nullable = true)
 |-- FBI Code_index: double (nullable = true)
 |-- Block_index: double (nullable = true)
 |-- mesDel: integer (nullable = true)
 |-- diaDel: integer (nullable = true)
 |-- horaDel: integer (nullable = true)
 |-- minutoDel: integer (nullable = true)
```

- **Almacenamiento de los datos:**
  Una vez limpiado todo el dataset, se exportó el dataset resultante, esto con la finalidad de no tener que limpiar el dataset cada vez que se desee trabajar con él en futuros proyectos. La librería de pySpark al exportar los datos los divide en varios archivos .csv, con el fin de poder paralelizar la información una vez se desee manipular.

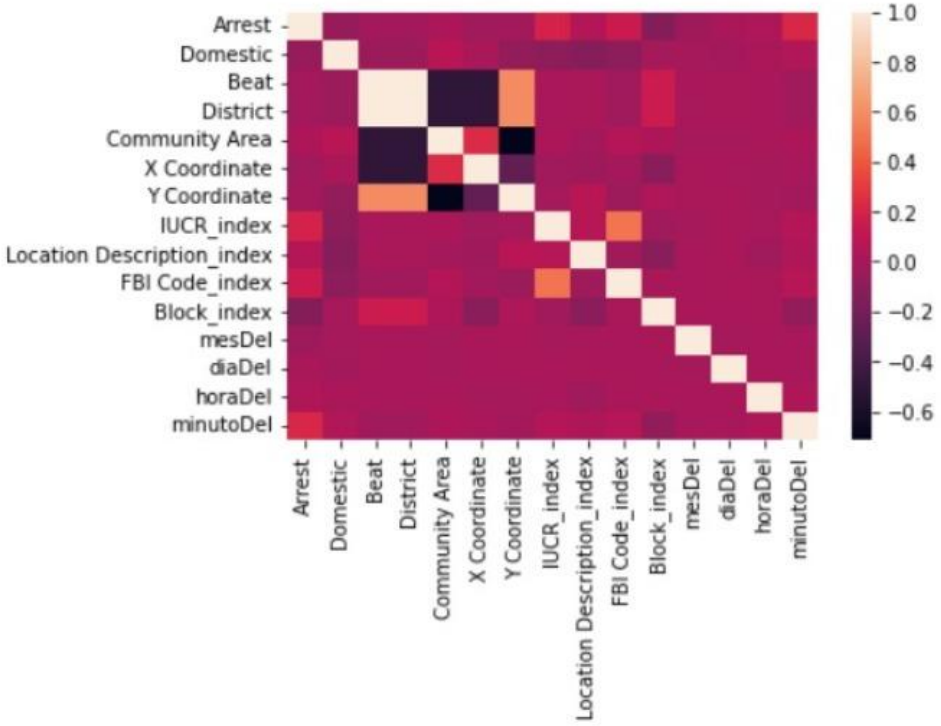
```
part-00000-617a955d-dc05-4c28-846a-df316565a832-c000.csv  part-00002-617a955d-dc05-4c28-846a-df316565a832-c000.csv  _SUCCESS
part-00001-617a955d-dc05-4c28-846a-df316565a832-c000.csv  part-00003-617a955d-dc05-4c28-846a-df316565a832-c000.csv
```

- **Vectorización de los Atributos:**
  Esta transformación se realizó justo antes de ingresar los datos a los modelos de machine learning correspondientes, todo esto con el fin de poder realizar un análisis gráfico y más profundo de las variables (ver siguiente sección). Cabe aclarar que antes de implementar los modelos de machine learning todas la columnas con un tipo de dato diferentes a enteros o dobles, fueron serializadas con el fin de que los datos sean aceptados por el modelo de machine learning.

```
root
 |-- atributos: vector (nullable = true)
 |-- Arrest: integer (nullable = true)
```

```
+--------------------+------+
|           atributos|Arrest|
+--------------------+------+
|[0.0,2515.0,25.0,...|     1|
|[1.0,1823.0,18.0,...|     0|
|[0.0,1832.0,18.0,...|     0|
|[0.0,2413.0,24.0,...|     0|
|[0.0,1824.0,18.0,...|     0|
|[0.0,1413.0,14.0,...|     1|
|[0.0,1633.0,16.0,...|     1|
|[0.0,1113.0,11.0,...|     1|
|[0.0,1235.0,12.0,...|     0|
|[0.0,1121.0,11.0,...|     1|
|[0.0,2423.0,24.0,...|     1|
|[0.0,2223.0,22.0,...|     0|
|[0.0,1434.0,14.0,...|     0|
|[0.0,531.0,5.0,49...|     0|
|[0.0,833.0,8.0,65...|     1|
|[0.0,2532.0,25.0,...|     1|
|[1.0,726.0,7.0,67...|     0|
|[0.0,1832.0,18.0,...|     0|
|[1.0,822.0,8.0,63...|     0|
|[0.0,1434.0,14.0,...|     0|
+--------------------+------+
only showing top 20 rows
```

- **Análisis de los Datos:**
    - **Matriz de Correlación:**

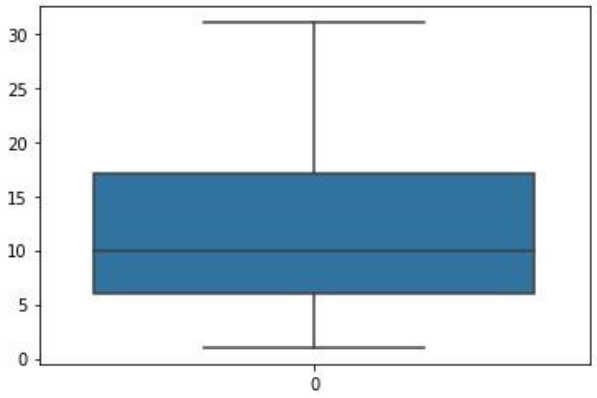


De la matriz de correlación anterior, podemos ver que destaca principalmente ciertas correlaciones negativas fuertes y correlaciones positivas fuertes entre ciertos atributos. Se puede observar que el área comunitaria posee una correlación negativa con el beat y los distritos. Por otro lado, se puede observar que el beat posee una correlación positiva fuerte con el atributo distrito, de igual manera aunque en menor medida, esto mismo pasa con los atributos IUCR y FBI code.

- **Diagramas de Cajas y Bigotes:**

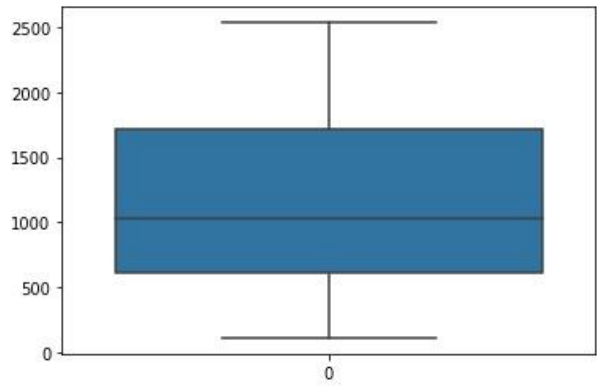
```
sns.boxplot(data=df.District)
```

```
<AxesSubplot:>
```



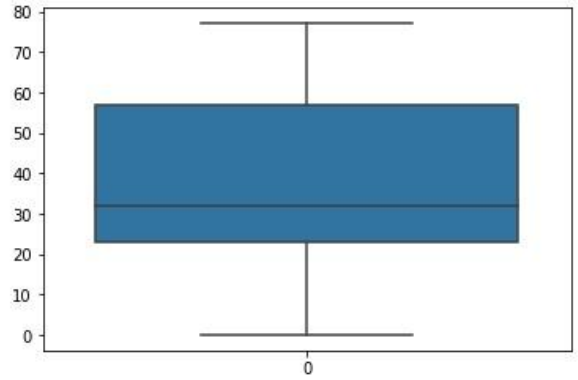

```
sns.boxplot(data=df.Beat)
```

```
<AxesSubplot:>
```



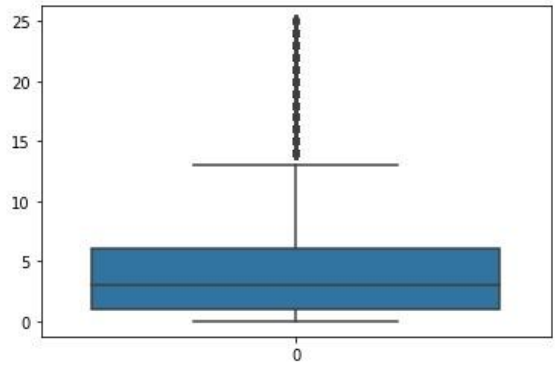

```
sns.boxplot(data=df.CommunityArea)
```

```
<AxesSubplot:>
```




```
sns.boxplot(data=df.FBICode_index)
```

```
<AxesSubplot:>
```

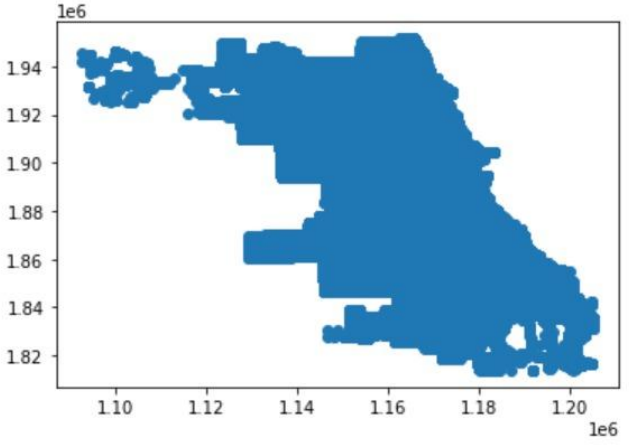


Las anteriores gráficas de cajas y bigotes nos permiten visualizar cómo están distribuidos y en qué rangos están ciertos atributos del dataset. Solamente se presentaron datos anómalos en la gráfica relacionada con los códigos de delitos del FBI. Analizando estos datos anómalos, se llegó a la conclusión de que estos datos hacen referencia a ciertos tipos de delitos que se cometieron en menor medida, o que son delitos muy poco comunes.

- **Gráfica de Coordenadas X Y:**
  Al realizar la gráfica de distribución sobre las coordenadas X Y de los delitos ocurridos dentro de la ciudad de Chicago, podemos confirmar que estos hacen referencia a coordenadas sobre la ubicación de delitos dentro de la ciudad de Chicago, pues la distribución de estos puntos configuran de cierta manera el mapa de la ciudad de Chicago.

```
filtro= df['XCoordinate']>0
df=df[filtro]
plt.scatter(df.XCoordinate,df.YCoordinate)
```

```
<matplotlib.collections.PathCollection at 0x22893a2bd90>
```



- **Implementación Algoritmos Machine Learning:**

	Precisión	Área bajo el ROC
Regresión Logística	0.74	0.72
Árboles de Decisión	0.78	0.54
Random Forest	0.84	0.85

Al implementar los modelos de machine learning sobre el dataset, podemos observar que los modelos con la mejor precisión son árboles de decisión y random forest, teniendo en cuenta que este estimador de precisión fue calculado a través de la tabla de predicciones por medio de métodos de pyspark sql. Sin embargo, podemos observar que el algoritmo de árboles de decisión presenta una baja precisión en el estimador del área bajo el ROC. Probablemente eso se deba a la gran cantidad de rangos que existen entre los atributos del dataset, una posible solución para mejorar la precisión de este algoritmo puede ser normalizar ciertos atributos.

Por último podemos observar que los algoritmos que mejor precisión poseen son la regresión logística y random forest siendo este último ,el que mejor se acomoda al dataset. Probablemente esto se deba a que ciertos algoritmos basados en el método de ensamble tienden a adaptarse mejor a dataset que poseen grandes rangos entre sus atributos.

- **¿Cómo vamos a utilizar este conjunto de datos para las entregas posteriores?:**

  Teniendo en cuenta los dos proyectos que se desarrollarán en el curso. A partir del dataset que escogimos y las técnicas de machine learning implementadas, se buscará extender las predicciones y análisis realizados previamente mediante el uso de streaming (Spark Streaming) y grafos (Spark GraphX). En cuanto a eso, se ha realizado un acercamiento en el que se detalla el funcionamiento de las distintas herramientas aplicadas a el dataset escogido.

  - Spark Streaming es una extensión de la API core de Spark que ofrece procesamiento de datos en streaming de manera escalable, alto rendimiento y tolerancia a fallos. Los datos pueden ser tomados de diferentes fuentes como Apache Kafka, Apache Flume, RabbitMQ, Amazon Kinesis, ZeroMQ o sockets TCP.

A diferencia de este primer proyecto en el cual se realiza el análisis y predicciones de los datos tomando todo el dataset, en Spark Streaming se reciben streams de datos en vivo los cuales se dividen en batches o lotes, que son procesados por el motor de Spark para generar un stream de salida o DStream, el cual es una secuencia de RDDs.

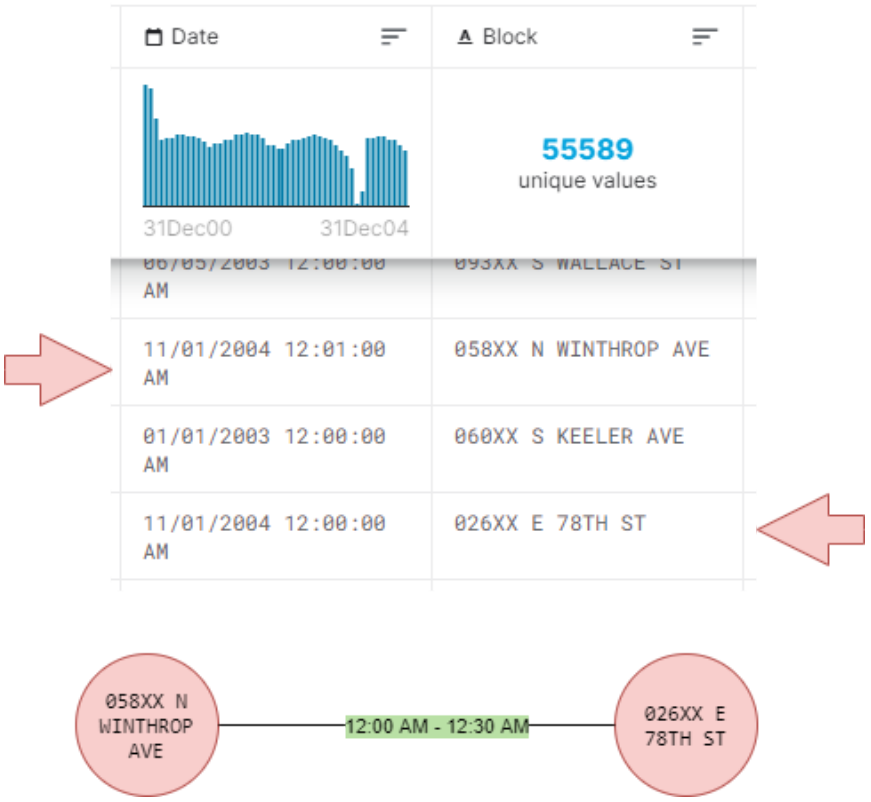
Para procesar dichos datos se llega a utilizar algoritmos complejos de machine learning expresados como funciones de alto nivel como lo son map, reduce, join y window. Una vez procesados, los datos son enviados a archivos en file systems o para dashboards en tiempo real.

- Spark GraphX es un motor para el análisis de grafos, construido sobre el núcleo de Spark, el cual utiliza los RDDs creados previamente con el fin de permitir a los usuarios crear, transformar y obtener conclusiones a partir de un grafo estructurado.

  GraphX se utiliza para explorar la naturaleza o las propiedades topológicas de un grafo. Además, esta herramienta permite ejecutar algoritmos que puedan encontrar subgrafos conectados y desconectados, realizar ciertos conteos o incluso realizar cálculos para hallar el camino más corto de un punto a otro en un grafo.

  Es por esto que para el último proyecto se propone utilizar GraphX junto con Neo4j, dado que los datos procesados en Spark con Mlib se puede llegar a conectar con Neo4j para poder visualizar y conformar los grafos que se pueden producir con estos datos, para después poder analizar las propiedades topológicas del grafo haciendo uso de GraphX.

Con respecto al diseño del grafo para la tercera entrega del proyecto, se planea crear el grafo tomando cierto sector de chicago ya que como contamos con aproximadamente 33.000 bloques mediante los cuales se delimita la ciudad de chicago, no se podrían ingresar todos en neo4j. Los nodos del grafo van a estar compuestos por los bloques de una sección de Chicago y estos van a estar relacionados por medio de ciertas franjas de tiempo en las que ocurre un crimen, es aquí donde se aprovechará que el dataset cuenta con las fechas, horas y minutos en los cuales esto ocurre; todo esto se realizará con el fin de encontrar ciertas similitudes entre dichos bloques. Una vez creado el grafo se plantea trabajar con el algoritmo de k-Nearest Neighbor con el fin de poder llegar a predecir en qué bloques del sector escogido previamente, va a ocurrir un arresto teniendo en cuenta los datos obtenidos de sus vecinos.



- **Cibergrafía**

❏ **How to Handle Imbalanced Classes in Machine Learning. Recuperado de la web en la siguiente Url: https://elitedatascience.com/imbalanced-classes**

❏ **What is Considered to Be a "Strong" Correlation?. Recuperado de la web en la siguiente Url: https://www.statology.org/what-is-a-strong-correlation/**

❏ **Classification and regression. Recuperado de la web en la siguiente Url: https://spark.apache.org/docs/2.2.0/ml-classification-regression.html**

❏ **Student Performance Data Set. Recuperado de la web en la siguiente Url: https://archive.ics.uci.edu/ml/datasets/Student+Performance#**

❏ **Un vistazo a Apache Spark Streaming. Recuperado de la web en la siguiente Url: https://sg.com.mx/revista/50/un-vistazo-apache-spark-streaming**

❏ **Spark Streaming. Recuperado de la web en la siguiente Url: https://bigdatadummy.com/2017/05/12/spark-streaming/**

❏ **Spark Streaming (procesamiento por lotes y tiempo real). Recuperado de la web en la siguiente Url: https://www.diegocalvo.es/spark-streaming/**

❏ **Neo4j and Apache Spark. Recuperado de la web en la siguiente Url: https://neo4j.com/developer/apache-spark/**

❏ **Graph data processing with Neo4j and Apache Spark. Recuperado de la web en la siguiente Url: https://medium.com/neo4j/graph-data-processing-30451b5b576f**