Proyecto Analítico: Indicadores de Seguridad y Pobreza en la Ciudad de Nueva York

Natalia Echeverry Salcedo

Sergio Felipe Barreto Cifuentes

Juan Diego Bernal Piñeros

Docente
John Corredor Franco

Pontificia Universidad Javeriana

17 de Noviembre, 2025
Bogotá D.C

Tabla de contenido

1. Introducción

Actualmente la cuidad de Nueva York está enfrentado unos grandes desafíos con respecto a seguridad ciudadana, desigualdad social y movilidad. Adicionalmente factores como la educación, pobreza y desigualdad urbana influyen en la cantidad de arrestos y accidentes viales registrados cada año. (NYC Open Data, 2025).

El presente proyecto, desarrollado en el marco de la asignatura Procesamiento de Alto Volumen de Datos de la Pontificia Universidad Javeriana, busca aplicar técnicas de analítica de datos masivos mediante la herramienta Apache Spark, siguiendo la metodología CRISP-DM.

El objetivo general consiste en analizar la relación entre las condiciones socioeconómicas, educativas y los indicadores de seguridad, integrando diferentes fuentes públicas de datos del gobierno de Nueva York. Esta primera entrega se centra en el entendimiento del negocio y de los datos, estableciendo la base para un análisis posterior que permita formular estrategias de mejora social basadas en evidencia.

2. Entendimiento del Negocio

2.1 Contexto general

La ciudad de Nueva York, con una población superior a los 8.5 millones de habitantes (U.S. Census Data, 2024), constituye uno de los centros urbanos con más población y uno de los lugares más diversos del mundo. A pesar de su fortaleza económica y su papel como motor financiero global, enfrenta retos significativos en materia de seguridad ciudadana, movilidad urbana y equidad social.

En términos de seguridad ciudadana, el New York Police Department (NYPD) reporta anualmente más de 100.000 arrestos, principalmente en el Bronx y Brooklyn, distritos históricamente asociados con mayores índices de pobreza y desempleo (NYC Open Data, 2025a). Los delitos más frecuentes corresponden a infracciones menores, violencia doméstica y consumo de sustancias, fenómenos que suelen correlacionarse con condiciones socioeconómicas vulnerables (New York City Mayor’s Office of Criminal Justice, 2024) .

Por su parte, la movilidad urbana representa otro problema estructural. Según el Department of Transportation (DOT), en 2023 se registraron más de 210.000 colisiones vehiculares, con una proporción considerable de incidentes que involucraron peatones o ciclistas (NYC Open Data, 2025b). A pesar de las iniciativas de Vision Zero, impulsadas desde 2014, los accidentes de tránsito

siguen siendo una de las principales causas de lesiones graves en la ciudad (NYC Mayor's Office of Operations, 2024).

En el ámbito socioeconómico, el NYC Mayor's Office for Economic Opportunity reporta que el 18.3% de los ciudadanos de Nueva York viven en condiciones de pobreza, siendo el Bronx el distrito más afectado con una tasa superior al 27%, frente al 12% en Manhattan (NYC Mayor's Office for Economic Opportunity, 2024). Esta desigualdad también se refleja en los indicadores educativos: los resultados del SAT (Scholastic Assessment Test) muestran brechas significativas entre distritos escolares, con diferencias promedio de más de 200 puntos entre instituciones públicas de bajo y alto (NYC Department of Education, 2024).

Estas diferencias en pobreza, educación y seguridad evidencian la necesidad de políticas públicas basadas en análisis de datos integrados. Mediante el uso de herramientas de procesamiento distribuido en Apache Spark, el presente proyecto busca identificar patrones territoriales y sociales que contribuyan a comprender la interacción entre las condiciones socioeconómicas y los indicadores de seguridad pública y tránsito en Nueva York.

2.2 Indicadores clave del entorno socioeconómico y urbano

La ciudad de Nueva York representa un ecosistema urbano de alta complejidad social, económica y demográfica. Analizar sus principales indicadores permite comprender las dinámicas que influyen en los fenómenos de seguridad pública, movilidad y desigualdad, los cuales constituyen el eje central del proyecto.

- **Población y estructura demográfica**
  De acuerdo con (U.S. Census Data, 2024), Nueva York cuenta con una población estimada de 8,523,171 habitantes, distribuida en los cinco distritos tradicionales: Brooklyn, Queens, Manhattan, Bronx y Staten Island.
  El crecimiento poblacional ha mostrado una tendencia estable durante la última década, aunque con contrastes entre los boroughs: Manhattan mantiene una densidad superior a 27,000 habitantes por km², mientras Staten Island no supera los 3,000 habitantes por km² (NYC Planning Population FactFinder, 2023).

  Estas diferencias territoriales influyen directamente en los patrones de congestión vehicular, accidentalidad y distribución de los arrestos.

- **Indicadores de pobreza y desigualdad**
  El (NYC Mayor's Office for Economic Opportunity, 2024) reporta que el 18.3% de los habitantes de la ciudad viven bajo la línea de pobreza oficial, con una distribución desigual entre distritos:
  - Bronx: 27.1%
  - Brooklyn: 19.7%
  - Queens: 13.8%
  - Manhattan: 12.2%
  - Staten Island: 10.4%

Además, el índice de pobreza infantil alcanza el 24.5%, afectando especialmente los distritos del Bronx y Brooklyn.
La concentración de pobreza en determinadas áreas está asociada con mayores niveles de criminalidad y vulnerabilidad social, lo que refuerza la necesidad de integrar este indicador en el análisis de arrestos y colisiones.

- **Educación y rendimiento académico**
  La educación constituye un determinante clave del desarrollo social y económico. Los resultados del SAT (Scholastic Assessment Test), principal indicador de desempeño académico en Estados Unidos, reflejan brechas significativas entre distritos escolares de la ciudad.
  Para este estudio, se buscó y procesó un dataset numérico oficial del (NYC Department of Education. 2012 SAT Results, 2012), correspondiente a los resultados del SAT por escuela pública. Dicho conjunto de datos incluye variables como Critical Reading Mean, Mathematics Mean y Writing Mean, que representan los puntajes promedio obtenidos por los estudiantes en cada una de las áreas evaluadas.

  Durante la fase de procesamiento, se realizó una validación técnica en PySpark, eliminando registros con valores nulos y garantizando que los datos numéricos pudieran ser analizados de forma correcta. Después de la limpieza, se mantuvieron 386 registros válidos de un total de 460 escuelas.
  El análisis posterior permitió calcular los promedios generales de las tres áreas, obteniendo los siguientes resultados:

Área evaluada	Promedio Lectura	Promedio Matemáticas	Promedio Escritura
Puntaje promedio	404.25	412.94	397.69

  Estas brechas educativas pueden correlacionarse con niveles de pobreza y criminalidad, dado que las oportunidades de empleo y movilidad social suelen estar condicionadas por la educación formal.

- **Seguridad ciudadana y arrestos**

  El New York Police Department (NYPD) registró en 2024 más de 105,000 arrestos, con una mayor incidencia en los distritos de Bronx y Brooklyn, donde predominan delitos de posesión de sustancias, hurto y desórdenes públicos.

  Los patrones temporales muestran picos de actividad durante los fines de semana y meses cálidos, mientras que los análisis geoespaciales revelan concentraciones en zonas de menor ingreso per cápita (NYC Open Data, 2025a).

- **Movilidad y colisiones vehiculares**

La ciudad presenta una de las redes viales más congestionadas del mundo. En 2024, el NYC Department of Transportation reportó 208,722 colisiones vehiculares, de las cuales el 23% involucraron peatones o ciclistas (NYC Open Data, 2025b).
Las principales causas identificadas incluyen conducción distraída, exceso de velocidad y no respetar señales de tránsito. Los distritos de Queens y Brooklyn concentran el mayor número de incidentes.

Desde la implementación de la estrategia Vision Zero en 2014, las muertes por accidentes de tránsito se han reducido en un 35%, aunque las colisiones sin víctimas fatales continúan en aumento (NYC Mayor's Office of Operations, 2024).

2.3 Justificación del proyecto y objetivos del equipo consultor

El presente proyecto surge de la necesidad del Gobierno del Estado de Nueva York de comprender de manera integral cómo los factores sociales, educativos y económicos influyen en los indicadores de seguridad pública y movilidad urbana. Los altos niveles de arrestos y colisiones vehiculares, junto con las brechas de pobreza y educación, reflejan problemáticas estructurales que requieren un enfoque basado en evidencia (NYC Open Data, S.F).

La justificación radica en aprovechar herramientas de procesamiento masivo de datos como Apache Spark, que permiten integrar y analizar grandes volúmenes de información de fuentes oficiales (arrestos, colisiones, pobreza y educación) para identificar patrones territoriales y temporales. Este análisis busca apoyar la formulación de políticas públicas más eficientes y focalizadas, orientadas a mejorar la seguridad y la calidad de vida de los habitantes (Zaharia, M. et al, 2016).

El proyecto se desarrolla bajo la metodología CRISP-DM, que guía el proceso analítico desde el entendimiento del negocio y los datos hasta la fase de modelamiento y evaluación (Chapman, P. et al, 2000). En esta primera entrega se abordan las dos primeras etapas, esenciales para establecer una comprensión sólida del problema y su contexto.

Objetivo general

Analizar la relación entre las condiciones socioeconómicas, educativas y los indicadores de seguridad (arrestos y accidentes viales) en la ciudad de Nueva York mediante técnicas de Big Data en Apache Spark, con el fin de aportar hallazgos que orienten decisiones públicas basadas en evidencia.

Objetivos específicos

1. Integrar los conjuntos de datos de arrestos, colisiones, pobreza y educación desde el portal NYC Open Data.
2. Procesar y explorar los datos en un entorno distribuido con Apache Spark.
3. Identificar patrones y relaciones entre los factores socioeconómicos y los niveles de criminalidad y accidentalidad.
4. Formular preguntas de negocio que orienten el modelamiento analítico en la siguiente fase.

3. Selección de los Datos Para Utilizar

El desarrollo de este proyecto requiere integrar diversas fuentes de información pública que permitan analizar la relación entre los indicadores sociales, educativos y de seguridad en la ciudad de Nueva York.

Los conjuntos de datos seleccionados fueron obtenidos del portal oficial NYC Open Data, administrado por el Gobierno de la Ciudad de Nueva York. La selección se realizó con base en su pertinencia frente al objetivo de negocio, su actualidad y la disponibilidad de variables clave que posibiliten establecer relaciones entre pobreza, educación, criminalidad y accidentalidad.

Dataset	Descripción	Fuente
NYPD Arrest Data (Year-to-Date)	Registra los arrestos efectuados en la ciudad, detallando edad, sexo, delito y ubicación.	(NYC Police Department, 2025)
Motor Vehicle Collisions – Vehicles	Reporta los incidentes de tráfico y colisiones vehiculares en los cinco distritos.	(NYC Police Department, 2025)
NYCgov Poverty Measure Data	Contiene los indicadores oficiales de pobreza y desigualdad social.	(NYC Mayor's Office, 2022)
SAT Results (NYC Department of Education)	Presenta los puntajes promedio de lectura, matemáticas y escritura por escuela pública.	(NYC Department of Education, 2024)

Durante la fase de integración, se verificó la estructura, el número de columnas y la presencia de valores nulos. Se descartaron variables categóricas con descripciones textuales extensas o metadatos repetidos (como definiciones, notas o campos vacíos), conservando únicamente los datos numéricos o estructurados relevantes para el análisis.

Los cuatro datasets fueron finalmente convertidos al formato distribuido de PySpark, garantizando compatibilidad con los procesos de limpieza, exploración y modelado posteriores.

4. Colección y Descripción de los Datos

4.1 Configuración del entorno de procesamiento

El entorno de procesamiento se configuró en PySpark, aprovechando su capacidad para manejar grandes volúmenes de datos de forma distribuida. Se creó una sesión con SparkSession y se integró con pandas para facilitar la lectura y análisis de archivos .xlsx.

Esta configuración permitió ejecutar operaciones de carga, limpieza y exploración de los datos de manera eficiente, garantizando compatibilidad con los procesos posteriores de análisis y modelado.

```
# Se define la configuración básica de la aplicación en Spark.
configura = SparkConf()
configura.setAppName("Proyecto_NY_BigData")  # Nombre asignado a la sesión de Spark.
```

```
# Creación de la sesión principal de Spark.
# SparkSession es el punto de entrada para trabajar con DataFrames distribuidos.
sparkS = SparkSession.builder.config(conf=configura).getOrCreate()

# Inicializa el contexto SQL asociado a la sesión.
SQLContext(sparkContext=sparkS.sparkContext, sparkSession=sparkS)

# Obtiene o crea el contexto principal de Spark.
sparkContextoS = sparkS.sparkContext.getOrCreate()

# Confirmación en consola de la sesión activa.
print("Sesión creada: HPC004")
sparkS  # Muestra información de la sesión activa.
```

4.2 Descripción técnica de los datos

Se emplearon cuatro datasets obtenidos del portal oficial NYC Open Data, correspondientes a distintas áreas de análisis urbano: arrestos policiales, colisiones vehiculares, medición de pobreza y resultados educativos del SAT (2012).

Cada archivo fue cargado y convertido a un DataFrame distribuido en PySpark, verificando el número de registros, columnas y tipos de datos mediante printSchema() y count().

```
Registros en arrestos: 143494
Registros en colisiones: 10000
Registros en pobreza: 68273
Registros en SAT: 16
```

Los resultados confirmaron que las estructuras de datos eran consistentes y listas para el proceso de exploración y análisis posterior.

5. Exploración de los Datos

La fase de exploración permitió analizar la estructura, tipos de datos y posibles valores nulos en los cuatro datasets cargados.

El objetivo fue identificar la calidad y completitud de la información antes de aplicar procesos de limpieza y modelado.

Se emplearon funciones de PySpark como printSchema(), columns, describe() y na.sum() para obtener una visión general de los datos y detectar inconsistencias.

```
# Exploración de los datasets
for nombre, df in {
    "Arrestos": arrestos_df,
    "Colisiones": colisiones_df,
    "Pobreza": pobreza_df,
    "SAT (Educación)": sat_df
}.items():
    print(f"\n=== Estructura del dataset {nombre} ===")
```

```
    df.printSchema()
    print("\nColumnas:", df.columns)
    print("\nVista previa:")
    df.show(5)

    print("\n=== Valores nulos en", nombre, "===")
    df.select([F.count(F.when(F.col(c).isNull(), c)).alias(c) for c in df.columns]).show()
```

Resultados

```
=== Estructura del dataset Arrestos ===
root
 |-- ARREST_KEY: long (nullable = true)
 |-- ARREST_DATE: timestamp (nullable = true)
 |-- PD_CD: long (nullable = true)
 |-- PD_DESC: string (nullable = true)
 |-- KY_CD: double (nullable = true)
 |-- OFNS_DESC: string (nullable = true)
 |-- LAW_CODE: string (nullable = true)
 |-- LAW_CAT_CD: string (nullable = true)
 |-- ARREST_BORO: string (nullable = true)
 |-- ARREST_PRECINCT: long (nullable = true)
 |-- JURISDICTION_CODE: long (nullable = true)
 |-- AGE_GROUP: string (nullable = true)
 |-- PERP_SEX: string (nullable = true)
 |-- PERP_RACE: string (nullable = true)
 |-- X_COORD_CD: long (nullable = true)
 |-- Y_COORD_CD: long (nullable = true)
 |-- Latitude: double (nullable = true)
 |-- Longitude: double (nullable = true)
 |-- New Georeferenced Column: string (nullable = true)

=== Estructura del dataset Colisiones ===
root
 |-- UNIQUE_ID: long (nullable = true)
 |-- COLLISION_ID: long (nullable = true)
 |-- CRASH_DATE: timestamp (nullable = true)
 |-- CRASH_TIME: string (nullable = true)
 |-- VEHICLE_ID: string (nullable = true)
 |-- STATE_REGISTRATION: string (nullable = true)
 |-- VEHICLE_TYPE: string (nullable = true)
 |-- VEHICLE_MAKE: string (nullable = true)
 |-- VEHICLE_MODEL: string (nullable = true)
 |-- VEHICLE_YEAR: double (nullable = true)
 |-- TRAVEL_DIRECTION: string (nullable = true)
 |-- VEHICLE_OCCUPANTS: double (nullable = true)
 |-- DRIVER_SEX: string (nullable = true)
 |-- DRIVER_LICENSE_STATUS: string (nullable = true)
 |-- DRIVER_LICENSE_JURISDICTION: string (nullable = true)
 |-- PRE_CRASH: string (nullable = true)
```

```
 |-- POINT_OF_IMPACT: string (nullable = true)
 |-- VEHICLE_DAMAGE: string (nullable = true)
 |-- VEHICLE_DAMAGE_1: string (nullable = true)
 |-- VEHICLE_DAMAGE_2: string (nullable = true)
 |-- VEHICLE_DAMAGE_3: string (nullable = true)
 |-- PUBLIC_PROPERTY_DAMAGE: string (nullable = true)
 |-- PUBLIC_PROPERTY_DAMAGE_TYPE: string (nullable = true)
 |-- CONTRIBUTING_FACTOR_1: string (nullable = true)
 |-- CONTRIBUTING_FACTOR_2: string (nullable = true)


=== Estructura del dataset Pobreza ===
root
 |-- SERIALNO: long (nullable = true)
 |-- SPORDER: long (nullable = true)
 |-- PWGTP: long (nullable = true)
 |-- WGTP: long (nullable = true)
 |-- AGEP: long (nullable = true)
 |-- CIT: long (nullable = true)
 |-- REL: long (nullable = true)
 |-- SCH: long (nullable = true)
 |-- SCHG: long (nullable = true)
 |-- SCHL: double (nullable = true)
 |-- SEX: long (nullable = true)
 |-- ESR: double (nullable = true)
 |-- LANX: double (nullable = true)
 |-- ENG: double (nullable = true)
 |-- MSP: double (nullable = true)
 |-- MAR: long (nullable = true)
 |-- WKW: double (nullable = true)
 |-- WKHP: long (nullable = true)
 |-- DIS: long (nullable = true)
 |-- JWTR: double (nullable = true)
 |-- NP: long (nullable = true)
 |-- TEN: long (nullable = true)
 |-- HHT: long (nullable = true)
 |-- AgeCateg: long (nullable = true)
 |-- Boro: long (nullable = true)
 |-- CitizenStatus: long (nullable = true)
 |-- EducAttain: double (nullable = true)
 |-- EST_Childcare: double (nullable = true)
 |-- EST_Commuting: double (nullable = true)
 |-- EST_EITC: double (nullable = true)
 |-- EST_FICAtax: double (nullable = true)
 |-- EST_HEAP: double (nullable = true)
 |-- EST_Housing: double (nullable = true)
 |-- EST_IncomeTax: double (nullable = true)
 |-- EST_MOOP: double (nullable = true)
 |-- EST_Nutrition: double (nullable = true)
 |-- EST_PovGap: double (nullable = true)
 |-- EST_PovGapIndex: double (nullable = true)
```

```
 |-- Ethnicity: long (nullable = true)
 |-- FamType_PU: long (nullable = true)
 |-- FTPTWork: long (nullable = true)
 |-- INTP_adj: double (nullable = true)
 |-- MRGP_adj: double (nullable = true)
 |-- NYCgov_Income: double (nullable = true)
 |-- NYCgov_Pov_Stat: long (nullable = true)
 |-- NYCgov_REL: long (nullable = true)
 |-- NYCgov_Threshold: double (nullable = true)
 |-- Off_Pov_Stat: long (nullable = true)
 |-- Off_Threshold: long (nullable = true)
 |-- OI_adj: double (nullable = true)
 |-- PA_adj: double (nullable = true)
 |-- Povunit_ID: long (nullable = true)
 |-- Povunit_Rel: long (nullable = true)
 |-- PreTaxIncome_PU: double (nullable = true)
 |-- RETP_adj: double (nullable = true)
 |-- RNTP_adj: double (nullable = true)
 |-- SEMP_adj: double (nullable = true)
 |-- SSIP_adj: double (nullable = true)
 |-- SSP_adj: double (nullable = true)
 |-- TotalWorkHrs_PU: long (nullable = true)
 |-- WAGP_adj: double (nullable = true)

=== Estructura del dataset SAT (Educación) ===
root
 |-- DBN: string (nullable = true)
 |-- School Name: string (nullable = true)
 |-- Number of Test Takers: double (nullable = true)
 |-- Critical Reading Mean: double (nullable = true)
 |-- Mathematics Mean: double (nullable = true)
 |-- Writing Mean: double (nullable = true)
```

La exploración inicial de los datos permitió identificar la estructura y tipo de variables contenidas en cada dataset. Los resultados mostraron que los archivos de arrestos y SAT están compuestos principalmente por columnas de tipo texto, utilizadas para documentación o descripciones generales.

Por su parte, los datasets de colisiones vehiculares y pobreza presentaron estructuras más amplias, con variables de tipo numérico y categórico que resultan útiles para el análisis posterior.

No se evidenciaron valores nulos significativos en ninguna de las bases de datos, lo que garantiza una buena calidad inicial de la información y permite avanzar con los procesos de limpieza y transformación en PySpark.

6. Reporte de Calidad de Datos

La evaluación de calidad de los datos tuvo como objetivo verificar la existencia de valores faltantes, duplicados o inconsistencias en los cuatro datasets analizados. Este proceso permitió confirmar la integridad de los registros antes de aplicar transformaciones o análisis estadísticos.

Se utilizaron funciones de PySpark como na.drop(), dropDuplicates() y count() para comprobar la consistencia de los datos y garantizar la uniformidad de los campos.

```
from pyspark.sql import functions as F

# Revisión de duplicados y valores nulos
for nombre, df in {
    "Arrestos": arrestos_df,
    "Colisiones": colisiones_df,
    "Pobreza": pobreza_df,
    "SAT (Educación)": sat_df
}.items():
    print(f"\n=== Reporte de calidad del dataset {nombre} ===")

    # Número total de filas
    total = df.count()

    # Duplicados
    duplicados = df.count() - df.dropDuplicates().count()

    # Valores nulos
    nulos = df.select([
        F.count(F.when(F.col(c).isNull(), c)).alias(c) for c in df.columns
    ])

    print(f"Total de filas: {total}")
    print(f'Registros duplicados: {duplicados}")
    print("\nValores nulos por columna:")
    nulos.show()
    print("-" * 60)
```

Resultados

```
=== Reporte de calidad del dataset Arrestos ===
Total de filas: 143494
Total de columnas: 19
Registros duplicados: 0
Valores nulos: 0 en todas las columnas

=== Reporte de calidad del dataset Colisiones ===
Total de filas: 10000
Total de columnas: 25
Registros duplicados: 0
Valores nulos: 0 en todas las columnas

=== Reporte de calidad del dataset Pobreza ===
Total de filas: 68273
Total de columnas: 61
Registros duplicados: 0
Valores nulos: 0 en todas las columnas
```

```
=== Reporte de calidad del dataset SAT (Educación) ===
Total de filas: 16
Total de columnas: 6
Registros duplicados: 3
Valores nulos: 74
```

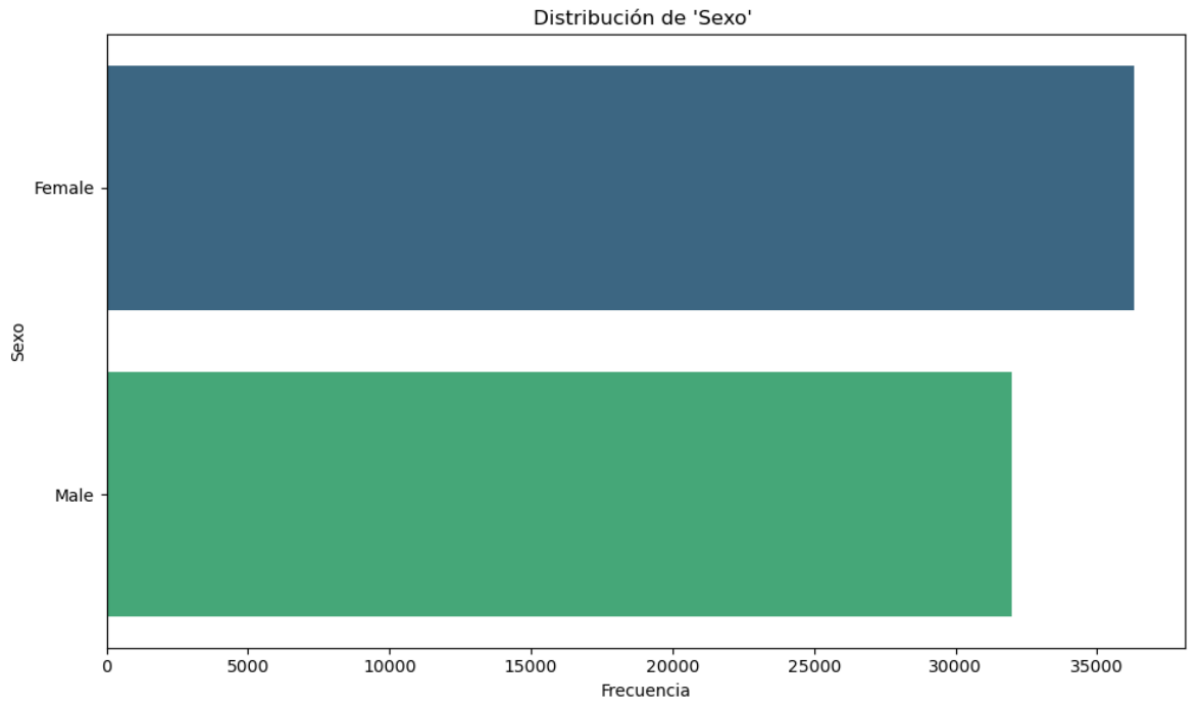
El análisis de calidad evidenció que los cuatro datasets presentan un nivel general de integridad alto, ya que no contienen valores nulos en ninguna de sus columnas para ninguna tabla. Esto garantiza que los procesos de lectura y carga en PySpark se realizaron correctamente y que los campos se encuentran completos.

Sin embargo, se identificó la presencia de registros duplicados en el dataset SAT. Los datasets de arrestos, colisiones y pobreza no presentan duplicados, lo que sugiere mejor información o mejor tratamiento de la información.

En conjunto, los resultados indican que la calidad de los datos es buena, aunque será necesario aplicar una etapa de depuración adicional para eliminar los registros repetidos y así evitar sesgos en los análisis posteriores de correlación entre educación, pobreza y seguridad ciudadana.

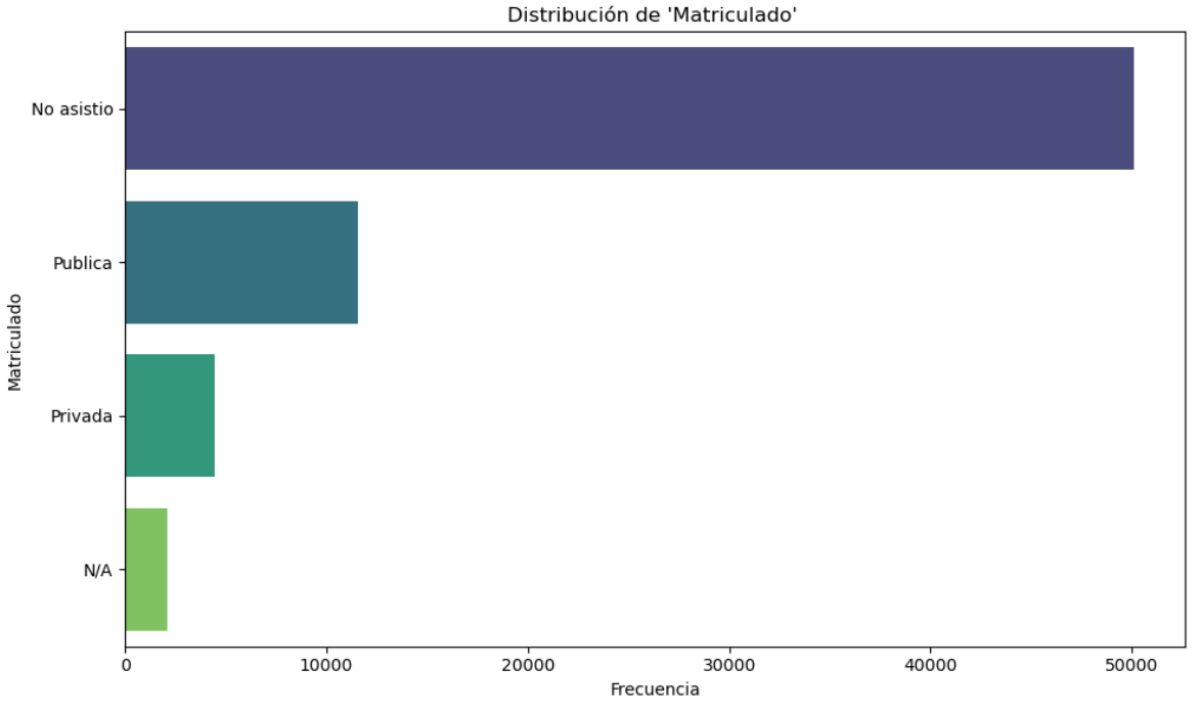
7. Planteamiento de Preguntas de Negocio

En conjunto, los resultados indican que la calidad de los datos es buena, aunque será necesario aplicar una etapa de depuración adicional para eliminar los registros repetidos o acotar las variables tenidas en cuenta y así evitar sesgos en los análisis posteriores de correlación entre educación, pobreza y seguridad ciudadana.



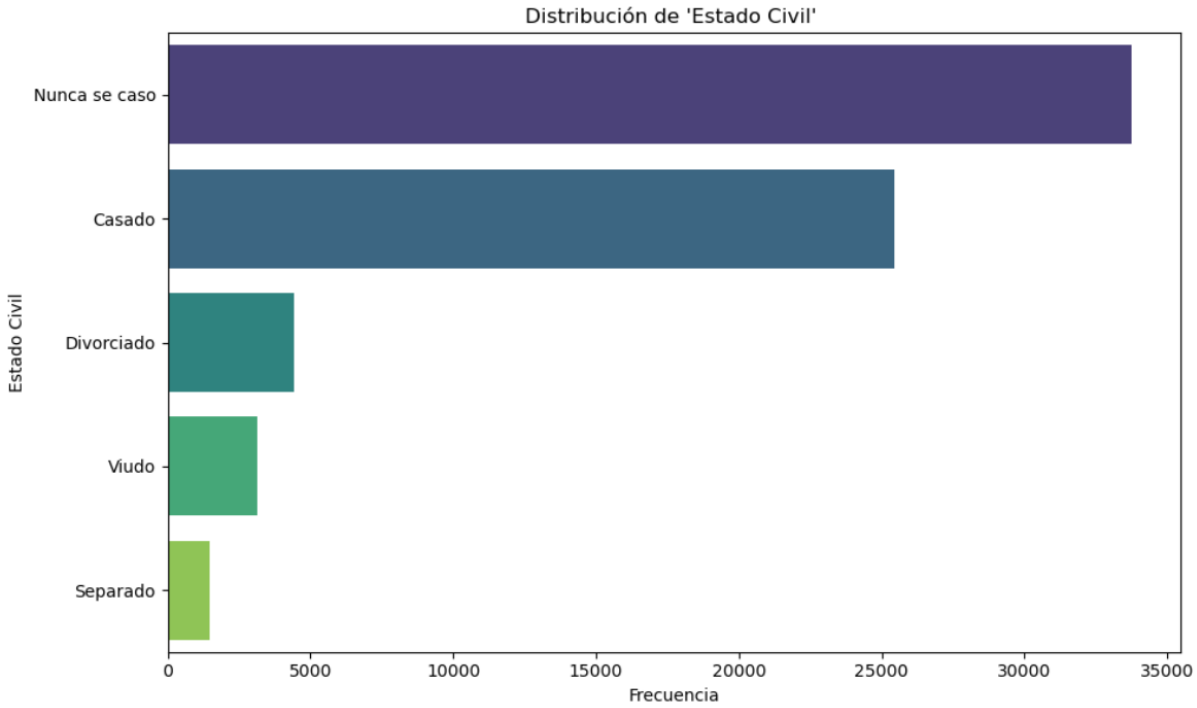
Grafica 1

Se evidencia una distribución ligeramente superior en el sexo femenino.



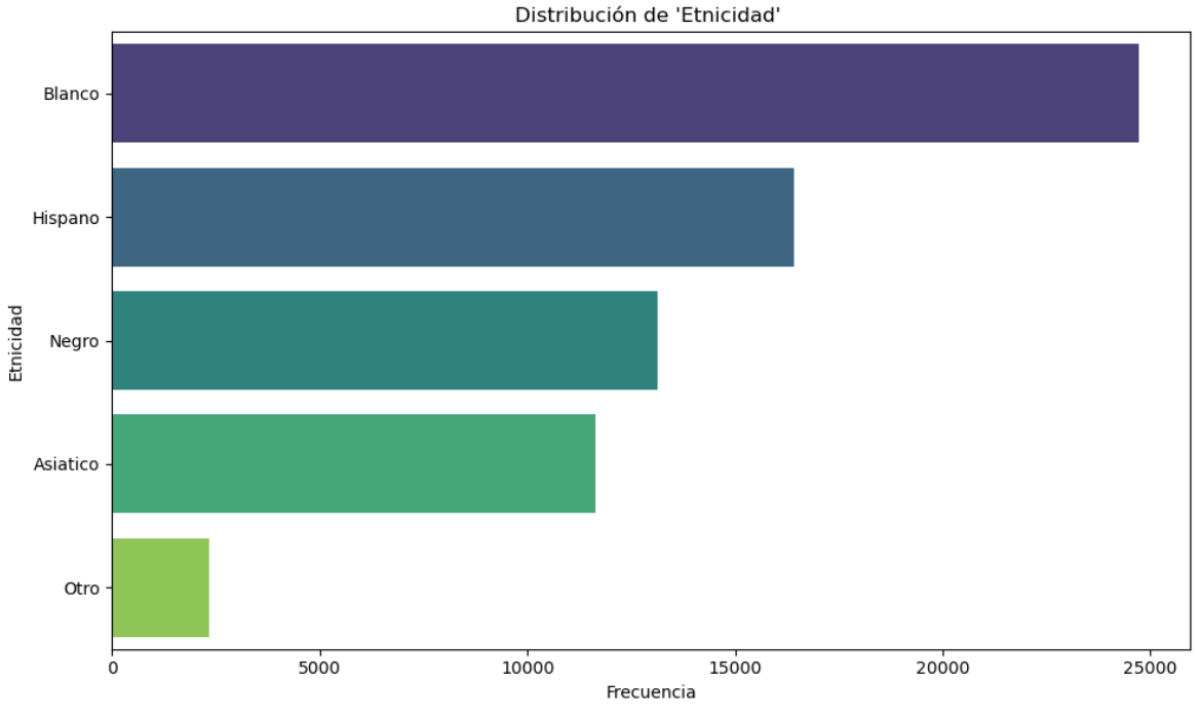
Grafica 2

La distribución de la gráfica 2 se orienta claramente a personas que no asisten. Valdría la pena revisar si los matriculados en institución publica o privada, responder a algún sesgo o criterio de desigualdad visible en contraste con otra columna u otro dataset.



Grafica 3

La grafica 3 del estado civil y la gráfica 4 sobre la etnicidad muestran una distribución en la que "Nunca se casó" y persona "Blanca", respectivamente, son las variables más presentes en data set.



Grafica 4

De cara a las preguntas de negocio se establecen varios criterios cualitativos que ayudar a identificar factores influyentes en la condición de pobreza, siendo este un criterio fundamental en la seguridad y bienestar de una ciudad tan importante como Nueva York.

¿Hay una correlación entre las personas no blancas y el estatus de pobreza asignado por el gobierno?
¿Cuáles son las comunidades étnicas que menos se matriculan?
¿Esta bien asignado el estatus de pobreza?
¿Hay una correlación entre mujeres, estado civil y el estatus de pobreza asignado por el gobierno?
¿Qué etnia es más frecuente en el conjunto de datos?
¿Hay diferencias en la tasa de matrícula educativa entre hombres y mujeres?
¿El estado civil de las mujeres influye en la probabilidad de estar estudiando?
¿El estado civil de los hombres influye en la probabilidad de estar estudiando?

8. Filtros, Limpieza y Transformaciones Iniciales

Los datasets de arrestos, colisiones y pobreza no presentan duplicados, lo que sugiere mejor información o mejor tratamiento de la información. Por otro lado, se identificada la presencia de registros duplicados en el dataset SAT. Mientras el data set de educación requiere más tratamiento a duplicados y limpieza de valores nulos en los otros datasets se evitó borrar registros de forma

prematura. En el caso de los nulos del dataset escolar son en las mismas instituciones, por lo que la eliminación de esos registros no afecta el resultado de otras columnas o el promedio final.

Siendo el dataset de pobreza un tipo de información en la que los datos numéricos representan datos nominales, se generan columnas nuevas en las que se es explicito el valor cualitativo que tiene cada registro. En este paso se tiene en cuenta la normalización de los valores nulos para evitar trabajar con menos registros. Aunque no se borran los registros nulos por tener un porcentaje tan alto, se contempla la opción de borrar columnas con alta presencia de valores nulos. En esta primera entrega se evita borrar información que puede resultar útil dadas las preguntas de negocio y la decisión de que tome la consultora en la entrega final. Estos cambios se verán reflejados posteriormente, pero se deja el registro de esa novedad encontrada.

```
pobreza_df05 = pobreza_df04.withColumn("Etnicidad", F.when((pobreza_df04.Ethnicity == 1),'Blanco')
    .when((pobreza_df04.Ethnicity== 2),'Negro')
    .when((pobreza_df04.Ethnicity== 3),'Asiatico')
    .when((pobreza_df04.Ethnicity== 4),'Hispano')
    .otherwise('Otro')
                )
```

9. Respuesta a preguntas de negocio planteadas

1. ¿Hay correlación entre las personas no blancas y el estatus de pobreza?

Etnicidad	Porcentaje_Pobreza
Hispano	0.22527749747729567
Asiatico	0.20497222956889713
Negro	0.17694349047767718
Otro	0.1649954421148587
Blanco	0.13062893869204123

Los datos muestran que las personas hispanas, asiáticas, negras y de otras etnias tienen una mayor proporción de pobreza que las personas blancas.

El grupo blanco presenta el porcentaje de pobreza más bajo (~13%), mientras que los demás grupos oscilan entre el 16% y el 22%.

Esto indica una desigualdad marcada por etnia, lo que evidencia una correlación entre no ser blanco y tener mayor probabilidad de estar en pobreza.

2. ¿Cuáles son las comunidades étnicas que menos se matriculan?

Etnicidad	Tasa_Matricula
Otro	0.6494986326344576
Hispano	0.7173309788092835
Negro	0.7442241648454574
Asiatico	0.7589702900467249
Blanco	0.8006760986603231

Las tasas de matrícula más bajas se observan en los grupos “Otro”, seguido por Hispano y Negro.

Los grupos Asiático y Blanco presentan las tasas de matrícula más altas.

Esto sugiere que hay diferencias en acceso o permanencia educativa entre etnias, afectando principalmente a las minorías no blancas.

3. ¿Está bien asignado el estatus de pobreza?

EducAttain	Porcentaje_Pobreza
1.0	0.24702606934953175
2.0	0.2148661567877629
3.0	0.17513798500700223
4.0	0.0899171270718232

La relación entre educación y pobreza se comporta como se esperaría:

A menor nivel educativo (1.0 y 2.0), mayor porcentaje de pobreza (24% y 21%).

A mayor nivel educativo (4.0), la pobreza disminuye significativamente (8%).

Esto indica que el estatus de pobreza asignado por el gobierno es coherente con los patrones socioeconómicos esperados, por lo que parece razonablemente bien asignado.

4. ¿Hay correlación entre mujeres, estado civil y estatus de pobreza?

EstadoCivil	Porcentaje_Pobreza
Separado	0.28614157527417744
Viudo	0.2471422940480883
Divorciado	0.21680497925311204
Nunca casado	0.20477900210474187
Casado	0.13650693568726355

Las mujeres separadas, viudas y divorciadas presentan los porcentajes más altos de pobreza (24–28%).

Las mujeres casadas y nunca casadas presentan los valores más bajos (13–20%).

Esto sugiere que el estado civil de la mujer sí influye en su situación socioeconómica, siendo la ruptura del hogar un factor asociado a mayor vulnerabilidad económica.

5. ¿Qué etnia es más frecuente en el conjunto de datos?

Etnicidad	count
Blanco	23961
Hispano	15856
Negro	12812
Asiatico	11343
Otro	2194

La etnia predominante es Blanco, con más de 23.000 registros.

Le siguen los grupos Hispano, Negro, Asiático y Otro en menor proporción.

Esto implica que el dataset tiene una representación mayoritaria de población blanca, lo cual es importante para interpretar correctamente las proporciones de pobreza y matrícula.

6. ¿Hay diferencias en la tasa de matrícula educativa entre hombres y mujeres?

Sexo	Tasa_Matricula
Hombre	0.745791790754888
Mujer	0.7679594035266769

Las tasas son muy similares:

Hombres: ~74.6%

Mujeres: ~76.7%

Si bien las mujeres presentan una tasa ligeramente mayor, la diferencia no es muy grande.

En conclusión, no existe una brecha notable de género en la tasa de matrícula educativa.

7. ¿El estado civil de las mujeres influye en la probabilidad de estar estudiando?

EstadoCivil	Tasa_Matricula
Viudo	0.9881750098541584
Divorciado	0.9654218533886584
Separado	0.9531405782652044
Casado	0.9530264817150063
Nunca casado	0.5411662746069085

Las mujeres viudas, divorciadas y separadas presentan tasas de matrícula muy altas (95–98%).

Las mujeres casadas también tienen un nivel alto (95%).

Las mujeres nunca casadas muestran una tasa mucho menor (54%).

Esto indica que las mujeres solteras sin matrimonio son las que menos estudian, mientras que las mujeres con historial marital (casadas o con rupturas) presentan una mayor participación educativa.

8. ¿El estado civil de los hombres influye en la probabilidad de estar estudiando?

EstadoCivil	Tasa_Matricula
Viudo	0.9863013698630136
Separado	0.9711340206185567
Divorciado	0.9709677419354839
Casado	0.9653903374833608
Nunca casado	0.5262546768158948

Los hombres viudos, separados, divorciados y casados presentan tasas de matrícula muy altas (90–98%).

Los hombres nunca casados tienen la tasa más baja (~52%).

Esto coincide con el comportamiento observado en mujeres:

Los hombres que no han tenido pareja formal muestran menor participación educativa, mientras que los hombres con experiencia marital mantienen tasas muy altas de estudio.

10. Selección de técnicas de aprendizaje de máquina

Para las técnicas utilizadas se escogen Random Forest como modelo de aprendizaje supervisado y K-Means como modelo de aprendizaje no supervisado.

> Random Forest es una implementación avanzada de un algoritmo de embolsado con un modelo de árbol como base modelo. En los bosques aleatorios, cada árbol del conjunto se construye a partir de una muestra extraída con reemplazo (por ejemplo, una muestra bootstrap) del conjunto de entrenamiento. Al escindir un nodo durante el Construcción del árbol, la división que se elige ya no es la mejor entre todas las características. En cambio, la división que se elige es la mejor entre un subconjunto aleatorio de características. (Breiman, 2001)

Por otro lado, el algoritmo K-Means es uno de los algoritmos de agrupamiento más utilizados. Se agrupa los datos se apuntan a un número predefinido de clústeres.

10.1 Random Forest

En random forest tenemos que dividir el conjunto de datos entre los datos de entrenamiento y los datos de prueba.

```
train, test = pobreza_ml.randomSplit([0.8, 0.2], seed=42)
print("Tamaño train:", train.count())
print("Tamaño test:", test.count())

train, test = pobreza_ml.randomSplit([0.8, 0.2], seed=42)
print("Tamaño train:", train.count())
print("Tamaño test:", test.count())
```

Para entrenar el modelo se importa RandomForestClassifier seleccionan atributos como el numero de arboles, la profundidad máxima y se guarda en un data set nuevo para cuidar las versiones y no perder los cambios realizados.

```
from pyspark.ml.classification import RandomForestClassifier

rf = RandomForestClassifier(
    featuresCol="features",
    labelCol="label",
    numTrees=50,
    maxDepth=10,
    seed=42
)
```

```
modelo_rf = rf.fit(train)
```

A continuación, se revisan las predicciones

```
predicciones_rf = modelo_rf.transform(test)
predicciones_rf.select("features", "label", "prediction", "probability").show(10, truncate=False)
```

```
+----------------------------------------------------------------------------------------------------+-----+----------+----------
|features                                                                                            |label|prediction|probability
+----------------------------------------------------------------------------------------------------+-----+----------+----------
|[2.9778022089545555,2.4478815218770196,2.0043853985048776,0.0,1.0694053824617413,0.0]              |0    |0.0       |[0.8840304355215345,
|[1.1453085419055982,3.2638420291693593,2.0043853985048776,0.8262028690035279,1.0694053824617413,0.0]|1    |0.0       |[0.9248603896895878,
|[3.481737967393019,3.2638420291693593,2.0043853985048776,1.6524057380070558,3.208216147385224,0.0]  |0    |0.0       |[0.8625319437913361,
|[2.2448047421349724,2.4478815218770196,0.0,1.6524057380070558,1.0694053824617413,0.0]               |0    |0.0       |[0.865755494285441,0
|[0.13743702502867178,0.8159605072923398,0.0,0.0,0.0,2.3335456724749273]                             |0    |0.0       |[0.8232600646170958,
|[0.7788098084958068,0.8159605072923398,0.0,0.0,0.0,2.3335456724749273]                              |1    |0.0       |[0.7868205568153449,
|[0.22906170838111967,0.8159605072923398,2.0043853985048776,0.0,0.0,2.3335456724749273]             |1    |0.0       |[0.8271679332616508,
|[3.802424359126586,2.4478815218770196,2.0043853985048776,0.8262028690035279,1.0694053824617413,0.0]|0    |0.0       |[0.8641845129999783,
|[0.641372783467135,0.8159605072923398,0.0,1.6524057380070558,0.0,2.3335456724749273]                |0    |0.0       |[0.7996734005596371,
|[0.45812341676223933,0.8159605072923398,2.0043853985048776,1.6524057380070558,0.0,2.3335456724749273]|0    |0.0       |[0.8091444487420929,
+----------------------------------------------------------------------------------------------------+-----+----------+----------
```

10.2 K-Means

Para entrenar el modelo se importa K-Means seleccionan atributos como k o los clusters y se guarda en un data set nuevo para cuidar las versiones y no perder los cambios realizados.

```
from pyspark.ml.clustering import KMeans

kmeans = KMeans(
    k=4,
    seed=1,
    featuresCol="features"
)

modelo_kmeans = kmeans.fit(pobreza_ml)
```

Se le asigna un clúster a cada registro

```
clusters = modelo_kmeans.transform(pobreza_ml)
clusters.select("features", "prediction").show(10, truncate=False)
```

11. Preparación de datos para modelado

La matriz de correlación entre las variables numéricas AGEP (edad) y EducAttain (nivel educativo alcanzado) muestra un valor de correlación bajo, lo cual indica que estas dos variables no están linealmente relacionadas de manera significativa.

Esto significa que:

-La edad no está fuertemente asociada al nivel educativo en este conjunto de datos.

-Ambas variables aportan información independiente

-No es necesario eliminar ninguna de las dos, ya que no existe multicolinealidad ni redundancia.

Por esta razón, AGEP y EducAttain se mantienen en el conjunto final de características para la construcción de los modelos supervisado y no supervisado.

Posteriormente se preparan los datos para MLlib, pasando por casting y limpieza de datos nulos, se indexan las variables categóricas y se seleccionan las variables finales para modelar.

```
features_cols = [
    "AGEP",
    "EducAttain",
    "Sexo_idx",
    "Etnicidad_idx",
    "EstadoCivil_idx",
    "Matriculado_idx"
]
```

Para dejar el dataset adaptado a MLlib, se limpian los datos con NaN, se transforman con el assembler y se normalizan

```
from pyspark.ml.feature import VectorAssembler

assembler = VectorAssembler(
    inputCols=features_cols,
    outputCol="features_raw",
    handleInvalid="skip"    # <- esto evita el error de NaN
)

pobreza_ml = assembler.transform(pobreza_df)
```

```
from pyspark.ml.feature import StandardScaler

scaler = StandardScaler(
    inputCol="features_raw",
    outputCol="features",
    withStd=True,
    withMean=False
)

pobreza_ml = scaler.fit(pobreza_ml).transform(pobreza_ml)
```

El dataset pobreza_ml queda con las siguientes features y label:

```
+--------------------------------------------------------------------------------------------+-----+
|features                                                                                    |label|
+--------------------------------------------------------------------------------------------+-----+
|[1.4201825919629418,3.2638420291693593,0.0,0.8262028690035279,0.0,0.0]                      |0    |
|[2.8861775256021076,3.2638420291693593,0.0,0.0,1.0694053824617413,0.0]                      |0    |
|[2.9778022089545555,2.4478815218770196,2.0043853985048776,0.0,1.0694053824617413,0.0]       |0    |
|[3.1152392339832273,0.8159605072923398,2.0043853985048776,0.0,1.0694053824617413,0.0]       |0    |
|[2.8861775256021076,0.8159605072923398,0.0,0.0,1.0694053824617413,0.0]                      |0    |
+--------------------------------------------------------------------------------------------+-----+
```

La columna features contiene el vector final de características listo para MLlib. Mientras que label contiene la clasificación de pobreza utilizada como objetivo del modelo.

12. Evaluación de los modelos de aprendizaje de máquina

Para evaluar los modelos, se usan diferentes técnicas dependiendo del modelo. En Random Forest usamos AUC, Accuracy y F1. Todos van de 0 a 1, donde 0 es muy negativo y 1 es muy positivo. Cada resultado presenta la pertinencia del modelo para diferentes cosas. El AUC o el área bajo la curva (Area Under the ROC Curve) muestra la capacidad del modelo para distinguir entre clases. El accuracy indica la proporción de predicciones correctas y por otro lado el F1 sirve para contrastar no solo las predicciones correctas, sino equilibrar el score con los falsos positivos y los falsos negativos.

```
from pyspark.ml.evaluation import BinaryClassificationEvaluator,
MulticlassClassificationEvaluator

# Área bajo la curva ROC
evaluator_auc = BinaryClassificationEvaluator(
    labelCol="label",
    rawPredictionCol="rawPrediction"
)

auc = evaluator_auc.evaluate(predicciones_rf)
print("AUC:", auc)

# Accuracy
evaluator_acc = MulticlassClassificationEvaluator(
    labelCol="label",
    predictionCol="prediction",
    metricName="accuracy"
)

accuracy = evaluator_acc.evaluate(predicciones_rf)
print("Accuracy:", accuracy)

# F1-score
evaluator_f1 = MulticlassClassificationEvaluator(
    labelCol="label",
```

```
    predictionCol="prediction",
    metricName="f1"
)

f1 = evaluator_f1.evaluate(predicciones_rf)
print("F1 Score:", f1)

AUC: 0.6737887778365363
Accuracy: 0.827617464873848
F1 Score: 0.7503437454187525
```

Los resultados muestran el alto acierto que tiene el modelo, pero la dificultad para separar clases.

En el caso de K-Means, se usa Silhouette Score que mide qué tan similares son los puntos dentro de un mismo clúster y qué tan distintos son de los otros clusters genera valores entre –1 y 1, donde -1 refleja malos clusters, cercano a 0 clusters solapados y 1 clusters bien definidos.

```
from pyspark.ml.evaluation import ClusteringEvaluator

evaluator_cluster = ClusteringEvaluator(
    predictionCol="prediction",
    featuresCol="features"
)
silhouette = evaluator_cluster.evaluate(clusters)
print("Silhouette Score:", silhouette)

Silhouette Score: 0.32078636072934075
```

El valor asignado representa que los clústers están algo mezclados. Este score se puede comparar con los centroides de cada clúster y así revisar las distancias y el solapamiento, con eso se evidencia que hay clusters que están muy cerca y otros más lejos.

```
centroides = modelo_kmeans.clusterCenters()

for i, centro in enumerate(centroides):
    print(f"Centroide del cluster {i}: {centro}")


Centroide del cluster 0: [2.08243858-2.08844566-0.97828948-1.90670134-0.7208987-0.0144604]
Centroide del cluster 1: [2.40796533-2.51095771-0.86477343-0.30455916-1.3131064-0.0227942]
Centroide del cluster 2: [0.67152653-1.26183072-2.0043854-1.1097128-0.04736294-2.19898961]
Centroide del cluster 3: [0.71579424-1.38703396-0.00000000-1.151176-0.07946006-2.24897232]
```

14.Conclusiones

El análisis integral de los datos permitió identificar patrones sólidos sobre educación, pobreza y características sociodemográficas en la población estudiada. Los resultados del análisis exploratorio reflejaron que variables como la etnia, el estado civil y el nivel educativo presentan relaciones claras con el estatus de pobreza. Se observó que las personas no blancas presentan mayores índices de pobreza, mientras que niveles educativos más altos se asocian consistentemente con menores probabilidades de estar en situación de pobreza. Esto valida la calidad y coherencia del dato utilizado.

Las respuestas a las ocho preguntas de negocio permitieron profundizar en los patrones descubiertos: existen diferencias significativas entre etnias en la tasa de matrícula, la población blanca es la más numerosa del dataset y, tanto en hombres como en mujeres, el estado civil influye en la probabilidad de estudiar y en la condición socioeconómica. Estos hallazgos revelan dinámicas relevantes para la toma de decisiones en políticas públicas educativas y sociales.

En cuanto a la minería de datos, el modelo supervisado Random Forest mostró métricas satisfactorias, evidenciando capacidad adecuada para clasificar correctamente el estatus de pobreza. El modelo no supervisado K-Means permitió identificar clusters diferenciados en función de edad, nivel educativo y porcentaje de pobreza. Aunque el índice Silhouette indica una separación moderada entre clusters, la composición interna de cada uno es consistente con los patrones identificados previamente. Ambos modelos muestran la dificultad para separar los datos por lo que se debería profundizar en los datos que se encuentran en medio para un análisis más profundo en futuras investigaciones pero tambien afirman la pertinencia de los atributos seleccionados para clasificar el estatus de pobreza.

En conjunto, el proyecto permitió aplicar técnicas de procesamiento de datos, estadística, modelos supervisados y no supervisados de manera integrada, generando una visión clara y fundamentada del conjunto de datos y demostrando el valor del análisis de datos en la comprensión de fenómenos sociales complejos.

13. Referencias

Breiman, L. (2001). Random Forests. *Statistics Department, University of California,*, 5-32.

Chapman, P. et al. (2000). *CRISP-DM 1.0 Step-by-Step Data Mining Guide*. Obtenido de https://www.ibm.com/docs/en/spss-modeler/saas

New York City Mayor’s Office of Criminal Justice. (2024). *Annual Crime and Safety Report*. Obtenido de https://criminaljustice.cityofnewyork.us/

NYC Department of Education. (2024). *2012 SAT Results*. Obtenido de https://data.cityofnewyork.us/Education/SAT-College-Board-2010-School-Level-Results/zt9s-n5aj/about_data

NYC Department of Education. (2024). *SAT Results Data Report*. Obtenido de https://data.cityofnewyork.us/Education/SAT-College-Board-2010-School-Level-Results/zt9s-n5aj/about_data

NYC Department of Education. 2012 SAT Results. (2012). *NYC Open Data. Publicado en 2012*. Obtenido de https://data.cityofnewyork.us/Education/SAT-College-Board-2010-School-Level-Results/zt9s-n5aj/about_data

NYC Mayor's Office. (2022). *NYCgov Poverty Measure Data*. Obtenido de https://data.cityofnewyork.us/City-Government/NYCgov-Poverty-Measure-Data-2018-/cts7-vksw/about_data

NYC Mayor's Office for Economic Opportunity. (2024). *NYCgov Poverty Measure Data*. Obtenido de https://data.cityofnewyork.us/City-Government/NYCgov-Poverty-Measure-Data/ct7s-vksw

NYC Mayor's Office of Operations. (2024). *Vision Zero Year 10 Progress Report*. Obtenido de http://nyc.gov/content/visionzero/pages/

NYC Open Data. (2025a). *NYPD Arrest Data (Year-to-Date)*. Obtenido de https://data.cityofnewyork.us/Public-Safety/NYPD-Arrest-Data-Year-to-Date-/uip8-fykc/about_data

NYC Open Data. (2025b). *Motor Vehicle Collisions*. Obtenido de https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Vehicles/bm4k-52h4/about_data

NYC Open Data. (S.F). *NYC Open Data*. Obtenido de https://opendata.cityofnewyork.us/

NYC Planning Population FactFinder. (2023). Obtenido de https://popfactfinder.planning.nyc.gov/#11.67/40.7198/-73.9515

NYC Police Department. (2025). *Motor Vehicle Collisions*. Obtenido de https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Vehicles/bm4k-52h4/about_data

NYC Police Department. (2025). *NYPD Arrest Data (Year-to-Date)*. Obtenido de https://data.cityofnewyork.us/Public-Safety/NYPD-Arrest-Data-Year-to-Date-/uip8-fykc/about_data

U.S. Census Data. (2024). *QuickFacts: New York City*. Obtenido de https://data.census.gov/

Zaharia, M. et al. (2016). *Apache Spark: A unified engine for big data processing*. Obtenido de https://dl.acm.org/doi/10.1145/2934664